Journal of Business Cycle Research
https://doi.org/10.1007/s41549-026-00144-3

RESEARCH PAPER

Robots Are Us

Seth G. Benzell[1] · **Laurence J. Kotlikoff**[2] · **Guillermo Lagarda**[3]

Received: 25 May 2026 / Accepted: 7 August 2026
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2026

Abstract

Will smart (programmed) machines do to humans what the internal combustion engine did to horses—render them obsolete? If so, can putting people out of work or, at least, good work leave them unable to buy what smart machines produce? The answer, as we illustrate in a simple, two-period, two-good life-cycle model, is yes. Our goods are output and prayers. Output is produced with code, new and legacy, and capital. This function is effectively a robot with code for brains and capital for brawn. "Robots are Us" captures our key assumption that code is the offspring of human brains, specifically that of high-skilled agents. Such agents are free to produce new code or pray (for others). Low-skilled agents only pray. An improvement in code tech, modeled as greater durability, increases current demand for (and supply of) new code. But it decreases future demand. What explains this dynamic reversal of Say's Law? When code lasts longer, today's fresh code means even more legacy code tomorrow. But tomorrow's greater legacy code reduces tomorrow's need for/demand for new code and, thus, coders. Hence, a code-tech shock generates a boom-bust cycle in the coders' wage, which, thanks to skilled-worker mobility equals their prayer wage. Output, saving, and investment boom and bust in unison with wages. The investment bust can leave the economy's capital stock and, potentially, output and lifetime utility below their starting values. Such immiserating growth is more likely when code can, as AI now does, self improve. Whether or not future generations are worse off, the big winners are early generations who ride the boom and avoid the bust. As we discuss, cross-cohort redistribution can share the gains from tech shocks and preclude immiseration. What about limiting IP rights? Surprisingly, doing so makes immiseration more likely. Better tech worsening the economy? This is as counterintuitive as Leamer (1999)'s finding that a minimum wage can raise GDP and lower inequality. Both studies pull rabbits out of hats, which was Edward Leamer's lifetime practice and delight.

∗University of Piraeus, Department of Banking and Financial Management, anthropel@unipi.gr.
†Babson College Finance Division, jhasanhodzic1@babson.edu. ‡Boston University, Department of Economics, kotlikoff@gmail.com.

Extended author information available on the last page of the article

Published online: 12 October 2026

Keywords Automation · Welfare · Wages · Technology · Digital Economy · OLG · Savings · AI

JEL Classification E32 · O3 · O1 · O4

1 Introduction

Smart machines, combining artificial intelligence and physical capital, are increasingly replacing humans in performing both routine (e.g., driving taxis) and cognitive (e.g., writing code) tasks. Indeed, current-generation AIs appear capable of completing more than one in four current job-specific tasks [see Eloundou et al. (2024)]. See also Aschenbrenner (2024). This paper addresses automation within a simple, two-period, two-good, two-skill OLG model. One good, corn, is produced with code and capital. We dub the combination of artificial brain (code) and physical brawn (capital) a robot. The other good, prayer, is produced solely by labor. Agents work only when young. Skilled agents can code or pray and move between these occupations to equalize their remuneration. Unskilled agents can only pray.

Coders develop robot brains, henceforth AIs, be they sensors on industrial arms or Large Language Models. Our title, "Robots Are Us" captures this key fact—that AIs are the offspring of human brains. We denote by δ the degree to which currently produced code remains useful from one period to the next. This retention factor will be less than 1 if AIs need to be upgraded, retrained, updated, etc. Δ will exceed 1 if AIs can self improve. Our focus is on the impact of increases in δ.

Though simple, our model admits a range of dynamic outcomes from robotization, some quite unpleasant. Indeed, improving technology can increase today's labor demand but decrease tomorrow's, creating a boom-bust cycle that makes all future generations worse off. Such immiserating growth is more likely when technological change is self-propelled, an emerging AI capacity. The answer can also depend on code's excludability and its rivalry in use.

The initial response to an increase in the code retention rate is a tech boom that raises the demand for new code, and thus coders. The resultant rise in the wages of coders leads enough high-skilled prayers/priests to move into coding to maintain wage equality across the two occupations.

The resulting rise in young workers' wages engenders more national saving and capital formation, reinforcing the boom. But over time, as the stock of legacy code grows, the demand for new code and, thus, coders falls. This depresses wages, reducing saving by the young and, thus, the amount of capital they accumulate for old age. It also reverses the occupational flow with would-be coders selling prayers, presumably for the return of good times. The investment bust can immiserate all future generations. Yes, a higher δ economy ends up with permanently more brain (code), but, potentially, too little brawn (capital) to sustain its initial level of output.

Because of our OLG structure, AI's dynamic intergenerational redistribution of resources, entailing a rise and then a fall in wages, spells a rise and then fall in national saving and investment. Since less capital means lower wages, general equilibrium feedback exacerbates any long-term decline in output. The race here is

not between men and machines as in Acemoglu et al. (2018), but between code and capital. If capital falls enough, output will ultimately be lower notwithstanding the higher quantity and durability of code.

We simulate our model for a variety of parameterizations, showing that in some situations all future generations are left worse off. Immiseration depends on saving preferences as well as the ability to substitute software (code) for hardware. If young workers save at high rates or if software and hardware can readily substitute for one another, immiseration won't arise.

Moreover, immiserating technological progress can't arise in our model if agents are infinitely-lived, intergenerational altruistic, representative agents. In such models, national saving is driven by the difference in interest and time preference rates, not the intergenerational distribution of resources. Indeed, capital shortages lead to higher interest rates, which immediately induce greater saving and capital formation. Yet, a host of studies, including Altonji et al. (1992, 1995, 1997), and Abel and Kotlikoff (1994) find no evidence of operative intergenerational altruism. This fact notwithstanding, the representative agent model has become, presumably for analytical tractibility, the workhorse of macro models of robots [e.g., Acemoglu et al. (2018)]. Unfortunately, as shown here, robots may have a far darker side than such studies portray.

2 Literature

Concern about new technology dates back to Ned Ludd's destruction of two stocking frames in 1779, near Leicester, England. Sixty-five years later, Marx (1992) reiterated Ludd's warning about machines replacing humans. Keynes and Maynard (1963) also raised concerns about technological unemployment during the Great Depression, but he saw it as a temporary phase.

In the mid-20th century, with high employment and rapid wage growth, concerns about technology were often dismissed. However, recent rapid job losses to smart machines have led economists to reconsider. Brynjolfsson and Mcafee (2014) in *The Second Machine Age* and Aghion et al. (2017) in *Artificial Intelligence and Economic Growth* highlight the role of machines in changing compensation across occupations, increasing inequality, and decreasing labor force participation. While Brynjolfsson and McAfee focus on rising inequality and job displacement, Aghion and co-authors view AI as a new form of automation subject to *Baumol's Cost Disease*, with long-term outcomes determined by what AI cannot do.

The immediate impacts of smart machines on employment and wages have been significant. Studies by Autor et al. (2003), Acemoglu and Autor (2011), and Autor and Dorn (2013) document significant outsourcing of middle-skilled jobs to smart machines. Similar trends are observed in Europe (Goos et al., 2011) and during America's early industrial revolution (Katz & Margo, 2014). Many economists connect robotization to the decline in labor's share of output, with Hemous and Olsen (2022) showing how capital substitutes for low-tech labor and complements high-tech labor, explaining trends in labor's share and income inequality since the 1960 s.

There are skeptics that question negative labor-market impacts of smart machines. Mishel et al. (2013) argue that robots cannot explain post-1970 s U.S. job polarization given the timing of wage and employment changes. Autor (2015) contends that technological advancements create new job opportunities, similar to how the automobile industry created jobs despite displacing equestrian drivers. As for increasing inequality, Eloundou et al. (2024) argue that high-skilled occupations, such as software engineers and data scientists, are now more exposed to the impact of artificial intelligence than lower-paid workers.

