Neural Attention in Heterogeneous-Agent Economies

Alessandro T. Villa

August 11, 2026

WP 2026-14

https://doi.org/10.21033/wp-2026-14

FEDERAL RESERVE BANK of CHICAGO

Neural Attention in Heterogeneous-Agent Economies

Alessandro T. Villa*

August 11, 2026

Abstract

Transformers, the neural-network architecture behind much of modern AI, combine flexible nonlinear approximation with an attention mechanism that learns how information is combined across inputs. Can neural attention reveal which parts of an agent distribution matter for aggregate dynamics? I represent economic states as tokens—such as wealth bins, capital, debt, and productivity—and embed a transformer in a recursive Krusell–Smith-style solution method. The transformer forecasts the continuation tokenized distribution entering individual Bellman equations. Its neural-network component approximates nonlinear aggregate policy functions, while attention learns how distributional information is aggregated and yields interpretable weights. In Krusell–Smith benchmarks, the transformer rediscovers approximate aggregation, assigning nearly uniform attention across the distribution when aggregate capital is nearly sufficient. In a heterogeneous-firm economy with endogenous default, attention instead concentrates near the default margin and rises during crises for financially constrained firms when their default risk becomes more consequential for aggregate dynamics. Attention therefore serves simultaneously as a solution device and a diagnostic of economic aggregation.

Keywords: Heterogeneous-agent economies; heterogeneous firms, firm dynamics; financial frictions; endogenous default; transformer attention; computational economics; distributional dynamics; neural networks; machine learning.

JEL Codes: C45; C63; C68; D22; E21; E32; E44; G33.

*Federal Reserve Bank of Chicago, 230 South LaSalle Street, Chicago, IL 60604, USA. Email: alessandro.tenzin.villa@gmail.com. Acknowledgments: I thank Christian Hellwig, Giuseppe Lopomo, and Nicolas Werquin, for helpful comments. Disclaimer: The views expressed in this paper do not represent the views of the Chicago Fed or the Federal Reserve System.

1 Introduction

Transformers (Vaswani et al., 2017) are a core architecture behind modern artificial intelligence. They combine neural-network functional approximation with attention, a mechanism that learns how inputs, or tokens, should be linked before forming a prediction. This paper brings transformers to heterogeneous-agent and heterogeneous-firm models. To my knowledge, this is the first application of transformer attention as a recursive state-reduction device in these economies. The economic state in these models is naturally distributional: it can be represented by wealth bins, capital, debt, productivity or balance-sheet cells whose aggregate implications depend on where mass is located in the cross section. I ask whether transformer attention can reduce this high-dimensional distributional state while preserving an interpretable map of which regions matter for aggregates and when. The contribution is twofold.

First, methodologically, I develop a recursive solution method, the Distributional State Transformer (DST), in the spirit of Krusell and Smith (1998) and building on recent machine-learning approaches to heterogeneous-agent models (Fernandez-Villaverde et al., 2023; Han et al., 2025; Kase et al., 2025; Gu et al., 2026). Rather than committing ex ante to a small set of moments, I represent the cross-sectional distribution through economic tokens: wealth-productivity bins in a household economy, or capital-debt-productivity cells in a firm economy. A transformer forecasts the next tokenized distribution, which enters the individual Bellman equation through continuation values. The transformer's neural-network component provides a flexible approximation to equilibrium laws of motion, while its attention layers determine which parts of the distribution are relevant and when.

Second, I apply the method to a heterogeneous-firm economy in which endogenous firm default and transformer attention interact in an economically meaningful way. While recent machine-learning methods for heterogeneous-agent macroeconomics have been applied primarily to household economies, firm balance sheets provide a natural setting for distributional attention. The model builds on Khan and Thomas (2013), adding costly equity issuance and debt priced under endogenous default. These frictions make aggregate dynamics depend on where firms are located relative to financing and default margins in the joint distribution of capital, debt, and productivity. The application delivers two findings. First, when equity issuance is costless and endogenous firm default is shut down, the transformer recovers the Krusell–Smith-like aggregation result obtained by Khan and Thomas (2013). Put in the position of a researcher observing the full firm distribution, the transformer assigns approximately uniform attention across firms . Once costly equity issuance and endogenous default are introduced, attention is no longer flat: it concentrates on firms

near the endogenous default margin, identifying the part of the distribution that becomes relevant when financial frictions are active. Second, transformer attention is state dependent. A negative aggregate productivity shock raises firm default and temporarily shifts attention toward financially fragile firms. A positive shock of the same size lowers default risk, hence does not generate a symmetric reallocation of attention away from the default margin. The aggregate responses mirror this shift in attention. In booms, lower default risk allows investment and hours to expand sharply. In busts, the rise in default absorbs part of the adjustment: investment and hours fall, and attention moves toward the firms whose balance sheets determine the strength and persistence of the downturn. This asymmetry shows that attention serves as a dynamic aggregate-state reduction method: it helps solve the model while also diagnosing which regions of the distribution matter across aggregate states and when.

Related Literature. This paper contributes to two main strands of the literature: (i) computational methods for heterogeneous-agent economies, especially recent machine-learning approaches that seek to improve on the Krusell–Smith approach, and (ii) firm dynamics with financial frictions. A key contribution is to connect these strands through a recursive algorithm that represents the cross-sectional distribution as economic tokens, learns their law of motion with a transformer, and uses the resulting attention weights as both a dimensionality-reduction device and an interpretable diagnostic.

Machine Learning and Computational Methods in Heterogeneous-Agent Models. An increasingly rich literature uses machine-learning tools to solve high-dimensional dynamic economic models, including Scheidegger and Bilionis (2019), Maliar et al. (2021), Azinovic et al. (2022), Valaitis and Villa (2024), Payne et al. (2025), and Gopalakrishna et al. (2026). Within this literature, the closest methodological papers for heterogeneous-agent economies are Fernandez-Villaverde et al. (2023), Han et al. (2025), Kase et al. (2025), and Gu et al. (2026). Fernandez-Villaverde et al. (2023) use neural networks to approximate nonlinear perceived laws of motion in a continuous-time heterogeneous-agent model with financial frictions. I complement these approaches in two ways. First, I use a transformer architecture that produces pairwise attention weights across distributional tokens. These weights are part of the forecasting rule itself: they determine how information from different regions of the cross section is combined before predicting the next state. Attention is therefore both a dimensionality-reduction mechanism and an economic diagnostic: after solving the model, the same weights reveal which regions of the state space matter for equilibrium forecasts and when. Second, I represent the distribution through tokens rather than committing ex ante to

a small set of moments. Each token describes one region of the cross section, the transformer forecasts the next vector of token moments, and the forecast is mapped to nearby library distributions when evaluating continuation values in the Bellman equation.

Han et al. (2025) use neural networks to solve rich heterogeneous-agent economies, while Kase et al. (2025) use neural networks to solve and estimate nonlinear heterogeneous-agent models. Both represent the cross section by a finite population of simulated agents.[1] By contrast, Gu et al. (2026) exploit the continuous-time master-equation formulation, represent the distribution by a high-dimensional finite-dimensional state, and use neural networks to obtain a global solution.

I complement these approaches in two ways. First, I use a transformer architecture that produces attention weights across distributional tokens. These weights are part of the perceived law itself and determine how information from different regions of the cross section is combined when forecasting future states. Attention therefore provides an endogenous diagnostic of economic aggregation. Second, I retain a continuum distribution throughout the solution and update it directly using Young (2010)'s non-stochastic transition, rather than approximating the cross section with a finite simulated population. The transformer forecasts the next tokenized distribution, while a sparse library of economically feasible distributions maps this continuous forecast into continuation values in the Bellman equation.

A final distinction is the domain of the quantitative application. I apply the method to a heterogeneous-firm economy that builds on Khan and Thomas (2013), adding costly equity issuance and defaultable debt under endogenous firm default. These financial frictions create financially fragile regions of the firm distribution, making the joint distribution of capital, debt, and firm productivity directly relevant for aggregate dynamics.

Firm Dynamics and Financial Frictions. A large literature studies firm dynamics with investment frictions and financial frictions, including Hopenhayn (1992), Hopenhayn and Rogerson (1993), Cooley and Quadrini (2001), Gomes (2001), Hennessy and Whited (2005), Clementi and Hopenhayn (2006), Cooper and Haltiwanger (2006), Hennessy and Whited (2007), Jermann and Quadrini (2012), Clementi and Palazzo (2016), Lanteri (2018), Ottonello and Winberry (2020), Villa (2026), and Ippolito et al. (2026). These papers show that firm-level balance sheets, financing constraints, and default risk can shape financing, investment, and aggregate dynamics. The quantitative application in this paper belongs to this tradition: firms choose capital and defaultable debt, face costly equity issuance, and

[1] These papers also relate to simulation-based methods that use machine learning. Duffy and McNelis (2001) introduced neural networks into parameterized-expectations algorithms, Judd et al. (2011) developed simulation-based techniques for high-dimensional dynamic economies, and Valaitis and Villa (2024) use supervised learning to approximate expectation terms in macro-finance models.

default endogenously.

The application builds directly on Khan and Thomas (2008) and Khan and Thomas (2013). Khan and Thomas (2008) study plant-level investment dynamics with idiosyncratic productivity and nonconvex adjustment costs. Khan and Thomas (2013) add financial frictions through collateralized borrowing and aggregate credit shocks, in a production-heterogeneity economy with partial irreversibility. An important lesson from this work is that, in general equilibrium, Krusell–Smith-style forecasting rules can remain highly accurate even in rich firm-dynamics environments.

Relative to Khan and Thomas (2013), I add defaultable debt, costly equity issuance, and endogenous firm default. These financial frictions create financially fragile regions of the firm distribution, making the joint distribution of capital, debt, and firm productivity directly relevant for aggregate dynamics. The results deliver two findings. First, once the model is solved with the DST algorithm, the transformer attention identifies the region near the endogenous default margin as the economically relevant part of the firm distribution. The attention profile is not flat: it is concentrated on firms close to default. This contrasts with the Krusell–Smith benchmark, where attention is close to uniform when mean sufficiency is essentially correct. Second, attention is state dependent. A negative aggregate productivity shock raises default and shifts attention toward financially fragile firms, whereas a positive shock of the same size generates a much weaker response. This asymmetry reveals attention as a dynamic diagnostic of which regions of the distribution matter across aggregate states.

Outline. The remainder of the paper is organized as follows. Section 2 introduces transformer attention through two economic aggregation problems: a two-period economy with mean sufficiency and an infinite-horizon Krusell–Smith economy. Section 3 presents the Distributional-State Transformer algorithm. Section 4 introduces a heterogeneous-firm economy with endogenous default and shows how the equilibrium can be reformulated using the transformer's perceived laws of motion. Section 5 presents the quantitative results. Section 6 concludes.

2 Economic Aggregation and Transformer Attention

This section introduces transformer attention directly through two economic aggregation problems. The first is a two-period economy in which the cross-sectional distribution matters only through mean capital. While this aggregation result is known analytically to the researcher, it is not known to the transformer. Instead, the transformer observes the entire distribution and is trained only to predict next-period aggregate capital. The example

therefore provides a transparent setting to ask whether the attention mechanism discovers that mean capital is the sufficient statistic.

The second example is an infinite-horizon Krusell–Smith economy in which the cross-sectional distribution is endogenous and enters the household value function as an aggregate state. There, the transformer forecasts the next tokenized distribution inside the equilibrium loop, and the forecast is used when evaluating continuation values. Across the two examples, the transformer provides flexible nonlinear approximation, while attention reveals how distributional information is organized before prediction.

2.1 Mean-Sufficient Attention in a Two-Period Economy

I begin with a benchmark in which the cross-sectional distribution matters only through its mean. The environment is deliberately simple, so that the economic aggregation result is known analytically and the attention diagnostic can be evaluated against a sharp benchmark.

Consider a two-period economy with a continuum of households indexed by $i \in [0, 1]$. Household i enters the period with capital k_i, and aggregate capital is

$$K = \int_0^1 k_i \, di.$$

There are no idiosyncratic productivity shocks. A representative firm rents aggregate capital and produces final output according to

$$Y = ZK^{\alpha}, \qquad 0 < \alpha < 1,$$

where Z is aggregate productivity. The competitive return on capital is

$$R_K = \alpha Z K^{\alpha - 1}.$$

Household i chooses next-period capital k_i' to solve

$$\max_{k_i'} \left\{ \log c_i + \beta \log c_i' \right\},$$

subject to

$$c_i + k_i' = R_K k_i, \qquad c_i' = R_K' k_i'.$$

The next-period return R_K' is taken as exogenous to the household and drops out of the saving first-order condition under logarithmic utility. Substituting the constraints into the

objective yields

$$\max_{k_i'} \left\{ \log(R_K k_i - k_i') + \beta \log(R_K' k_i') \right\}.$$

The first-order condition gives the closed-form saving rule

$$k_i' = \kappa R_K k_i, \qquad \kappa \equiv \frac{\beta}{1+\beta}. \tag{1}$$

Thus individual saving is linear in individual capital. Aggregating equation (1) across households gives

$$K' = \int_0^1 k_i' \, di = \kappa R_K \int_0^1 k_i \, di = \kappa R_K K.$$

Using the firm first-order condition,

$$K' = \kappa \alpha Z K^{\alpha}. \tag{2}$$

Hence the entire distribution of capital is summarized by the scalar K. Conditional on K and Z, reallocating capital across households has no effect on aggregate capital accumulation.

The prediction problem. The objective is to approximate the aggregation rule (2) mapping the current aggregate state into next-period aggregate capital. Let X denote a collection of economic inputs describing the aggregate state. The transformer is a parameterized mapping

$$\widehat{y} = \mathcal{T}_\theta(X),$$

where θ denotes the network parameters. In the present example, $\widehat{y} = \widehat{K}'$ although the same architecture can be used to predict multiple aggregate objects.

The network is trained using observations generated from the analytical law

$$K' = \kappa \alpha Z \left(\sum_{j=1}^{J} \mu_j k_j \right)^{\alpha},$$

which serves only to construct the training targets and is not supplied to the model.

Mapping the model to the transformer. The input X consists of a collection of vectors, called *tokens*, where each token represents one economic object. Representing the aggregate state as a collection of tokens allows the transformer to learn mappings over complex distributional states, where tokens may encode capital, debt, productivity, or other characteristics of the cross-sectional distribution.

In this example, I approximate the wealth distribution with J wealth bins. Bin j has representative capital k_j and mass μ_j. Together with aggregate productivity, these define the input sequence

$$X = \{ \underbrace{x_1, \ldots, x_J}_{\text{wealth-bin tokens}} , \quad \underbrace{x_Z}_{\text{aggregate-productivity token}} \}.$$

For $j = 1, \ldots, J$, the wealth-bin token is

$$x_j = \left(\underbrace{k_j, \mu_j}_{\text{wealth-bin state}} \,\middle|\, \underbrace{0}_{\text{aggregate productivity}} \,\middle|\, \underbrace{0}_{\text{token type}} \right),$$

while the aggregate-productivity token is

$$x_Z = \left(\underbrace{0, 0}_{\text{wealth-bin state}} \,\middle|\, \underbrace{Z}_{\text{aggregate productivity}} \,\middle|\, \underbrace{1}_{\text{token type}} \right).$$

Thus each token has dimension $d_x = 4$. The first two entries encode wealth-bin information, the third aggregate productivity, and the last identifies the token type.

Processing the tokenized state. The transformer processes the tokenized state through three successive operations. First, it constructs informative features for each economic object. Second, it learns how information should be exchanged across those objects to form a representation of the aggregate state. Finally, it maps this learned state representation into the desired economic prediction using a flexible nonlinear approximator. Figure 1 summarizes these three stages.

