Broken Ladders: AI, Teamwork, and the Dynamics of Skill Formation in the Workplace*

Zsófia L. Bárány and Miklós Koren†

December 22, 2025

[Preliminary and Incomplete, Please Do Not Distribute]

Abstract

We study AI adoption efficiency in knowledge work using a model of team production and learning. Seniors pick higher-value problems and solve the hardest tasks; juniors learn by working with them. AI can solve problems at a lower cost than juniors, but it lacks the ability to recognize the value of the problems it is solving. Its output has to be reviewed by a human; this supervision time is an important bottleneck in adoption. AI raises output in the short run, but it can limit learning and lower long-run output. Inefficient AI adoption can be the result of seniors not internalizing the value of the mentoring they provide, or juniors wasting the supervision time of seniors with AI-generated work. We propose to measure the key learning mechanisms in knowledge sectors using Hungarian administrative data.

*This research was funded by the European Research Council (ERC Advanced Grant agreement number 101097789) and by the National Research, Development and Innovation Office (Forefront Research Excellence Program contract number 144193). The views expressed in this project are those of the authors and do not necessarily reflect the official view of the European Union, the European Research Council, or the National Research, Development and Innovation Office.

†*Bárány*: Central European University, CEPR, Hungarian Society of Economics and ROA. baranyzs@ceu.edu *Koren*: Central European University, HUN-REN KRTK, CEPR and CESifo. korenm@ceu.edu

1 Introduction

Artificial intelligence (AI), especially generative AI, is reshaping how work is organized. Evidence from the U.S. Census Bureau suggests that 12–14% of large businesses already use AI tools in their operations (Census 2025). A growing concern is that these tools will displace entry-level jobs. Major technology companies—including Microsoft, Amazon, and Anthropic—have reportedly reduced hiring of interns and junior employees, choosing instead to rely on AI for tasks that junior staff used to perform. A similar mechanism might be driving the fact that employment in the youngest age group fell, while employment in other age groups did not (Brynjolfsson et al. 2025). This pattern extends beyond technology: senior lawyers now use AI to draft contracts, and expert developers leverage AI coding agents rather than delegating to junior colleagues.

In knowledge organizations, juniors learn from seniors through apprenticeship. A junior collaborating with a senior acquires expertise—technical skills and judgment about what problems matter—to become a senior. If AI replaces juniors in production, it improves the productive efficiency of seniors in the short run. But AI also threatens to break the career ladder, halting the development of human skills and potentially lowering long-run output.

We develop a model to study this trade-off. Building on the framework of hierarchical team production (Garicano 2000, Garicano and Rossi-Hansberg 2006), we add two key features. First, seniors and juniors differ not only in problem-solving skills but also in *value recognition*—the ability to identify which problems generate high revenue. Second, juniors can learn both skills by working alongside seniors (*learning-by-mentoring*). We model AI as a tool with zero marginal cost that can solve problems up to some difficulty level, but lacks value recognition and requires human supervision. This supervision creates a binding resource constraint: AI adoption scales only up to what humans can oversee.

The Michelangelo thought experiment

To build intuition, consider a sculptor's workshop in Renaissance Florence. Senior sculptors—masters like Michelangelo—create beautiful, life-like sculptures that sell for $y = 2$ Golden Ducats. Junior sculptors can do preparatory work, carving marble into rough shapes. Working solo in the suburbs, a junior produces crude sculptures selling for 1 Ducat each.

Crucially, Michelangelo knows which commissions to accept. When the Medici request a tomb sculpture, he recognizes this as a 2-Ducat opportunity and commits. When a minor guild offers the same nominal fee for a decorative fountain, he knows the project will drag on, specifications will change, and payment will never arrive. An apprentice would accept both indiscriminately—or worse, spend months on a cancelled project. This ability to identify which problems are worth solving is *value recognition*. It is distinct from the technical skill of carving marble: a junior can learn to carve beautifully, but learning to read patrons, anticipate their tastes, and select profitable commissions takes years of experience working alongside a master.

Michelangelo can finish $n = 3$ sculptures per year if he does not have to carve the marble himself. With three apprentices, each carving one marble block, the workshop produces 6 Ducats of output. Junior wages are pinned by their outside option (1 Ducat), so Michelangelo earns $3 \times 2 - 3 \times 1 = 3$ Ducats—triple what a solo junior earns. Apprentices also learn: by year two, they become masters themselves. This is the career ladder.

Now Leonardo invents a marble-carving machine, rented at zero cost. In one scenario, Michelangelo fires all apprentices and uses the machine directly. He still produces 6 Ducats of output, but now keeps it all—his earnings double to 6 Ducats. This is the static gain from AI. But no one is learning. In year two, there is still only one master in Florence instead of four.

In another scenario, juniors also have access to the machine. Each junior with AI can carve 3 marble blocks per year instead of one. A junior approaches Michelangelo: "I can give you 3 carved marbles per year. Pay me 3 Ducats"—the new outside option. Michelangelo is indifferent; he used to pay 1 Ducat each to 3 juniors, now he pays 3

Ducats to a single, more productive junior. The economics are unchanged. But only one person watches Michelangelo work. Only one person learns. Only one person becomes a master. The career ladder has shrunk.

Note what the marble-carving machine cannot do: it cannot judge which commissions Michelangelo should accept. The machine prepares marble as skillfully as any apprentice, but has no sense of which patrons are worth cultivating. Even with a perfect carving machine, Michelangelo must still review each block, decide which projects to pursue, and manage client relationships. This is why the senior's capacity constraint persists: AI does not free up Michelangelo's time for more projects, because the bottleneck was never the carving. The bottleneck is his judgment about what to carve.

Two sources of inefficiency

The thought experiment reveals two sources of inefficiency. First, when Michelangelo fires apprentices to use AI directly, he does not internalize the value of the mentoring he provides. This is a pecuniary externality: if juniors could pay for training (via wage cuts), the externality would be internalized. Second, when juniors use AI and become more productive, fewer are needed in teams. Learning and work are bundled together in the apprenticeship model—you learn by being present while you work. AI unbundles productivity from learning: a junior with AI produces more, but does not learn faster. In fact, because teams shrink, fewer juniors learn at all.

Wage adjustments alone cannot restore the first-best. Knowledge is non-rival: training three apprentices is not three times as costly as training one. With AI, optimal production calls for small teams (each junior handles more problems), but optimal learning calls for large teams (many juniors watch the master). These objectives conflict. Part-time apprenticeships could help—each junior works some time for Michelangelo and some time solo with AI—but require divisible jobs and a learning technology that scales linearly with time. If there is a minimum time threshold for effective learning, or if matching frictions prevent such arrangements, the bundling cannot be undone.

Main results

We formalize these intuitions in a model where seniors have both higher problem-solving skills ($z_1 > z_0$) and better value recognition ($v_1 > v_0$). AI has skill z_A but no value recognition; all of its output must be reviewed by a human. AI does not resolve the senior's capacity constraint: the maximum number of junior problems a senior can handle is unchanged. Therefore, team output stays the same when juniors use AI—teams simply shrink.