Our model addresses these issues by featuring an endogenous technological and growth response to an exogenous technology shock—the rise in the rate of code retention. This connects to the endogenous growth literature, including contributions by Schumpeter and Cycles (1939), Arrow (1962), Lucas (Jul 1988), Romer (1990), and others. Unlike models where robots substitute perfectly for labor, our model sees durable capital and code as complements, with immiseration arising from high-tech workers producing code that competes with future high-tech workers.

Ours is hardly the first approach to modeling automation. Zeira (1998) posits labor-substituting machines, while Zuleta (2004, 2008) considers the degree of capital intensiveness. Acemoglu et al. (2018) models a continuum of tasks that are automated or invented over time. Sachs et al. (2012) and Sachs (2015) model robots as substituting perfectly for labor. In contrast, our model treats durable capital and code as complementary, with immiseration arising from high-tech workers producing additional code that competes with future high-tech workers.

3 The Model

Agents consume two products—corn and prayers. Corn, which can be consumed or invested, is produced using capital and code. The CES function governing corn production can be viewed as a smart machine or robot since it combines physical capital and code. Prayer is the non-automatable sector. It's produced via a CES function of low- and high-tech labor.[1] Prayers are ephemeral, i.e., consumed when produced. Supplies of high- and low-tech workers remain fixed through time.[2]

3.1 Production

Time-t production of corn, Y_t, and prayers, S_t, satisfy (1) and (2),

$$Y_t = D_Y \left[\alpha (K_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} + (1-\alpha)(A_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} \right]^{\frac{\varepsilon_y}{\varepsilon_y - 1}}, \tag{1}$$

[1] Adding labor to the production of corn, or capital to the production of prayer, would not alter our qualitative findings.

[2] Our model can readily accommodate long-run balanced growth arising from population growth or labor-augmenting technological change.

$$S_t = D_S \left[\gamma (H_{S,t})^{\frac{\varepsilon_S - 1}{\varepsilon_S}} + (1-\gamma)(G_t)^{\frac{\varepsilon_S - 1}{\varepsilon_S}} \right]^{\frac{\varepsilon_S}{\varepsilon_S - 1}}, \tag{2}$$

where $H_{S,t}$ is the number of high-tech workers in the prayer sector and G_t references low-tech workers. D_S and D_Y are total factor productivity terms, γ and α are CES share parameters, and ε_y and ε_s are CES elasticities. The stock of code A_t grows according to,

$$A_t = \delta A_{t-1} + z H_{A,t}, \tag{3}$$

where the "depreciation" factor is $\delta \in [0, 1)$. Higher δ means that legacy code is useful for longer.[3] With this specification we have in mind both the explicit efforts of AI scientists and engineers, and the way that the labor of workers in automatable sectors can be used as training data for future AI systems. $H_{A,t}$ stands for the supply of high-tech coders and z is their productivity coefficient.

The demands for code, high-tech workers, and capital satisfy

$$\max_{K_t, A_t} Y_t (A_t, K_t) - m_t A_t - r_t K_t, \tag{4}$$

where corn is the numeraire, m_t is the rental rate for code, and r_t is the interest rate. Factor demands for prayer reflect,

$$\max_{H_{S,t}, G_t} q_t S_t \left(H_{S,t}, G_t\right) - w_t^G G_t - w_t^H H_{S,t}, \tag{5}$$

where q_t is the price of prayer, w_t^H is a high-tech worker's wage in the prayer sector, and w_t^G is a low-tech worker's wage.

Factor prices satisfy

$$w_t^H = q_t D_S \left[\gamma \left(H_{S,t}\right)^{\frac{\varepsilon_S - 1}{\varepsilon_S}} + (1-\gamma)(G_t)^{\frac{\varepsilon_S - 1}{\varepsilon_S}} \right]^{\frac{1}{\varepsilon_S - 1}} \left[\gamma (H_{S,t})^{-\frac{1}{\varepsilon_S}} \right], \tag{6}$$

$$w_t^G = q_t D_S \left[\gamma \left(H_{S,t}\right)^{\frac{\varepsilon_S - 1}{\varepsilon_S}} + (1-\gamma)(G_t)^{\frac{\varepsilon_S - 1}{\varepsilon_S}} \right]^{\frac{1}{\varepsilon_S - 1}} \left[(1-\gamma)(G_t)^{-\frac{1}{\varepsilon_S}} \right], \tag{7}$$

$$r_t = D_Y \left[\alpha (K_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} + (1-\alpha)(A_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} \right]^{\frac{1}{\varepsilon_y - 1}} \left[\alpha (K_t)^{-\frac{1}{\varepsilon_y}} \right], \tag{8}$$

[3] Some of our simulations will assume a 30 percent depreciation rate per period (30–40 years), meaning about 1 percent of the code base must be replaced annually. The actual rate of code depreciation is unclear. The IRS allows a 3-year lifespan for licensed software, 15 years for in-house or bespoke software, and software bundled with hardware depreciates at the hardware's rate.

and

$$m_t = D_Y \left[\alpha (K_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} + (1-\alpha)(A_t)^{\frac{\varepsilon_y - 1}{\varepsilon_y}} \right]^{\frac{1}{\varepsilon_y - 1}} \left[(1-\alpha)(A_t)^{-\frac{1}{\varepsilon_y}} \right]. \tag{9}$$

3.2 Households

Whether high- or low-tech, households maximize

$$u = (1-\phi)\left[(1-\kappa)\log c_{y,t} + \kappa \log s_{y,t}\right] + \phi\left[(1-\kappa)\log c_{o,t+1} + \kappa \log s_{o,t+1}\right]. \tag{10}$$

subject to,

$$c_{y,t} + q_t s_{y,t} + \frac{c_{o,t+1} + q_{t+1} s_{o,t+1}}{1 + r_{t+1}} = i_{j,t}, \tag{11}$$

where $c_{y,t}$, $c_{o,t}$, $s_{y,t}$, $s_{o,t}$, are consumption of corn and prayer by the young and old, respectively, and $i_{j,t}$ is total resources of group j. For low-tech workers,

$$i_{G,t} = w_t^G. \tag{12}$$

For high-tech workers in the prayer sector,

$$i_{(H,S),t} = w_t^H, \tag{13}$$

and for high-tech workers writing code,

$$i_{(H,A),t} = z\left(m_t + \delta p_t\right), \tag{14}$$

where zm_t is revenue from renting out newly produced code and $z\delta p_t$ is the proceeds from the sale of rights to future use of this fresh code.

Households save in the form of capital and code. Capital and code accumulation obeys

$$\phi I_t = K_{t+1} + p_t \delta A_t, \tag{15}$$

where I_t is the total resources of those born in t, ϕ is the saving propensity of the young, and $p_t \delta A_t$ is the value of code retained from the current period. In equilibrium the return to both investments is identical.

Demands satisfy,

$$s_{y,t} = \frac{\kappa(1-\phi) i_{j,t}}{q_t}, \tag{16}$$

$$c_{y,t} = (1-\kappa)(1-\phi)i_{j,t}, \tag{17}$$

$$s_{o,t+1} = \frac{1+r_{t+1}}{q_{t+1}}\left[\kappa\phi i_{j,t}\right], \tag{18}$$

and

$$c_{o,t+1} = \left[1+r_{t+1}\right]\left[(1-\kappa)\phi i_{j,t}\right]. \tag{19}$$

3.3 Equilibrium

Since high-tech workers are mobile between sectors,

$$w_t^H = z\left(m_t + \delta p_t\right). \tag{20}$$

Asset-market clearing entails equal returns on capital and code, i.e.,

$$p_t = \sum_{s=t}^{\infty} R_{s+1,t}^{-1}\delta^{s-t}m_{s+1}, \tag{21}$$

where $R_{s,t}$ is the compound interest factor between t and s, i.e.,

$$R_{s,t} = \prod_{j=t}^{s}\left(1+r_j\right). \tag{22}$$

Finally, equilibrium requires

$$Y_t = C_{y,t} + C_{o,t} + K_{t+1} - K_t, \tag{23}$$

$$H_t = H_{A,t} + H_{S,t}, \tag{24}$$

and

$$S_t = S_{y,t} + S_{o,t}, \tag{25}$$

where C_y, C_o, S_y, S_o, are total consumption demand of corn and payer by the young and old respectively.

