The Macroeconomic Effect of AI: Sizing the Software Engineering Channel*

Alex Blumenfeld **Jonathon Hazell** **Chen Lian** **Andreas Schaab**

July 12, 2026

Preliminary (Numbers may change).

Abstract

What is the macroeconomic effect of AI? One important channel is that AI will increase the productivity of software engineering, which this paper measures using asset prices. Asset prices capitalize not only current productivity gains due to AI, but also a forecast of future gains expected by the market. We estimate a cross-sectional relationship between the exposure of firms' stock returns to an AI index and their software engineering intensity, and develop a model that maps this relationship into software engineering productivity and GDP. Under our baseline calibration, news about AI from November 2022 to December 2025 implies a market forecast for present-value software engineering productivity gains equivalent to a permanent increase of 30.5 percent. These gains imply one-time permanent increases in GDP of 3.3 percent through production alone and 6.0 percent when higher software engineering productivity also raises R&D productivity. A further benefit of our approach is that it can be updated in real time: incorporating returns from the first months of 2026, a period of rapid progress in AI coding agents, implies substantially larger market-implied gains in software engineering productivity, with a correspondingly larger GDP impact.

*Blumenfeld: UC Berkeley; alex.blumenfeld@berkeley.edu. Hazell: London School of Economics; j.hazell@lse.ac.uk. Lian: UC Berkeley and NBER; chen_lian@berkeley.edu. Schaab: UC Berkeley and NBER; schaab@berkeley.edu. We thank Xavier Jaravel, Benjamin Moll, Christina Patterson, Maarten De Ridder, and Eben Lazarus for valuable comments and suggestions; Gregor Schubert for sharing data; and Sven van Holten Charria and Nicholas Tokay for outstanding research assistance.

1 Introduction

The rapid advance of artificial intelligence has prompted intense debate about its macroeconomic consequences. Forecasts of AI's effect on GDP span an enormous range, from modest gains of roughly one percent over a decade (Acemoglu, 2025) to scenarios in which broad AI automation could raise growth rates by an order of magnitude (Trammell and Korinek, 2023; Davidson et al., 2026). This uncertainty reflects a fundamental measurement problem: AI's economic effects operate through many channels: task automation, new product creation, and scientific discovery. The aggregate impact of each channel is difficult to observe directly, not the least because many of AI's advances have yet to play out. At the same time, the stakes of this question are high. Whether AI will meaningfully accelerate productivity growth or leave only a modest footprint shapes decisions ranging from monetary policy to corporate investment to workforce training.

Among the channels through which AI may affect the macroeconomy, software engineering (SWE) stands out as both concrete and potentially large. Software engineers produce and maintain the digital infrastructure that underpins operations across virtually every industry, and randomized trials find that developers using AI coding assistants complete individual tasks 21 to 56 percent faster (Peng et al., 2023; Paradis et al., 2025). However, these experimental estimates have two limitations as a basis for sizing productivity and macroeconomic impacts. First, they measure the effect of today's tools and cannot speak to future AI advances. Second, they capture speed-ups on isolated tasks, not the overall productivity of a software engineer whose job bundles many complementary activities. Because total productivity is disproportionately constrained by the least-improvable bottleneck tasks (Jones and Tonetti, 2026), even large gains on individual tasks can translate into modest gains in a software engineer's overall productivity.

This paper takes a financial-markets approach that addresses both limitations, using forward-looking asset prices to infer the total effect of AI on software engineering productivity and, through it, on GDP. Because stock prices capitalize the entire expected future path of AI-driven productivity gains, including not only the improvements delivered by today's tools but also anticipated future advances, they provide a present-value measure of AI's impact on software engineering productivity and on GDP through the software engineering channel. And because stock prices reflect markets' assessments of total software engineering productivity inclusive of all bottleneck complementarities, they sidestep the difficult aggregation from task-level speed-ups to overall software engineering productivity gains.

The central idea is that if AI primarily raises the productivity of software engineers, firms that rely more heavily on software engineers should benefit disproportionately from AI advances. In particular, an AI-induced increase in software engineering productivity lowers the effective cost of software engineering labor. Firms with higher software engineering intensity experience a larger reduction in their unit costs and capture outsized profit gains. This differential profit exposure is capitalized into stock prices, generating a cross-sectional pattern: firms with higher software engineering intensity should exhibit higher stock return sensitivity to an AI index. The magnitude of this cross-sectional relationship, i.e. how steeply stock return sensitivity to the AI index rises with software engineering intensity, is informative about the size of the underlying AI-induced present-value software engineering productivity growth per unit increase in the AI index.

Specifically, our approach to sizing the GDP impact of AI through the software engineering channel combines an empirical step with a model-based step. The empirical step estimates how firms' exposure to AI advances rises with their software engineering payroll share; the structural step maps this cross-sectional gradient into present-value software engineering productivity and GDP effects.

The empirical step proceeds in two stages. In the first stage, we estimate each firm's stock return sensitivity to an AI index, controlling for standard risk factors. In the second stage, we regress these sensitivities on firms' software engineering payroll shares, controlling for industry fixed effects and firm-level controls, to obtain the cross-sectional slope δ. A key advantage of this cross-sectional approach is that it isolates the software engineering productivity channel, even though movements in the AI index can also reflect changes in overall productivity or discount rates. Because these channels affect all firms uniformly, they do not generate cross-sectional profit variation correlated with software engineering payroll shares. Only SWE-augmenting productivity shocks create a differential cost advantage for SWE-intensive firms, generating profit variation proportional to firms' software engineering payroll shares. By projecting firms' AI-index exposures onto their software engineering payroll shares, the cross-sectional regression retains only the component driven by AI-induced software engineering productivity growth.

Although the cross-sectional slope isolates the software engineering productivity channel, translating it into an implied productivity gain requires a structural model. The slope reflects both the magnitude of the underlying AI-induced present-value software engineering productivity news per unit movement in the AI index and the structural gradient that maps a given software engineering productivity increase into cross-sectional differences in profit and, hence, value. To recover the underlying AI-induced present-value software engineering productivity news, we develop a model that expresses the structural gradient in terms of a few key parameters, such as the elasticity of substitution across product varieties, which determines how aggressively consumers shift expenditure toward firms that lower prices, and the substitution elasticity between SWE and non-SWE labor, which controls how much firms shift their input mix toward cheaper software engineers. One can then recover the AI-induced present-value software engineering productivity news per unit movement in the AI index from the cross-sectional slope δ and this structural gradient. Multiplying this object by the cumulative (residualized) AI-index return over a given period yields the event-window present-value software engineering productivity news implied by the AI news in that period.

Our data combine stock returns from CRSP, firm-level accounting variables from Compustat, and employee-level occupation data from Revelio Labs, which provide granular information on each firm's software engineering workforce. We proxy for the AI index using the ROBO Global Artificial Intelligence ETF (THNQ), and control for standard risk factors using the Fama-French three-factor model. The key firm-level regressor is the software engineering payroll share: the fraction of a firm's total payroll attributable to software engineers, constructed from Revelio's occupation and compensation data.