First, an *embedding layer* maps each raw token into a learned feature vector,

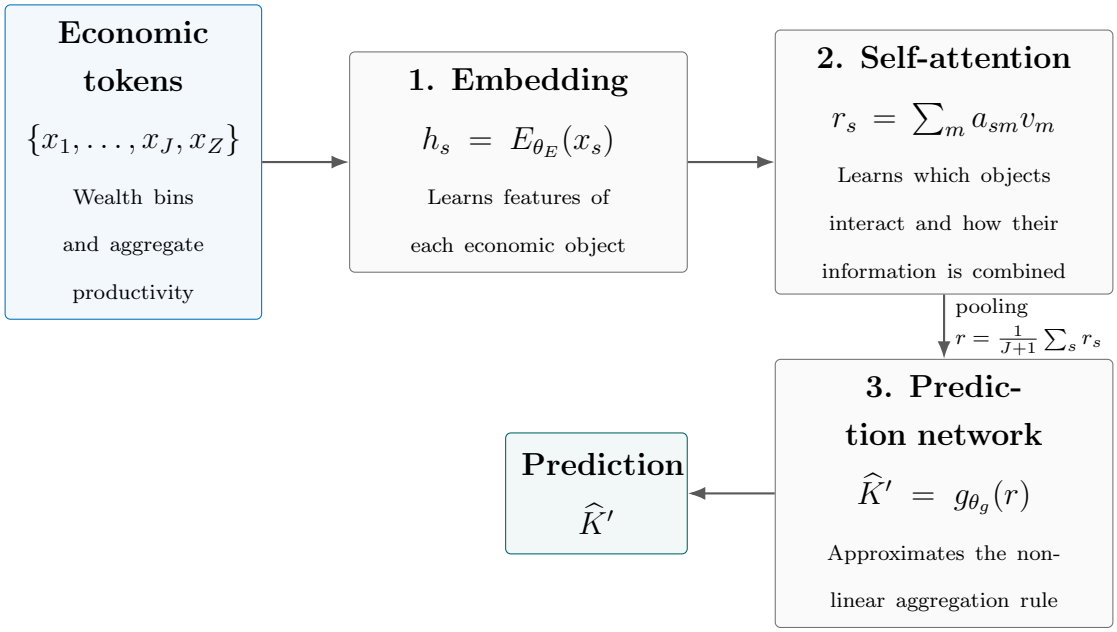
$$h_j = E_{\theta_E}(x_j), \qquad h_Z = E_{\theta_E}(x_Z).$$

The raw wealth-bin token contains only k_j and μ_j, together with placeholders and a type indicator. These coordinates describe the bin, but need not be the most informative features for prediction or for comparing the token with the remaining economic objects. For example, in richer environments a token may contain capital, debt, and productivity, while prediction depends on nonlinear combinations such as leverage or net worth. The embedding therefore constructs a learned representation of each token before interactions across tokens are evaluated. A simple embedding is

$$E_{\theta_E}(x) = \sigma(W_E x + b_E),$$

where the researcher specifies the functional form and embedding dimension, while W_E and b_E are estimated jointly with the remaining network parameters. Thus, the researcher does not specify which transformations of the raw token variables are economically relevant; these features are learned directly from the prediction objective.

Figure 1: Transformer Processing of the Tokenized Economic State



Notes: The embedding layer constructs learned features from each raw economic token. Self-attention then updates each token using information from the remaining wealth-bin and aggregate-productivity tokens. After pooling, a feedforward neural network maps the resulting representation into the prediction of next-period aggregate capital.

Second, a *self-attention layer* allows each embedded token to interact with every other token. In this benchmark, it learns which regions of the wealth distribution are most informative for predicting K', how information should be aggregated across wealth bins, and how the distribution interacts with aggregate productivity. More generally, the attention mechanism learns both which economic objects matter for prediction and how information should flow across them. The output is therefore a collection of context-dependent token representations: each wealth-bin token is represented not in isolation, but conditional on the rest of the distribution and on Z.

Finally, the context-dependent token representations are pooled into a single aggregate representation and passed to a standard feedforward neural network,

$$\widehat{K}' = g_{\theta_g}(r).$$

At this stage, the relevant features have already been constructed and the economically

important interactions identified by the attention layer. The role of the prediction network is therefore to provide a flexible nonlinear approximation of the equilibrium mapping from the learned aggregate-state representation to the desired prediction. All network parameters are estimated jointly to minimize prediction errors.

The complete transformer architecture is summarized in the following box. As mentioned, the embedding h_s is a learned feature representation of token s. The query q_s determines what information token s seeks from the remaining tokens; the key k_m determines how strongly token m matches that query; and the value v_m contains the information that token m contributes. The scalar a_{sm} therefore measures how much token s attends to token m: the row index gives attention and the column index receives attention. The vector r_s is the resulting context-dependent representation of token s. Mean pooling combines these representations into r, which the prediction neural network g_{θ_g} maps into $\widehat{K}'$.

Full self-attention architecture

Let

$$\mathcal{I} = \{1, \ldots, J, Z\}$$

index the J wealth-bin tokens and the aggregate-productivity token. Given $x_s \in \mathbb{R}^{d_x}$, $s \in \mathcal{I}$, the transformer computes

$$\begin{aligned}
h_s &= E_{\theta_E}(x_s) \in \mathbb{R}^d, \\
q_s &= W_Q h_s \in \mathbb{R}^d, \qquad k_s = W_K h_s \in \mathbb{R}^d, \qquad v_s = W_V h_s \in \mathbb{R}^d, \\
e_{sm} &= \frac{q_s^\top k_m}{\sqrt{d}} \in \mathbb{R}, \qquad m \in \mathcal{I}, \\
a_{sm} &= \frac{\exp(e_{sm})}{\sum_{\ell \in \mathcal{I}} \exp(e_{s\ell})} \in \mathbb{R}, \qquad a_{sm} \geq 0, \qquad \sum_{m \in \mathcal{I}} a_{sm} = 1, \\
r_s &= \sum_{m \in \mathcal{I}} a_{sm} v_m \in \mathbb{R}^d, \qquad s \in \mathcal{I}, \\
r &= \frac{1}{J+1} \sum_{s \in \mathcal{I}} r_s \in \mathbb{R}^d, \\
\widehat{K}' &= g_{\theta_g}(r).
\end{aligned}$$

Results. The benchmark gives a precise prediction for attention. Since each household's saving is proportional to k_i, each wealth bin contributes to K' only through its contribution

to mean capital,

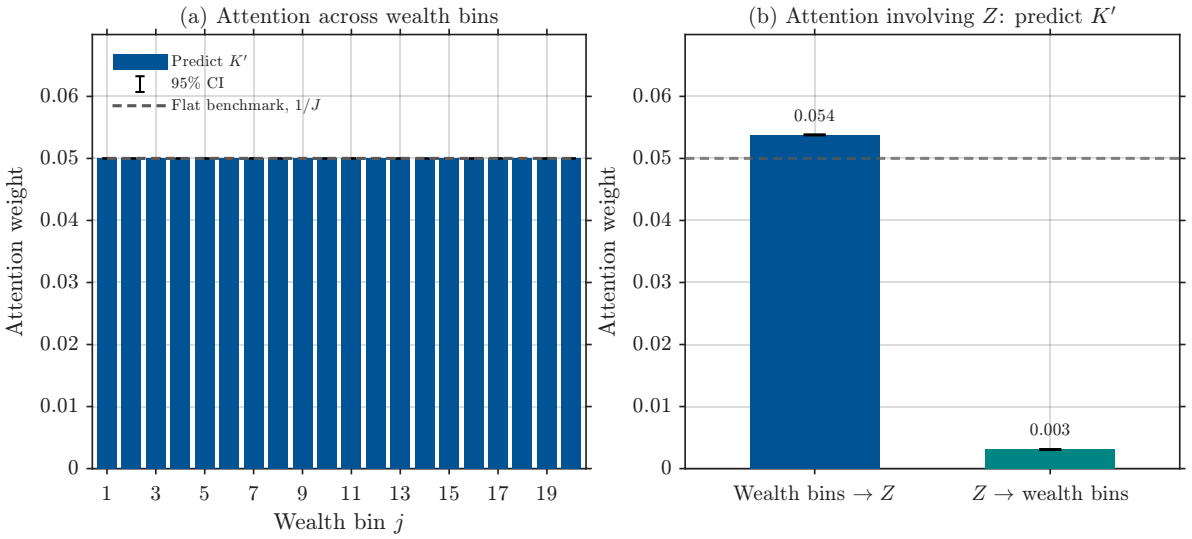
$$K = \sum_{j=1}^{J} \mu_j k_j.$$

No particular region of the distribution is special. The attention profile across wealth-bin tokens should therefore be symmetric.

I simulate 100,000 economies, train on 80,000, and evaluate attention on held-out test economies. The numerical experiments use $J = 20$ wealth bins, $\beta = 0.96$, $\alpha = 0.36$, aggregate productivity $Z \sim U[0.9, 1.1]$, and lognormal bin capital draws. The transformer has one full self-attention layer with internal dimension $d = 128$, a one-hidden-layer prediction head with 64 hidden units, batch size 256, learning rate 10^{-3}, and 80 training epochs. The model fits the analytical law of motion closely, with test $R^2 = 0.99994$.

Figure 2 reports the attention diagnostics. Panel (a) shows that received attention across wealth-bin tokens is essentially flat and tracks the $1/J$ benchmark. This is the attention analogue of mean sufficiency: the model does not single out any region of the wealth distribution when the economic model aggregates through mean capital. In other words, the flat distributional profile in Panel (a) is the key diagnostic: the transformer learns the aggregate law of motion without assigning special importance to any particular wealth bin.

Figure 2: Transformer Attention



Notes: Panel (a) reports received attention across wealth-bin tokens, normalized within wealth bins; the dashed line is $1/J$. Panel (b) reports average attention from wealth-bin queries to Z and from the Z query to wealth-bin tokens. Bars report test-sample averages; error bars denote 95% confidence intervals.

Panel (b) reports attention involving the aggregate-productivity token. Wealth-bin tokens attend to Z, reflecting that the mapping from the distribution to K' depends on ag-

gregate productivity. By contrast, attention from Z back to the wealth bins is small. This asymmetry reflects the two-period structure: Z shifts the aggregate mapping but does not affect the evolution of the wealth distribution itself. As the infinite-horizon example below shows, this asymmetry disappears when aggregate productivity also affects individual policies and hence the transition of the distribution.

2.2 Attention in an Infinite-Horizon Krusell–Smith Economy

I next apply the same transformer architecture to a canonical infinite-horizon Krusell–Smith economy. The purpose is to move from the analytical two-period benchmark, where mean sufficiency holds exactly, to a recursive environment in which the cross-sectional distribution is endogenous. The distribution is an explicit state variable in the household problem. I ask the transformer to forecast the next tokenized distribution and use that forecast inside value-function iteration. The exercise therefore puts the network in the position of a researcher who observes the cross-section and must discover whether a low-dimensional aggregate representation is sufficient.

Time is discrete, $t = 0, 1, \ldots$. Household i enters the period with assets $a_{i,t}$ and idiosyncratic labor productivity $\varepsilon_{i,t} \in \mathcal{E}$. In the quantitative exercise, $\mathcal{E}$ is a seven-state Rouwenhorst approximation to a persistent log-productivity process. Aggregate productivity is $Z_t \in \{Z_B, Z_G\}$. The two shocks follow independent Markov chains, so their joint transition kernel is

$$\Pr(\varepsilon_{i,t+1} = \varepsilon', Z_{t+1} = Z' \mid \varepsilon_{i,t} = \varepsilon, Z_t = Z) = P_\varepsilon(\varepsilon, \varepsilon') P_Z(Z, Z').$$

Let μ_t denote the cross-sectional distribution over (a, ε). Given prices $R(Z, \mu)$ and $w(Z, \mu)$, the household solves

$$V(a, \varepsilon, Z, \mu) = \max_{c, a', n} \left\{ u(c) + \beta \mathbb{E} \left[V(a', \varepsilon', Z', \mu') \mid \varepsilon, Z \right] \right\},$$

subject to

$$c + a' = R(Z, \mu) a + w(Z, \mu) \varepsilon n, \qquad a' \geq \underline{a}, \qquad c \geq 0, \qquad 0 \leq n \leq 1.$$

There is no disutility from labor. Hence the household supplies the full labor endowment,

$$n(a, \varepsilon, Z, \mu) = 1,$$

and the problem reduces to the standard consumption-saving problem with idiosyncratic

labor-income risk.

Aggregate capital and effective labor are

$$K_t = \int a \, d\mu_t(a, \varepsilon), \qquad N_t = \int \varepsilon n_t(a, \varepsilon) \, d\mu_t(a, \varepsilon) = \int \varepsilon \, d\mu_t(a, \varepsilon).$$

A representative firm operates

$$Y_t = Z_t K_t^{\alpha} N_t^{1-\alpha}, \qquad 0 < \alpha < 1,$$

so equilibrium factor prices are

$$R_t = 1 - \delta + \alpha Z_t K_t^{\alpha-1} N_t^{1-\alpha}, \qquad w_t = (1-\alpha) Z_t K_t^{\alpha} N_t^{-\alpha}.$$

A recursive equilibrium is characterized by an asset policy function

$$g_a(a, \varepsilon, Z, \mu),$$

factor-price functions $R(Z, \mu)$ and $w(Z, \mu)$, and a law of motion for the distribution,

$$\mu' = \mathcal{H}(\mu, Z, Z'),$$

such that households optimize, firms optimize, markets clear, and the law of motion induced by individual policies coincides with $\mathcal{H}$.

The usual Krusell–Smith approximation replaces the distribution in the aggregate state by a small set of moments. In the baseline one-moment case,

$$\log K_{t+1} = b_0(Z_t) + b_1(Z_t) \log K_t.$$

Mapping the model to the transformer. The numerical state remains the full joint distribution μ_t over the asset grid and idiosyncratic productivity states. As in the two-period example, the transformer operates on a tokenized representation of this distribution. Partition the asset support into J fixed wealth bins and let

$$m_{j,t} = \left(\bar{a}_{j,t}, \omega_{j,t}, \bar{\varepsilon}_{j,t}, \overline{\varepsilon^2}_{j,t} \right),$$

where $\omega_{j,t}$ is the mass in bin j and the remaining entries are conditional moments. Stacking the distributional content of the J bins gives

$$m_t = \mathrm{vec}\left(m_{1,t}, \ldots, m_{J,t}\right).$$

The transformer input is

$$X_t = \{\underbrace{x_{1,t}, \ldots, x_{J,t}}_{\text{wealth-bin tokens}}, \quad \underbrace{x_{Z,t}}_{\text{aggregate-productivity token}}\},$$

where, for $j = 1, \ldots, J$,

$$x_{j,t} = \left(\underbrace{\bar{a}_{j,t}, \omega_{j,t}, \bar{\varepsilon}_{j,t}, \overline{\varepsilon^2}_{j,t}}_{\text{wealth-bin state } m_{j,t}} \,\middle|\, \underbrace{0}_{\text{aggregate productivity}} \,\middle|\, \underbrace{1,0}_{\text{token type}} \right),$$

and

$$x_{Z,t} = \left(\underbrace{0,0,0,0}_{\text{wealth-bin state}} \,\middle|\, \underbrace{Z_t}_{\text{aggregate productivity}} \,\middle|\, \underbrace{0,1}_{\text{token type}} \right).$$

Thus the collection of wealth-bin tokens provides a low-dimensional representation of the entire cross-sectional distribution, while the underlying distribution μ_t remains the state used to update the economy.

The key difference from the two-period benchmark is that the prediction target is now itself distributional. Rather than forecasting a scalar such as K_{t+1}, the transformer learns a perceived law of motion for the entire tokenized distribution,

$$\widehat{m}_{t+1} = m_t + \mathcal{T}_{\theta,m}(X_t), \tag{3}$$

where $\mathcal{T}_{\theta,m}(X_t)$ predicts the change in the distributional vector. In the implementation, inputs and targets are standardized and the raw forecast is projected onto the economically admissible set; Section 3 gives the general formulation.