3.4 The Steady State with Cobb–Douglas Production

Suppose both sectors operate with Cobb–Douglas technology. Define the capital-to-code ratio

$$k \equiv \frac{K}{A}. \tag{26}$$

Let

$$h \equiv \frac{H}{A}, \qquad g \equiv \frac{G}{A}, \tag{27}$$

denote the supplies of high-tech and low-tech labor per unit of code.

Factor prices in the corn sector become

$$m = (1-\alpha) D_Y k^{\alpha}, \tag{28}$$

$$r = \alpha D_Y k^{\alpha-1}, \tag{29}$$

and the ex-dividend price of code satisfies

$$p = \frac{(1-\alpha) D_Y k^{\alpha}}{1 + r - \delta}. \tag{30}$$

In steady state the goods-market condition can be written

$$\begin{aligned} D_Y k^{\alpha} = \; & \frac{(1-\phi)(1-\kappa)}{\phi}\left[k + \frac{(1-\alpha) D_Y k^{\alpha} \delta}{1 + \alpha D_Y k^{\alpha-1} - \delta}\right] \\ & + (1-\kappa)\left[k + \frac{(1-\alpha) D_Y k^{\alpha} \delta}{1 + \alpha D_Y k^{\alpha-1} - \delta}\right]\left(1 + \alpha D_Y k^{\alpha-1}\right). \end{aligned} \tag{31}$$

The asset-market condition equates the value of assets held by the young to their savings:

$$k + p\delta = \phi \left\{ z(m + p\delta) h + (1-\gamma) g \left(\frac{\gamma}{z(m + p\delta)} \right)^{\frac{\gamma}{1-\gamma}} (q D_S)^{\frac{1}{1-\gamma}} \right\}. \tag{32}$$

These two equations determine the steady-state values of (k, q).

The mechanism for immiseration operates through wages. In the Cobb–Douglas steady state, a higher code-retention rate δ lowers the capital-to-code ratio k. Because the marginal product of code equals

$$m = (1-\alpha) D_Y k^{\alpha}, \tag{33}$$

a lower k reduces the marginal product of code and, therefore, coder compensation, given by

$$w^H = z(m + \delta p). \tag{34}$$

If the resulting decline in wages sufficiently reduces aggregate savings, the accumulation of physical capital may fall, potentially lowering output and welfare.

Proposition 1 *In the Cobb–Douglas steady state, the capital-to-code ratio $k \equiv K/A$ is a decreasing function of the code-retention rate δ:*

$$\frac{dk}{d\delta} < 0. \tag{35}$$

Proof Using

$$y \equiv D_Y k^{\alpha}, \qquad r \equiv \alpha D_Y k^{\alpha-1}, \qquad p = \frac{(1-\alpha)D_Y k^{\alpha}}{1+r-\delta}, \tag{36}$$

the steady-state goods-market condition can be written as

$$y = (1-\kappa)(k + p\delta)\left(\frac{1-\phi}{\phi} + 1 + r\right). \tag{37}$$

Dividing by k and using $y/k = r/\alpha$ yields

$$\frac{r}{\alpha} = (1-\kappa)\left(\frac{1}{\phi} + r\right)\left[1 + \frac{(1-\alpha)r\delta}{\alpha(1+r-\delta)}\right]. \tag{38}$$

This equation implicitly defines r as a function of δ. Rearranging gives

$$\delta(r) = \frac{(1+r)\left(r - \alpha(1-\kappa)(\phi^{-1}+r)\right)}{r - \alpha(1-\kappa)(\phi^{-1}+r) + (1-\kappa)(\phi^{-1}+r)(1-\alpha)r}. \tag{39}$$

Differentiating shows that $\delta'(r) > 0$ for all admissible parameter values, so $dr/d\delta > 0$. Since

$$r = \alpha D_Y k^{\alpha-1}, \tag{40}$$

and $0 < \alpha < 1$, it follows that k is a decreasing function of r. Hence

$$\frac{dk}{d\delta} < 0. \tag{41}$$

□

3.5 Analytic Results in a Simplified Model

Consider the special case of only one good (corn) produced using Leontief technology (i.e. $\varkappa = 0$, $G = 0$, $\epsilon_y \to 0$).[4] Under the right conditions, a sufficiently large increase in the code retention rate zeros out the wage, producing full economic collapse. This occurs either after two periods or at a specific date in the future. This simplified model exhibits many key features of the full model. These include the potential for, in this case, an abrupt boom and bust.

Assume production satisfies

$$Y_t = D_y min\left[(1-\alpha)A_t, \alpha K_t\right]. \tag{42}$$

Since all workers are coders and their period-specific labor supply, H, is fixed, A_t is easy to derive. Assuming δ is zero prior to time zero and fixed at $\hat{\delta}$ afterwards,

$$A_t = \frac{zH\left(1-\hat{\delta}^{t+1}\right)}{1-\hat{\delta}}. \tag{43}$$

When code is scarce, production is linear in code and the marginal product of code and the wage are positive. The return to capital is zero. The opposite holds when capital is scare. I.e.,

$$m_t = \begin{cases} D_y(1-\alpha) & \text{if } (1-\alpha)A_t < \alpha K_t \\ 0 & \text{if } (1-\alpha)A_t > \alpha K_t \end{cases} \tag{44}$$

and

$$r_t = \begin{cases} 0 & \text{if } (1-\alpha)A_t < \alpha K_t \\ D_y\alpha & \text{if } (1-\alpha)A_t > \alpha K_t. \end{cases} \tag{45}$$

Economic collapse, if it occurs, arises in the period after code switches from scarce to abundant. Let T reference the first date at which code becomes abundant, assuming this occurs. Then $m_t = D_y(1-\alpha) \forall t < T$ and $m_t = 0 \forall t \geq T$. The economy permanently collapses at $T+1$.

The condition for code abundance is $\alpha K_T < (1-\alpha)A_T$. Inserting equations (13) and (14) and rearranging yields

$$\alpha\left(\phi H z m_{t-1} + p_{T-1}\hat{\delta}\left(\phi zH - A_{T-1}\right)\right) < (1-\alpha)A_T. \tag{46}$$

Note, from Eq. (43) that A_{T-1} is greater than or equal to zH. Hence, the second term on the left-hand-side of Eq. (46) is negative. Because only a fraction ϕ of newly produced code's value is saved, a positive price of code serves, on balance, to crowd out capital. When ϕ, the saving preference parameter, is large, immiseration is less likely.

[4] Unlike the main model, there's no physical capital depreciation. Investment -a storage technology- motivates saving in the absence of a positive marginal product of capital.

Also note that $m_T = 0$ implies $p_{T-1} = 0$ since code is valueless if it is expected to have no return in subsequent periods. Setting $p_{T-1} = 0$ and incorporating formulas for m_t and A_t yields a condition for the economy to collapse at time $T + 1$ as a function of parameters

$$\alpha\phi D_y < \frac{1 - \hat{\delta}^{T+1}}{1 - \hat{\delta}}, \tag{47}$$

As Eq. (47) indicates, if collapse occurs, it occurs faster the smaller the product of ϕ,α , and D_y, and the larger the value of $\hat{\delta}$. Output rises until the economy collapses since A_t is strictly increasing over time. Neither the timing nor occurrence of economic collapse is affected by coder productivity, z.

What happens to welfare? Those born at $t = 0$ through $t = T - 2$ are better off than those born prior to period 0. The reason is their ability to sell their durable code raises their total wage. Those born in $T - 1$ are also better off. They receive the $t = 0$ wage while young, but a high interest rate on their physical capital investments while old. As for generations born in $t = T$ or later, they starve.