Our main efficiency results are:

- *Without learning*, AI adoption is always efficient. Seniors' decisions to maintain or dissolve teams maximize GDP.
- *With external learning*, such that juniors learn but do not internalize the gains, seniors may over-adopt AI. They might dissolve teams even when doing so lowers long-run output.
- *With internalized learning*, such that juniors accept wage cuts that fully internalize the gains from learning, seniors' decisions are efficient. However, AI can still be dynamically inefficient—it raises short-run output but lowers the long-run senior share enough that steady-state GDP falls.

Dynamic inefficiency is most likely when learning is fast, productivity gaps between seniors and juniors are large, and AI's static gains are modest. As AI improves, static gains grow and inefficiency becomes less likely. The value recognition gap persists even with perfect problem-solving AI, so human mentorship retains value.

Related literature

Our model highlights a dynamic externality in AI adoption via lost learning opportunities. This connects to the literature on learning-by-doing and human capital formation (Arrow 1962; Lucas 1988). In those models, workers or firms do not internalize the benefit

of accumulated skills. Acemoglu and Restrepo (2018, 2020) emphasize that automation must be sufficiently productive and counterbalanced by new tasks, otherwise displacement costs outweigh gains. Beraja and Buera (2024) show that dynamic competition can lead private incentives to deviate from social optima, though not always—some equilibria are constrained-efficient. Our welfare results have a similar flavor: private incentives can deviate from social optima for seniors if learning is not priced, and for juniors because the higher productivity of juniors with AI crowds out other juniors from learning.

Recent empirical work explores AI's impact on training. Hess et al. (2023) find reduced training in high-automation-risk jobs. Mühlemann (2024) documents that German firms adopting AI cut training for current workers but increased apprenticeships. Brynjolfsson et al. (2025) show that generative AI can accelerate learning: novice customer support workers improved with AI assistance. Noy and Zhang (2023) find similar effects for writing. These studies suggest AI could serve as a training device rather than a substitute for mentorship—an avenue we do not explore but that could mitigate the dynamic losses we identify.

The remainder of the paper proceeds as follows. Section 2 presents the model, we then analyze equilibrium without learning in Section 3 and with learning in Section 4, and we discuss sources of inefficiency in Section 4.3. Section 5 concludes.

2 Model setup

This section presents the model, describing the environment.

Problems and worker types. Problems have varying difficulties and values. Formally, let task difficulty Z be uniformly distributed on $[0, 1]$. A worker's skill level $z \in (0, 1)$ represents the hardest problem that they can solve: a person with skill z_i can solve all problems with difficulty $Z \leq z_i$ with certainty, but cannot solve any problem with $Z > z_i$.

We assume that there are two types of workers, who differ in their skill levels: juniors with skill z_0 and seniors with skill z_1, where $z_1 > z_0$. Seniors and juniors also differ in

their ability to recognize problem value, v_i. The value of a problem is the revenue it generates if solved. Seniors draw problems with higher expected values than juniors, that is $v_1 > v_0$. This could be micro-founded by individuals drawing problems from some value distribution and deciding whether to attempt to solve them or to discard them: if seniors are better at judging value, they end up with higher expected value of attempted problems.

Demographics and learning. Time is continuous. At any given point in time, δL people are born (we normalize $L = 1$). A fraction ϕ of individuals are born with senior skills and value recognition (z_1, v_1), and fraction $1 - \phi$ with junior attributes (z_0, v_0). All individuals die with a Poisson arrival rate of δ, independent of skill.

The model allows for stochastic learning by juniors, in which case their skill level changes to z_1 and their value recognition to v_1. We consider two learning scenarios, no learning and learning-by-mentoring. In the no learning scenario, the skill distribution is static. In the learning-by-mentoring scenario, juniors working in teams learn from seniors at rate $\lambda(n)$, where n is the size of the team of juniors that the senior leads.

Working solo. Working on a problem requires time, committed before knowing the difficulty. One problem takes one unit of time to work on. If a person is working alone, called solo work, they can solve it with probability z_i, so their expected output per unit of time is $y_i \equiv v_i z_i$. These values also pin down their solo wage in a competitive market: working alone, a junior earns $w_{0,solo} = y_0$ per unit of time, and a senior earns $w_{1,solo} = y_1$.

Working in teams. Now consider teamwork: a senior can collaborate with several juniors. Seniors draw problems for everyone on their team, at expected value v_1, and juniors attempt to solve all the problems first. Juniors solve the easier problems that they are capable of, and escalate the unsolved harder problems, those with $Z > z_0$ to the senior. The senior then spends time handling those tougher problems.

We assume that whenever a junior brings a problem to the senior, the senior spends

$h < 1$ units of time on it, whether or not the senior eventually manages to solve it. The parameter h captures the time cost per problem of communication, mentoring, and problem-solving. The condition $h < 1$ means it is more efficient for a senior to solve a problem brought by a junior than to solve it alone. Intuitively, the junior filters and only forwards the harder subset of problems to the senior. The senior's time constraint requires that, in expectation, they must be able to handle all the problems that juniors send to them.

Working with AI. AI systems work like subordinates in a team: humans give problems to the AI, the AI attempts to solve all problems, and passes on the output to the human. The AI provides a solution to all problems, but it can solve problems only up to difficulty z_A. Because the AI pretends to solve all problems, each attempted problem by the AI requires human time. For problems with $Z \leq \min\{z_A, z_i\}$, the review takes $h_{A1} < 1$ time. This captures the insight that when the AI provides a correct solution *and* the difficulty of the problem does not exceed the human's skill level, the review process is fast. This review time is important, because it captures the resource constraint that AI creates through the need for human verification. For problems with $Z > \min\{z_A, z_i\}$, reviewing and solving takes $h_{A2} \in (h_{A1}, 1]$ time. When the AI cannot provide a correct solution *or* the difficulty of the problem is above the human's skill level, reviewing and/or solving the problem takes long. This implies that workers can effectively use AI capabilities up to their own skill level.

3 Equilibrium in the no learning economy

In our baseline model there is no learning: each individual keeps the skill they were born with. This is a static model, where the measure of juniors is $L_0^{NL} = 1 - \phi$ and the measure of seniors is $L_1^{NL} = \phi$ at all times. We first analyze this economy without AI, then introduce AI and study how it affects outcomes.

3.1 Equilibrium without AI

Solo output and wages. As discussed before, when working solo, expected output equals the probability of solving a random problem times its expected value:

$$Q_{i,solo}^{NL} = w_{i,solo}^{NL} = v_i z_i = y_i. \tag{1}$$

This says that a solo junior produces $y_0 = v_0 z_0$ and a solo senior produces $y_1 = v_1 z_1$ per unit of time.