Training targets are generated by applying the Young (2010) transition to the current household policy. Aggregate capital is therefore an implication of the predicted distribution rather than a separate transformer target:

$$\widehat{K}_{t+1}^{\mathrm{DST}} = \sum_{j=1}^{J} \widehat{\omega}_{j,t+1} \widehat{\bar{a}}_{j,t+1}. \tag{4}$$

The comparison with the one-moment Krusell–Smith law therefore asks whether a transformer that observes the distribution itself rediscovers approximate aggregation through aggregate capital while looking at the entire tokenized distribution.

Embedding the distributional forecast in the household problem. I briefly describe how the transformer forecast enters the household problem, leaving the construction of the distribution library, the interpolation scheme, and the general equilibrium algorithm to Section 3. The purpose here is only to make clear how a forecast of the future distribution affects individual decisions.

Because μ_{t+1} enters the household continuation value, the transformer's distributional forecast must enter directly into the individual Bellman equation. A Cartesian grid over all coordinates of m_t is infeasible. I therefore store value and policy functions on a sparse library of economically feasible distributions,

$$\mathcal{L} = \{\mu^1, \ldots, \mu^S\}.$$

At a current library node (μ^s, Z), the transformer produces the continuous forecast

$$\widehat{m}'_{sZ} = m(\mu^s) + \mathcal{T}_{\theta,m}(X_{sZ}).$$

Because this forecast will generally lie between stored distributions, it determines a set of nearby library nodes $\mathcal{N}(s, Z)$ and interpolation weights $\{\omega_{sZ,r}\}_{r\in\mathcal{N}(s,Z)}$. The resulting continuation value is

$$\mathcal{C}_{sZ}(a', \varepsilon) = \sum_{Z'} P_Z(Z, Z') \sum_{\varepsilon'} P_\varepsilon(\varepsilon, \varepsilon') \sum_{r\in\mathcal{N}(s,Z)} \omega_{sZ,r} V(a', \varepsilon', Z', \mu^r)\,, \tag{5}$$

so that the household problem at library node μ^s is

$$V(a, \varepsilon, Z, \mu^s) = \max_{a'\geq \underline{a}} \{u(R(Z, \mu^s)a + w(Z, \mu^s)\varepsilon - a') + \beta \mathcal{C}_{sZ}(a', \varepsilon)\}\,. \tag{6}$$

The transformer forecast therefore enters the household problem through the continuation value. The equilibrium iteration follows the logic of Krusell and Smith. Given a fixed transformer—the perceived aggregate law of motion—I solve the household problem and simulate the resulting policies using the non-stochastic Young transition. The simulated distributional transitions provide supervised targets for updating the transformer. I then resolve the household problem under the updated law and repeat until convergence. Section 3 presents the general algorithm in detail, including library construction, interpolation,

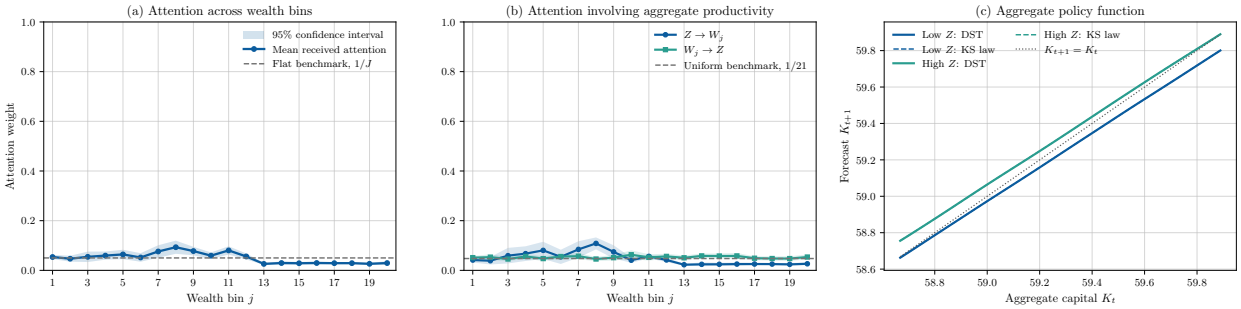
training, and convergence. Appendix A provides implementation details for this example.

Results. The state-contingent Krusell–Smith law and the aggregate-capital forecast implied by the transformer-predicted distribution are evaluated on the same chronological holdout sample. As expected, the linear law already performs extremely well: across the ten converged seeds, its mean out-of-sample R^2 is 0.99996. The aggregate-capital forecast implied by the DST distributional law has mean $R^2 = 1.00000$ to five decimal places, even though aggregate capital is not supplied as an input and the network is trained to predict all 80 distributional coordinates. Thus, when placed in the role of a researcher who observes the distribution rather than its mean, the transformer recovers an aggregate law of motion that is quantitatively indistinguishable from the one-moment Krusell–Smith rule.

Figure 3 reports the corresponding attention diagnostics and aggregate policy functions. The contrast with the two-period benchmark is useful. In the two-period economy, the distribution is exogenous and only its mean matters exactly, so the transformer assigns essentially flat received attention across wealth bins. In the recursive economy, the distribution is endogenous and enters the continuation value. Attention is broadly dispersed, but it is not exactly uniform: the transformer places greater weight on a middle-to-lower region of the wealth distribution and less weight on the sparsely populated upper tail. The important aggregation result is not literal uniformity of every attention weight. It is that this nonuniform internal representation nevertheless produces aggregate policy functions almost identical to those implied by the Krusell–Smith law.

Panel (b) isolates directional attention between the wealth tokens and aggregate productivity. Attention from the Z query to wealth-bin keys, $Z \to W_j$, is concentrated around bins 5–9 and falls sharply in the upper tail. Attention from wealth-bin queries to the Z key, $W_j \to Z$, is substantially more even. These two directions are distinct objects: the former asks which regions of the distribution enter the update of the Z-token representation, whereas the latter asks how strongly each distributional token conditions on the aggregate state. Panel (c) makes the aggregation comparison directly. For both productivity states, the transformer-implied forecast of K_{t+1} closely tracks the state-contingent Krusell–Smith law over the common support of simulated aggregate capital.

Figure 3: Transformer Attention and Aggregate Policy



Notes: Panels (a) and (b) report attention diagnostics for the infinite-horizon Krusell–Smith benchmark. Panel (a) shows received attention across wealth bins; Panel (b) shows directional attention between wealth bins and aggregate productivity Z. Dashed horizontal lines denote the uniform attention benchmarks. Lines are cross-seed means and shaded regions are 95% confidence intervals. Panel (c) compares aggregate-capital policies implied by the transformer-predicted distribution (solid) with the corresponding Krusell–Smith laws (dashed); the dotted line is $K_{t+1} = K_t$.

The directional attention involving aggregate productivity also differs across the two benchmarks. In Panel (b) of Figure 2, attention is strongly asymmetric: wealth-bin tokens attend to Z, whereas attention from Z back to the wealth bins is negligible. In the two-period economy, Z acts primarily as an aggregate shifter: conditional on mean capital, it does not alter how the wealth distribution is aggregated to predict K'. In Panel (b) of Figure 3, by contrast, attention operates in both directions. In the recursive economy, Z_t affects household saving policies and therefore the transition of the entire wealth distribution. The relevance of a given region of the distribution for predicting μ_{t+1} can consequently depend on the aggregate state, generating attention in both directions between Z_t and the wealth-bin tokens.

3 The Distributional State Transformer

The examples in Section 2 show that attention can recover economically meaningful aggregation patterns. This section turns that idea into a general solution method. The *Distributional State Transformer* (DST) is an iterative algorithm for recursive heterogeneous-agent economies in which the cross-sectional distribution is part of the aggregate state. The method represents the distribution and the aggregate environment as tokens, learns a perceived law for the next distribution from model-consistent simulations, and inserts that law directly into the Bellman equation.

3.1 Recursive Economy and Tokenized State

Let $a \in \mathcal{X}$ denote an individual state and let $z_t \in \mathcal{Z}$ collect the aggregate shocks. The cross-sectional distribution of individual states at date t is denoted by μ_t. The recursive aggregate state is

$$\Omega_t = (\mu_t, z_t)\,. \tag{7}$$

Given individual policy rules g, the model implies the distributional transition

$$\mu_{t+1} = \Gamma\left(\mu_t, z_t, z_{t+1}; g\right), \tag{8}$$

where Γ is the transition operator induced by individual policy rules, idiosyncratic shocks, and the transition of the aggregate shocks from z_t to z_{t+1}.

The purpose of the DST is to approximate the equilibrium transition of this distribution without reducing it to a small, prespecified set of aggregate moments. To do so, the method represents the current aggregate state as a fixed-length sequence of economic tokens,

$$X_t = \tau(\Omega_t) = \{x_{1,t}, \ldots, x_{J,t}\}\,, \qquad x_{j,t} \in \mathbb{R}^{d_x}, \tag{9}$$

where τ denotes the tokenization map and J is the fixed number of tokens.

A token is an economic object written in numerical format. Distribution tokens represent regions of the cross section, such as wealth bins or firm balance-sheet cells. For example, in the two-period economy of Section 2.1, each distribution token represents a wealth bin with its capital and mass, while aggregate productivity is included as a separate token. In the recursive Krusell–Smith economy of Section 2.2, each wealth-bin token contains the bin mass and conditional moments of assets and idiosyncratic productivity.

Tokens may represent also aggregate shocks, prices, policy variables, or structural parameters if the purpose is also to estimate the model, as in Kase et al. (2025). A generic token has the form

$$x_{j,t} = (\text{economic content} \mid \text{placeholders} \mid \text{type indicators})\,. \tag{10}$$

Placeholders allow economically different objects to have the same numerical dimension, while the type indicators identify the economic role of each token.

Let $m_{j,t}$ denote the economic content of distribution token j. Depending on the application, $m_{j,t}$ may contain a cell mass, mass-weighted asset holdings, balance-sheet positions, productivity moments, or other statistics describing that region of the distribution. Stacking

these contents gives

$$m_t \equiv \text{vec}\left(m_{1,t}, \ldots, m_{J,t}\right). \tag{11}$$

The vector m_t is the tokenized representation of μ_t whose transition is learned by the transformer. It is not restricted to a hand-chosen aggregate statistic such as mean wealth or aggregate capital. Aggregate moments can instead be computed from m_t.

3.2 The Perceived Distributional Law

The transformer maps the current tokenized state X_t into a perceived law of motion for the distribution. Because distributional states are persistent, the network predicts the change in the distributional vector,

$$\Delta m_{t+1} \equiv m_{t+1} - m_t, \tag{12}$$

rather than its next-period level. This residual formulation asks the network to learn the typically smaller and less variable adjustment to the current distribution. The resulting forecast is

$$\widehat{m}_{t+1} = m_t + \mathcal{T}_{\theta,m}(X_t), \tag{13}$$

where $\mathcal{T}_{\theta,m}(X_t)$ denotes the transformer prediction of Δm_{t+1}.

Some applications also require auxiliary equilibrium objects inside the recursive problem. Let ψ_t collect these objects, with prediction

$$\widehat{\psi}_t = \mathcal{T}_{\theta,\psi}(X_t). \tag{14}$$

For example, in the heterogeneous-firm economy of Section 4, it is convenient to track the log marginal utility as an auxiliary target,

$$\psi_t = \ell_t \equiv \log U_C(C_t),$$

from which consumption, wages, and stochastic discount factors are recovered.

Training targets are generated by applying the distributional transition operator to the current distribution under the current policy functions. The parameters θ of the transformer are found to minimize the following loss

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{T}_{\text{tr}}|} \sum_{t \in \mathcal{T}_{\text{tr}}} \left[\left\| \Delta m_{t+1} - \mathcal{T}_{\theta,m}(X_t) \right\|_2^2 + \lambda_\psi \left\| \psi_t - \mathcal{T}_{\theta,\psi}(X_t) \right\|_2^2 \right], \tag{15}$$

where $\lambda_\psi = 0$ when no auxiliary target is used. Inputs and targets are standardized during training and transformed back into economic units before entering the recursive problem.

3.3 Sparse Library of Distributional States

A Cartesian grid over all coordinates of the tokenized distribution is infeasible, as even a small number of grid points along each coordinate would generate an exponentially large aggregate state space. A complementary approach is to solve the model directly over the ergodic set generated by stochastic simulation. For example, Valaitis and Villa (2024) combine stochastic simulation with parameterized expectations in a representative-agent setting. Han et al. (2025) solve heterogeneous-agent models by training neural networks over simulated equilibrium trajectories, while Kase et al. (2025) approximate the distribution with a finite simulated population and train over the resulting simulated state space.

Here, instead, I retain complete distributions over the underlying individual-state grid and use the non-stochastic simulation method of Young (2010), which preserves the continuum-agent representation. The remaining challenge is to approximate equilibrium objects over the resulting high-dimensional distribution space. I do so by replacing the Cartesian grid with a sparse library of feasible distributions,

$$\mathcal{L} = \left\{\mu^1, \ldots, \mu^S\right\}, \tag{16}$$

where each node μ^s is an entire cross-sectional distribution over the underlying individual-state grid. The transformer operates on the corresponding tokenized representation and learns to interpolate across distributions. This sparse representation creates two challenges: selecting a small set of distributions that adequately covers the relevant region of the aggregate state space, and interpolating equilibrium objects between sparse library nodes.

Selecting the library nodes. The library $\mathcal{L}$ may be fixed at initialization or enriched during the outer fixed-point iteration with newly simulated distributions that expand its coverage of the aggregate state space.

To construct the initial library, I first generate a candidate set of feasible distributions,

$$\mathcal{C}_{\text{sim}} = \{\nu^1, \ldots, \nu^N\}, \qquad N \geq S.$$

These candidates are obtained from a bootstrap simulation under a preliminary perceived law of motion. In the heterogeneous-firm application of Section 5, I initialize this law using a low-dimensional Krusell–Smith approximation based on a small set of aggregate moments. The candidate set may be augmented with anchor distributions,

$$\mathcal{C} = \mathcal{C}_{\text{sim}} \cup \mathcal{C}_{\text{anc}},$$

where the anchors are chosen to improve the coverage of economically relevant regions of the state space that may be visited only rarely during the bootstrap simulation. The library $\mathcal{L} = \{\mu^1, \ldots, \mu^S\}$ is then selected as a space-filling subset of $\mathcal{C}$.

To compare candidate distributions, let

$$m(\nu) = \mathrm{vec}\big(m_1(\nu), \ldots, m_J(\nu)\big)$$

denote the tokenized representation of ν. For example, in the heterogeneous-firm application, token j corresponds to a capital–debt–productivity cell $\mathcal{B}_j$ and contains

$$m_j(\nu) = \left(\bar{k}_j(\nu), \omega_j(\nu), \bar{b}_j(\nu), \bar{\varepsilon}_j(\nu)\right), \qquad \omega_j(\nu) = \int_{\mathcal{B}_j} d\nu,$$

where the remaining entries are conditional means within the cell.[2] For distance calculations, I use the corresponding mass-weighted moments,

$$\widetilde{m}_j(\nu) = \left(\omega_j(\nu)\bar{k}_j(\nu), \omega_j(\nu), \omega_j(\nu)\bar{b}_j(\nu), \omega_j(\nu)\bar{\varepsilon}_j(\nu)\right),$$

which converge continuously to zero as the mass of a cell vanishes. Let

$$\widetilde{m}(\nu) = \mathrm{vec}\big(\widetilde{m}_1(\nu), \ldots, \widetilde{m}_J(\nu)\big).$$

The coordinates used for library selection may also include aggregate moments that are particularly important for continuation values and prices. In the firm application, these are aggregate capital, aggregate firm debt, and leverage:

$$K(\nu) = \int k \, d\nu, \qquad B(\nu) = \int b \, d\nu, \qquad \ell(\nu) = \frac{B(\nu)}{K(\nu)}.$$

After standardizing each coordinate across the candidate set, define

$$c(\nu) = \left[\frac{\widetilde{m}(\nu) - \bar{m}}{s_m}, \lambda_K \frac{K(\nu) - \bar{K}}{s_K}, \lambda_B \frac{B(\nu) - \bar{B}}{s_B}, \lambda_\ell \frac{\ell(\nu) - \bar{\ell}}{s_\ell}\right], \tag{17}$$

where $\lambda_K, \lambda_B, \lambda_\ell$ determine the relative weight placed on aggregate capital, debt, and leverage.