Equation (47) also specifies the condition for permanent scarcity of code, namely,

$$1 - \frac{1}{\alpha\phi D_y} > \hat{\delta} \tag{48}$$

If code is permanently scarce, then $w_{H,t} = zm_t + \frac{zm_t}{1-\hat{\delta}} = z(1-\alpha)D_y + \frac{z(1-\alpha)D_y}{1-\hat{\delta}}$. Hence, so long as $\hat{\delta}$ is sufficiently small (or D_y, ϕ, and α are sufficiently large), a rise in δ leads to an immediate and permanent increase in compensation and welfare.[5] The size of the increase in wages is increasing in δ, z and D_y.

This simplified model also illustrates the role of the elasticity of substitution in the corn sector on the possibility of immiseration. If code and capital were perfect substitutes rather than complements, an increase in δ from zero to positive would never lower the marginal product of code, capital, or the wage. Therefore, strong complementarity between code and capital makes immiseration more likely.

4 Numerical Solution and Results

We calculate the economy's perfect foresight transition path and equilibrium following an immediate and permanent increase in the rate of code retention via Gauss-Seidel iterations.

[5] Compensation of coders jumps immediately to a higher level where it remains indefinitely if the economy never collapses. If it does collapse in the future, the present discounted value of code falls as collapse nears. Interestingly, although new coders get paid the same through time and capital income is zero, output rises, albeit at a decelerating rate. In the steady state, total wage income is equal to total output. Along the transition, the high wage is financed by decummulation of physical capital, which is increasingly crowded out.

4.1 Simulation Results

When the code retention rate, δ, is zero, corn production relies on contemporaneous capital and labor (code writers). As δ rises, production also depends on the contributions of deceased high-tech workers. Initially, an increase in δ raises coder compensation, attracting more high-tech workers into coding. This raises compensation in both sectors and increases the price of prayers. The wages of low-tech workers may rise or fall depending on their complementarity with high-tech workers.

Over time, the situation for high-tech workers worsens. As durable code increases, the marginal productivity of code falls, reducing the demand for new coders. High-tech workers may end up earning less than in the initial steady state, working primarily in the prayer sector. For low-tech workers, the return of high-tech workers to the prayer sector initially raises, then lowers prayer prices, which can also affect low-tech wages. When high- and low-tech workers are close substitutes, their wages move in tandem.

The dynamic reaction to a higher δ impacts capital formation. The initial rise in high-tech worker earnings can lead to increased saving and investment, making code and high-tech workers more productive. However, if total worker compensation eventually falls, so does the saving of the young and the economy's capital supply. Less capital reduces the marginal productivity of code and raises interest rates, lowering the price of code and coder wages.

We next explore alternative transition paths under varying technology or preference assumptions, examining both immiserating growth and welfare-improving growth scenarios.

4.2 Long-Run Immiseration

Table 1 reports model parameters for our first simulation, assuming workers save 20 percent of their wages. While this saving rate seems low, it results in an initial steady-state marginal product of capital of 1.66, or 1.02 percent annually. This is below empirical estimates for the U.S. economy's marginal product of capital.[6] Thus saving is arguably calibrated to be over abundant in this illustration

Figure 1 shows the negative long-term consequences of the code-retention rate, δ , rising from 0 to.7. Initially, real national income rises, peaking at 7.8 percent above baseline, but ultimately declines to 4.2 percent below its initial steady-state value. In the long run, corn output decreases by 28.0 percent.

After the shock, the output of prayer dips as workers move to lucrative coding jobs, then increases steeply as automated workers return. The price of prayers does the opposite. While long-run production of prayers is 27.4 percent higher, the price drops by 43.4 percent. Both types of workers are perfect substitutes in spirituality, and their common compensation initially jumps 7.6 percent before gradually falling. In the long run, all workers earn 24.8 percent less than initially.

The first generation impacted is young before the code retention shock and old during it. They benefit from a higher marginal product of capital but are harmed by

[6] See, for example, Poterba (1998)

Table 1 This table gives parameter values for both simulation scenarios in Fig. 1

	Model	Value		
		Simulation 1	Simulation 2	Simulation 3.1−3.3
	Parameter	(1)	(2)	(3)
Elasticity in Prayer Sector	ϵ_s	∞	∞	1
Elasticity in Corn Sector	ϵ_y	1	1	1
Prayer High-Tech Input Share Parameter	γ	0.5	0.5	0.5
Corn Capital Input Share Parameter	α	0.5	0.5	0.5
Code Retention Share	δ	$0 \rightarrow 0.7$	$0 \rightarrow 0.8$	$0 \rightarrow 0.25$
Saving Preference Parameter	ϕ	0.2	0.85	0.5
High-Tech Worker Quantity	H	1	1	1
Low-Tech Worker Quantity	G	1	1	1
Prayer Consumption Share	κ	0.5	0.5	0.5
Code Writing Productivity	z	1	1	1
TFP in Corn	D_y	1	1	1
TFP in Prayer	D_s	1	1	1
Firm Setup Cost	C	NA	NA	0.055
Exogenous Free Code	$\overline{A}$	NA	NA	0.25

Note how we depict the changes on the retention rate, δ, from 0 to 0.7 and to 0.85, respectively

a higher price of prayer, ending slightly worse off. The next generation benefits from the shock, with high compensation when young and a high marginal product of capital when old. However, they face higher prayer prices, resulting in a 14.2 percent rise in lifetime utility compared to initial steady-state generations. Long-term generations are worse off by 16.5 percent, with national income 4.2 percent lower.

The significant long-term decline in the capital stock and associated rise in its marginal product have two causes: falling wages, which finance capital acquisition, and the crowding out of capital by durable code. Initially, increased compensation leads to more saving, but much of it is used to acquire legacy code rather than capital. The long-term decline in both physical capital and labor's shares of national income is consistent across all simulations.

4.3 Long-Run Welfare Improvement

Figure 1 demonstrates that a tech boom does not necessarily lead to long-term economic decline. The key factor is a higher saving propensity. Previously, we assumed a saving propensity, ϕ, of 0.2. Here, with all other parameters fixed, we assume ϕ equals 0.85 (see Table 1). With this higher saving rate, workers are better off in the long run, although the generation already old when the code-retention shock hits is worse off due to a short-term rise in the price of prayer.

National income peaks after the technological shock, rising 25.6 percent above baseline, and in the long run, it remains 16.8 percent higher. This reflects a temporary rise and subsequent fall in the capital stock, but not enough to reduce long-term wel-