We aim to measure firms that might benefit from software engineering as an input into production, as opposed to firms that sell output related to software engineering (e.g. firms selling software-as-a-service). Therefore, following Acemoglu et al. (2022), we drop all firms in software-producing industries (NAICS sectors 51 and 54); we also drop firms that are part of the semiconductor supply chain. In our sample, there are firms across various industries that use software engineering intensively, including online marketplaces (e.g. eBay), high-end electronics manufacturers (e.g. Sonos), and travel booking platforms (e.g. Expedia or Airbnb).

In our baseline specification with NAICS3 industry fixed effects and size controls, the second-stage regression yields an estimated slope of $\delta = 1.52$ on the software engineering payroll share, so that comparing two firms whose payroll shares differ by one percentage point, a one percentage point movement in the AI index predicts a return differential of 1.52 basis points. Year-by-year regressions show that the positive relationship between software engineering intensity and AI-index exposure is present in every year from 2022 through 2025. The result is robust and stable across a range of additional specifications, which account for many possible confounders; we also consider specifications with fine industry fixed effects (NAICS4 and NAICS6 levels). This stability suggests limited scope for unobserved heterogeneity, other than the software engineering share, to explain the cross-sectional relationship (Oster, 2019).

We then use the structural step described above to convert the cross-sectional slope into an AI-induced productivity gain. Under our baseline calibration, news about AI from November 2022 to December 2025 implies a market-implied present-value increase in software engineering productivity equivalent to a permanent increase of 30.5 percent, within the range of the 21–56 percent task-level speed-ups found in the experimental literature (Peng et al., 2023; Paradis et al., 2025). Two offsetting forces can explain this alignment. On the one hand, complementarities between tasks imply that total productivity gains should be smaller than task-level gains (Jones and Tonetti, 2026; Demirer, Musolff and Yang, 2026): when a software engineer's workflow combines many complementary activities, output is constrained by the least-improvable bottleneck tasks, so even large gains on individual tasks translate into modest gains in overall productivity. On the other hand, our estimate reflects the normalized present value of the entire future path of software engineering productivity growth, so it capitalizes anticipated future AI advances and should therefore exceed the gains delivered by today's tools alone. We then conduct sensitivity analysis over the key structural parameters. The estimates are most sensitive to the within-firm substitution elasticity between SWE and non-SWE labor. When firms can more easily substitute toward SWE labor, the relative SWE wage absorbs more of a SWE productivity shock, attenuating the cost advantage of SWE-intensive firms. The same cross-sectional slope then implies a larger underlying productivity gain. When SWE and non-SWE labor are more complementary, this wage response is weaker and the inferred productivity gain is smaller.

We next translate this AI-induced software engineering productivity gain into GDP impact through the software engineering channel. The baseline calculation applies Hulten's theorem, in the spirit of Acemoglu (2025) and Aghion and Bunel (2024): a software engineering productivity gain raises GDP in proportion to the final expenditure-weighted aggregate software engineering cost share. This Hulten weight, combined with the inferred present-value productivity gain, implies that news about AI from November 2022 to December 2025 corresponds to a present-value GDP increase equivalent to a one-time permanent increase of 3.31 percent.

We discuss two potential threats to our use of the cross-sectional slope δ to infer AI-induced SWE productivity gains. The first is an aggregate shock that shifts expenditure toward SWE-intensive goods, generating profit differentials proportional to firms' software engineering payroll shares and thereby mimicking a true productivity shock. The second is heterogeneity in how strongly individual firms' valuations respond to discount-rate news, which may conflate the productivity channel with discount-rate-driven valuation effects. We state the conditions under which our inversion remains valid despite these concerns and provide empirical support for those conditions.

We then consider two extensions and show that the inferred AI-induced software engineering productivity gain and GDP impact are largely unchanged. First, we introduce a task-based framework with an explicit upstream AI sector. The AI sector supplies an input to a software supplier, which bundles a continuum of digital tasks performed either by software engineering labor or by AI under comparative advantage. A cost-side isomorphism leaves the baseline mapping from the cross-sectional slope to software productivity and GDP largely intact. Second, we embed the framework in the full U.S. input-output network and recompute the structural gradient. The network extension again changes the inferred productivity gain and GDP impact only moderately.

We finally evaluate how the GDP impact changes when AI-induced software engineering productivity gains feed into innovation. We embed the model in a semi-endogenous growth model with an R&D sector in which software engineering is part of R&D. This extension leaves the inversion from the cross-sectional slope to software engineering productivity unchanged, but significantly amplifies the mapping from software engineering productivity gains to GDP because higher software engineering productivity raises effective R&D input. Under our baseline calibration of the innovation-augmented model, the implied permanent GDP impact rises from 3.31 percent to 5.98 percent, a 1.81-fold amplification.

Finally, a benefit of our approach is that it permits real-time quantification of new AI-related information. Applying our baseline multipliers to residualized AI-index returns from January through May 2026 implies additional market-implied present-value software engineering productivity gains equivalent to a permanent increase of 46.5 percent and a corresponding increase in GDP of 5.05 percent through the software engineering production channel. This acceleration is consistent with the rapid development and adoption of AI coding agents during this period.

Literature review. A growing literature constructs occupation-level AI exposure measures that classify tasks by their susceptibility to automation (Eloundou et al., 2024; Felten, Raj and Seamans, 2021; Webb, 2020; Acemoglu et al., 2022; Hampole et al., 2025). Building on these measures, existing estimates of AI's aggregate impact on GDP span a wide range (see also Karger et al., 2026 on disagreement among experts over AI's macroeconomic effects). Acemoglu (2025) bounds the TFP effect at 0.53–0.66% over a decade; Aghion and Bunel (2024) estimate 0.68% per year using a task-based framework with broader assumptions (see also Briggs and Kodnani, 2023). Yet these top-down calculations rely on task-exposure measures that say little about how quickly firms adopt AI (McElheran et al., 2024), whether exposure predicts actual worker use (Lindenlaub et al., 2026), how task-level gains aggregate to firm-level productivity (Brynjolfsson, Rock and Syverson, 2019, 2021), or how AI capabilities will evolve in the future.