[2] Conditional moments are set to zero when $\omega_j(\nu) = 0$.

The S library nodes are selected from $\mathcal{C}$ using a farthest-point rule. Define the distance

$$d(\nu, \nu') = \left[\frac{1}{d_c} \left\| c(\nu) - c(\nu') \right\|_2^2 \right]^{1/2},$$

where d_c is the dimension of $c(\nu)$. The first library node is the candidate closest to the center of the candidate cloud,

$$\mu^1 \in \arg\min_{\nu \in \mathcal{C}} \left\| c(\nu) - \bar{c} \right\|_2^2, \qquad \bar{c} = \frac{1}{|\mathcal{C}|} \sum_{\nu \in \mathcal{C}} c(\nu).$$

Given the first $r-1$ selected nodes, $\mathcal{L}_{r-1} = \{\mu^1, \ldots, \mu^{r-1}\}$, the next node is

$$\mu^r \in \arg\max_{\nu \in \mathcal{C}} \min_{\mu \in \mathcal{L}_{r-1}} d(\nu, \mu), \qquad r = 2, \ldots, S. \tag{18}$$

Thus, each new node is the candidate farthest from the portion of the state space already represented in the library. Anchor distributions are selected when they expand the library's coverage, but need not be retained when they are redundant relative to the other candidates. If the library is enriched during the outer iteration, the same distance criterion is applied to newly simulated distributions.

Interpolating over the sparse library. For a current state, the transformer produces a continuous forecast $\widehat{m}'$. Because $\widehat{m}'$ will generally lie between library nodes, the method approximates equilibrium objects using nearby stored distributions. The distance between the forecast and library node r is

$$d_r(\widehat{m}') = \left[\frac{1}{d_c} \left\| \Xi(\widehat{m}') - c(\mu^r) \right\|_2^2 \right]^{1/2}, \tag{19}$$

where d_c is the dimension of the standardized coordinate vector. Let $\mathcal{N}(\widehat{m}')$ denote the set of nearest library nodes. In the absence of an exact match, the interpolation weights are

$$\omega_r(\widehat{m}') = \frac{d_r(\widehat{m}')^{-2}}{\sum_{\ell \in \mathcal{N}(\widehat{m}')} d_\ell(\widehat{m}')^{-2}}, \qquad r \in \mathcal{N}(\widehat{m}'), \tag{20}$$

so that

$$\sum_{r \in \mathcal{N}(\widehat{m}')} \omega_r(\widehat{m}') = 1.$$

An exact match receives unit weight. For any equilibrium object F stored at the library nodes, its value at the predicted distribution is approximated by

$$\widehat{F}(\widehat{m}') = \sum_{r \in \mathcal{N}(\widehat{m}')} \omega_r(\widehat{m}') F(\mu^r). \tag{21}$$

Thus, the continuous transformer forecast determines which stored distributions enter the approximation and how much weight each receives. The same interpolation weights can be applied to continuation values, policies, default probabilities, and other equilibrium objects stored on the library.

3.4 Embedding the Law in the Bellman Equation

The perceived distributional law enters the recursive problem. At library node μ^s and aggregate state z, the transformer produces the distributional forecast

$$\widehat{m}'_{sz} = m(\mu^s) + \mathcal{T}_{\theta,m}(X_{sz}).$$

As described in Section 3.3, this forecast determines a set of nearby library nodes $\mathcal{N}(s,z)$ and interpolation weights $\{\omega_{sz,r}\}_{r \in \mathcal{N}(s,z)}$.

Let a denote the current individual state and let α denote an individual choice. Suppressing application-specific constraints, the continuation value associated with (a,α) is

$$\mathcal{C}_{sz}(a,\alpha) = \sum_{z'} P_z(z,z') \sum_{r \in \mathcal{N}(s,z)} \omega_{sz,r}\, \mathbb{E}\left[M_{sz,rz'} V_r(a',z') \mid a,\alpha,z,z'\right]. \tag{22}$$

Here $V_r(a',z')$ is the value function stored at library node μ^r, and the expectation integrates over the transition of the individual state. The stochastic discount factor $M_{sz,rz'}$ is determined by the economic model and may depend on auxiliary objects predicted by the transformer.

The Bellman equation at (μ^s, z) is then

$$V_s(a,z) = \max_{\alpha \in \mathcal{A}(a,z,\mu^s)} \left\{ u\left(a,\alpha,z,\mu^s;\widehat{\psi}_{sz}\right) + \mathcal{C}_{sz}(a,\alpha) \right\}. \tag{23}$$

The learned law therefore affects current decisions. through changes in the predicted library nodes and interpolation weights entering the continuation value.

3.5 Equilibrium Iteration

The equilibrium is a fixed point between the perceived distributional law and the policy functions, as in Krusell and Smith (1998). The sparse library $\mathcal{L} = \{\mu^1, \ldots, \mu^S\}$ is constructed before the equilibrium iteration and then held fixed.

At outer iteration k, the transformer is trained on transitions generated by the previous policy functions. It is then evaluated at each current aggregate state (μ^s, z), producing the forecast $\widehat{m}'_{sz}$, the neighboring library nodes $\mathcal{N}(s, z)$, the interpolation weights $\omega_{sz,r}$, and any auxiliary predictions $\widehat{\psi}_{sz}$. Holding these objects fixed, the individual Bellman equation is solved to convergence. The resulting policy functions are then simulated using the non-stochastic distributional transition of Young (2010), generating the training sample for the next outer iteration. Computationally, the transformer training step is parallelized on GPU, while the Bellman solves are parallelized across library nodes and aggregate states on CPU.[3]

Algorithm 1 DST Equilibrium Iteration

1: Construct a bootstrap policy under a simple perceived law of motion.
2: Simulate the policy using the non-stochastic distributional transition operator of Young (2010).
3: Select and fix the sparse library $\mathcal{L} = \{\mu^1, \ldots, \mu^S\}$.
4: Initialize value and policy functions at all library nodes.
5: Construct the initial training sample $\{X_t, \Delta m_{t+1}, \psi_t\}_{t \in \mathcal{I}_{\text{tr}}}$.
6: **for** outer iteration $k = 1, 2, \ldots, K_{\max}$ **do**
7: Train or fine-tune $\mathcal{T}_{\theta^{(k)}}$ on the current sample. ▷ **Parallelized on GPU**
8: Evaluate $\mathcal{T}_{\theta^{(k)}}$ at every (μ^s, z) and compute $\widehat{m}'_{sz}$, $\mathcal{N}(s, z)$, $\omega_{sz,r}$, and $\widehat{\psi}_{sz}$.
9: Solve the individual problem at all library nodes. ▷ **Parallelized across nodes and aggregate states on CPU**
10: Simulate the resulting policy functions using the non-stochastic distributional transition operator.
11: Update the training sample with the simulated transitions.
12: Compute the policy change $\Delta_g^{(k)}$ and the perceived-law change $\Delta_{\mathcal{T}}^{(k)}$.
13: **if** $\Delta_g^{(k)} < \tau_g$ and $\Delta_{\mathcal{T}}^{(k)} < \tau_{\mathcal{T}}$ **then**
14: Stop.
15: **end if**
16: **end for**

The policy criterion $\Delta_g^{(k)}$ measures the change in policy functions across successive outer iterations on the fixed library. The law criterion $\Delta_{\mathcal{T}}^{(k)}$ measures the corresponding change in

[3] Transformer training is implemented in PyTorch and parallelized on GPU, using CUDA or Apple's Metal backend; see Vaswani et al. (2017) for the parallel attention architecture and Paszke et al. (2019) for the automatic-differentiation framework used in the implementation.

transformer forecasts at the same states.

3.6 Attention Diagnostics

The perceived law in Section 3.2 is produced by repeatedly combining the input tokens through self-attention. The same attention weights used to construct the transformer outputs can therefore be inspected to study how the network organizes the economic state.

For observation t, layer ℓ, and head h, let

$$A_t^{\ell h} = \left(a_{jm,t}^{\ell h}\right)_{j,m=1}^{J}, \qquad \sum_{m=1}^{J} a_{jm,t}^{\ell h} = 1. \tag{24}$$

The row index j denotes the query token whose representation is being updated, while the column index m denotes the key-value token used in that update. Thus, $a_{jm,t}^{\ell h}$ measures the weight placed on token m when updating token j in layer ℓ and head h.

To summarize attention across test observations, layers, and heads, define the average attention matrix

$$\bar{A} = (\bar{a}_{jm})_{j,m=1}^{J}, \qquad \bar{a}_{jm} = \frac{1}{|\mathcal{I}_{\text{te}}|LH} \sum_{t \in \mathcal{I}_{\text{te}}} \sum_{\ell=1}^{L} \sum_{h=1}^{H} a_{jm,t}^{\ell h}. \tag{25}$$

I use two attention diagnostics. The first diagnostic measures *received attention*. For token m in a group of economic tokens $\mathcal{G}$, define

$$\text{Rec}_m(\mathcal{G}) = \frac{\sum_{j \in \mathcal{Q}} \bar{a}_{jm}}{\sum_{m' \in \mathcal{G}} \sum_{j \in \mathcal{Q}} \bar{a}_{jm'}}, \qquad m \in \mathcal{G}, \tag{26}$$

where $\mathcal{Q}$ is the set of query tokens included in the diagnostic. This statistic measures how the attention received by group $\mathcal{G}$ is distributed across its members. It satisfies $\sum_{m \in \mathcal{G}} \text{Rec}_m(\mathcal{G}) = 1$. If the $J_{\mathcal{G}}$ tokens in the group enter symmetrically, the corresponding benchmark is $1/J_{\mathcal{G}}$.

The second diagnostic measures *directional group attention*. For two token groups $\mathcal{A}$ and $\mathcal{B}$, define

$$\text{Attn}(\mathcal{A} \to \mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} \sum_{m \in \mathcal{B}} \bar{a}_{jm}. \tag{27}$$

This statistic measures the average share of attention that query tokens in $\mathcal{A}$ assign to key-value tokens in $\mathcal{B}$. Because the first group supplies the queries and the second supplies the

key-value tokens,

$$\text{Attn}(\mathcal{A} \to \mathcal{B}) \neq \text{Attn}(\mathcal{B} \to \mathcal{A})$$

in general. In the heterogeneous-firm application that follows, these attention measures show how the transformer combines regions of the distribution associated with different levels of capital, debt, productivity, and default risk. In this sense, attention could be interpreted as a description of the network's learned aggregation rule.

4 Heterogeneous Firms and Endogenous Default

I next apply the DST to a heterogeneous-firm economy with capital and debt. The environment builds on the production-heterogeneity framework of Khan and Thomas (2013): firms differ in idiosyncratic productivity, choose physical capital subject to partial irreversibility, face borrowing constraints, and operate under aggregate productivity shocks. I extend that benchmark by adding costly equity issuance and endogenous firm default. This makes the joint distribution of capital, debt, and productivity relevant for aggregate dynamics. Appendix B gives the computational details for this application.

4.1 Model

Environment. Time is discrete and one model period is one year. Aggregate productivity Z_t follows an AR(1) process in logs, i.e. $\log Z_t = \rho_Z \log Z_{t-1} + \sigma_Z \eta_t^Z$, with $\eta_t^Z \sim \mathcal{N}(0,1)$. A continuum of firms is indexed by idiosyncratic productivity ε_t, physical capital k_t, and one-period debt b_t. Idiosyncratic productivity ε_t also follows an AR(1) process in logs, i.e. $\log \varepsilon_t = \rho_\varepsilon \log \varepsilon_{t-1} + \sigma_\varepsilon \eta_t^\varepsilon$, with $\eta_t^\varepsilon \sim \mathcal{N}(0,1)$. The production technology of an active firm is

$$y_t(\varepsilon, k, n; Z) = Z_t \varepsilon_t k_t^\alpha n_t^\nu, \qquad \alpha + \nu < 1. \tag{28}$$

Operating profits, net of a fixed operating cost f, are

$$\pi_t(\varepsilon, k; Z, w) = Z_t \varepsilon_t k_t^\alpha n_t^\nu - w_t n_t - f. \tag{29}$$

Given the real wage w_t, labor demand is given by the marginal product of labor:

$$n_t(\varepsilon, k; Z, w) = \left(\frac{\nu Z_t \varepsilon_t k_t^\alpha}{w_t} \right)^{1/(1-\nu)}. \tag{30}$$

The representative household owns both firms and competitive lenders. Its preferences are

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left[\frac{C_t^{1-\gamma}}{1-\gamma} - \chi N_t \right], \tag{31}$$

with the logarithmic case obtained as $\gamma \to 1$. Its budget constraint is

$$C_t = w_t N_t + \int \mathcal{D}_t(\varepsilon, k, b)\, d\mu_t(\varepsilon, k, b) + \Pi_t^L, \tag{32}$$

where μ_t denotes the time-t distribution of firms over capital, debt, and idiosyncratic productivity, $\mathcal{D}_t(\varepsilon, k, b)$ denotes the net payout to shareholders by a firm in state (ε, k, b), and Π_t^L denotes aggregate profits from lending. Because lenders are perfectly competitive, debt prices satisfy their zero-profit condition, and hence $\Pi_t^L = 0$ in equilibrium.

The representative household's labor-supply first-order condition is

$$w_t = \chi C_t^{\gamma}, \tag{33}$$

so aggregate consumption determines the wage. Because the household owns both firms and lenders, the stochastic discount factor relevant for their intertemporal payoffs is

$$M_{t,t+1} = \beta \left(\frac{C_{t+1}}{C_t} \right)^{-\gamma}. \tag{34}$$

Firm Finance and Default. A firm enters the period with idiosyncratic productivity, capital, and debt (ε, k, b). Conditional on continuing through the period, it operates, repays current debt b, chooses next-period capital k', and issues one-period debt b'. Default is modeled as part of the transition into the next period: after the continuation value is computed, equity holders compare it with the value of default. New debt sells at price $q_t(\varepsilon, k', b')$. Let net investment be

$$i(k, k') \equiv k' - (1-\delta)k. \tag{35}$$

Following Khan and Thomas (2013), capital is subject to partial irreversibility. The real resource cost of investing from k to k' is

$$\mathcal{I}(k, k') = \begin{cases} i(k, k'), & i(k, k') \geq 0, \\ \theta_k i(k, k'), & i(k, k') < 0, \end{cases} \tag{36}$$

with $\theta_k < 1$. The payout before external-equity costs is therefore

$$d_t = \pi_t(\varepsilon, k; Z, w) + q_t(\varepsilon, k', b')b' - b - \mathcal{I}(k, k'). \tag{37}$$

Debt is subject to a borrowing constraint,

$$b' \leq \bar{b}(k') \equiv \theta_b \theta_k k'. \tag{38}$$

I keep this constraint to preserve comparability with the debt-capacity margin in Khan and Thomas (2013). Economically, it captures borrowing-base rules, loan-size limits, and covenant restrictions that cap debt issuance ex ante. In the baseline calibration with endogenous default, the constraint is effectively slack: the stationary share of firms at or near the borrowing limit is numerically zero. Debt pricing disciplines leverage before the borrowing limit becomes active, because firms approaching states with high default risk face sharply lower bond prices.

If the payout in (37) is negative, the firm raises external equity and pays a quadratic issuance cost. Equity holders receive

$$\mathcal{D}(d_t) = \begin{cases} d_t, & d_t \geq 0, \\ d_t - \dfrac{\phi_e}{2}(-d_t)^2, & d_t < 0. \end{cases} \tag{39}$$

This friction makes internal finance valuable and gives leverage a role beyond the borrowing constraint.