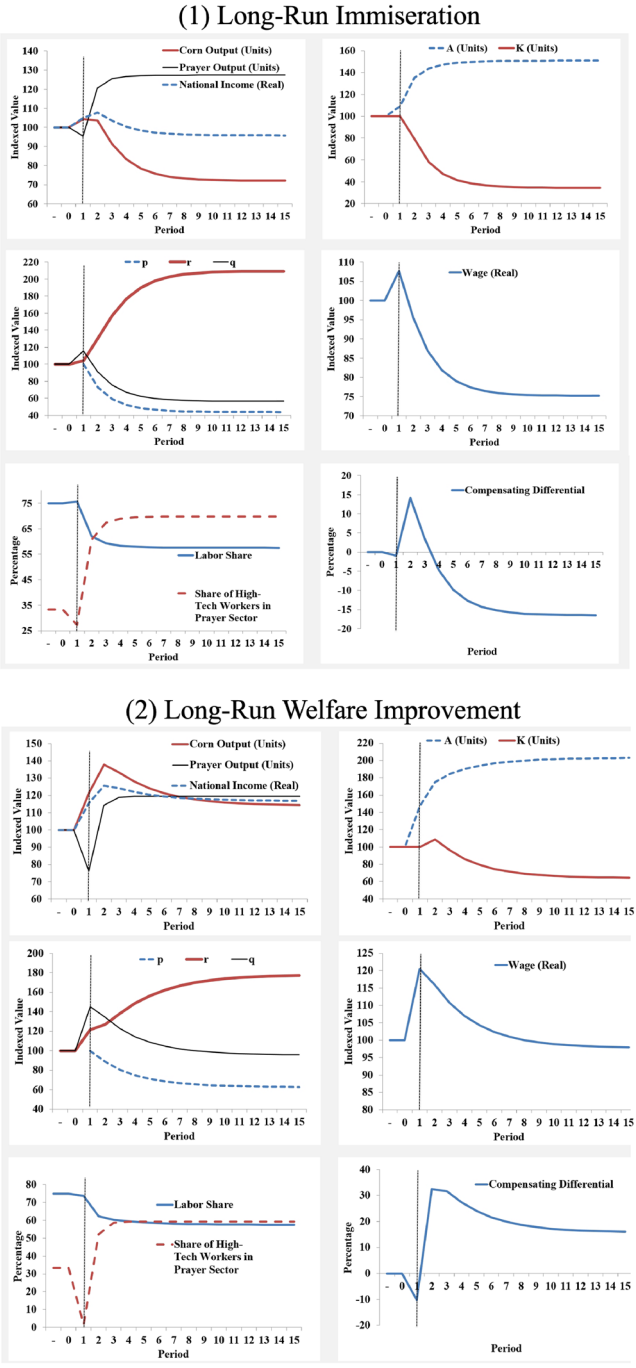

Fig. 1 Transition paths based on Table 1. In this and all subsequent figures, compensating differential references the percentage change in steady-state consumption needed for initial steady-state households' lifetime utility to equal the lifetime utility of households who are old in a given period along the transition path. Output, capital and code stocks are in units. Wage and national income are deflated

fare. Unlike the previous simulation where the capital stock declined immediately, here it temporarily increases by 8.7 percent.

A short-term rise in both the capital stock and prayer price boosts the common wage in the short run, keeping it roughly at its initial value in the long run. After peaking 20.5 percent above its initial value, the wage ultimately falls, ending 2.2 percent lower. The stock of code more than doubles, while the capital stock eventually declines by 35.8 percent.

The respective increase in the stock of code and decrease in the stock of capital result in a significant long-term rise in the interest rate, up 78.0 percent. Although the common wage of high and low-tech workers decreases slightly, the higher interest rate allows future generations to consume significantly more.

A high saving rate leads the δ shock to improve long-run welfare because a higher saving rate supports a higher capital stock, mitigating the decline in wages after code accumulates. We will next discuss the sensitivity of welfare changes to the combination of saving propensity and code retention rates, finding that low saving and high code retention rates lead to the worst outcomes, while high saving and high code retention rates lead to the best outcomes.

4.4 Alternative Possible Outcomes

The model's reaction to the δ shock is sensitive to parameter values. Figure 2 illustrates the dependence of outcomes on the saving preference parameter and code writing productivity. The top half shows combinations of saving preference parameters ϕ and shocks to δ, assuming Table 1's other parameter values. Red areas indicate higher long-run utilities, while blue areas indicate lower utilities. Long-run utility increases most when δ is large and the saving rate is high, and decreases most when the δ shock is high and the saving rate is low. The lower half of Fig. 2 considers joint shocks to the saving rate and code-writing productivity, z. Higher values of both reinforce positive impacts on long-run utility. Unlike δ shocks, increases in code-writing productivity (z) always enhance welfare since this shock boosts productivity for living high-tech workers. Policies aiming to raise wages by reducing labor supply, such as increasing the minimum wage, are likely to backfire in this model.

These simulations provide several insights. High-tech workers benefit from higher substitutability between capital and code in the corn sector, stabilizing the marginal product of code and preventing long-run immiseration. The path of the capital-to-code ratio in response to a rise in δ is independent of the absolute and relative numbers of each type of worker, suggesting that transforming more low-tech into high-tech workers might have less impact than expected. An increase in δ always triggers a tech boom, raising both the price of code and high-tech wages. However, this boom is typically short-lived, leading to a major tech and saving bust. Generally, the δ shock raises labor's share in the short run but lowers it in the long run.

Finally, if the shock to δ is sufficiently high to raise its value above 1, code will be self-generating. If the production function for output has an elasticity of substitution between code and capital equal to or exceeding 1, the marginal product of code will fall, over time, to zero. This would reduce wages from coding to zero. Absent a

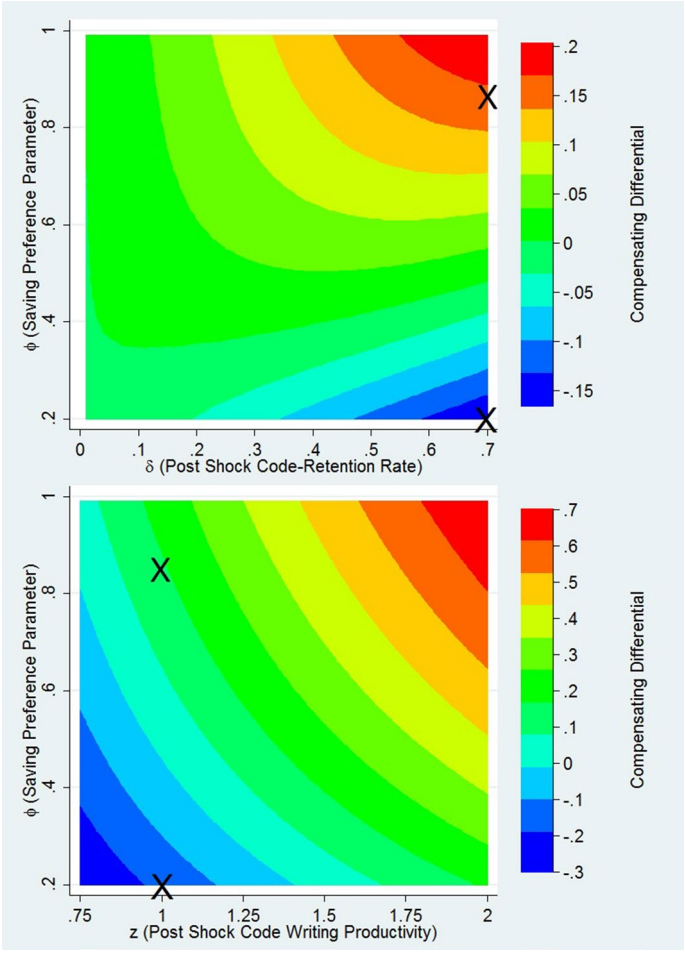

Fig. 2 Long-Run Compensating Differential for Alternative Saving and Code-Retention and Productivity Shocks. Parameters not on axes are given in Table 1. X's denote parameter combinations with transition paths discussed in the text

prayer sector, workers' saving and the capital stock will also fall zero. With no capital, there will be no output. I.e., immiseration is guaranteed.

5 The Role of Property Rights and Rivalry

To this point, we've assumed that code is private and rival, meaning it is unavailable for rent or use by others when one firm uses it. However, code, unlike physical capital, may represent stored information that can be non-rival in use, although non-rivalry doesn't imply non-excludability. Patents, copyrights, trade secrets, and other measures can limit unlicensed distribution, while government mandates or non-profit contributors can make code an open source public good. Current LLMs span the

gamut of these possibilities: LLama's model weights are public and ChatGPT 3.5 is provided free as a public good—these are non-rival and non-excludable; ChatGPT 4.0 can be hired at a fixed rate per token—it is non-rival and excludable; the AI being developed by various militaries to power their proprietary 'loyal wingman' aircraft are rival (in the sense they will not be shared across organizations) and excludable. This section modifies the baseline model to explore these issues by adding a firm entry decision. This is crucial in the context of intellectual property (IP) as the number of firms using code simultaneously must be determined.