Randomized experiments find large task-level speed-ups from AI coding assistants (e.g. Peng et al., 2023; Paradis et al., 2025; Cui et al., 2026; Gambacorta et al., 2024). Similar gains have been documented in customer support (Brynjolfsson, Li and Raymond, 2025), professional writing (Noy and Zhang, 2023), and management consulting (Dell'Acqua et al., 2026). However, translating task-level gains into effects on firm-level or aggregate-level productivity and output is difficult. Acemoglu et al. (2022) find that AI-exposed establishments reduce non-AI hiring, yet detect no aggregate employment or wage effects at the occupation or industry level, illustrating the gap between micro-level adoption and macro-level consequences. Hampole et al. (2025) find strong AI-labor substitution at the task level, but show that its implied effects on labor demand are muted by within-occupation task reallocation and productivity-driven increases in labor demand at AI-adopting firms. Demirer, Musolff and Yang (2026) provide direct evidence of this attenuation in software development: large gains in coding activity attenuate substantially before reaching projects, releases, or app usage. Jones and Tonetti (2026) provide a theoretical benchmark for why such aggregation losses can be severe: in their model the economy's production structure features complementary tasks ("weak links") so that even fully automating all current software tasks yields a small GDP effect. Consistent with these aggregation challenges, Bick, Blandin and Deming (2026) show that generative AI diffused rapidly by late 2024 but still delivered only modest average time savings in the aggregate, despite high adoption rates. Our paper takes a different route: because asset prices capitalize not only current AI-driven productivity gains but also anticipated future advances, and already embed expectations about adoption speed and task-to-job aggregation, we can sidestep these difficulties.

A growing literature uses asset price data to measure the economic value of AI at the firm level. Eisfeldt et al. (2026) show that firms with high workforce exposure to generative AI gained roughly 5% in market value relative to low-exposure firms after ChatGPT's release. Bertomeu et al. (2025) use Italy's temporary ChatGPT ban and find that high-exposure firms lost roughly 6% in market value relative to low-exposure firms during the ban. Babina et al. (2024) show that AI-investing firms have higher market valuations. Andrews and Farboodi (2025) use bond yield responses to AI model releases to assess whether markets price in transformative growth. We instead use asset price data to back out AI-induced productivity gains and the associated macroeconomic impact through the software engineering channel.

Finally, our work connects to the theoretical literature on AI, automation, and growth. The task-based framework of Acemoglu and Restrepo (2018, 2019) provides the foundation for analyzing how automation displaces workers from existing tasks while new tasks reinstate labor demand, with Acemoglu and Restrepo (2022) showing that task displacement accounts for 50–70% of US wage structure changes over four decades. Recent work emphasizes that AI-driven changes in task composition interact with workers' comparative advantage to produce heterogeneous labor market effects even within highly exposed occupations (Freund and Mann, 2026; Althoff and Reichardt, 2026), and that AI may augment workers' capabilities rather than automate their tasks (Agrawal, McHale and Oettl, 2026). Growth-theoretic models show that AI's long-run impact depends critically on whether it can automate the research process itself (Aghion, Jones and Jones, 2019; Besiroglu, Emery-Xu and Thompson, 2024; Korinek and Suh, 2024; Davidson et al., 2026), consistent with our findings on the importance of accounting for the innovation channel when sizing the GDP impact of AI-induced software engineering productivity gains.

2 Sizing the Macroeconomic Impact of AI via Software Engineering

This section presents our approach to sizing the macroeconomic impact of AI through the software engineering channel. We combine cross-sectional estimates with a simple general equilibrium model to infer AI-induced software engineering productivity change. To ease exposition and directly deliver closed-form expressions for our questions of interest, we use the simplest environment here, but illustrate that our inversion procedure is robust to multiple extensions: Section 2.4 shows robustness to aggregate shocks that shift the composition of demand across goods and to heterogeneity in firms' exposure to discount-rate news, Section 4.2 microfounds the labor-augmenting shock via an upstream AI sector with task-based production, Section 4.3 embeds the framework in the U.S. input-output network, and Section 5 extends the environment to allow AI to operate through an additional innovation channel. Sections 4 and 5 also study how to translate the implied productivity change into a GDP impact.

2.1 Environment

Time is discrete, indexed by $t \in \{0, 1, \ldots\}$. The economy comprises J firms, indexed by j, each producing a distinct good. We maintain the assumption that each firm is small throughout ($J \to \infty$). A representative household owns all firms, and financial markets are complete. There are three aggregate shocks: S_t, A_t, and β_t. S_t denotes the productivity of software engineers (SWE). A_t denotes the Hicks-neutral aggregate productivity. β_t is a time-varying one-period discount factor. The aggregate shocks may be arbitrarily correlated.

Households. The representative household is endowed with two factors, supplied inelastically: software engineers in quantity s and non-SWE labor in quantity L. Household consumption preferences are

$$\mathbb{E}_0\Big[\sum_{t=0}^{\infty}\Big(\prod_{\ell=0}^{t-1}\beta_\ell\Big)u(c_t)\Big],$$

with the convention $\prod_{\ell=0}^{-1} = 1$, where $\beta_t \in (0,1)$ and has steady-state value β. The consumption bundle is a standard CES aggregate with fixed demand weights ω_j summing to one,

$$c_t = \Big[\sum_j \omega_j^{1/\varepsilon} c_{jt}^{(\varepsilon-1)/\varepsilon}\Big]^{\varepsilon/(\varepsilon-1)}, \qquad \sum_j \omega_j = 1.$$

Cost minimization implies

$$c_{jt} = \omega_j\Big(\frac{p_{jt}}{P_t}\Big)^{-\varepsilon} c_t \qquad \text{and} \qquad P_t = \Big[\sum_j \omega_j p_{jt}^{1-\varepsilon}\Big]^{1/(1-\varepsilon)},$$

where P_t is the consumer price index and p_{jt} denotes the price of good j. The aggregate price index serves as numeraire, $P_t = 1$ for all t. The stochastic discount factor (SDF) is

$$\Xi_{t,t+1} = \beta_t \frac{u_c(c_{t+1})}{u_c(c_t)}.$$

Production. Each firm j produces output using a technology with constant returns to scale. A CES labor composite combines SWE and non-SWE labor,

$$H_{jt} = \left[\phi_j^{1/\sigma}(S_t s_{jt})^{(\sigma-1)/\sigma} + (1-\phi_j)^{1/\sigma} L_{jt}^{(\sigma-1)/\sigma}\right]^{\sigma/(\sigma-1)},$$

where s_{jt} and L_{jt} are firm j's employment of SWE and non-SWE labor, and $\sigma > 0$ is the elasticity of substitution across the two types of labor. The parameter $\phi_j \in (0,1)$ governs software engineering intensity within the labor composite and varies across firms. S_t is software engineering productivity, which augments SWE labor. The firm then combines the labor composite with materials (in units of final goods) through a Cobb-Douglas function:

$$y_{jt} = A_t\left(\frac{H_{jt}}{\alpha}\right)^{\alpha}\left(\frac{M_{jt}}{1-\alpha}\right)^{1-\alpha}, \tag{1}$$

where M_{jt} is firm j's use of final-good materials and α is the labor share of total costs. A_t is a Hicks-neutral aggregate productivity shock. The total cost of firm j is $\mathscr{C}_{jt} = w_t s_{jt} + W_t L_{jt} + M_{jt}$. Let q_{jt} denote the unit cost of the labor composite,

$$q_{jt} = \left[\phi_j (w_t/S_t)^{1-\sigma} + (1-\phi_j) W_t^{1-\sigma}\right]^{1/(1-\sigma)}.$$

With final-good materials priced at the numeraire, firm j's unit cost is

$$mc_{jt} = \frac{1}{A_t} q_{jt}^{\alpha},$$

where w_t/S_t is the effective software engineering wage. Firms face CES demand and charge a constant markup $p_{jt} = \frac{\varepsilon}{\varepsilon-1} mc_{jt}$. Per-period profits are a constant fraction of nominal revenue, $\pi_{jt} = \frac{1}{\varepsilon} p_{jt} y_{jt}$.