Let $V_t^c(\varepsilon, k, b)$ denote the value of continuing, after optimizing over (k', b'), and let V^d denote the value of default. I normalize $V^d = 0$. Equity holders default whenever the value of continuing is below the value of default. Hence,

$$V_t(\varepsilon, k, b) = \max\left\{V_t^c(\varepsilon, k, b), V^d\right\}, \tag{40}$$

and the default rule is

$$\mathbf{1}_t^d(\varepsilon, k, b) = \mathbf{1}\left\{V_t^c(\varepsilon, k, b) < V^d\right\}. \tag{41}$$

In the numerical implementation, I use a smooth approximation to the default rule. Appendix C describes the approximation and shows that it is quantitatively close to the hard-threshold rule.

Defaulting firms exit and are replaced by entrants so that the mass of firms remains constant. Entrants start with capital $k^e = K^{ss}$, where K^{ss} is the stationary aggregate capital stock of the deterministic economy, zero debt, and idiosyncratic productivity drawn from the

stationary idiosyncratic distribution. Entrants do not inherit the capital of exiting firms; the difference between the post-exit entrant distribution and undepreciated incumbent capital is accounted for through aggregate investment. In the baseline calibration, debt recovery is set to zero. This keeps the resource accounting transparent and lets default risk show up directly in debt prices.

A continuum of identical competitive lenders price debt using the household stochastic discount factor. For $b' > 0$, the one-period debt price satisfies

$$q_t(\varepsilon, k', b') = \mathbb{E}_t \left[M_{t,t+1} \left(1 - p^d_{t+1}(\varepsilon_{t+1}, k', b') \right) \right], \tag{42}$$

where recovery in default is zero. The expectation in (42) is over next-period aggregate productivity, idiosyncratic productivity, and the transformer-implied distributional transition described below. Thus default pricing is one of the channels through which the forecasted aggregate distribution affects current firm decisions.

The continuing-firm Bellman equation is

$$V^c(\varepsilon, k, b; \Omega_t) = \max_{k', b'} \left\{ \mathcal{D}\big(d_t(\varepsilon, k, b, k', b'; \Omega_t)\big) + \mathbb{E}_t \left[M_{t,t+1} V(\varepsilon', k', b'; \Omega_{t+1}) \right] \right\}, \tag{43}$$

subject to (38). The object V on the right-hand side is the default-adjusted inclusive value in (40). The default-adjusted value summarizes the transition decision: firms with low continuation value exit and are replaced in the next-period distribution, while continuing firms carry their chosen (k', b') forward.

Aggregation and Market Clearing. Recall that $\mu_t(\varepsilon, k, b)$ denote the cross-sectional distribution of active firms. Aggregate capital and debt are

$$K_t = \int k \, d\mu_t, \qquad B_t = \int b \, d\mu_t. \tag{44}$$

Given policies, default probabilities, and the wage, output and labor are obtained by integrating (28) and (30). Aggregate capital is computed from the distributional transition, including replacement of defaulting firms,

$$K_{t+1} = \int k \, d\mu_{t+1}, \qquad I_t \equiv K_{t+1} - (1 - \delta) K_t. \tag{45}$$

The resource constraint of the economy is

$$C_t + I_t + \int \Psi(k, k')\, d\mu_t + \int f\, d\mu_t + \int \Phi_e(d_t)\, d\mu_t = Y_t, \tag{46}$$

where $\Psi(k, k') = (1 - \theta_k)[-i(k, k')]_+$ is the resource cost from partial irreversibility and $\Phi_e(d_t) = \frac{\phi_e}{2}[-d_t]_+^2$. The equilibrium wage is the fixed point that satisfies (33) and (46). In the simulation step, this wage is solved exactly from the current distribution and firm policies. The distribution evolves by applying the firm policy rules, the idiosyncratic transition matrix, the default probabilities, and the entrant distribution.

4.2 Tokenizing the Firm Distribution

The cross-sectional distribution μ_t is defined over idiosyncratic productivity, capital, and debt. Passing every point of the numerical state grid to the transformer would be computationally costly. I therefore partition the firm state space into capital bins, debt bins, and productivity groups. The baseline uses 15 capital bins, 4 debt bins, and 7 idiosyncratic-productivity groups, yielding

$$J = 15 \times 4 \times 7 = 420 \tag{47}$$

distributional cells.

This partition is used only to represent the distribution to the transformer. It need not coincide with the finer capital, debt, and productivity grids used to solve the firm problem and update the distribution. Each token aggregates all numerical grid points belonging to the same capital–debt–productivity cell.

For cell j, let

$$\mu_{j,t} = \int_{\mathcal{B}_j} d\mu_t(\varepsilon, k, b) \tag{48}$$

denote its mass, where $\mathcal{B}_j$ is the corresponding region of the firm state space. Let $\bar{k}_{j,t}$, $\bar{b}_{j,t}$, and $\bar{\varepsilon}_{j,t}$ denote the conditional means within that cell. I represent its economic content using mass-weighted moments,

$$m_{j,t} = \left(\mu_{j,t}\bar{k}_{j,t}, \mu_{j,t}, \mu_{j,t}\bar{b}_{j,t}, \mu_{j,t}\bar{\varepsilon}_{j,t}\right). \tag{49}$$

This representation is numerically convenient for cells with little or no mass. As $\mu_{j,t}$ approaches zero, all components of $m_{j,t}$ also approach zero, whereas conditional means alone can vary sharply or require arbitrary values for empty cells. Mass-weighting also makes

aggregate quantities additive across tokens. For example,

$$K_t = \sum_{j=1}^{J} \mu_{j,t} \bar{k}_{j,t}, \qquad B_t = \sum_{j=1}^{J} \mu_{j,t} \bar{b}_{j,t}. \tag{50}$$

The full distribution token appends a placeholder for aggregate productivity and two token-type indicators:

$$x_{j,t}^{\mu} = (m_{j,t}, 0, 1, 0)\,. \tag{51}$$

A separate aggregate-productivity token is

$$x_t^{Z} = (0, 0, 0, 0, Z_t, 0, 1)\,. \tag{52}$$

The final two entries identify the token type: $(1, 0)$ denotes a firm-distribution token and $(0, 1)$ denotes the aggregate-productivity token. The economic input sequence is therefore

$$X_t = \left\{ x_{1,t}^{\mu}, \ldots, x_{J,t}^{\mu}, x_t^{Z} \right\}, \tag{53}$$

which contains $J + 1 = 421$ economic tokens. A learned readout token is prepended inside the transformer. The encoder has two transformer layers, four attention heads per layer, and model dimension 64.

The transformer forecasts the next tokenized distribution. Rather than predicting levels, it predicts changes in the mass-weighted token contents,

$$\Delta m_{j,t+1} \equiv m_{j,t+1} - m_{j,t}. \tag{54}$$

The level forecast is then constructed as

$$\widehat{m}_{j,t+1} = m_{j,t} + \widehat{\Delta m}_{j,t+1}. \tag{55}$$

Predicting changes is numerically more stable because the distribution generally moves gradually from one period to the next, making changes smaller and more centered than levels.

The firm Bellman equation also requires household marginal utility, because it determines the stochastic discount factor and, through the household labor-supply condition, the wage. Although these objects could be recovered by resolving the aggregate equilibrium associated with each predicted distribution, it is computationally convenient to predict marginal utility directly as an auxiliary equilibrium object. Define

$$\ell_t \equiv \log U_C(C_t) = -\gamma \log C_t. \tag{56}$$

The target ℓ_t is computed from the model-consistent consumption level generated in the equilibrium simulation. The transformer output is therefore

$$\mathcal{T}_\theta(X_t) = \left(\widehat{\Delta m}_{1,t+1}, \ldots, \widehat{\Delta m}_{J,t+1}, \widehat{\ell}_t\right). \tag{57}$$

The predicted marginal utility implies

$$\widehat{C}_t = \exp\left(-\frac{\widehat{\ell}_t}{\gamma}\right), \qquad \widehat{w}_t = \chi \widehat{C}_t^{\gamma}. \tag{58}$$

Finally, the predicted distribution is projected back onto the economically admissible support. Cell masses are constrained to be nonnegative and to sum to one. Whenever a cell has positive predicted mass, its conditional means are recovered from the corresponding mass-weighted moments and restricted to the capital, debt, and productivity support of that cell.

4.3 Embedding the Transformer in the Firm Problem

The transformer maps the current tokenized distribution into a continuous forecast of the next tokenized distribution. This forecast cannot be used directly as a grid point for value function iteration. The distributional state is high dimensional: even if each of the $J = 420$ distribution tokens were allowed to take only four discrete values, a Cartesian grid would contain 4^{420} points. Storing value and computing policy functions on such a grid is therefore computationally infeasible.

I instead select a sparse set of complete firm distributions,

$$\mathcal{L} = \left\{\mu^1, \ldots, \mu^S\right\}, \tag{59}$$

called the distributional library. See Section 3.3 for details on the selection criteria. Each node μ^s is a full probability distribution over the underlying numerical grid for (ε, k, b). Applying the tokenization map to μ^s produces the corresponding sequence of J distribution tokens,

$$X^\mu(\mu^s) = \left\{x_1^\mu(\mu^s), \ldots, x_J^\mu(\mu^s)\right\}.$$

The baseline uses $S = 200$ library nodes.

At the current aggregate state (μ^s, Z), the transformer takes as input the tokenized representation

$$X(\mu^s, Z) = \left\{x_1^\mu(\mu^s), \ldots, x_J^\mu(\mu^s), x^Z\right\}.$$

Its distributional output is a forecast of the change in the stacked vector of mass-weighted moments. Let

$$m^s \equiv \text{vec}\left(m_1(\mu^s), \ldots, m_J(\mu^s)\right). \tag{60}$$

The forecasted next-period moment vector is

$$\widehat{m}'_{sZ} = m^s + \mathcal{T}_{\theta,m}\left(X(\mu^s, Z)\right), \tag{61}$$

where $\mathcal{T}_{\theta,m}$ denotes the distributional component of the transformer output. The auxiliary component produces the current marginal-utility prediction

$$\widehat{\ell}_{sZ} = \mathcal{T}_{\theta,\ell}\left(X(\mu^s, Z)\right). \tag{62}$$

The forecast $\widehat{m}'_{sZ}$ will generally not coincide with the moment vector associated with any library distribution. I therefore identify the library nodes whose tokenized distributions are closest to the forecast and interpolate their stored continuation values. Distances are computed in a standardized coordinate system containing the full distributional-moment vector. Let $\mathcal{N}(s, Z)$ denote the selected neighboring library nodes and let $\omega_{sZ,r} \geq 0$ denote their inverse-distance weights, with

$$\sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} = 1. \tag{63}$$

These weights define the transformer-implied transition from the current distribution μ^s to future library distributions μ^r.

The perceived continuation value associated with a firm choosing (k', b') is therefore

$$\mathcal{C}_{sZ}(\varepsilon, k', b') = \sum_{Z', \varepsilon'} P_Z(Z, Z') P_\varepsilon(\varepsilon, \varepsilon') \times \sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} \widehat{M}_{sZ,rZ'} V_r(\varepsilon', k', b'; Z'), \tag{64}$$

where $V_r(\varepsilon', k', b'; Z')$ is the firm value evaluated at the future aggregate state (μ^r, Z'), and

$$\widehat{M}_{sZ,rZ'} = \beta \left(\frac{\widehat{C}_{rZ'}}{\widehat{C}_{sZ}}\right)^{-\gamma} \tag{65}$$

is the stochastic discount factor implied by the transformer predictions

$$\widehat{C}_{sZ} = \exp\left(-\frac{\widehat{\ell}_{sZ}}{\gamma}\right), \qquad \widehat{C}_{rZ'} = \exp\left(-\frac{\widehat{\ell}_{rZ'}}{\gamma}\right).$$

Competitive lenders use the same perceived transition. At the current aggregate state (μ^s, Z), the price of one unit of debt issued by a firm with productivity ε and choices (k', b') is

$$q_{sZ}(\varepsilon, k', b') = \sum_{Z', \varepsilon'} P_Z(Z, Z') P_\varepsilon(\varepsilon, \varepsilon') \times \sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} \widehat{M}_{sZ,rZ'} \mathcal{R}_{rZ'}(\varepsilon', k', b'), \tag{66}$$

where, under zero recovery in default,

$$\mathcal{R}_{rZ'}(\varepsilon', k', b') = 1 - p^d_{rZ'}(\varepsilon', k', b'). \tag{67}$$

Equations (64) and (66) identify the points at which the transformer enters the firm problem. Its distribution forecast determines the interpolation weights placed on future library states, while its marginal-utility prediction determines current wages and stochastic discount factors.

At aggregate state (μ^s, Z), the payout of a continuing firm is

$$d_{sZ}(\varepsilon, k, b, k', b') = \pi\left(\varepsilon, k; Z, \widehat{w}_{sZ}\right) + q_{sZ}(\varepsilon, k', b') b' - b - \mathcal{I}(k, k'). \tag{68}$$

Using the transformer-implied continuation value $\mathcal{C}_{sZ}(\varepsilon, k', b')$, the value of continuing is

$$V^c_s(\varepsilon, k, b; Z) = \max_{k', b'} \left\{ \mathcal{D}\left(d_{sZ}(\varepsilon, k, b, k', b')\right) + \mathcal{C}_{sZ}(\varepsilon, k', b') \right\}, \tag{69}$$

subject to

$$b' \leq \bar{b}(k'). \tag{70}$$

The default-adjusted firm value is

$$V_s(\varepsilon, k, b; Z) = \max\left\{V^c_s(\varepsilon, k, b; Z), V^d\right\}, \tag{71}$$

with $V^d = 0$, and the associated default rule is

$$d^d_s(\varepsilon, k, b; Z) = \mathbf{1}\left\{V^c_s(\varepsilon, k, b; Z) < V^d\right\}. \tag{72}$$

The outer equilibrium loop follows Algorithm 1. Given a current transformer law, I solve the firm problem at every library node. I then simulate the economy using Young (2010)'s lottery method, solving current market clearing and the equilibrium wage exactly at each date. The simulated path provides new supervised targets for the next-period distributional moments and current marginal utility. The transformer is then retrained, its law is damped, and firm policies are updated. Convergence requires stability of both the transformer law

and the firm policy functions.

5 Quantitative Results

This section presents the quantitative analysis of the heterogeneous-firm economy. After calibrating the model and characterizing its stationary equilibrium, I use transformer attention to identify which regions of the firm distribution matter most for aggregate dynamics. Attention concentrates on financially constrained firms near the default margin, and shifts toward these firms during downturns as their default risk becomes more consequential for aggregate dynamics.

5.1 Calibration

The real and stochastic primitives follow Khan and Thomas (2013) as closely as possible in an annual discrete-time implementation. I set

$$\alpha = 0.27, \qquad \nu = 0.60, \qquad \beta = 0.96, \qquad \delta = 0.065.$$

Aggregate productivity is discretized with three Rouwenhorst nodes, with persistence $\rho_Z = 0.852$ and innovation standard deviation $\sigma_Z = 0.014$. Idiosyncratic productivity is discretized with seven Rouwenhorst nodes. The labor-disutility scale χ is chosen so that stationary hours are approximately one third.

Five parameters are jointly calibrated to the five moments reported in Table 1. The investment irreversibility parameter and idiosyncratic productivity process are

$$\theta_k = 0.945, \qquad \rho_\varepsilon = 0.653, \qquad \sigma_\varepsilon = 0.135,$$

and are disciplined by the three establishment-level investment moments used by Khan and Thomas (2013), based on Cooper and Haltiwanger (2006). The financial frictions introduce two additional parameters: the operating fixed cost $f = 0.0751$, disciplined by the default frequency, and the equity-issuance cost $\phi_e = 1.500$, disciplined by the frequency of equity issuance. The remaining parameters follow Khan and Thomas (2013), including the borrowing-limit parameter $\theta_b = 1.350$.