Team size and output. The senior's time constraint pins down the maximum team size. Since juniors pass on problems with difficulty $Z > z_0$, which occurs with probability $1-z_0$, and each escalated problem takes h of the senior's time, a senior with n_0 juniors spends $n_0(1-z_0)h$ time on escalated problems. Setting this equal to the senior's total time yields:

$$n_0 = \frac{1}{h(1-z_0)}. \tag{2}$$

This is the *maximum number of junior problems* that a senior can handle. Team output is the sum of the value of problems that juniors solve and those that the senior solves:

$$Q_{team}^{NL} = n_0 v_1 z_0 + n_0 v_1 (1-z_0)\frac{z_1 - z_0}{1-z_0} = n_0 v_1 z_1 = \frac{y_1}{h(1-z_0)}. \tag{3}$$

The senior essentially multiplies their expertise across n_0 problems by hiring n_0 juniors. Seniors can be highly productive when supported by a team: if communication is more efficient, h is smaller, and juniors pass on only the harder problems, z_0 is larger, a senior can leverage a larger team.

Participation constraint. Teamwork is better than solo work if team output exceeds the sum of individual outputs, $n_0 y_0 + y_1$:

$$\frac{y_1}{y_0} > \frac{n_0}{n_0 - 1} = \frac{1}{1 - h(1 - z_0)}. \tag{PC}$$

This requires the senior's productivity to be sufficiently large relative to the junior's productivity. The threshold depends on the efficiency of teamwork: if h is smaller or z_0 is larger, teamwork is more efficient, and the productivity gap between seniors and juniors need not be as large. We assume that this *participation constraint* holds throughout.

Labor market equilibrium. If teamwork is better than solo work, as many teams form as possible. We assume that *juniors are abundant*, meaning that all seniors can lead teams of n_0 juniors, and there are some juniors who cannot join teams. This case arises if $L_0 > n_0 L_1$, which requires $\phi < h(1 - z_0)/[1 + h(1 - z_0)]$. We assume thorughout that this condition holds. The excess juniors work solo and earn $w^{NL}_{0,solo} = y_0$. Competition among juniors implies that juniors in teams also earn y_0; if offered less, they would work solo; if offered more, juniors outside of teams would undercut them. In this case, seniors extract all the surplus and their wage is given by

$$w^{NL}_{1,team} = Q^{NL}_{team} - n_0 w^{NL}_{0,solo} = n_0(y_1 - y_0) = \frac{y_1 - y_0}{h(1 - z_0)}. \tag{4}$$

Seniors prefer teams over solo work if $w^{NL}_{1,team} > w^{NL}_{1,solo}$, which holds exactly when the participation constraint (PC) is satisfied.

Aggregate output and efficiency. Aggregate output is the sum of team output and solo output of juniors who could not join teams:

$$Y^{NL}_{teams} = L^{NL}_1 n_0 y_1 + (L^{NL}_0 - n_0 L^{NL}_1) y_0 = \phi n_0(y_1 - y_0) + (1 - \phi) y_0, \tag{5}$$

which is equal to total wages paid to seniors and to juniors. If everyone worked solo, output would be $Y_{solo}^{NL} = \phi y_1 + (1-\phi)y_0$. Comparing these immediately yields that aggregate output with teams is higher than aggregate output from solo work if and only if (PC) holds. This means that *team formation is efficient*: the participation constraint that makes seniors willing to form teams is exactly the condition under which team formation raises GDP.

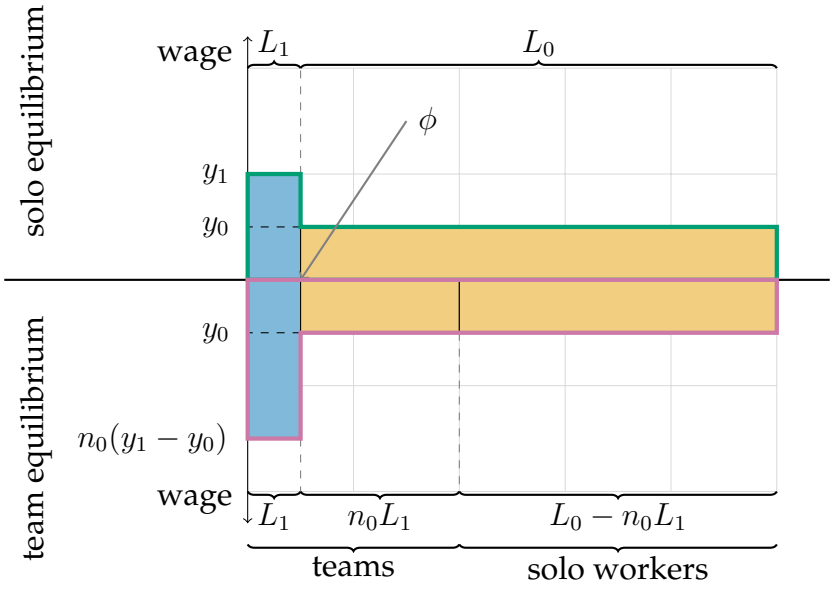


Figure 1: Output and wages in the baseline economy

Notes: The horizontal axis shows the share of seniors and juniors in the economy. The vertical axis shows wages when working solo above the horizontal axis, and when working in teams below the horizontal axis.

Figure 1 illustrates the equilibrium. The horizontal axis represents the share of seniors, ϕ, and the share of juniors, $1-\phi$. The top part, above the horizontal axis, shows on the vertical axis income for all workers when everyone works solo: seniors earn y_1, shown in blue, and juniors earn y_0, shown in yellow. Aggregate output is the sum of all earnings and equals Y_{solo}^{NL}, and is shown with the green outline. The bottom part shows on the vertical axis income for all workers in case of team formation. Each senior hires n_0 juniors, and seniors earn $n_0(y_1 - y_0)$ (in blue), while juniors in teams as well as when solo earn y_0 (in yellow). Aggregate output is the sum of total earnings and equals Y_{teams}^{NL}, and is shown

with the red outline. When the participation constraint holds team output exceeds solo output, because due to (PC) senior wage in teams exceed senior solo wage, and juniors earn the same whether teams are formed or not.

3.2 Equilibrium with AI

Now suppose AI is available to all workers, with capability $z_A \leq z_0$. Recall that for problems with difficulty $Z \leq \min\{z_A, z_i\}$ it takes h_{A1} time to review AI's output. For problems exceeding this difficulty, the time required is h_{A2}, and only problems up to difficulty z_i are solved. Therefore, given the capacity of each worker, if $z_A \leq z_0$, each can handle

$$n_A = \frac{1}{h_{A1} z_A + h_{A2}(1 - z_A)} \tag{6}$$

AI instances. Solo output and wages with AI are therefore

$$Q_{i,solo}^{NL,AI} = w_{i,solo}^{NL,AI} = n_A z_A v_i + n_A (1 - z_A) \frac{z_i - z_A}{1 - z_A} v_i = n_A y_i. \tag{7}$$

AI amplifies skills: when $n_A > 1$, both juniors and seniors produce and earn more when using AI than without.