We define firm j's software engineering payroll share as $\gamma_{jt} = w_t s_{jt}/(w_t s_{jt} + W_t L_{jt})$. Since α is the labor share of total cost, the corresponding software engineering cost share is $\alpha\gamma_{jt} = (w_t s_{jt})/\mathscr{C}_{jt}$. The CES labor composite implies $\gamma_{jt} = \phi_j (w_t/(S_t q_{jt}))^{1-\sigma}$, which is strictly increasing in ϕ_j. The value of firm j is the present discounted value of its profit stream,

$$V_{jt} = \mathbb{E}_t\left[\sum_{\tau=0}^{\infty} \Xi_{t,t+\tau}\pi_{j,t+\tau}\right]. \tag{2}$$

Here $\Xi_{t,t} = 1$ and $\Xi_{t,t+\tau} = \prod_{\ell=0}^{\tau-1} \Xi_{t+\ell,t+\ell+1}$ for $\tau \geq 1$.

Market clearing. Goods market clearing requires the production of each differentiated good to equal household consumption demand plus firms' final-good materials demand for that good. Labor market clearing requires $\sum_j s_{jt} = s$ and $\sum_j L_{jt} = L$. We define firm j's final-expenditure share as $f_{jt} = p_{jt}c_{jt}/c_t = \omega_j (p_{jt}/P_t)^{1-\varepsilon}$, with $\sum_j f_{jt} = 1$. These are the weights the price index and demand system operate on. In this model they are also sales shares, $f_{jt} = p_{jt}y_{jt}/\sum_k p_{kt}y_{kt}$, and value-added shares, $f_{jt} = (p_{jt}y_{jt} - M_{jt})/\sum_k (p_{kt}y_{kt} - M_{kt})$. The aggregate (final-expenditure-weighted) software engineering payroll share is $\gamma_t = \sum_j f_{jt}\gamma_{jt}$.[1]

2.2 Responses to Aggregate Shocks

We characterize the first-order response of the economy to a perturbation $(d\log S_t, d\log A_t, d\log\beta_t)$. We drop t subscripts when referring to steady-state values. Under CES demand with constant markups and the numeraire $P_t = 1$, per-period profits satisfy:

$$d\log\pi_{jt} = (1-\varepsilon)d\log mc_{jt} + d\log c_t.$$

By Shephard's lemma, the unit cost moves as a cost-share-weighted average of effective input prices.

$$d\log mc_{jt} = -d\log A_t + \alpha\gamma_j(d\log w_t - d\log S_t) + \alpha(1-\gamma_j)d\log W_t,$$

where materials are priced in units of the final good, whose price is the numeraire, and therefore do not contribute a separate price term. Software engineering productivity S_t enters by reducing the effective SWE wage $d\log w_t - d\log S_t$, with an impact proportional to the firm's SWE payroll share γ_j scaled by the labor cost share α. The constant price index ($d\log P_t = 0$) implies the final-expenditure-weighted average of unit cost changes satisfies $0 = -d\log A_t + \alpha\big[\gamma(d\log w_t - d\log S_t) + (1-\gamma)d\log W_t\big]$, where $\gamma = \sum_j f_j\gamma_j$ is the final-expenditure-weighted aggregate software engineering payroll share. Subtracting this zero from $d\log mc_{jt}$ isolates the relative cost change, $d\log mc_{jt} = -\alpha(\gamma_j-\gamma)[d\log S_t - (d\log w_t - d\log W_t)]$. Substituting into the profit equation yields:

$$d\log\pi_{jt} = (\varepsilon-1)\alpha(\gamma_j-\gamma)\big(d\log S_t - (d\log w_t - d\log W_t)\big) + d\log c_t. \tag{3}$$

Equation (3) separates the profit response into a firm-specific component and a firm-invariant level shift in aggregate demand $d\log c_t$ that does not generate cross-sectional variation. The term $d\log S_t - (d\log w_t - d\log W_t)$ captures the net reduction in the effective cost of software engineering labor relative to other labor, which can come from either a direct increase in software engineering productivity $d\log S_t$ or a decline in the relative software engineering wage $d\log w_t - d\log W_t$. When this effective cost falls, firms with a higher-than-average reliance on software engineers ($\gamma_j > \gamma$) experience a relative marginal cost decline. Since demand is elastic ($\varepsilon > 1$), consumers substitute toward these newly cheaper goods, translating the cost advantage into profit gains. The aggregate productivity shock A_t drops out of relative costs because it scales all production frontiers uniformly, leading to $d\log w_t = d\log W_t$ in general equilibrium. The time-varying discount factor β_t shifts the stochastic discount factor uniformly across firms. Neither generates cross-sectional variation in profits.

[1] Firm j's sales-Domar weight is $\lambda_{jt} = p_{jt}y_{jt}/c_t$. Because each good's sales equal final consumption plus materials demand, $\lambda_{jt} = \theta f_{jt}$. Thus $\sum_j\lambda_{jt} = \theta$, where $\theta \equiv 1/[1-(1-\alpha)(\varepsilon-1)/\varepsilon] \geq 1$ is the ratio of gross use of the final-good composite, household consumption plus firms' materials purchases, to GDP.

In contrast, S_t augments SWE labor and therefore creates a differential cost advantage for SWE-intensive firms. When S_t increases, SWE labor becomes more abundant in efficiency units, so the relative effective SWE wage $\log w_t - \log S_t - \log W_t$ falls in general equilibrium to clear the labor market. The size of this adjustment is

$$d\log w_t - d\log S_t - d\log W_t = -\frac{1}{\Sigma} d\log S_t, \qquad \text{with} \quad \Sigma = \sigma + \left[\alpha(\varepsilon-1) - (\sigma-1)\right]\frac{\sigma_s^2}{\gamma(1-\gamma)},$$

where $\sigma_s^2 = \sum_j f_j(\gamma_j - \gamma)^2$ is the cross-sectional variance of software engineering payroll shares. A higher aggregate elasticity of substitution between SWE and non-SWE labor Σ dampens the equilibrium wage response: when the economy substitutes more easily between SWE and non-SWE labor, less of a relative wage decline is needed to clear the labor market after a SWE productivity gain.