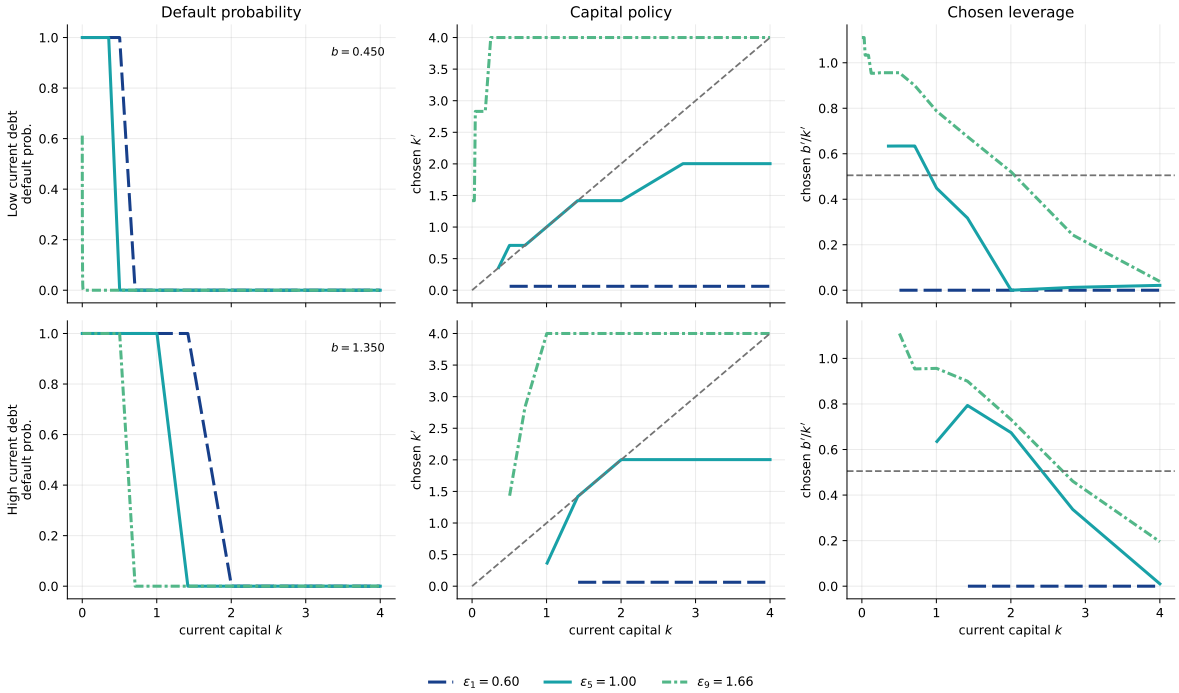
Table 1: Targeted Moments

Moment	Model	Target	Source
Mean investment rate, i/k	0.114	0.130	Khan and Thomas (2013)
Std. investment rate, i/k	0.335	0.337	Khan and Thomas (2013)
Serial correlation of i/k	0.079	0.058	Khan and Thomas (2013)
Default frequency	0.019	0.020	Ippolito et al. (2026)
Equity-issuance frequency	0.038	0.040	Ippolito et al. (2026)

Notes: The five moments jointly discipline $\theta_k, \rho_\varepsilon, \sigma_\varepsilon, f, \phi_e$. Investment is $i/k = (k' - (1-\delta)k)/k$; its standard deviation and serial correlation are winsorized at $[-1, 1]$. Model moments are computed from the stationary distribution. Equity issuance is counted when gross external equity exceeds 10 percent of current capital.

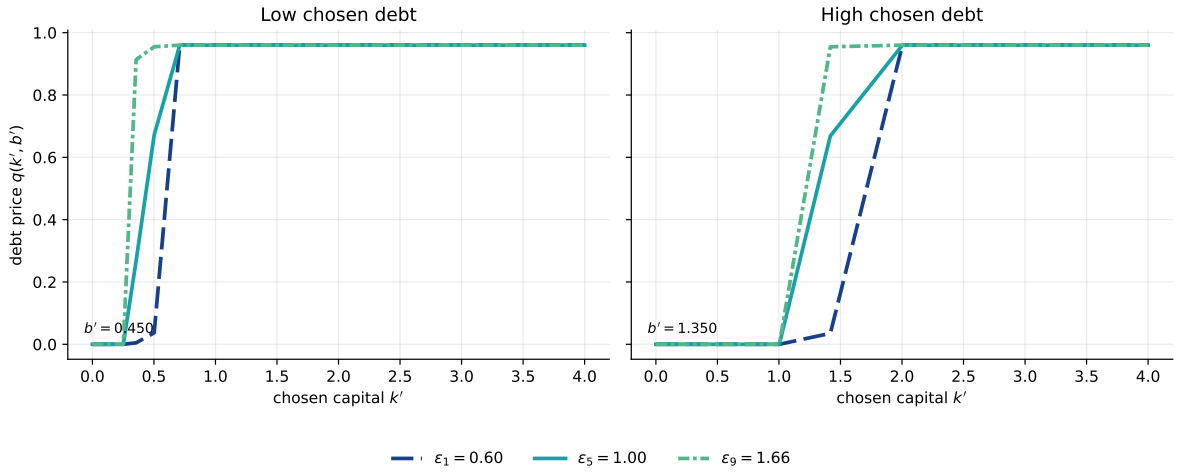
Stationary policies. Figures 4 and 5 characterize the financial margins that will be central to the dynamic analysis. Firms with low capital, high inherited debt, and low productivity are financially constrained: they are more likely to require costly equity issuance and operate close to the default margin. Their higher default probability is reflected in worse borrowing terms, further raising the cost of financing. As firms grow and mature, they accumulate capital, deleverage, and move away from the default region; lower default risk then improves their access to debt finance. The dynamic analysis below asks whether transformer attention identifies these financially constrained firms as particularly relevant for aggregate dynamics, and whether their importance changes with aggregate conditions.

Figure 4: Stationary Firm Policies



Notes: The figure reports stationary firm policies as functions of current capital k for selected idiosyncratic productivity states ε. Columns report the default probability, chosen next-period capital k', and chosen leverage b'/k'. The top and bottom rows correspond to low and high current debt, respectively.

Figure 5: Stationary Debt Prices



Notes: The figure reports the equilibrium debt price $q(k', b')$ as a function of chosen next-period capital k', for selected idiosyncratic productivity states ε. The left and right panels correspond to low and high chosen debt b', respectively.

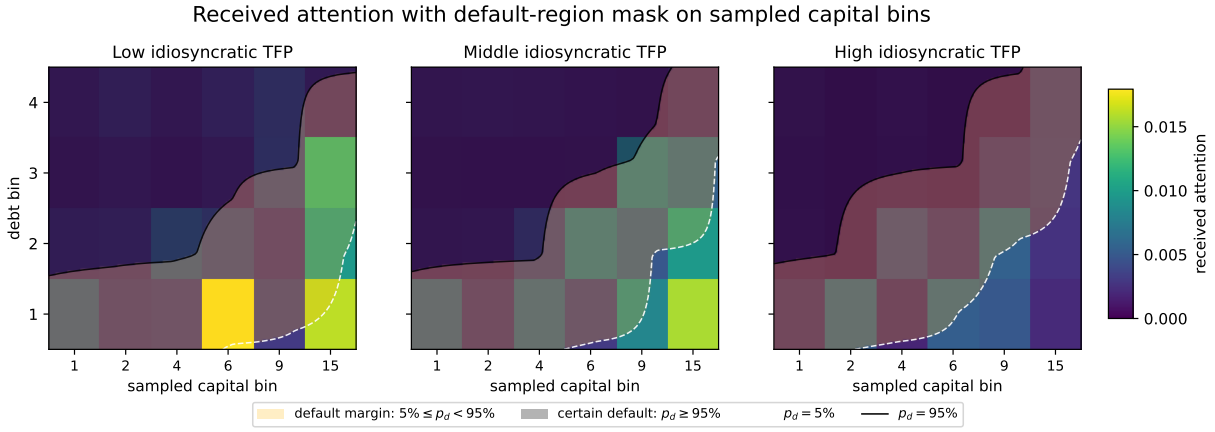
5.2 Quantitative Results

Before turning to the attention diagnostics, I verify that the learned distributional law reproduces the aggregate objects implied by the model. The model is solved using simulated paths of 50,000 periods. Along the simulated path from the final DST iteration, I aggregate each transformer-predicted distribution to recover implied aggregate capital and leverage and compare them with their realized counterparts generated by the model. The resulting R^2 values are 0.9993 for K_{t+1} and 0.9925 for B_{t+1}/K_{t+1}. The auxiliary marginal-utility prediction, from which consumption is recovered, yields an R^2 of 0.9996. Thus, the transformer accurately reproduces aggregate outcomes while forecasting the distribution itself rather than separate aggregate laws.

Where does attention concentrate? Figure 6 provides the main cross-sectional diagnostic. The heatmaps report ergodic received attention across the firm distribution over capital, debt, and idiosyncratic productivity, averaged over all 50,000 periods of the final simulated path. Attention is sharply nonuniform and concentrates around the endogenous default region, particularly among low- and middle-productivity firms. These are precisely the balance-sheet states in which firms are financially fragile and continuation is close to default.

Importantly, the default margin is not supplied to the transformer as an input. The network observes tokens describing the distribution of capital, debt, productivity, and firm mass; the default region shown in the figure is constructed ex post from the equilibrium default probabilities. The concentration of attention near this region is therefore an endogenous feature of the learned distributional law.

Figure 6: Attention and the Endogenous Default Region

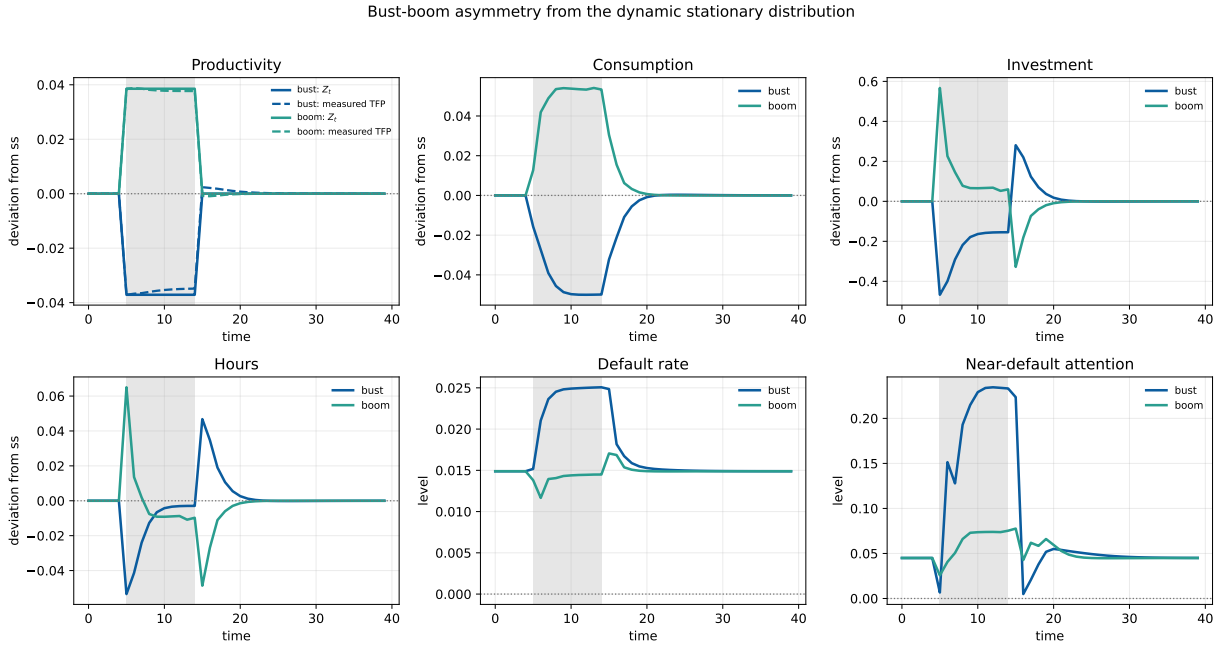


Notes: The figure reports ergodic received attention across capital–debt cells, grouping idiosyncratic productivity into low, middle, and high states. The overlay is constructed from equilibrium default probabilities: yellow denotes the default margin, $5\% \leq p^d < 95\%$, and gray denotes $p^d \geq 95\%$. Capital bins outside the ergodic support are omitted.

State-dependent attention. The previous result identifies where attention is allocated on average. I next ask whether the importance of this region changes with aggregate conditions. Starting from the dynamic stationary distribution associated with the learned law, I subject the economy to temporary aggregate-productivity busts and booms of equal magnitude. Figure 7 reports the responses.

The adjustment is asymmetric. A downturn generates a sharp and persistent rise in default, together with larger contractions in investment and hours than the corresponding expansions generated by the boom. At the same time, attention received by firms near the default margin rises markedly during the bust, while the response in the boom is much smaller. The learned aggregation rule is therefore state dependent: adverse aggregate conditions shift attention toward financially fragile regions of the distribution precisely when default becomes more consequential for aggregate dynamics.

Figure 7: Bust–Boom Asymmetry and Transformer Attention



Notes: The economy starts from the dynamic stationary distribution implied by the final transformer law. The shaded region denotes the temporary aggregate-productivity shock. All variables except default and attention are deviations from their dynamic stationary values. Near-default attention is attention received by firm tokens close to the endogenous default margin.

6 Conclusion

This paper asks whether transformers can be used not only to fit equilibrium laws of motion, but also to uncover the distributional information that makes those laws work. The

contribution is twofold.

First, I propose a transformer-based recursive solution method that preserves the fixed-point logic of Krusell and Smith (1998) while replacing pre-specified aggregate moments with a tokenized representation of the cross-sectional distribution and using a flexible nonlinear function class for the perceived laws of motion. The cross-sectional distribution is represented as a collection of economic tokens–wealth bins in the household benchmark and capital-debt-productivity cells in the firm application. The transformer forecasts the next tokenized distribution, which determines the continuation values used in individual Bellman equations. Within the transformer, the neural-network component provides nonlinear approximation of equilibrium laws of motion. The attention component performs endogenous state-space reduction by weighting distributional tokens before prediction, while also producing weights that can be inspected after the model is solved.

Second, I apply the method to a heterogeneous-firm economy with endogenous firm default. The application delivers two findings. The first concerns where attention is allocated in the cross section. When placed in the position of a researcher observing the full firm distribution, the transformer recovers the Krusell–Smith-like aggregation logic in the benchmark with costless equity issuance and no endogenous default: attention is approximately uniform across firms. Once costly equity issuance and endogenous default are introduced, the attention profile is no longer flat. The transformer assigns the highest weight to firms near the endogenous default margin, indicating that this region of the distribution is most informative for aggregate dynamics. The second finding is that transformer attention is state dependent. A negative aggregate productivity shock raises firm default and temporarily shifts attention toward financially fragile firms. A positive shock of the same size lowers default risk and therefore does not generate a symmetric reallocation of attention away from the default margin. The aggregate responses mirror this shift in attention: in booms, lower default risk allows investment and hours to expand sharply; in busts, the rise in default absorbs part of the adjustment, investment and hours fall, and attention moves toward the firms whose balance sheets shape the downturn. This asymmetry shows that attention serves as a dynamic aggregate-state reduction method: it helps solve the model while also diagnosing which regions of the distribution matter across aggregate states and when.

References

Azinovic, M., Gaegauf, L., and Scheidegger, S. (2022). Deep equilibrium nets. *International Economic Review*, 63(4):1471–1525.

Clementi, G. L. and Hopenhayn, H. A. (2006). A theory of financing constraints and firm dynamics. *Quarterly Journal of Economics*, 121(1):229–265.

Clementi, G. L. and Palazzo, B. (2016). Entry, exit, firm dynamics, and aggregate fluctuations. *American Economic Journal: Macroeconomics*, 8(3):1–41.

Cooley, T. F. and Quadrini, V. (2001). Financial markets and firm dynamics. *American Economic Review*, 91(5):1286–1310.

Cooper, R. W. and Haltiwanger, J. C. (2006). On the nature of capital adjustment costs. *Review of Economic Studies*, 73(3):611–633.