Corn-producing firms enter by paying a fixed cost each period to access free code, covering both business overhead and the cost of discovering new AI applications. When code is excludable, firms may rent additional supplies at a market price, with strong IP protections incentivizing more code creation but limiting existing AI use. We first present the modified model for private (rival and excludable) code, then explore two alternatives: non-rival and non-excludable (public good) code, and non-rival but excludable code, where developers can rent it out to multiple companies without friction.

5.1 Rival, Excludable (Private) Code

Corn firms maximize their profit, which is equal to

$$\pi_{j,t} = F\left(k_{j,t}, zH_{j,t} + a_{j,t} + \overline{A}\right) - C - r_t k_{j,t} - m_t a_{j,t} - w^H H_{j,t}, \tag{49}$$

where $\pi_{j,t}$ are profits for firm j at time t, $F(\bullet)$ is a constant elasticity of substitution production function, $k_{j,t}$ is the amount of capital rented by the firm, $a_{j,t}$ is the amount of code rented by the firm, $H_{j,t}$ is the amount of high-tech labor hired by the firm, $\overline{A}$ is the exogenously set amount of free code in the economy, and C is the per-period fixed operating cost. In equilibrium all firms have zero profits.

Market clearing conditions are,

$$\sum a_{j,t} = \delta A_{t-1}, \tag{50}$$

$$\sum k_{j,t} = K_t, \tag{51}$$

$$\sum H_{j,t} = H_{A,t}, \tag{52}$$

$$Y = c_{o,t} + c_{y,t} - K_t + K_{t+1} - NC, \tag{53}$$

where N is the number of firms. All other equations are as in the baseline model. Since all firms are identical, Eq. (53) can be converted into an equation for N, the number of firms.

$$0 = NF\left(\frac{K_t}{N}, \frac{zH_{A,t}}{N} + \frac{1}{N}\delta A_{t-1} + \overline{A}\right) - NC - r_t K_t - m_t \delta A_{t-1} \tag{54}$$

Firms enter up to the point that the value of the public code they obtain for free, namely $\overline{A}$, equals their fixed cost of production. Thus,

$$\overline{A}F_{a,t} = C. \tag{55}$$

This fixes the marginal product of code at $\frac{C}{\overline{A}}$ in every period. Intuitively, new firms can acquire a perfect substitute for new code, and, thus, new coders at a fixed cost by setting up shop and gaining access to $\overline{A}$ in free code. Given that corn's production obeys constant returns to scale, fixing code's marginal product means fixing the ratio of capital to code. This, in turn, fixes the interest rate. Hence, the rental rates of code and capital are invariant to the increase in δ.

To solve the model an additional step is added to the iteration procedure. Given a guess of prices and stocks in a period, Eq. (54) is used to calculate N. This guess of N in each period is included in the next iteration to calculate new prices.

Figure 3 shows transition paths for this economy, with excludable, non-rival code, after an increase in the code retention rate. Parameter values are provided in Table 1. Although an increase in δ does not change the marginal productivity of code, it does raise coder compensation. The reason is that coders can now sell property rights to the future use of their invention. Immediately after the shock, the wage of high-tech workers increases 10.2 percent, decreasing to 6.5 percent higher in the long run due to a decrease in the relative price of corn.

The number of firms decreases as a result of the technology shock, by 11.3 percent in the long run. Were the number of firms to remain fixed, the jump in δ would entail a higher code to capital ratio (in the short term the capital stock is fixed, and in the long run it increases by less than the code stock). This would mean a lower marginal productivity of code, which equation (55) precludes. It would also mean a negative payoff to setting up a new firm. Another way of viewing this relationship is that as the rental price of code decreases, the attractiveness of acquiring code by setting up a new firm decreases. As the latter is fixed, the rental price of code is fixed, with the margin of adjustment being fewer firms created.

In the period of the shock, welfare for the old decreases by 1.8 percent. This is because the cost of prayer increases 8.0 percent while the interest rate remains fixed. In the long run, high and low-tech workers are 6.5 percent better off. As the wage of high-tech workers can only increase as a function of the technology shock, this model variant does not admit long-run immiserating growth absent additional assumptions. For example, if the number of firms were to be fixed due to oligopilization of the industry, or if the fixed cost of firm entry were increasing in the number of firms, (55) would not hold, in which case the marginal productivity of code would decrease as code accumulates. This would reintroduce the possibility of immiserating growth.

5.2 Non-rival, Non-excludable (Public) Code

Assume code, in the period after it is produced, becomes a pure public good used simultaneously by every firm. This could arise by government edict, the wholesale pirating of code, or reverse engineering.

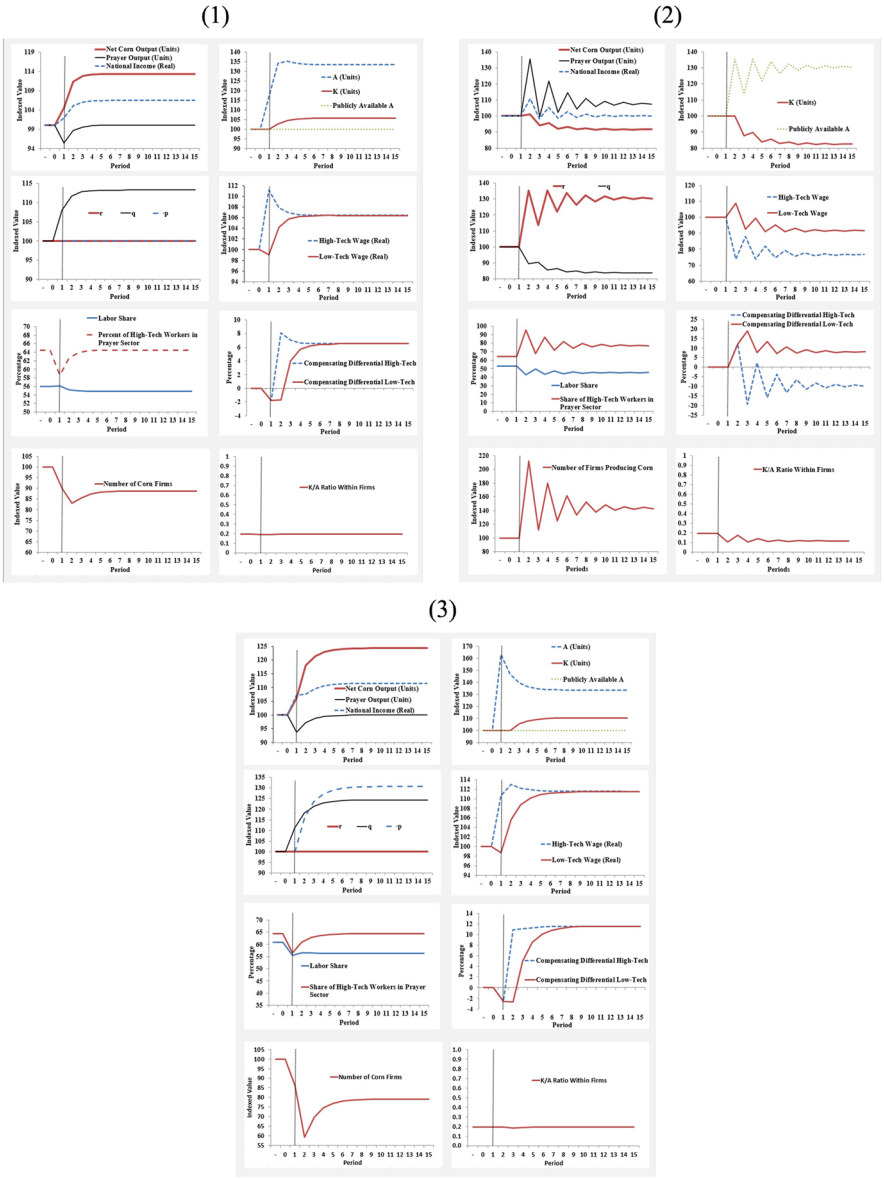

Fig. 3 Transition paths based on the third column of Table 1. Note: (1) Rival, Excludable (Private) Code, (2) Non-rival, Non-excludable (Public), and (3) Non-rival, Excludable code