The aggregate elasticity of substitution Σ combines two channels through which the economy absorbs a SWE productivity shock. The first is within-firm substitution, governed by σ: as σ rises, firms substitute more easily toward the now-cheaper SWE labor, so aggregate substitutability Σ rises as well. The second is between-firm reallocation: consumers shift spending toward SWE-intensive firms whose prices fall by more. Holding payroll shares and σ fixed, a higher demand elasticity ε or a higher labor share α raises Σ by strengthening the reallocation channel. The cross-sectional variance σ_s^2 enters because this reallocation channel operates only when firms differ in SWE intensity; if all firms had the same payroll share, Σ would equal σ.

Substituting this general equilibrium wage adjustment back into (3) yields

$$d\log\pi_{jt} = \delta^S(\gamma_j - \gamma)\, d\log S_t + d\log c_t, \qquad \delta^S \equiv \alpha\frac{\varepsilon-1}{\Sigma}. \tag{4}$$

The structural gradient δ^S is the profit elasticity with respect to S_t per unit of the software engineering payroll share, which we measure directly in the data. The firm-invariant term $d\log c_t$ does not affect this gradient.

Since (4) holds at every future horizon, log-linearizing firm value (2) aggregates these responses into a present discounted value:

$$d\log V_{jt} = \delta^S(\gamma_j - \gamma)(1-\beta)\mathbb{E}_t\Big[\sum_{k=0}^{\infty}\beta^k d\log S_{t+k}\Big] + (1-\beta)\mathbb{E}_t\Big[\sum_{k=0}^{\infty}\beta^k\big(d\log c_{t+k} + d\log \Xi_{t,t+k}\big)\Big], \tag{5}$$

where the second term is firm-invariant. To connect with the empirical strategy based on stock returns, let $R_{jt} \equiv d\log V_{jt} - d\log V_{j,t-1}$ denote firm j's realized return. The following lemma, obtained by taking the period-t revision of (5), links its unexpected component to the period-t news in normalized present-value software engineering productivity,

$$\mathscr{S}_t \equiv (1-\beta)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}\beta^k d\log S_{t+k}\Big].$$

The factor $1-\beta$ normalizes units so that, under a random-walk process for $\log S_t$, $\mathscr{S}_t$ equals the period-

t innovation in software engineering productivity, $d\log S_t - \mathbb{E}_{t-1}[d\log S_t]$.

Lemma 1 (Firm Value Response). *To first order, each firm j's unexpected period-t return is linked to the period-t news in normalized present-value software engineering productivity by:*

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \mathscr{S}_t \gamma_j + \mu_t, \tag{6}$$

where μ_t is firm-invariant, collecting unexpected responses common to all firms.

The key feature of (6) is that Hicks-neutral productivity and discount-rate shocks move all firm values proportionally and are therefore absorbed into μ_t. Hence, for any two firms j and i, $R_{jt} - R_{it} = \delta^S \mathscr{S}_t(\gamma_j - \gamma_i)$. That is, cross-sectional return differences depend only on software-engineering productivity news, differences in software-engineering payroll shares, and the structural gradient δ^S, which governs how strongly software-engineering productivity news translates into cross-sectional return differences per unit of the payroll share. A cross-sectional regression of firm returns on software engineering payroll shares therefore has slope $\delta^S \mathscr{S}_t$, reflecting both the structural gradient δ^S and software engineering productivity news $\mathscr{S}_t$. We can then use the model-implied gradient δ^S to recover the software engineering productivity news $\mathscr{S}_t$ from the slope.

2.3 AI-induced SWE Productivity Gain through Inversion

Our objective is to use the cross-sectional return differences to infer AI-driven news about software engineering productivity gain. For this purpose, we project onto the excess return of the AI index and infer news in normalized present-value software-engineering productivity conditional on the excess AI-index return:

$$\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\mathrm{AI}}] = \kappa \cdot \tilde{R}_t^{\mathrm{AI}}, \qquad \text{where} \qquad \kappa \equiv \frac{\mathrm{Cov}_t(\mathscr{S}_t, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})}, \tag{7}$$

where $\mathbb{E}^*[\cdot \mid \cdot]$ denotes the best linear predictor, $\tilde{R}_t^{\mathrm{AI}}$ is the residualized excess AI-index return (assumed to be unforecastable at $t-1$), defined as the residual from projecting $R_t^{\mathrm{AI}} - R_t^f$ onto F_t, where F_t denotes a vector of risk factors (e.g., the Fama-French market, SMB, and HML factors), and κ is news in normalized present-value software-engineering productivity implied by a unit movement in the residualized excess AI-index return.

To infer κ, we project the firm-level return onto the residualized AI-index return:

$$\mathbb{E}^*[R_{jt} \mid \tilde{R}_t^{\mathrm{AI}}] = \delta^S \kappa \tilde{R}_t^{\mathrm{AI}} \gamma_j + \tilde{\mu}_t, \tag{8}$$

where $\tilde{\mu}_t$ is firm-invariant. From this expression, a regression of the observed firm-level return R_{jt} on the observed residualized AI-index return interacted with payroll shares, $\tilde{R}_t^{\mathrm{AI}}\gamma_j$, gives an estimate of $\kappa\delta^S$. Combining this estimated coefficient with the structural gradient δ^S based on (4) then gives κ, the news in software-engineering productivity implied by a unit movement in the AI-index return.

To implement (8) empirically, we use the two-step approach described further in Section 3.1. In the first stage, we regress firm j's excess return $R_{jt} - R_t^f$ on the excess AI-index return, $R_t^{\mathrm{AI}} - R_t^f$, and control factors F_t to obtain its first-stage AI-index beta. By the Frisch–Waugh–Lovell theorem, firm j's first-stage AI-index beta equals

$$\beta_j = \frac{\mathrm{Cov}_t(R_{jt} - R_t^f, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})},$$

The second stage regresses each firm j's first-stage AI-index beta β_j on its payroll share γ_j and a vector of firm-level controls X_j, yielding the second-stage coefficient

$$\delta = \frac{\mathrm{Cov}_j(\beta_j, \gamma_j \mid X_j)}{\mathrm{Var}_j(\gamma_j \mid X_j)}.$$

(8) implies that firm j's first-stage AI-index beta β_j and its payroll share γ_j are linked by

$$\beta_j = \bar{\beta} + \kappa \delta^S \gamma_j$$

where $\bar{\beta}$ is firm-invariant and absorbs the risk-free rate. Because β_j is affine in γ_j, partialling out X_j does not affect the slope, so the second-stage coefficient δ satisfies

$$\delta = \kappa \delta^S.$$

Hence, our two-stage procedure effectively regresses firm returns R_{jt} on $\tilde{R}_t^{\mathrm{AI}} \gamma_j$, as in (8), and obtains the coefficient $\kappa \delta^S$. Combining this coefficient with the structural gradient δ^S recovers κ, which isolates news in software-engineering productivity implied by a unit movement in the AI-index return. Movements in the AI index may also capitalize Hicks-neutral productivity or discount-rate shocks, but these common effects enter only through the firm-invariant intercept $\bar{\beta}$. They therefore generate no cross-sectional variation correlated with γ_j conditional on X_j and do not affect δ (and hence κ).