Duffy, J. and McNelis, P. D. (2001). Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. *Journal of Economic Dynamics and Control*, 25(9):1273–1303.

Fernandez-Villaverde, J., Hurtado, S., and Nuno, G. (2023). Financial frictions and the wealth distribution. *Econometrica*, 91(3):869–901.

Gomes, J. F. (2001). Financing investment. *American Economic Review*, 91(5):1263–1285.

Gopalakrishna, G., Gu, Z., and Payne, J. (2026). Institutional asset pricing with segmentation and household heterogeneity. Working paper, Rotman School of Management.

Gu, Z., Laurière, M., Merkel, S., and Payne, J. (2026). Global solutions to master equations for continuous time heterogeneous agent macroeconomic models. *Journal of Political Economy Macroeconomics*. Accepted.

Han, J., Yang, Y., and E, W. (2025). Deepham: A global solution method for heterogeneous agent models with aggregate shocks. Working paper.

Hennessy, C. A. and Whited, T. M. (2005). Debt dynamics. *Journal of Finance*, 60(3):1129–1165.

Hennessy, C. A. and Whited, T. M. (2007). How costly is external financing? evidence from a structural estimation. *Journal of Finance*, 62(4):1705–1745.

Hopenhayn, H. and Rogerson, R. (1993). Job turnover and policy evaluation: A general equilibrium analysis. *Journal of Political Economy*, 101(5):915–938.

Hopenhayn, H. A. (1992). Entry, exit, and firm dynamics in long run equilibrium. *Econometrica*, 60(5):1127–1150.

Ippolito, F., Steri, R., Tebaldi, C., and Villa, A. T. (2026). Mind the gap: The market price of financial (in)flexibility. *Journal of Finance, forthcoming*.

Jermann, U. and Quadrini, V. (2012). Macroeconomic effects of financial shocks. *American Economic Review*, 102(1):238–271.

Judd, K. L., Maliar, L., and Maliar, S. (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. *Quantitative Economics*, 2(2):173–210.

Kase, H., Melosi, L., and Rottner, M. (2025). Estimating nonlinear heterogeneous agent models with neural networks. Working paper.

Khan, A. and Thomas, J. K. (2008). Idiosyncratic shocks and the role of nonconvexities in plant and aggregate investment dynamics. *Econometrica*, 76(2):395–436.

Khan, A. and Thomas, J. K. (2013). Credit shocks and aggregate fluctuations in an economy with production heterogeneity. *Journal of Political Economy*, 121(6):1055–1107.

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

Lanteri, A. (2018). The market for used capital: Endogenous irreversibility and reallocation over the business cycle. *American Economic Review*, 108(9):2383–2419.

Maliar, L., Maliar, S., and Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101.

Ottonello, P. and Winberry, T. (2020). Financial heterogeneity and the investment channel of monetary policy. *Econometrica*, 88(6):2473–2502.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Payne, J., Rebei, A., and Yang, Y. (2025). Deep learning for search and matching models. Research Paper 25-05, Swiss Finance Institute.

Scheidegger, S. and Bilionis, I. (2019). Machine learning for high-dimensional dynamic stochastic economies. *Journal of Computational Science*, 33:68–82.

Valaitis, V. and Villa, A. T. (2024). A machine learning projection method for macro-finance models. *Quantitative Economics*, 15(1):145–173.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Villa, A. T. (2026). Macro shocks and firm dynamics with oligopolistic financial intermediaries. *Review of Economic Studies, forthcoming*.

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

Online Appendix

Appendix A provides implementation details for the infinite-horizon household benchmark, including tokenization, projection, library interpolation, and the non-stochastic distributional transition. Appendix B specializes the DST algorithm to the heterogeneous-firm application, describing the firm tokenization, library construction, debt pricing, market clearing, and equilibrium iteration. Appendix C documents the smooth approximation to the endogenous default rule used in the quantitative implementation and its relation to the deterministic default problem.

A Implementation of the Household Benchmark

This appendix provides the implementation details for the infinite-horizon Krusell–Smith benchmark in Section 2.2. The general DST algorithm, including the perceived distributional law, sparse library, interpolation operator, and equilibrium iteration, is described in Section 3. Here I specialize those objects to the household economy and document the numerical treatment of the distribution.

Let

$$\mathcal{A} = \{a_1, \dots, a_{N_a}\}, \qquad \mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_{N_\varepsilon}\}, \qquad \mathcal{Z} = \{Z_B, Z_G\}$$

denote the asset, idiosyncratic-productivity, and aggregate-productivity grids. The numerical aggregate state is the complete distribution

$$\mu \in \Delta(\mathcal{A} \times \mathcal{E}).$$

Labor supply is fixed at one, so the individual policy is the asset policy

$$g_a(a, \varepsilon, Z, \mu).$$

The distribution μ, rather than its tokenized representation, is propagated through the economy. Tokenization is used to construct the input to the transformer's perceived distributional law.

A.1 Tokenization

As in Section 2.2, partition the asset support into $J = 20$ fixed wealth bins

$$\{\mathcal{B}_j\}_{j=1}^{J}.$$

For distribution μ, the mass in bin j is

$$\omega_j(\mu) = \sum_{a \in \mathcal{B}_j} \sum_{\varepsilon \in \mathcal{E}} \mu(a, \varepsilon).$$

Whenever $\omega_j(\mu) > 0$, define the conditional moments

$$\bar{a}_j(\mu) = \frac{1}{\omega_j(\mu)} \sum_{a \in \mathcal{B}_j} \sum_{\varepsilon \in \mathcal{E}} a \, \mu(a, \varepsilon),$$

$$\bar{\varepsilon}_j(\mu) = \frac{1}{\omega_j(\mu)} \sum_{a \in \mathcal{B}_j} \sum_{\varepsilon \in \mathcal{E}} \varepsilon \, \mu(a, \varepsilon),$$

and

$$\overline{\varepsilon^2}_j(\mu) = \frac{1}{\omega_j(\mu)} \sum_{a \in \mathcal{B}_j} \sum_{\varepsilon \in \mathcal{E}} \varepsilon^2 \mu(a, \varepsilon).$$

The economic content of wealth-bin token j is therefore

$$m_j(\mu) = \left(\bar{a}_j(\mu), \omega_j(\mu), \bar{\varepsilon}_j(\mu), \overline{\varepsilon^2}_j(\mu) \right),$$

and the tokenized representation of the distribution is

$$m(\mu) = \operatorname{vec}\left(m_1(\mu), \ldots, m_J(\mu) \right).$$

If $\omega_j(\mu) = 0$, the asset coordinate is set to the midpoint of $\mathcal{B}_j$, while the productivity coordinates are assigned fixed feasible values. These coordinates do not affect aggregate moments because the corresponding bin mass is zero.

Together with the aggregate-productivity token, these objects generate the transformer input $X(\mu, Z)$ defined in Section 2.2. Aggregate capital is computed directly from the tokenized distribution as

$$K(\mu) = \sum_{j=1}^{J} \omega_j(\mu) \bar{a}_j(\mu).$$

A.2 Standardization and Admissibility Projection

The perceived law in equation (3) is implemented by predicting changes in the tokenized distribution,

$$\Delta m_{t+1} = m_{t+1} - m_t.$$

Inputs and target changes are standardized using moments computed from the current training sample. Let

$$\bar{\Delta}_m \qquad \text{and} \qquad s_{\Delta_m}$$

denote the coordinate-specific mean and standard deviation of Δm_{t+1}, and let $\widetilde{X}_t$ denote the standardized transformer input. If $\widetilde{\mathcal{T}}_{\theta,m}(\widetilde{X}_t)$ is the transformer prediction in standardized units, the raw forecast in economic units is

$$m_{t+1}^{\text{raw}} = m_t + \bar{\Delta}_m + s_{\Delta_m} \odot \widetilde{\mathcal{T}}_{\theta,m}(\widetilde{X}_t).$$

The implementation of the perceived distributional law is then

$$\widehat{m}_{t+1} = \Pi_H \left(m_{t+1}^{\text{raw}} \right), \tag{73}$$

where Π_H projects the raw network forecast onto the economically admissible token set.

The projection imposes

$$\widehat{\omega}_{j,t+1} \geq 0, \qquad \sum_{j=1}^{J} \widehat{\omega}_{j,t+1} = 1,$$

keeps each predicted conditional asset mean inside its wealth bin,

$$\widehat{\bar{a}}_{j,t+1} \in \mathcal{B}_j,$$

and restricts the predicted conditional productivity moments to the support implied by $\mathcal{E}$. Aggregate capital is not included as a separate prediction target. It is reconstructed from the projected distributional forecast:

$$\widehat{K}_{t+1}^{\text{DST}} = \sum_{j=1}^{J} \widehat{\omega}_{j,t+1} \widehat{\bar{a}}_{j,t+1}.$$

A.3 Sparse Library and Interpolation

Value and policy functions are stored on the fixed library

$$\mathcal{L} = \{\mu^1, \ldots, \mu^S\},$$

constructed before the final equilibrium iteration as described in Section 3. At each library node the implementation stores

$$V^s(Z, \varepsilon, a) \equiv V(a, \varepsilon, Z, \mu^s)$$

and

$$g_a^s(Z, \varepsilon, a) \equiv g_a(a, \varepsilon, Z, \mu^s).$$

For the household benchmark, distances between distributions are computed in standardized token space. Let

$$c_H(\mu)$$

denote the standardized coordinate vector constructed from $m(\mu)$. At current library node (μ^s, Z), the transformer forecast

$$\widehat{m}'_{sZ} = m(\mu^s) + \mathcal{T}_{\theta,m}(X_{sZ})$$

is first projected according to equation (73). Let $\mathcal{N}(s, Z)$ denote the $M = 4$ nearest library distributions to this forecast. For $r \in \mathcal{N}(s, Z)$, define

$$d_{sZ,r} = \left[\frac{1}{d_c} \left\| c_H(\widehat{m}'_{sZ}) - c_H(\mu^r) \right\|_2^2\right]^{1/2},$$

where d_c is the dimension of the standardized coordinate vector. In the absence of an exact match, the interpolation weights are

$$\omega_{sZ,r} = \frac{d_{sZ,r}^{-2}}{\sum_{\ell \in \mathcal{N}(s,Z)} d_{sZ,\ell}^{-2}}, \qquad r \in \mathcal{N}(s, Z), \tag{74}$$

with unit weight assigned to an exact match.

These weights generate the continuation value used in the household problem,

$$\mathcal{C}_{sZ}(a', \varepsilon) = \sum_{Z' \in \mathcal{Z}} P_Z(Z, Z') \sum_{\varepsilon' \in \mathcal{E}} P_\varepsilon(\varepsilon, \varepsilon') \sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} V^r(Z', \varepsilon', a'). \tag{75}$$

The Bellman equation at library node μ^s is therefore

$$V^s(Z, \varepsilon, a) = \max_{a' \geq \underline{a}} \left\{ u(R(Z, \mu^s)a + w(Z, \mu^s)\varepsilon - a') + \beta \mathcal{C}_{sZ}(a', \varepsilon) \right\}. \tag{76}$$

Current factor prices are always computed from the current distribution μ^s; the transformer

affects the household problem only through the perceived transition to future distributions.

During equilibrium simulation, the realized distribution μ_t generally does not coincide with a library node. I therefore locate the nearest library distributions to $m(\mu_t)$ using the same standardized distance metric and interpolate the stored asset policies g_a^s using the corresponding inverse-distance weights. This produces the policy

$$g_a(a, \varepsilon, Z_t, \mu_t)$$

used to update the complete distribution.

A.4 Non-Stochastic Distribution Transition

Given the interpolated policy g_a, current distribution μ_t, and aggregate state Z_t, the next distribution is computed using the non-stochastic transition of Young (2010),

$$\mu_{t+1} = \mathcal{Y}_H\left(\mu_t, g_a, Z_t, Z_{t+1}\right).$$

For each current grid point (a_i, ε_j), the chosen next-period asset position is represented as a lottery over its neighboring points on $\mathcal{A}$. The mass $\mu_t(a_i, \varepsilon_j)$ is split across these asset points using the corresponding interpolation weights and across next-period idiosyncratic states using

$$P_\varepsilon(\varepsilon_j, \varepsilon').$$

The operator therefore updates the complete continuum distribution directly; it does not approximate the cross section with a finite population of simulated households.

The equilibrium iteration follows Algorithm 1. A bootstrap perceived law is first used to generate feasible candidate distributions and construct the fixed library $\mathcal{L}$. Given a current transformer law, I solve equation (76) at every library node, interpolate the resulting policies along the simulated distribution path, and propagate the distribution using $\mathcal{Y}_H$. The resulting pairs

$$\{X_t, \Delta m_{t+1}\}$$

form the supervised training sample for the next transformer update. The household problem is then resolved under the updated perceived law, and the procedure is repeated until both the policy functions and the perceived law satisfy the convergence criteria defined in Section 3.

A.5 Numerical Implementation

The economic and machine-learning components are implemented separately. NumPy is used for array operations, Numba for JIT compilation of the Bellman and distribution-transition routines, and PyTorch for transformer training, inference, and attention extraction. The Bellman and Young-transition routines are executed on the CPU, while transformer operations are run on CUDA when available, on Apple Silicon through the `mps` backend when available, and otherwise on the CPU. Thus GPU acceleration is used for self-attention and neural-network optimization, while the economic transition and value-function iterations operate on the complete numerical distribution.

B Implementation of the Heterogeneous-Firm Application

This appendix provides implementation details for the heterogeneous-firm application in Section 4. The general DST algorithm is described in Section 3. Here I specialize the tokenization, interpolation, debt-pricing, and distributional transition operators to the firm economy.

A firm state is (ε, k, b), and the complete aggregate distribution is

$$\mu \in \Delta(\mathcal{E} \times \mathcal{K} \times \mathcal{B}).$$

Value and policy functions are stored on a fixed sparse library of complete firm distributions. The transformer supplies a perceived law for the next tokenized distribution and an auxiliary prediction of household marginal utility. The former determines continuation values and debt prices, while the latter determines the perceived stochastic discount factor and wage inside the firm problem.

B.1 Tokenization and Admissibility Projection

As described in Section 4, the firm state space is partitioned into $J = 420$ cells, corresponding to 15 capital bins, 4 debt bins, and 7 idiosyncratic-productivity groups. Let $\mathcal{C}_j$ denote cell j. Its probability mass under distribution μ is

$$\mu_j(\mu) = \sum_{(\varepsilon,k,b)\in\mathcal{C}_j} \mu(\varepsilon, k, b).$$

For cells with positive mass, let $\bar{k}_j(\mu)$, $\bar{b}_j(\mu)$, and $\bar{\varepsilon}_j(\mu)$ denote the corresponding conditional means.

Consistent with equation (49), the economic content of token j is the vector of mass-weighted moments

$$m_j(\mu) = \left(\mu_j(\mu)\bar{k}_j(\mu), \mu_j(\mu), \mu_j(\mu)\bar{b}_j(\mu), \mu_j(\mu)\bar{\varepsilon}_j(\mu)\right),$$

and

$$m(\mu) = \operatorname{vec}\left(m_1(\mu), \ldots, m_J(\mu)\right).$$

Mass weighting makes empty cells well defined and implies that aggregate capital and debt can be recovered additively from the tokenized distribution:

$$K(m) = \sum_{j=1}^{J} m_{j,k}, \qquad B(m) = \sum_{j=1}^{J} m_{j,b},$$

where

$$m_{j,k} = \mu_j \bar{k}_j, \qquad m_{j,b} = \mu_j \bar{b}_j.$$

The perceived law in Section 4 is implemented by predicting standardized changes in m. Let $\bar{\Delta}_m$ and s_{Δ_m} denote the coordinate-specific mean and standard deviation of the token changes in the current training sample. If $\widetilde{\mathcal{T}}_{\theta,m}(\widetilde{X})$ denotes the transformer prediction in standardized units, the raw forecast in economic units is

$$m'^{,\text{raw}} = m(\mu) + \bar{\Delta}_m + s_{\Delta_m} \odot \widetilde{\mathcal{T}}_{\theta,m}(\widetilde{X}).$$

The forecast used by the equilibrium algorithm is

$$\widehat{m}'(\mu, Z; \theta) = \Pi_F\left(m'^{,\text{raw}}\right), \tag{77}$$

where Π_F projects the prediction onto the economically admissible token set. Cell masses are constrained to be nonnegative and normalized to sum to one. For cells with positive projected mass, conditional capital, debt, and productivity means are recovered from the mass-weighted coordinates and restricted to the support of the corresponding cell. Mass-weighted coordinates are set to zero for cells with zero projected mass.