Profits are

$$\pi_{j,t} = F\left(k_{j,t}, zH_{j,t} + a_{j,t} + \overline{A}\right) - C - r_t k_{j,t} - w^H H_{j,t}, \tag{56}$$

as firms no longer need to rent their stock of code ($a_{j,t}$), where

$$a_{j,t} = \delta A_{t-1} \forall j \tag{57}$$

As before, firm entry and exit imply zero profits,

$$0 = NF\left(\frac{K_t}{N}, zH_{A,t} + \delta A_{t-1} + \overline{A}\right) - NC - r_t K_t - w^H H_{A,t}. \tag{58}$$

and, because the amount of free code available to newly set up firms changes over time, (55) is modified to

$$\left(\delta A_{t-1} + \overline{A}\right) F_{a,t} = C. \tag{59}$$

Finally, with investment in code no longer crowding out investment in capital,

$$K_{t+1} = \phi I_t. \tag{60}$$

Figure 3 shows the transition path after a δ increase for the case of non-excludable code. The initial steady state is the same as in the prior case of excludable rival code. However, the response to the jump in δ is dramatically different. It has no immediate effect on the high-tech wage because workers no longer hold copyright to their code. They, therefore, have no incentive to move to the corn sector, leaving the economy unresponsive to the shock in the short term.

The period after the shock, the economy begins to react. The stock of free public code, which now includes both $\overline{A}$ plus all of the economy's legacy code, is larger. This induces more firm entry. The number of firms immediately more than doubles to 112.4 percent above its initial level in the short run and is 43.4 percent higher in the long run. As (59) shows, with more free code available, the break-even condition entails a lower code rental rate. In equilibrium, this entails more firms operating with less capital per unit of code.

The lower marginal product of code and, thus, of coders leads 30.3 percent of high-tech workers to move from coding into prayer in the period after the shock. National income peaks at 10.8 percent above its initial level in this period. The interest rate rises by 35.5 percent and the wage of low-tech workers increases by 15.0 percent.

The economy's transition is characterized by a series of oscillations. Periods of relatively high coder hiring and fewer firms is followed by periods of plentiful free code, more firm entry, and relatively low coder hiring. In the long run, the share of high-tech workers coding is 12.6 percent higher than its initial level and the high-tech wage is 23.4 percent lower. Welfare in the long run for high-tech workers is 9.7 percent lower. For low-tech workers, welfare is 8.0 percent higher. It is easy to select

parameters such that both groups are worse off. As in the baseline model, the main mechanism for immiseration is the reduction of the high-tech wage leading to less capital accumulation. A contributing factor is the inefficiency introduced due to coders no longer being able to internalize the full value of their work.

5.3 Non-rival, Excludable Code

Now assume code is excludable, but non-rival in its use, permitting high-tech workers to license all their code to all firms in the period after it is produced. The equations for the rival, excludable model hold with the following exceptions. First, profits are given by

$$\pi_{j,t} = F\left(k_{j,t}, zH_{j,t} + \delta A_{t-1} + \overline{A}\right) - C - r_t k_{j,t} - m_t \delta A_{t-1} - w^H H_{j,t} \tag{61}$$

Second, the price of code reflects its use by all firms.

$$p_t = \sum_{s=t}^{\infty} R_{s+1,t}^{-1} \delta^{s-t} m_{s+1} N_{s+1}. \tag{62}$$

As Fig. 3 shows, the δ shock produces a long-run, welfare-improving growth path, indeed a significantly better path than in the rival, excludable case. As in the rival, excludable case, firm entry satisfies Eq. (55). Hence, the interest rate and marginal product of new code are fixed, the wage of high-tech workers must increase, and a long-term welfare improvement is ensured. Welfare for the old in the period of the shock decreases by 2.6 percent, slightly more than in the rival, excludable case. Thereafter, however, households are much better off, enjoying 11.5 percent higher utility in the long run, measured, as always, as a compensating differential.

This case features less entry. In the long-run, there are 20.8 percent fewer firms than before the δ shock. In the public code case, there is a 43.4 percent increase in the number of firms in response to the shock. In the private code case, the shock reduces the number of firms by 11.3 percent. Intuitively, since each firm can use all available code, fewer firms are needed. More surprising is the decrease relative to the rival code case. The reason is that with non-rival, excludable code, the effective supply and value of code is very high. The higher value crowds out capital investment. As can be seen from Eq. (55), the relatively small number of firms entering is due to a relatively higher effective stock of code and lower stock of capital.

6 Conclusion

Will smart machines, rapidly replacing workers in various jobs, lead to economic misery or prosperity? Our two-period OLG model suggests both outcomes are possible but firmly predicts a long-run decline in labor's share of income, tech-booms followed by tech-busts, and a growing dependency of current output on past software investment.

In our simple model, long-run immiseration can be caused by increases in the code retention rate. Increases in the code retention rate could correspond to innovations in the ability of LLMs to learn and generalize from human data. It could also arise from advances in the ability of AI to self-improve –potentially raising the code retention rate above unity.

The mechanism for immiseration is a reduction in labor income and, thereby, saving and capital formation. Yes, the economy has better technology. But it has less capital. With the right parameters, the latter factor can outweigh the former. Immiseration is more likely with a smaller propensity to save and greater complementarity between capital and code.

Making higher code retention a win win for all current and future generations requires making sure the national saving rate is sufficiently high. This could be done through taxing high-skilled cohorts who benefited from technological breakthroughs and transferring resources to young people and other agents with a high propensity to save. This keeps the capital stock and wages from falling. Other policies for managing the rise of smart machines may backfire. For example, policies restricting labor supply will reduce total labor income. While this may temporarily raise wages, it will reduce saving, investment, and the capital formation on which wages depend.

To the extent that AI is non-rival, countries must decide on property rights for AI as well. Excludabilty increases the return to software development, increasing long-run levels of code, wages, national income, and welfare. It also lowers the potential for technologically driven business cycles. On the other hand, treating AI as a public good by, for example, denying AI software patents, ignoring AI copyrights, failing to prosecute software piracy, or mandating that new code be open-source, produces worse results along these dimensions. This said, such a policy increases short-term welfare as well as firm entry. This last implication is an increasingly important desiradatum as countries become concerned about the concentration of power in the hands of a small number of superstar technology companies.

Our simple model illustrates the range of things that smart machines can do for us as well as do to us. Its central message is disturbing. Absent appropriate fiscal policy, which redistributes from winners to losers, smart machines can mean collective long-term misery. Ironically, the same AI that helps us produce more in the present can do the opposite in the future.

This is the type of inconvenient truth that Edward Leamer generated and relished generating throughout his career. His long list of burst bubbles include: A minimum wage can make the economy both more productive and more equitable. A single market, namely housing, is central to most business cycles. The Hecksher-Ohlin model explains patterns of trade once productive factors are properly measured. And, econometric findings are largely in the eyes of their designers, beholders, and believers. This devastating critique of econometric practices is the mother of all of Leamer's burst bubbles. His statement in Leamer (1983), "Hardly anyone takes anyone else's data analysis seriously.", has had, in the opinion of an arguably unbiased major LLM, "a profound disciplinary impact on how empiricists think about specification search, fragility, and research credibility."

Our paper does not drop the big one as did *Let's Take the Con Out of Econometrics*. It doesn't, in 13 pages, call into question essentially all econometric find-

ings—past, present, and, given ongoing practice, future. Yet, we hope *Robots Are Us* clarifies how effectively embedding our brains in AI may be undermining our children's economic futures—something that standard robot models with their pretend intergenerational altruism appear to ignore.