Multiplying κ by the movement in the residualized AI index $\tilde{R}_t^{\mathrm{AI}}$ at period t gives period-t news in software engineering productivity driven by AI.

Proposition 1 (Financial-Market-Implied Software Engineering Productivity Change)**.** *The best linear predictor of the period-t news in normalized present-value software engineering productivity given the contemporaneous movement in the residualized AI index is*

$$\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{AI}] = \kappa \cdot \tilde{R}_t^{AI}, \qquad \textit{where} \quad \kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon - 1)} \cdot \delta. \tag{9}$$

Because firm values are forward-looking, the software engineering productivity news recovered by this inversion is not the contemporaneous productivity gain alone. It is the news $\mathscr{S}_t$ in the normalized present-value productivity object, which revises with both current and anticipated future software engineering productivity gains.

Summing over cumulative excess returns of the AI index over a period (e.g., from the November 2022 ChatGPT launch to the end of 2025) then gives changes in the present value of software engineering productivity implied by AI news over that period:

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] = \kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}}.$$

2.4 Identification

Beyond the environment above, two potential threats to identification emerge: an aggregate shock that shifts expenditure toward SWE-intensive goods, and heterogeneity in how strongly individual firms' valuations respond to discount-rate news. We address each in turn.

Demand composition. Let Z_t be a demand shifter. The consumption bundle becomes

$$c_t = \left[\sum_j \omega_j(Z_t)^{\frac{1}{\varepsilon}} c_{jt}^{\frac{\varepsilon-1}{\varepsilon}}\right]^{\frac{\varepsilon}{\varepsilon-1}},$$

where the CES demand weights $\omega_j(Z_t)$, normalized so that $\sum_j \omega_j(Z_t) = 1$, generalize the constant benchmark weights ω_j by letting the shifter Z_t reallocate expenditure across goods. Production is unchanged from (1). The full set of aggregate shocks is (S_t, A_t, β_t, Z_t).

The demand shifter Z_t can contaminate the payroll-share slope δ through two channels. First, a direct demand channel: if Z_t shifts expenditure weights $\omega_j(Z_t)$ in a way that, conditional on controls X_j, correlates with SWE payroll share γ_j, the demand reallocation generates firm-specific profit variation that the regression attributes to the software engineering channel. Second, a factor-price channel: if Z_t shifts demand toward SWE-intensive goods in a final-expenditure-weighted sense, demand for software engineering labor rises, driving up the relative SWE wage and generating cost differentials proportional to the SWE payroll share γ_j. Assumption 1 shuts down both channels.

Assumption 1. *(Factor-neutral demand composition) The demand-weight elasticities $\omega'_j = \partial \log \omega_j / \partial \log Z|_{ss}$ have zero covariance with software engineering payroll shares in two distinct senses,*[2]

$$Cov_j(\omega'_j, \gamma_j \mid X_j) = 0 \qquad \textit{and} \qquad Cov_{f,j}(\omega'_j, \gamma_j) = 0.$$

The two conditions shut down the two channels one at a time. The first condition targets the direct demand channel: the demand shift may raise the profits of high-γ_j firms simply because consumers now want their products more. The condition is stated conditional on X_j because the regression controls for X_j: only demand variation that survives the controls can bias δ. When X_j includes industry fixed effects, the regression already absorbs any demand reallocation that merely moves spending

[2] We define $\text{Cov}_{f,j}(\omega'_j, \gamma_j) \equiv \sum_j f_j(\omega'_j - \bar{\omega}'_f)(\gamma_j - \gamma)$, where $\bar{\omega}'_f = \sum_j f_j \omega'_j$ and $\gamma = \sum_j f_j \gamma_j$. Since the final-expenditure weights sum to one, this equals $\sum_j f_j(\gamma_j - \gamma)\omega'_j$.

across industries, so the condition asks only that no reallocation survive within industries, conditional on any additional firm-level controls in X_j. We verify this condition using an empirical proxy for the demand elasticity ω'_j, namely product-market exposure to generative AI; the estimate of δ is not sensitive to adding this exposure to our second-stage controls.

The second condition targets the factor-price channel at its source: when the final-expenditure-weighted covariance is zero, the demand reallocation does not raise aggregate demand for SWE labor, so the relative SWE wage does not move and no cost differentials arise. These weights appear here, and not in the first condition, because the labor market clears in weighted terms: each firm's demand for SWE labor scales with its sales (equivalently, its final-expenditure share), so large firms contribute more to aggregate SWE demand and the wage responds to the final-expenditure-weighted tilt of demand toward SWE-intensive firms. The test in Appendix C.1 assesses this condition and cannot reject the null that the final-expenditure-weighted covariance between SWE payroll share γ_j and the empirical proxy for product-market exposure to generative AI ω'_j is zero.

Discount-rate heterogeneity. In the benchmark, the time-varying discount factor β_t shifts all firm valuations proportionally and is absorbed into the common component μ_t. In practice, firms have heterogeneous valuation exposure to common discount-rate news. If that heterogeneity covaries with the payroll share γ_j conditional on firm-level controls, and the residualized AI index captures discount-rate news that is not spanned by the first-stage factors, the estimated slope δ could conflate the software engineering productivity channel with discount-rate-driven valuation effects.

Specifically, the firm-return response (6), allowing for such heterogeneous exposure to discount-rate news, extends to

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j \mathscr{S}_t + \eta_j \mathscr{D}_t + \mu_t, \tag{10}$$

where $\mathscr{D}_t$ is the period-t discount-rate news, defined as in (D.4) by applying the revision operator $(\mathbb{E}_t - \mathbb{E}_{t-1})$ to the normalized cumulative SDF path. The equation says that the firm return responds to software engineering productivity news through $\delta^S \gamma_j$ (as before) and to common SDF news through the duration loading η_j.

Appendix D microfounds this heterogeneity through firms' franchise duration. At each date, firm j's incumbent legal entity survives to the next period with probability $\varrho_j \in (0,1]$, and is otherwise replaced at fair value by an identical entity operating the same technology under the same demand curve. The static allocation, the representative-household SDF, and per-period profits are identical to the benchmark; firm j's equity becomes a duration-ϱ_j claim on the common SDF. The loading η_j is a strictly increasing closed-form function of firm j's franchise duration alone. When $\varrho_j = 1$, firm j's equity is a perpetual claim and we recover the benchmark: $\eta_j = 1$. When $\varrho_j < 1$, the firm's cash flows are effectively shorter-lived, so its value is less sensitive to discount-rate news.