Impact on team work. When juniors can use AI in teams, each junior can handle at most n_A problems with AI, passing on problems with $Z > z_0$ to the senior. The senior's capacity constraint still binds: the senior can handle at most n_0 escalated problems that have been previously handled by juniors. To minimize wage costs, the senior asks juniors to use as many AI instances as they can handle. Since each junior now escalates $n_A(1 - z_0)$ problems, the capacity constraint of the senior implies a lower team size than without AI:

$$n_0^{AI} = \frac{n_0}{n_A} = \frac{h_{A1} z_A + h_{A2}(1 - z_A)}{h(1 - z_0)}. \tag{8}$$

Team output remains unchanged:

$$Q_{team}^{NL,AI} = n_0^{AI} n_A z_A v_1 + n_0^{AI} n_A (1 - z_A) \frac{z_0 - z_A}{1 - z_A} v_1 + n_0^{AI} n_A (1 - z_0) \frac{z_1 - z_0}{1 - z_0} v_1 = n_0 y_1. \quad (9)$$

Therefore AI does not resolve the senior's capacity constraint. The maximum number of junior problems the senior can handle remains n_0. AI simply allows each junior to process more problems before escalating, which means fewer juniors are needed for the same team output. But since each junior's wage is now higher due to AI increasing their productivity, seniors' team wage also remains unchanged:

$$w_{1,team}^{NL,AI} = Q_{team}^{NL,AI} - n_0^{AI} w_{0,solo}^{NL,AI} = n_0 (y_1 - y_0).$$

Seniors' decision to maintain or dissolve teams. Seniors compare their team wage to their solo wage with AI. They choose to dissolve teams if

$$n_A y_1 > n_0 (y_1 - y_0). \quad \text{(DISS)}$$

When this condition holds, seniors prefer to work alone with AI rather than lead teams of juniors who use AI.

Efficiency of seniors' decision. If seniors maintain teams, aggregate output is:

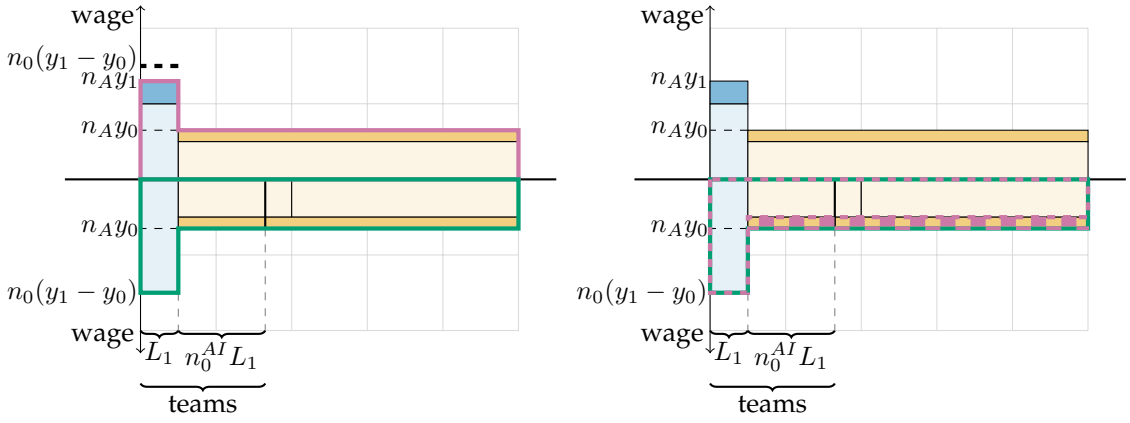
$$Y_{teams}^{NL,AI} = \phi n_0 (y_1 - y_0) + (1 - \phi) n_A y_0.$$

If seniors dissolve teams, aggregate output is:

$$Y_{solo}^{NL,AI} = \phi n_A y_1 + (1 - \phi) n_A y_0.$$

Aggregate output when everyone works solo exceeds aggregate output from teamwork if and only if seniors' solo wage is higher than their wage in teams. This implies that

seniors' decision is efficient: they maintain teams exactly when doing so maximizes GDP.



(a) Seniors' decision (b) Efficiency of AI

Figure 2: Impact of AI on output.

Notes: The horizontal axis shows the share of seniors and juniors in the economy. The vertical axis shows wages when working solo above the horizontal axis, and when working in teams below the horizontal axis. The left figure compares aggregate solo output with AI to aggregate team output with AI. The right figure compares aggregate team output without AI to aggregate output with AI.

The efficiency of seniors' decision is illustrated in Figure 2a. The darker blue and yellow regions show that solo output increases both for seniors to $n_A y_1$ and for juniors to $n_A y_0$. Below the horizontal axis we can see that the wages of solo and team juniors increases to $n_A y_0$. Senior team wages remain unchanged in the economy with AI use, but the size of teams shrinks to n_0^{AI}, implying fewer workers in teams. Since seniors decide to maintain teams if and only if $n_0(y_1 - y_0) > n_A y_1$, and in this case aggregate output is also higher with teams (shown in green outline) than without (shown with red outline), the seniors' decision is efficient.

Efficiency of AI adoption. To evaluate whether AI use is efficient, we need to compare aggregate output with AI, whether teams are maintained or dissolved, to output without AI. The income of solo and team juniors is strictly higher with AI than without, as $n_A y_0 > y_0$. Senior wage is weakly higher with AI. If teams are maintained, then senior wages are the same as before, whereas if teams are dissolved, then senior wages are strictly higher with AI, as that is the condition for dissolving teams. Therefore, AI adoption is always

efficient in the no-learning economy. This is shown in Figure 2b, for a case where with AI teams are maintained. If teams are maintained, senior wage, and total senior income is the same as without AI. But all juniors' productivity and earnings are higher with AI than without, which leads to a gain in aggregate output, indicated by the red checkered area.

Result 1: Without learning, seniors make efficient decisions about whether to maintain or dissolve teams, and AI adoption is always efficient.

AI capability improvements. The impact of improvements in AI capability depends on the level of z_A relative to human skills. As long as $z_A \leq z_0$, junior and senior solo productivity with AI increases as z_A increases, and the size of teams shrinks further. At $z_A = z_0$, junior AI use is maximized at $n_{A0} = 1/[h_{A1}z_0 + h_{A2}(1 - z_0)]$, and no further improvement in junior productivity can be obtained. For AI improvements in the $z_0 < z_A \leq z_1$ range, senior solo productivity with AI keeps on improving, and is maximized at $n_{A1} = 1/[h_{A1}z_1 + h_{A2}(1 - z_1)]$. Beyond this point, AI improvements do not generate any productivity gains, as verification becomes the bottleneck. If $n_{A1}y_1 > n_0(y_1 - y_0)$, then there exists a threshold AI capability z_A^* at which seniors optimally decide to dissolve teams.