The auxiliary transformer output is predicted log marginal utility,

$$\widehat{\ell}(\mu, Z) = \log \widehat{U_C(C)} = -\gamma \log \widehat{C}(\mu, Z),$$

which implies

$$\widehat{C}(\mu, Z) = \exp\left[-\frac{\widehat{\ell}(\mu, Z)}{\gamma}\right], \qquad \widehat{w}(\mu, Z) = \chi \widehat{C}(\mu, Z)^{\gamma}. \tag{78}$$

B.2 Fixed Library and Interpolation

The firm application uses the fixed distributional library

$$\mathcal{L} = \{\mu^1, \ldots, \mu^S\}, \qquad S = 200.$$

The library is constructed before the final transformer fixed-point iteration. Candidate distributions include the stationary distribution, capital and leverage perturbations around the stationary state, and distributions generated by a preliminary low-dimensional Krusell–Smith-style solution. The latter provides a warm start for the library and initial policy arrays only; the perceived law used in the final equilibrium is the transformer law.

For interpolation, I augment the standardized distributional coordinates with aggregate capital, debt, and leverage implied by the same token vector. Define

$$L(m) = \frac{B(m)}{K(m)}$$

and

$$c_F(m) = (\mathsf{S}_m m,\, \lambda_K K(m),\, \lambda_B B(m),\, \lambda_L L(m))\,,$$

where S_m standardizes the distributional coordinates and $(\lambda_K, \lambda_B, \lambda_L)$ determine the relative scaling of the aggregate coordinates.

At library node (μ^s, Z), the transformer generates

$$\widehat{m}'_{sZ} = \widehat{m}'(\mu^s, Z; \theta).$$

Let $\mathcal{N}(s, Z)$ contain the $M = 4$ library distributions nearest to this forecast under the Euclidean distance in c_F-space. For $r \in \mathcal{N}(s, Z)$, let

$$d_{sZ,r} = \left\| c_F(\widehat{m}'_{sZ}) - c_F(m(\mu^r)) \right\|_2 .$$

In the absence of an exact match, the interpolation weights are

$$\omega_{sZ,r} = \frac{d_{sZ,r}^{-2}}{\displaystyle\sum_{\ell \in \mathcal{N}(s,Z)} d_{sZ,\ell}^{-2}}, \qquad r \in \mathcal{N}(s, Z), \tag{79}$$

with unit weight assigned to an exact match.

B.3 Continuation Values and Debt Pricing

At current library state (μ^s, Z), the transformer provides both $\widehat{m}'_{sZ}$ and the auxiliary marginal-utility prediction $\widehat{\ell}_{sZ}$. For a future library node $r \in \mathcal{N}(s, Z)$ and aggregate state Z', the perceived stochastic discount factor is

$$\widehat{M}_{sZ,rZ'} = \beta \left(\frac{\widehat{C}_{rZ'}}{\widehat{C}_{sZ}} \right)^{-\gamma}, \tag{80}$$

where $\widehat{C}_{sZ} \equiv \widehat{C}(\mu^s, Z)$.

For a continuing firm choosing (k', b'), the continuation value is

$$\mathcal{C}_{sZ}(\varepsilon, k', b') = \sum_{Z'} P_Z(Z, Z') \sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} \sum_{\varepsilon'} P_\varepsilon(\varepsilon, \varepsilon') \widehat{M}_{sZ,rZ'} V^r(Z', \varepsilon', k', b'). \tag{81}$$

The same perceived transition prices one-period defaultable debt. Under the baseline zero-recovery assumption,

$$q^d_{sZ}(\varepsilon, k', b') = \sum_{Z'} P_Z(Z, Z') \sum_{r \in \mathcal{N}(s,Z)} \omega_{sZ,r} \sum_{\varepsilon'} P_\varepsilon(\varepsilon, \varepsilon') \widehat{M}_{sZ,rZ'} \left[1 - p^{d,r}(Z', \varepsilon', k', b') \right]. \tag{82}$$

Given these objects, the continuing-firm problem at library node μ^s is

$$V^{c,s}(Z, \varepsilon, k, b) = \max_{k', b'} \left\{ \mathcal{D}\left(d_{sZ}(\varepsilon, k, b, k', b') \right) + \mathcal{C}_{sZ}(\varepsilon, k', b') \right\}, \tag{83}$$

subject to the borrowing constraint and partial irreversibility. The default-adjusted value and default probabilities are then updated using the default rule described in Section 4. Debt prices, default probabilities, and firm values are iterated jointly to convergence for fixed transformer forecasts and interpolation weights.

B.4 Simulation and Market Clearing

The realized distribution generally does not coincide with a library node. At each simulated date, I therefore compute $c_F(m(\mu_t))$, identify the nearest library distributions, and use the same inverse-distance interpolation scheme to recover the current capital and debt policies and default probabilities. Denote the resulting policies by

$$g_k(\varepsilon, k, b, Z_t, \mu_t), \qquad g_b(\varepsilon, k, b, Z_t, \mu_t).$$

Given these policies, the complete firm distribution is updated using the non-stochastic operator

$$\mu_{t+1} = \mathcal{Y}_F\left(\mu_t, g_k, g_b, p^d, Z_t, Z_{t+1}\right).$$

For continuing firms, chosen capital and debt are represented as lotteries over neighboring points on the numerical (k, b) grid. Firm mass is then allocated across those grid points and next-period idiosyncratic-productivity states according to P_ε. Defaulting mass exits and is replaced by entrants with the entry capital stock, zero debt, and productivity drawn from the stationary idiosyncratic distribution. Thus, as in the household benchmark, the simulation propagates the complete distribution rather than a finite population of firms.

The auxiliary wage predicted by the transformer is used inside the perceived recursive problem, but not to impose current-period market clearing along the simulated equilibrium path. For each realized (μ_t, Z_t), the current wage is obtained from

$$\mathcal{R}_w(w; \mu_t, Z_t) \equiv w - \chi C(w; \mu_t, Z_t)^\gamma = 0, \tag{84}$$

where $C(w; \mu_t, Z_t)$ is aggregate consumption implied by firm production, investment, partial-irreversibility costs, operating costs, equity-issuance costs, and the distributional transition. Current market clearing is therefore imposed directly in the simulation, while transformer predictions determine the perceived future aggregate states entering firm decisions.

B.5 Equilibrium Iteration

The initial library and policy arrays are obtained from the stationary economy and the low-dimensional warm start described above. The library is then held fixed throughout the final DST equilibrium iteration.

At each outer iteration, the simulated path generates supervised observations of

$$(X_t, \Delta m_{t+1}, \log U_C(C_t)).$$

I supplement these observations with one-step transitions generated at the library nodes under the current policy functions. This ensures that the transformer is trained not only along the simulated path but also at the distributional states at which the Bellman equation is evaluated.

After training, the transformer is evaluated at every (μ^s, Z). For numerical stability, the resulting perceived distributional forecasts and marginal-utility predictions are damped across outer iterations:

$$\widehat{m}'^{,(h)}_{sZ} = (1 - \lambda_\theta)\widehat{m}'^{,(h-1)}_{sZ} + \lambda_\theta \widehat{m}'^{,\text{raw}}_{sZ},$$

$$\widehat{\ell}_{sZ}^{(h)} = (1-\lambda_\theta)\widehat{\ell}_{sZ}^{(h-1)} + \lambda_\theta \widehat{\ell}_{sZ}^{\text{raw}},$$

where h indexes the outer equilibrium iteration. The firm problem is then resolved using the associated interpolation weights, debt prices, and default probabilities.

The updated capital and debt policies are also damped,

$$g^{(h)} = (1-\lambda_g)g^{(h-1)} + \lambda_g g^{\text{raw},(h)}, \qquad g = (g_k, g_b).$$

The resulting policies are simulated using $\mathcal{Y}_F$, imposing the wage market-clearing condition (84) at each date. The new distributional transitions update the transformer training sample, and the procedure is repeated until the policy and perceived-law convergence criteria in Section 3 are satisfied.

B.6 Numerical Implementation

The economic and machine-learning components are implemented separately. NumPy is used for dense array operations, Numba for JIT-compiled dynamic programming and distribution transitions, and PyTorch for transformer training, inference, and attention extraction. The Bellman updates, expected-continuation arrays, debt-price schedules, and distributional transition are parallelized across CPU cores where possible. Transformer operations are run on CUDA, Apple `mps`, or CPU depending on available hardware.

Attention matrices are extracted under no-gradient evaluation from the same transformer that supplies the perceived distributional law used in (81) and (82).

C Smooth Approximation to the Default Decision

In the model presented in the main text, equity holders compare the value of continuing, $V_t^c(\varepsilon, k, b; \Omega_t)$, with the value of default, V^d, which is normalized to zero. Under the deterministic default rule, the firm value is

$$V_t^{\text{hard}}(\varepsilon, k, b; \Omega_t) = \max\left\{V_t^c(\varepsilon, k, b; \Omega_t), V^d\right\}, \tag{85}$$

and the default decision is

$$d_t^{\text{hard}}(\varepsilon, k, b; \Omega_t) = \mathbf{1}\left\{V_t^c(\varepsilon, k, b; \Omega_t) < V^d\right\}. \tag{86}$$

This rule introduces a kink in the firm value at $V_t^c(\varepsilon, k, b; \Omega_t) = V^d$ and a discontinuity in the default policy. For the numerical implementation, I replace the hard maximum with a

smooth log-sum-exp approximation.

Specifically, consider transitory choice-specific shocks to the continuation and default options. Equity holders compare

$$V_t^c(\varepsilon, k, b; \Omega_t) + \xi_t^c \qquad \text{and} \qquad V^d + \xi_t^d, \tag{87}$$

where ξ_t^c and ξ_t^d are independent type-I extreme-value random variables with common scale parameter $\sigma_d > 0$. The shocks are independent across firms and over time and affect only the current default decision. They are integrated out before the continuation value is evaluated and are not included as additional state variables.

Conditional on the economic state

$$s_t \equiv (\varepsilon, k, b; \Omega_t),$$

the firm defaults whenever

$$V^d + \xi_t^d > V_t^c(s_t) + \xi_t^c. \tag{88}$$

Because the difference of two independent type-I extreme-value variables follows a logistic distribution, the conditional default probability is

$$\begin{aligned} p_t^d(s_t) &\equiv \Pr\left[V^d + \xi_t^d > V_t^c(s_t) + \xi_t^c \mid s_t\right] \\ &= \frac{\exp\left(V^d/\sigma_d\right)}{\exp\left(V_t^c(s_t)/\sigma_d\right) + \exp\left(V^d/\sigma_d\right)} \\ &= \left[1 + \exp\left(\frac{V_t^c(s_t) - V^d}{\sigma_d}\right)\right]^{-1}. \end{aligned} \tag{89}$$

The corresponding continuation probability is

$$\begin{aligned} p_t^c(s_t) &\equiv 1 - p_t^d(s_t) \\ &= \frac{\exp\left(V_t^c(s_t)/\sigma_d\right)}{\exp\left(V_t^c(s_t)/\sigma_d\right) + \exp\left(V^d/\sigma_d\right)} \\ &= \left[1 + \exp\left(\frac{V^d - V_t^c(s_t)}{\sigma_d}\right)\right]^{-1}. \end{aligned} \tag{90}$$

After integrating over the choice-specific shocks, the expected firm value is, up to an additive constant that is common across choices,

$$V_t^{\sigma_d}(s_t) = \sigma_d \log\left[\exp\left(\frac{V_t^c(s_t)}{\sigma_d}\right) + \exp\left(\frac{V^d}{\sigma_d}\right)\right]. \tag{91}$$

This is the smooth counterpart of the hard maximum in (85). An equivalent expression is

$$V_t^{\sigma_d}(s_t) = \max\left\{V_t^c(s_t), V^d\right\} + \sigma_d \log\left[1 + \exp\left(-\frac{\left|V_t^c(s_t) - V^d\right|}{\sigma_d}\right)\right]. \tag{92}$$

Equation (92) is also the numerically stable expression used in the implementation, since it avoids exponentiating large value-function levels.

The derivative of the smoothed value with respect to the continuation value is

$$\frac{\partial V_t^{\sigma_d}(s_t)}{\partial V_t^c(s_t)} = \frac{\exp\left(V_t^c(s_t)/\sigma_d\right)}{\exp\left(V_t^c(s_t)/\sigma_d\right) + \exp\left(V^d/\sigma_d\right)} = p_t^c(s_t). \tag{93}$$

Thus, the marginal effect of a change in the continuation value on the ex-ante firm value equals the probability that the firm continues. Similarly,

$$\frac{\partial V_t^{\sigma_d}(s_t)}{\partial V^d} = p_t^d(s_t). \tag{94}$$

The second derivative is

$$\frac{\partial^2 V_t^{\sigma_d}(s_t)}{\partial V_t^c(s_t)^2} = \frac{1}{\sigma_d} p_t^c(s_t) p_t^d(s_t), \tag{95}$$

which is largest near the default boundary and converges to zero away from it.

Relation to the hard default rule. The smooth formulation nests the deterministic model as a limiting case. For any state such that $V_t^c(s_t) \neq V^d$,

$$\lim_{\sigma_d \to 0^+} p_t^d(s_t) = \mathbf{1}\left\{V_t^c(s_t) < V^d\right\}, \tag{96}$$

and

$$\lim_{\sigma_d \to 0^+} p_t^c(s_t) = \mathbf{1}\left\{V_t^c(s_t) > V^d\right\}. \tag{97}$$

At the indifference point $V_t^c(s_t) = V^d$, both choices receive probability one half. Moreover,

$$\lim_{\sigma_d \to 0^+} V_t^{\sigma_d}(s_t) = \max\left\{V_t^c(s_t), V^d\right\}. \tag{98}$$

The approximation error satisfies

$$0 \leq V_t^{\sigma_d}(s_t) - \max\left\{V_t^c(s_t), V^d\right\} \leq \sigma_d \log 2. \tag{99}$$

Hence, the smoothed value converges uniformly to the hard maximum as $\sigma_d \to 0^+$.

Use in the quantitative implementation. The baseline implementation sets

$$\sigma_d = 0.002. \tag{100}$$

This value is small relative to the variation in firm continuation values and therefore makes the smooth policy close to the deterministic default rule. Its purpose is numerical: it replaces the discontinuous default indicator with a continuous probability, stabilizes the distributional transition, and avoids abrupt changes in debt prices and policy functions when a firm moves across the default boundary.

In particular, the debt-pricing equation uses the smoothed repayment probability,

$$q_t(\varepsilon, k', b') = \mathbb{E}_t \left[M_{t,t+1} p^c_{t+1}(\varepsilon_{t+1}, k', b'; \Omega_{t+1}) \right], \tag{101}$$

where

$$p^c_{t+1} = 1 - p^d_{t+1}.$$

Likewise, the Young distributional transition assigns the mass $p^c_t(\varepsilon, k, b; \Omega_t)$ to the continuing-firm transition and the mass $p^d_t(\varepsilon, k, b; \Omega_t)$ to the default-and-entry transition. Thus, the smoothing does not alter the set of economic choices or introduce an additional persistent state. It only replaces the hard allocation of firm mass at the default boundary with its smooth probabilistic counterpart. The quantitative implementation uses (89)–(91) as a numerical approximation. As σ_d becomes small, the smoothed economy converges to the deterministic-default economy.