Let $\tilde{R}_t^{\text{AI}}$ denote the AI index return residualized on the first-stage factor span. Because the first stage partials out the spanned component of $\mathscr{D}_t$, the population two-stage coefficient on the payroll share satisfies

$$\delta = \kappa\delta^S + \kappa^\Xi\delta^\Xi, \tag{11}$$

where

$$\delta^\Xi \equiv \frac{\text{Cov}_j(\eta_j, \gamma_j \mid X_j)}{\text{Var}_j(\gamma_j \mid X_j)} \qquad \text{and} \qquad \kappa^\Xi \equiv \frac{\text{Cov}_t(\mathscr{D}_t, \tilde{R}_t^{\text{AI}})}{\text{Var}_t(\tilde{R}_t^{\text{AI}})}.$$

The slope δ^Ξ measures whether SWE-intensive firms have systematically different equity duration, while κ^Ξ measures whether the residualized AI index still loads on common discount-rate news. The bias $\kappa^\Xi\delta^\Xi$ is nonzero only when both ingredients are present. Assumption 2 requires that at least one is absent.

Assumption 2. *(Discount-rate orthogonality) At least one of the following conditions holds:*

$$Cov_t(\mathscr{D}_t, \tilde{R}_t^{AI}) = 0 \qquad or \qquad Cov_j(\eta_j, \gamma_j \mid X_j) = 0.$$

The first condition (factor spanning) requires that the first-stage factors absorb the priced discount-rate variation in the AI index. The second condition (cross-sectional orthogonality) requires that discount-rate loadings are conditionally uncorrelated with SWE payroll shares.

The factor spanning condition is strengthened by including richer controls in the first-stage regression (13). The baseline Fama-French three factors already absorb the component of discount-rate news spanned by broad market, size, and value-factor returns. In Section 3.3, we show that δ is stable when the first stage additionally controls for the NASDAQ 100 return (QQQ) and the Fama-French five factors, which further capture profitability- and investment-related valuation news, supporting the first condition.[3]

The cross-sectional orthogonality condition is strengthened by including proxies for equity duration in the second-stage controls. In Section 3.3, we show that δ is stable when the second stage controls for book-to-market, operating profitability, investment, and leverage, standard cross-sectional pricing controls that proxy for equity duration (e.g., Lettau and Wachter, 2007; Campbell, Polk and Vuolteenaho, 2010), supporting the second condition.

Proposition 2 (Identification)**.** *Under Assumptions 1 and 2,*

$$\kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon - 1)} \cdot \delta. \tag{12}$$

The inversion in Proposition 1 continues to hold.

[3] Babina et al. (2025) show that firms' AI investment leads to heterogeneous exposure to the market factor. Because the market factor is always controlled for in our first-stage regression, the heterogeneous exposure they study does not bias our estimates of δ isolating the software engineering productivity channel.

3 Empirical Results and the SWE Productivity Gain from AI

This section implements the strategy of Section 2 to infer the AI-induced software engineering productivity gain. Section 3.1 describes the detailed empirical strategy. Section 3.2 describes the data sources and the construction of the analysis sample. Section 3.3 presents the baseline estimates of the cross-sectional slope δ in our two-step procedure. Section 3.4 applies the inversion of Section 2 to translate this slope into the implied AI-induced software engineering productivity gain.

3.1 Empirical Strategy

As described above, our empirical strategy proceeds in two stages. In the first stage, following the spirit of Fama and MacBeth (1973), we estimate each firm's stock return sensitivity to an AI index, controlling for standard risk factors, to obtain a firm-level AI beta. In the second stage, we regress these estimated betas on firms' software engineer payroll shares, controlling for industry fixed effects and other firm-level controls, to estimate the cross-sectional slope δ.

First stage: firm-level AI betas. We start by estimating each firm's sensitivity, or "beta", with respect to the return of an AI-focused exchange-traded fund. For each firm j in the regression sample, which we will describe in Section 3.2, we run the following time-series regression using weekly data:

$$R_{jt} - R_t^f = \alpha_j + \beta_j \left(R_t^{\text{AI}} - R_t^f\right) + \sum_i \psi_{ji} F_{it} + \epsilon_{jt}, \tag{13}$$

where R_{jt} is firm j's stock return, R_t^{AI} is the return on the AI index, R_t^f is the risk-free rate, and F_{it} are risk factors. In the baseline specification, the controls are the Fama-French three factors (Fama and French, 1993): the excess market return, SMB (small minus big), and HML (high minus low book-to-market). The coefficient β_j measures firm j's AI-factor loading: a one percentage point increase in the AI index excess return is associated with a β_j percentage point movement in the firm's excess return. The risk factors absorb each stock's exposure to broad market forces, such as the market risk premium, size, and value.

Second stage: explaining cross-sectional betas. The cross-sectional model relates estimated firm betas to software-engineering intensity:

$$\hat{\beta}_j = \delta \cdot \gamma_j + \Gamma \cdot X_j + u_j, \tag{14}$$

where $\hat{\beta}_j$ is the first-stage estimate of firm j's AI beta, γ_j is the software engineering payroll share, and X_j denotes the full vector of second-stage controls. In the baseline specification, X_j includes size controls together with a full set of industry indicators. The coefficient δ measures the increase in the AI beta per unit increase in a firm's software engineering payroll share, conditional on this common control set X_j. In return units, comparing two firms whose payroll shares differ by one percentage point, a one percentage point movement in the AI index predicts a return differential of δ basis points. Throughout, the focus is on firms that use software as an input into production. Therefore in what follows we will exclude firms that either sell software, or form part of the semiconductor supply chain.

3.2 Data

Our core data sources are resume history data from Revelio Labs (Revelio Labs, 2026), financial statement variables from Compustat (S&P Global Market Intelligence, 2025), historical data on the Fama-French factors from Kenneth French's website, stock and exchange-traded fund (ETF) price data from CRSP (Center for Research in Security Prices, 2025), and returns on AI-related ETFs that serve as our aggregate AI index.

Revelio Labs aggregates publicly available professional profiles from platforms such as LinkedIn to construct a comprehensive, worker-level dataset covering workers' employment histories, job titles, employers, and compensation (Revelio Labs, 2026). Each record includes a standardized occupation classification based on O*NET codes (e.g., 15-1252.00, "Software Developers"), which Revelio then aggregates into approximately ten broad role categories such as "Software Engineer," "Data Scientist," "Sales," and "Operations." Each record also includes an imputed measure of total compensation, which combines base salary and variable pay such as bonuses but excludes stock options. Compensation is not observed directly but is instead predicted by Revelio using a model that takes as inputs features of the job such as role, seniority, and firm. The dataset also provides sampling weights that adjust for the fact that workers in some occupations are more likely to have public professional profiles than others. We use these data to construct the key firm-level regressor in our analysis: the share of total payroll attributable to software engineers.[4]

Compustat North America, maintained by S&P Global Market Intelligence, provides standardized financial statement variables drawn from firms' annual and quarterly public filings with the SEC (S&P Global Market Intelligence, 2025). We use the Compustat annual fundamentals file to obtain firm-level measures of total assets, net sales, operating profitability, investment, leverage, and book-to-market ratios. These variables serve as controls in the regressions below. We also use the Compustat-CRSP link table to merge accounting data with stock return data at the firm level.