4 Equilibrium in the economy with learning

We now add learning by assuming that juniors can become seniors over time if they work in teams. This captures career progression through on-the-job learning: a junior who collaborates with a senior gradually acquires senior-level skills and value recognition. We model this as a Poisson process: while working in a team, a junior becomes a senior at rate λ.[1]

[1]For now assume that λ is a constant, and in particular that it is independent from the size of teams, n.

4.1 Steady state without AI

The stock of seniors evolves according to:

$$\dot{L}_1 = \phi\delta - \delta L_1 + \lambda L_1 n_0,$$

where $\phi\delta$ is the inflow of individuals born as seniors, δL_1 is the outflow due to death, and $\lambda L_1 n_0$ is the inflow from learning. We assume that juniors remain abundant even in the economy with learning, that is $L_0 > n_0 L_1$ at all times, implying that each senior has a full team of seniors, and $L_1 n_0$ juniors work in teams. The steady state share of seniors is given by

$$L_1^* = \frac{\phi}{1 - \frac{\lambda}{\delta}\frac{1}{h(1-z_0)}} = \frac{\phi}{1 - \frac{\lambda}{\delta}n_0}. \tag{10}$$

This is well-defined and juniors remain abundant if $\lambda/\delta < (1-\phi)h(1-z_0) - \phi$. Learning by mentoring raises the long-run proportion of seniors: $L_1^* > \phi$. The faster the learning rate λ or the larger teams are, higher n_0, the greater the boost to human capital.

Aggregate output with learning. Steady state output with learning is the sum of total team output and total solo junior output:[2]

$$Y_{teams}^{*LE} = L_1^* n_0 y_1 + (1 - L_1^* - L_1^* n_0) y_0 = L_1^* n_0 (y_1 - y_0) + (1 - L_1^*) y_0. \tag{11}$$

It can be verified that this is equal to the sum of earnings in the no-learning economy, $w_{1,team}^{NL}$ for seniors, and $w_{0,solo}^{NL}$ for all juniors, but with a higher senior share. Note that when there is learning, junior wages in teams are not necessarily equal to solo junior wages. If juniors anticipate that by working in teams they might become seniors, they can internalize these gains and accept a wage cut as team juniors in exchange for learning. Whether learning gains are internalized or not does not change aggregate output, it just

[2]Calculated as the number of seniors times team output, plus the number of solo juniors times solo junior output.

changes how team output is split between the senior and the team of juniors.

Wages when learning is fully internalized. Suppose juniors anticipate career progression and accept lower wages for learning opportunities. In a competitive labor market with forward-looking workers and abundant juniors, a junior's expected lifetime utility from working in a team should equal that from working solo.

Let $J_{0,team}$ denote the expected present value of being a junior in a team:

$$\delta J_{0,team} = w_{0,team} + \lambda \left[J_{1,team} - J_{0,team} \right],$$

where $J_{1,team} = w_{1,team}/\delta$ is the present value of earnings as a senior. Solving:

$$J_{0,team} = \frac{w_{0,team} + \frac{\lambda}{\delta} w_{1,team}}{\delta + \lambda}.$$

The value of being a solo junior is $J_{0,solo} = y_0/\delta$. Indifference for juniors implies $J_{0,team} = J_{0,solo}$, leading to

$$w_{0,team} = y_0 - \frac{\lambda}{\delta}(w_{1,team} - y_0). \tag{12}$$

This shows that juniors in teams accept a wage *below* their solo marginal product y_0, because they expect to recoup the difference when they become seniors. We can write this wage cut as $\Delta = \frac{\lambda}{\delta}(w_{1,team} - y_0)$, so $w_{0,team} = y_0 - \Delta$. Juniors effectively pay for training via a wage discount.

The senior's wage is team output minus wages paid to juniors:

$$w_{1,team} = n_0(y_1 - w_{0,team}),$$

from which after substituting the expression for $w_{0,team}$ and solving, we get that senior

wage in teams is

$$w_{1,team}^{IL} = \frac{n_0\left(y_1 - y_0(1+\frac{\lambda}{\delta})\right)}{1-\frac{\lambda}{\delta}n_0}. \tag{13}$$

This exceeds the senior wage without internalized learning, $n_0(y_1 - y_0)$, because juniors accept wage cuts. The senior wage is increasing in the learning speed λ/δ, as faster learning means juniors accept larger wage cuts. Junior wage in teams is given by

$$w_{0,team}^{IL} = y_0 - \frac{\lambda}{\delta}\frac{n_0(y_1 - y_0) - y_0}{1-\frac{\lambda}{\delta}n_0}.$$

Juniors' wage cannot be negative, thus the wage cut cannot be larger than y_0. The condition under which wages can fully internalize the gains from learning is

$$\frac{y_1}{y_0} < \frac{1}{n_0}\frac{1+\frac{\lambda}{\delta}}{\frac{\lambda}{\delta}}.$$

If this condition does not hold, then juniors will work for free in teams, and wages do not fully internalize the gains from learning.

Efficiency of team formation with learning. Steady state aggregate output if teams are formed, Y_{teams}^{*LE}, exceeds steady state solo output, Y_{solo}, if

$$\frac{\phi}{1-\frac{\lambda}{\delta}n_0}n_0(y_1 - y_0) + \left(1 - \frac{\phi}{1-\frac{\lambda}{\delta}n_0}\right)y_0 > \phi y_1 + (1-\phi)y_0,$$

which can be shown to be equivalent to $w_{1,team}^{IL} > w_{1,solo}$. This implies that as long as wages fully internalize the dynamic gains from learning, seniors' decision to form teams is dynamically inefficient: they form teams precisely when team formation raises long-run GDP. However, if wages do not fully internalize learning gains, for example because learning gains are too large relative to junior solo productivity, then teams might not be formed, even though steady state aggregate output would be higher if teams were

formed.

Result 2: With internalized learning, team formation is efficient. The wage mechanism internalizes the value of mentoring.

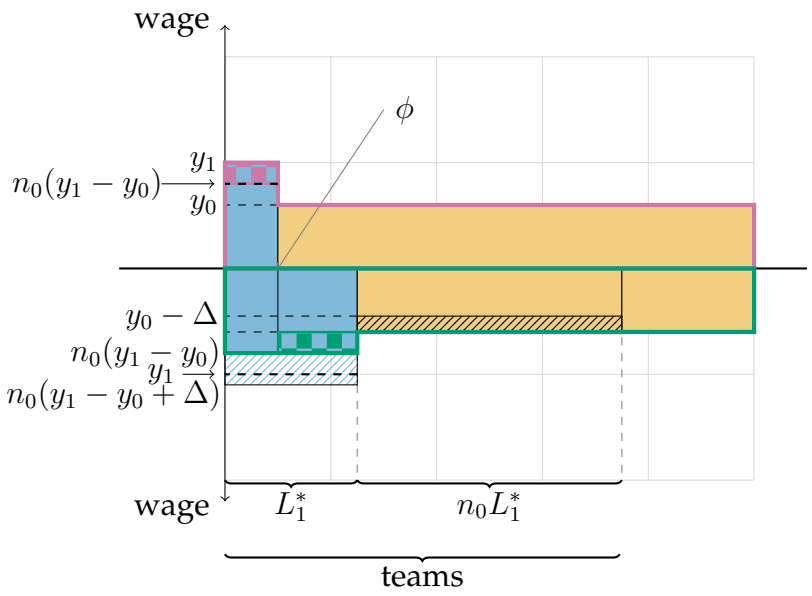


Figure 3: Learning-by-mentoring and internalized learning.