Appendix 1: Mathematical Details—Proof of Proposition 1

In the Cobb–Douglas steady state,

$$y \equiv D_Y k^{\alpha}, \qquad r \equiv \alpha D_Y k^{\alpha-1}, \qquad p = \frac{(1-\alpha)D_Y k^{\alpha}}{1+r-\delta}. \tag{63}$$

Using the goods-market steady-state condition,

$$y = (1-\kappa)(k+p\delta)\left(\frac{1-\phi}{\phi}+1+r\right), \tag{64}$$

and dividing by k gives

$$\frac{y}{k} = (1-\kappa)\left(1+\frac{p\delta}{k}\right)\left(\frac{1}{\phi}+r\right). \tag{65}$$

Since

$$\frac{y}{k} = \frac{r}{\alpha} \qquad \text{and} \qquad \frac{p}{k} = \frac{(1-\alpha)D_Y k^{\alpha-1}}{1+r-\delta} = \frac{(1-\alpha)r}{\alpha(1+r-\delta)}, \tag{66}$$

it follows that

$$\frac{r}{\alpha} = (1-\kappa)\left(\frac{1}{\phi}+r\right)\left[1+\frac{(1-\alpha)r\delta}{\alpha(1+r-\delta)}\right]. \tag{67}$$

Equation (67) implicitly defines r as a function of δ. Rearranging yields

$$\delta(r) = \frac{(1+r)\big(r-\alpha(1-\kappa)(\phi^{-1}+r)\big)}{r-\alpha(1-\kappa)(\phi^{-1}+r)+(1-\kappa)(\phi^{-1}+r)(1-\alpha)r}. \tag{68}$$

Differentiating (68), one obtains

$$\delta'(r) = \frac{N(r)}{D(r)^2}, \tag{69}$$

where the denominator is strictly positive and

$$N(r) = \alpha\phi^{-2}(1-\kappa)^2 - 2\alpha\phi^{-1}(1-\kappa)\kappa\, r + \big[\kappa\big(1-\alpha(1-\kappa)\big)+\phi^{-1}(1-\kappa)(1-\alpha)\big]r^2. \tag{70}$$

The coefficient on r^2 is positive, and the discriminant of this quadratic is

$$\Delta = -4\alpha\phi^{-2}(1-\kappa)^2(1-\alpha)\left[\kappa + \phi^{-1}(1-\kappa)\right] < 0. \tag{71}$$

Hence

$$N(r) > 0 \qquad \text{for all } r > 0, \tag{72}$$

which implies

$$\delta'(r) > 0 \qquad \Longrightarrow \qquad \frac{dr}{d\delta} > 0. \tag{73}$$

Finally, because

$$r = \alpha D_Y k^{\alpha-1} \qquad \text{with} \qquad 0 < \alpha < 1, \tag{74}$$

the capital-to-code ratio k is a decreasing function of r. Therefore

$$\boxed{\frac{dk}{d\delta} < 0}. \tag{75}$$

This establishes that in the Cobb–Douglas steady state, a higher code-retention rate always lowers the capital-to-code ratio.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Abel, A., & Kotlikoff, L. J. (1994). Intergenerational altruism and the effectiveness of fiscal policy: New tests based on cohort data, chapter 7. In *Savings and bequests* (pp. 31–42).

Acemoglu, D., & Autor, D. (2011). Chapter 12—Skills, tasks and technologies: Implications for employment and earnings. In S. Ram (Ed.), *Handbook of labor economics* (Vol. 4, pp. 1043–1171).

Acemoglu, D., Autor, D., & Restrepo, P. (2018). The race between man and machine: Implications of technology for growth, factor shares, and employment. *American Economic Review,108*(6), 1488–1542.

Aghion, P., Jones, B. F., & Jones, C. I. (2017). Artificial intelligence and economic growth.

Altonji, J. G., Hayashi, F., & Kotlikoff, L. J. (1992). Is the extended family altruistically linked? direct tests using micro data. *American Economic Review,82*(5), 1177–98.

Altonji, J. G., Hayashi, F., & Kotlikoff, L. J. (1995). Parental altruism and inter vivos transfers: theory and evidence.

Altonji, J. G., Hayashi, F., & Kotlikoff, L. J. (1997). Parental altruism and inter vivos transfers: Theory and evidence. *Journal of Political Economy,105*(6), 1121–1166.

Arrow, K. J. (1962). The economic implications of learning by doing. *The Review of Economic Studies,3*(June), 155–173.
Aschenbrenner, L. (2024). Situational awareness: The decade ahead. https://situational-awareness.ai. Accessed Sep 2026
Autor, D. H. (2015). Why are there still so many jobs? The history and future of workplace automation. *Journal of Economic Perspectives,29*(3), 3–30.
Autor, D. H., & Dorn, D. (2013). The growth of low-skill service job and the polarization of the U.S. labor market. *The American Economic Review,103*(5), 1553–1597.
Autor, D. H., Dorn, D., Levy, F., & Murnane, R. J. (2003). The skill content of recent technological change: An empirical exploration. *The Quarterly Journal of Economics,118*(4), 1279–1333.
Brynjolfsson, E., & Andrew, M. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company.
Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2024). GPTs are GPTs: Labor market impact potential of LLMs. *Science,384*(6702), 1306–1308.
Goos, M., Alan M., & Anna S. (2011). Explaining job polarization in Europe: The roles of technology, globalization and institutions. *CEP Discussion Papers.*
Hemous, D., & Morten O . (2022). The rise of the machines: Automation, horizontal innovation and income inequality. *CEPR Discussion Papers.*
Katz, L. F., & Margo, R. A (2014). Technical change and the relative demand for skilled labor: The United States in historical perspective.
Keynes, J. M. (1963). Economic possibilities for our grandchildren. In *Essays in persuasion* (pp. 358–373). W.W. Norton & Co.
Leamer, E. E. (1983). Let's take the con out of econometrics. *American Economic Review,73*(1), 31–43.
Leamer, E. E. (1999). Effort, wages, and the international division of labor. *Journal of Political Economy,107*(6), 1127–1162.
Lucas, R. E. (1988). On the mechanics of economic development. *Journal of Monetary Economics,22*(1), 3–42.
Marx, K. (1992). *Capital: A critique of political economy* (p. 1887). Penguin Classics.
Mishel, L., Schmitt, J., & Shierholz, H. (2013). Don't blame the robots. Assessing the job polarization explanation of growing wage inequality.
Poterba, J. M. (1998). The rate of return to corporate capital and factor shares: New estimates using revised national income accounts and capital stock data. In S. Ram (Ed.), *Carnegie-Rochester Conference series on public policy* (Vol. 48, pp. 211–246). Elsevier.
Romer, P. M. (1990). Endogenous technological change. *Journal of Political Economy,98*(5), S71–S102.
Sachs, J.D., & Kotlikoff, L.J. (2012). Smart machines and long-term misery.
Sachs, J. D., Benzell, S. G., & LaGarda, G. (2015). *Robots: Curse or blessing? A basic framework,*. Technical Report: National Bureau of Economic Research.
Schumpeter, J. A., & Cycles, B. (1939). *A theoretical, historical and statistical analysis of the capitalist process*. McGraw Hill Book Co.
Zeira, J. (1998). Workers, machines, and economic growth. *The Quarterly Journal of Economics,113*(4), 1091–1117.
Zuleta, H. (2004). Factor saving innovations and capital income share in OLG models.
Zuleta, H. (2008). Factor saving innovations and factor income shares. *Review of Economic Dynamics,11*(4), 836–851.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Seth G. Benzell[1] · Laurence J. Kotlikoff[2] · Guillermo Lagarda[3]

✉ Laurence J. Kotlikoff
kotlikoff@gmail.com

[1] Argyros School of Business and Economics, Chapman University, MIT Initiative on the Digital Economy, Stanford Digital Economy Lab, Beckman Hall, Orange, California 92866, USA

[2] Department of Economics, Boston University, 270 Bay State Road, Boston, MA 02215, USA

[3] Inter-American Development Bank Washington, D.C., 1300 New York Avenue NW, Washington, D.C. 20577, USA