Notes: The horizontal axis shows the share of seniors and juniors in the economy. The vertical axis shows wages when working solo above the horizontal axis, and when working in teams below the horizontal axis.

Figure 3 shows the steady state of the economy if everyone works solo above the horizontal axis, and the steady state of the economy if teams are formed. The senior solo wage is y_1, junior solo wage is y_0. The red outline shows steady state aggregate solo output, Y_{solo}. If there is learning, then working in teams expands the senior stock to $L_1^* > \phi$. If wages do not internalize learning gains, then senior team wage is $n_0(y_1 - y_0)$, and junior team and solo wage is y_0. The green outline shows steady state aggregate output if working in teams, Y_{teams}^{*LE}. With internalized learning, juniors accept a wage cut Δ (yellow hatched area) in exchange for career progression, while seniors receive a wage premium (blue hatched area). The senior team wage rises to $n_0(y_1 - y_0 + \Delta)$. In this configuration, steady state output is higher if teams are formed, which can be seen by comparing the red checkered area (the gain from higher senior solo wages) to the green

checkered area (the gain from having more seniors with teams). If learning gains are not sufficiently internalized in wages, then since $y_1 > n_0(y_1 - y_0)$ seniors choose not to form teams. However, if wages fully internalize learning gains, senior wages are $n_0(y_1 - y_0 + \Delta) > y_1$, and therefore seniors prefer to lead teams of juniors rather than working solo. The wage cut Δ exactly internalizes the dynamic gains from team formation, making team formation efficient.

4.2 Equilibrium with AI and learning

Now suppose AI, with $z_A \leq z_0$, arrives in an economy at its learning steady state with teams. Solo juniors always use AI, as $n_A > 1$, raising their outside option to $n_A y_0$. Two things can happen, either seniors maintain teams, but teams shrink and seniors hire $n_0^{AI} = n_0/n_A$ juniors, or seniors decide to dismiss teams and use AI solo. In either case the inflow of seniors decreases, either learning slows down because fewer juniors work in teams, or learning ceases altogether. In what follows we analyze whether seniors make the efficient decision, and whether AI use in general is dynamically efficient.

Outcomes if teams are maintained. If teams are maintained, the steady state share of seniors is:

$$L_1^{*AI} = \frac{\phi}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}} < L_1^*. \tag{14}$$

The senior share is lower with AI because each team has fewer juniors learning.

Steady state output in teams is the sum of total team output and total solo junior output:[3]

$$Y_{teams}^{*LE,AI} = L_1^{*AI} n_0 y_1 + (1 - L_1^{*AI} - L_1^{*AI}\frac{n_0}{n_A}) n_A y_0 = L_1^{*AI} n_0 (y_1 - y_0) + (1 - L_1^{*AI}) n_A y_0. \tag{15}$$

[3]Calculated as the number of seniors times team output plus the number of solo juniors times solo junior output.

Which again can be expressed as the sum of earnings in the no-learning economy, $w^{NL}_{1,team}$ for seniors, and $w^{AI}_{0,solo}$ for juniors, but with the relevant steady state senior share. Note that again, team output can be split in different ways between seniors and team juniors. If workers do not internalize the gains from learning, then team juniors earn $n_A y_0$, and seniors keep the rest of team output, $n_0(y_1 - y_0)$. If workers internalize the gains, then team junior wages will be lower, and senior wages will be higher.

When learning gains are fully internalized in wages, following the same steps as before, we can show that junior team wage with AI satisfies:

$$w_{0,team} = n_A y_0 - \frac{\lambda}{\delta}(w_{1,team} - n_A y_0),$$

while senior team wage is given by $w_{1,team} = n_0 y_1 - n_0^{AI} w_{0,team}$, and solving for the senior wage we get that:

$$w^{IL,AI}_{1,team} = \frac{n_0\left(y_1 - y_0(1 + \frac{\lambda}{\delta})\right)}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}}. \tag{16}$$

The senior's team wage is higher when learning is internalized, making dissolution less attractive. Senior wage is increasing in the learning speed λ/δ, in the number of junior problems the senior can handle, n_0, and is decreasing in the productivity enhancement of AI, n_A. The junior team wage that fully internalizes learning gains is given by

$$w^{IL,AI}_{0,team} = n_A y_0 - \frac{\lambda}{\delta}\frac{n_0(y_1 - y_0) - n_A y_0}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}}.$$

For junior wages to fully internalize learning gains, the wage cut cannot exceed junior solo wages, $n_A y_0$, as juniors cannot earn negative wages. This boils down to

$$\frac{y_1}{y_0} < \frac{n_A}{n_0}\frac{1 + \frac{\lambda}{\delta}}{\frac{\lambda}{\delta}}.$$

Seniors dissolve teams if $n_A y_1 > w_{1,team}^{IL,AI}$, which boils down to the following:

$$\frac{\lambda}{\delta} < \frac{n_A y_1 - n_0(y_1 - y_0)}{n_0(y_1 - y_0)}. \tag{17}$$

Efficiency of seniors' decision. To determine whether seniors' decision is efficient we need to compare $Y_{teams}^{*LE,AI}$ with $Y_{solo}^{AI} = \phi n_A y_1 + (1-\phi) n_A y_0$. Steady state aggregate team output exceeds solo output if

$$\frac{\phi}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}} n_0(y_1 - y_0) + \left(1 - \frac{\phi}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}}\right) n_A y_0 > \phi n_A y_1 + (1-\phi) n_A y_0,$$

which can be shown to be equivalent to $w_{1,team}^{IL,AI} > w_{1,solo}^{AI}$. Therefore, if wages fully internalize the gains from learning, senior's decision to form teams is dynamically efficient. On the other hand, if wages do not fully internalize the gains from learning, then seniors might choose to dissolve teams even when team formation would raise long-run GDP.

Result 3: With internalized learning, seniors' AI adoption decisions are efficient. If learning gains are not fully internalized, seniors may over-adopt AI and dissolve teams, even if it lowers output in the long-run.

Efficiency of AI adoption. Even if seniors' decision is efficient, given AI availability, AI use itself may be dynamically inefficient. To determine whether this is the case, we need to compare long-run output with AI to long-run output without AI.

When teams are maintained with AI. AI use is dynamically inefficient if $Y_{teams}^{*LE,AI} < Y_{teams}^{*LE}$:

$$\frac{\phi}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}} n_0(y_1 - y_0) + \left(1 - \frac{\phi}{1 - \frac{\lambda}{\delta}\frac{n_0}{n_A}}\right) n_A y_0 < \frac{\phi}{1 - \frac{\lambda}{\delta} n_0} n_0(y_1 - y_0) + \left(1 - \frac{\phi}{1 - \frac{\lambda}{\delta} n_0}\right) y_0,$$

which boils down to

$$(1-\phi)n_A\left(1-\frac{\lambda}{\delta}\left(1+\frac{1}{n_A}\right)\right) < \frac{\lambda}{\delta}n_0^2\left(\phi\frac{y_1-y_0}{y_0}+1\right).$$

The gain of AI comes from higher solo junior productivity, $n_A y_0 > y_0$, while the loss comes from there being fewer seniors, $L_1^{*AI} < L_1^*$. AI use is dynamically inefficient if the loss dominates the gains. Dynamic inefficiency of AI is increasing in the learning speed, λ/δ, the productivity gap, y_1/y_0, seniors' capacity to handle junior problems, n_0, share of born seniors, ϕ, and is decreasing in AI's productivity multiplier, n_A.

When teams dissolve with AI. AI use is dynamically inefficient if $Y_{solo}^{AI} < Y_{teams}^{*LE}$:

$$\phi n_A y_1 + (1-\phi)n_A y_0 < \frac{\phi}{1-\frac{\lambda}{\delta}n_0}n_0(y_1-y_0) + \left(1-\frac{\phi}{1-\frac{\lambda}{\delta}n_0}\right)y_0,$$

which is equivalent to

$$\frac{\lambda}{\delta}n_0 > 1-\phi\frac{n_0(y_1-y_0)-y_0}{n_A\phi(y_1-y_0)+(n_A-1)y_0}.$$

For juniors to be abundant in the learning economy's steady state, we required that $n_0\lambda/\delta < 1-\phi$. For the above condition to hold it is required that $\frac{n_0(y_1-y_0)-y_0}{n_A\phi(y_1-y_0)+(n_A-1)y_0}$ be large enough. This term is increasing in n_0 and y_1/y_0, and decreasing in n_A and ϕ.

Result 4: AI can be dynamically inefficient even when adoption decisions are privately optimal. This occurs when learning is fast, productivity gaps and teams without AI are large, and AI's static gains are modest.

The table below summarizes the efficiency of seniors' decision and of AI adoption in the different model specifications.

Table 1: Summary of Efficiency Results

Learning regime	**Pricing regime**	**Efficiency of**	
		Seniors' decision	**AI use**
No learning		Yes	Yes
Learning-by-mentoring	non-internalized gains	No	No
Learning-by-mentoring	internalized gains	Yes	No

Notes: The column 'Seniors' decision' refers to whether the decision to maintain or dissolve teams maximizes GDP given AI availability. The column 'AI use' refers to whether AI adoption raises long-run GDP compared to the no-AI steady state.

4.3 Sources of Inefficiency

This subsection explains why AI can cause dynamic inefficiency even when individual decisions are rational. Our analysis identifies two fundamentally different sources of inefficiency.

Source 1: Incorrectly priced learning. Seniors do not internalize the value of the mentoring they provide. When they dissolve teams in favor of AI, they ignore the loss of human capital formation. This pecuniary externality can be solved with wage adjustments: if juniors accept lower wages to compensate seniors for training, the externality is internalized, and seniors' decisions become efficient. If junior wages are too low to fully internalize learning gains, then seniors might make dynamically inefficient decision for the aggregate economy.

Source 2: Bundling of learning and work. AI raises junior productivity, increasing their outside option. But the senior's capacity constraint, the maximum number of junior problems they can handle, is unchanged. Therefore, fewer juniors are needed in teams, and fewer juniors learn. This crowding out of learning opportunities *cannot* be solved with wages alone. Even when learning is fully internalized, AI can reduce long-run output by

shrinking teams.

The missing market for learning. The economy has three goods: (1) problem solutions, (2) senior supervision, and (3) learning. But there are only two prices: junior wage and senior wage. Learning and work are bundled in the apprenticeship model: a junior cannot learn separately, they can only learn from working in a team.

AI unbundles productivity from learning. A junior with AI produces more output but does not acquire senior skills faster. In fact, because AI makes solo work more attractive and teams smaller, AI *reduces* learning opportunities per junior. The market cannot separately price these two effects.

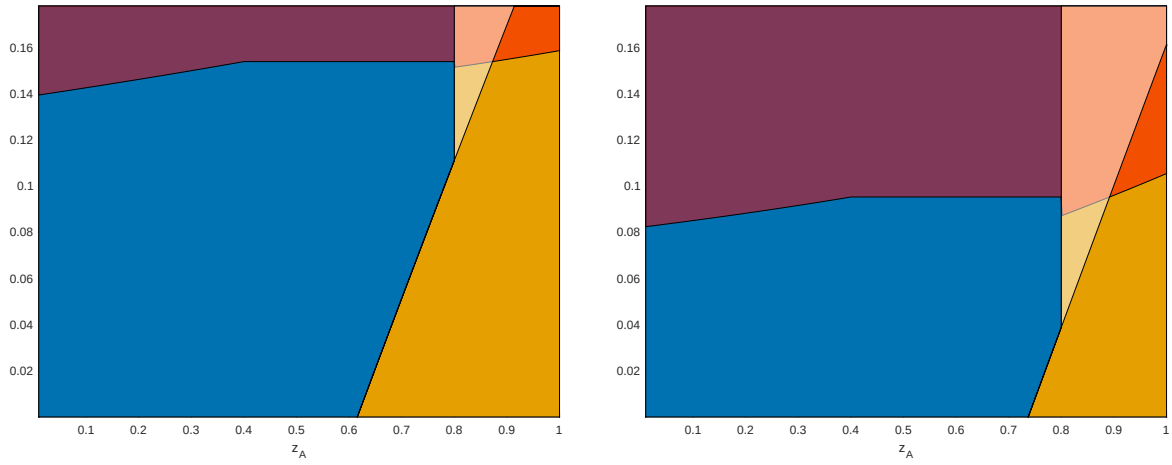
Frictions that prevent the first-best outcome. There are two fundamental frictions in this economy.

Non-rivalry of knowledge. Knowledge is non-rival: training n juniors is not n times as costly as training one junior. The senior's mentoring input (supervision time) scales less than proportionally with team size. For learning, teams should be large. But for production purposes, with AI, teams should be small (each junior handles more problems). These two objectives conflict.

Indivisibility of apprenticeships. Part-time apprenticeships could help: each junior could work $1/n_A$ time for a senior and $(n_A - 1)/n_A$ time solo with AI. This would maintain learning while capturing AI productivity gains. But this requires divisible jobs and assumes learning scales linearly with time in teams. If there is a minimum learning time threshold, or if matching frictions prevent such arrangements, this solution is infeasible.

Parameter regions. In Figure 4 we explore how the equilibrium of the model and dynamic inefficiency depends on the value of z_A and λ/δ. The left panel shows this for a lower value of $v_1 = 3$ ($v_0 = 1$), and the right panel for a higher value of $v_1 = 8$. The blue regions on the left of each panel shows where teams are maintained. If AI is not sufficiently skilled relative to the $z_0 = 0.4$ skill of juniors, then seniors prefer to maintain

teams. Whereas if z_A is high enough, then teams are dissolved, this is the yellow region on the right of each panel. The lighter colored triangular region is where teams are dissolved, but this decision creates a dynamic inefficiency in the aggregate economy. Note that in this example gains from learning are internalized in wages, but negative wages for juniors' is not allowed. Thus when the speed of learning, λ/δ is too high, solo junior wages are not high enough to fully internalize the gains, and dynamic inefficiency can and does arise. The top reddish region is where the adoption of AI creates a dynamic inefficiency. For any value of z_A there is a high enough learning speed, such that the adoption of AI is dynamically inefficient.[4] This shows that AI can be dynamically inefficient whether teams are maintained or not, and whether seniors' decision is dynamically efficient or not.



Low v_1 High v_1

Figure 4: Equilibrium with AI.

Notes: The figure shows the equilibrium with AI for different values of z_A on the horizontal axis, and different values of λ/δ on the vertical axis. The panel on the left is for $v_1 = 3$, on the right if for $v_1 = 8$. In both panels, the blue regions on the left show the region where teams are maintained, the yellow regions the right where teams are dissolved. The top reddish region (composed of 3 different colors) are where AI adoption leads to dynamic inefficiency. The triangular lighter colored region in the middle is where seniors' decision to dissolve teams is dynamically inefficient. The rest of the parameters are: $\phi = 0.05$, $z_0 = 0.4, z_1 = 1, v_0 = 1, h_{A1} = 0.2, h_{A2} = h = 0.4$.

[4]We only consider λ/δ values where juniors remain abundant even in the steady state of the team equilibrium.

5 Conclusion

We developed a model of AI adoption in knowledge work where seniors and juniors differ in problem-solving skills and value recognition, and juniors learn by working alongside seniors. The model yields three main insights.

First, AI creates static gains but can cause dynamic losses. When seniors use AI instead of hiring juniors, or when juniors use AI and teams shrink, fewer workers acquire senior skills through mentoring. The long-run supply of seniors falls, and steady-state GDP can decline even as short-run output rises. Second, dynamic inefficiency arises from two distinct sources: seniors not internalizing the value of mentoring (a pecuniary externality), and the bundling of learning with work (a real friction). The first source can be addressed through wage adjustments; the second cannot. Third, efficiency depends critically on institutional arrangements. When learning is internalized through wage negotiations, seniors' decisions are efficient. But AI can still lower long-run output by shrinking teams and reducing learning opportunities—a consequence of the bundling friction that no market mechanism can fully resolve.

Key mechanisms for policy. Several features matter for policy. The value recognition gap ($v_1 > v_0$) persists even as AI's problem-solving capability improves. Seniors' comparative advantage in selecting high-value problems ensures that human mentorship retains value even in a high-AI future. The senior capacity constraint—the maximum number of junior problems a senior can handle—is unchanged by AI. This means AI does not improve team output; it only shrinks teams. The gains from AI come from smaller teams and higher solo productivity, not from enhanced team performance. Finally, learning and work are bundled in the apprenticeship model. A junior learns by being present while working for a senior. AI unbundles productivity from learning: a junior with AI produces more but does not learn faster. Part-time arrangements could help—each junior works some time for a senior and some time solo with AI—but require divisible jobs and a learning technology that scales linearly with time.

Policy implications. Several policies could mitigate the adverse effects of AI on learning. Tax credits for training expenses or apprenticeship programs would help firms internalize the long-run cost of lost human capital. Subsidies for firms that maintain robust junior training, or industry-specific requirements for minimum mentoring ratios, could preserve learning opportunities. Historically, professions like medicine and law have guild-like systems ensuring knowledge transfer through residencies and clerkships. In an AI era, updated versions of these arrangements could focus on tasks AI cannot do—particularly value recognition and customer relationships.

Limitations and extensions. We treated skill levels, communication costs, and AI capability as exogenous. In reality, these evolve together. AI productivity improves as models learn from data (Wang and Wong 2025); human skills respond to AI presence through changes in education and training. Incorporating such feedback is an important extension.

We assumed learning happens only through mentoring. Alternative pathways exist: Brynjolfsson et al. (2025) show that AI can accelerate learning by codifying expert knowledge and providing real-time feedback. If juniors could acquire skills faster with AI assistance, the dynamic losses we identify would be mitigated. Exploring when AI complements rather than substitutes for mentorship is a promising direction.

Finally, we abstracted from firm heterogeneity and strategic considerations. Firms differ in their reliance on tacit knowledge and mentorship; some may find AI more disruptive than others. Strategic interactions—where one firm's AI adoption affects the labor market for all—could generate additional externalities. We leave these extensions for future work.

References

Acemoglu, D., and 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.

Acemoglu, D., and Restrepo, P. (2020). Robots and jobs: Evidence from US labor markets. *Journal of Political Economy*, 128(6), 2188–2244.

Arrow, K. J. (1962). The economic implications of learning by doing. *The Review of Economic Studies*, 29(3), 155–173.

Beraja, M., and Buera, F. (2024). The life-cycle of concentrated industries. *Unpublished manuscript*.

Brynjolfsson, E., Li, D., and Raymond, L. R. (2025). Generative AI at work. *The Quarterly Journal of Economics*, 140(2), 889–942.

Garicano, L. (2000). Hierarchies and the organization of knowledge in production. *Journal of Political Economy*, 108(5), 874–904. doi:10.1086/316101

Garicano, L., and Rossi-Hansberg, E. (2006). Organization and inequality in a knowledge economy. *Quarterly Journal of Economics*, 121(4), 1383–1435.

Hess, P., Janssen, S., and Leber, U. (2023). The effect of automation technology on workers' training participation. *Economics of Education Review*, 96, 102438.

Lucas, R. E., Jr. (1988). On the mechanics of economic development. *Journal of Monetary Economics*, 22(1), 3–42.

Mühlemann, S. (2024). *AI adoption and workplace training* (IZA Discussion Paper No. 17367). Institute of Labor Economics.

Noy, S., and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654), 187–192.

Wang, P., and Wong, T. N. (2025). *Artificial intelligence and technological unemployment* (NBER Working Paper No. 33867). National Bureau of Economic Research.