The Fama-French factor returns, available from Kenneth French's data library, provide daily and weekly time series of the standard risk factors used in asset pricing (Fama and French, 1993, 2015). In our baseline results, we use the three factors of Fama and French (1993): the excess market return

[4]Appendix Table A.6 shows the top 5 firms in each NAICS2 sector according to our software engineer compensation measure, as well as the 5 largest firms in each sector with zero software engineers identified in the Revelio data.

(MKT), the size factor (SMB), and the value factor (HML). In robustness exercises, we also include the investment factor (CMA) and the profitability factor (RMW) from the five-factor model of Fama and French (2015). The excess market return (MKT) is the return on a broad value-weighted portfolio of US equities minus the risk-free rate, proxied by the one-month Treasury bill rate. The size factor (SMB, "Small Minus Big") is constructed by sorting firms into groups based on market capitalization and taking the difference in returns between portfolios of small-cap and large-cap stocks. The value factor (HML, "High Minus Low") sorts firms by their ratio of book equity to market equity and takes the difference in returns between high book-to-market (value) firms and low book-to-market (growth) firms. The investment factor (CMA, "Conservative Minus Aggressive") sorts firms by asset growth and takes the difference in returns between those with low asset growth rates (investing conservatively) and high asset growth rates (investing aggressively). Finally, the profitability factor (RMW, "Robust Minus Weak") sorts firms on operating profitability and takes the difference in returns between highly profitable and less profitable firms. These factors are designed to capture exposure to broad market forces and to well-documented cross-sectional return premia.

The Center for Research in Security Prices (CRSP) US Stock and US Index databases provide historical security-level data on prices, returns, shares outstanding, and market capitalization for all securities listed on major US exchanges (Center for Research in Security Prices, 2025). We use weekly holding-period returns from CRSP for the firm-level time-series regressions in our first stage. CRSP also provides returns on exchange-traded funds, which we use to obtain the return series for the AI-related ETFs that serve as our aggregate AI index. We link CRSP security data to Compustat firm identifiers using the CRSP-Compustat merged database.

In the main results, we use THNQ, the ROBO Global Artificial Intelligence ETF, as our measure of the stock market returns of AI-intensive firms. THNQ tracks the ROBO Global Artificial Intelligence Index, which selects constituents based on their revenue exposure to artificial intelligence and classifies them into two broad categories: firms providing AI infrastructure (such as cloud computing, semiconductors, and data centers) and firms developing or deploying AI applications (such as healthcare AI, autonomous systems, and business analytics). The index uses a modified equal-weight methodology, in which holdings are assigned to scoring tiers based on the depth of their AI revenue exposure and then approximately equally weighted within each tier, giving higher aggregate weight to firms with more direct AI exposure. The index typically contains around 50 to 60 holdings and is rebalanced quarterly. Appendix Table A.1 lists the twenty largest holdings and their weights in the index.

We also run robustness checks that replace THNQ with AIQ, the Global X Artificial Intelligence and Technology ETF, to ensure that our results are not driven by ROBO Global's particular choices regarding which firms to label as representative of innovations in AI. AIQ tracks the Indxx Artificial Intelligence and Big Data Index, which selects firms that derive a meaningful share of their revenue from

AI, big data, or related technology segments. Unlike THNQ, AIQ uses a capped market-capitalization weighting scheme, in which individual holdings are subject to concentration caps of approximately 3 percent. This gives the index a heavier tilt toward large-cap technology and semiconductor firms relative to THNQ. The index typically contains around 85 to 90 holdings and is rebalanced semi-annually. Appendix Table A.2 lists the twenty largest holdings and their index weights.

Our main measure of the importance of software engineers at a firm is the share of total compensation in Revelio that goes to all workers in the "Software Engineer" category, specifically during 2021, the year before our stock-price panel begins.[5] We compute the total number of workers in each occupation at a firm using Revelio's individual-level weights (addressing measurement error from some occupations being more likely to have public profiles). Since compensation is imputed, we also report results using raw employment shares of software engineers rather than payroll shares.

Sample selection. Table 2 shows how we arrive at our main analysis sample. The time period is from the start of 2022 to the end of 2025. We start with the initial CRSP-Compustat merge (Row 1). We filter to Compustat firms with strictly positive assets and sales (Row 2). In Row 3, we restrict to nonfinancial firms by dropping NAICS sectors 52 (Finance and Insurance) and 53 (Real Estate and Rental and Leasing). In Row 4, we retain only firms that match to Revelio in 2021. In Row 5, we compute average assets and average sales from 2017 to 2021, ignoring missing years, then restrict to firms with strictly positive average assets and sales over this period, as well as non-missing market capitalization data for the end of 2021. In Row 6, we restrict to a balanced panel of firms with stock price data for all weeks of 2022 through 2025. In Row 7, we drop all firms that are among the holdings of either THNQ or AIQ as of March 2026, to eliminate any mechanical correlation between returns on our chosen AI ETFs and returns on the stocks in our final dataset.

In Row 8, we drop from our baseline sample a list of firms classified by GPT 5.5 as being involved in the development or production of semiconductors. These firms' stock returns will likely covary with returns on an AI ETF because they are integral suppliers of hardware required to run AI models, not because they stand to benefit from AI innovations through the software channel. Specifically, we ask GPT 5.5 to identify firms that are part of the semiconductor supply chain, based on their NAICS codes and their business model descriptions from Compustat. The majority of the 71 firms in our final list are in NAICS 334413 (Semiconductor and Related Device Manufacturing), although the classification excludes firms in that sector whose business models are focused on solar panel manufacturing rather than semiconductors; the list also includes some firms from other sectors whose `busdesc` field in Compustat mentions that they manufacture semiconductors or produce inputs to the semiconduc-

[5]Revelio's "Software Engineer" role category aggregates 125 O*NET occupation codes. Appendix Table A.3 lists the twenty largest by compensation share.

tor manufacturing process. Appendix Table A.4 summarizes the full set of dropped semiconductor-related firms.

In addition, the focus of the paper is on firms that use software as an input to production, not firms that sell software products. Thus, in Row 9, we drop the latter group: as in Acemoglu et al. (2022), we exclude firms in NAICS sectors 51 (Information) and 54 (Professional, Scientific, and Technical Services), as many firms in these sectors are producers of AI rather than users. Finally, in Row 10, we drop the bottom 5% of remaining firms ranked by total employment in Revelio: sample selection is a concern for these firms, because they may have few resumes posted online. At this point, we have the final dataset.

The table also lists the percentiles of 2021 sales. Even as we make the sample restrictions, the distribution of firm sales remains reasonably similar. Appendix Figure B.1 shows the distribution of firms across NAICS2 sectors after the NAICS exclusions and in our cleaned final sample; the distributions are similar in each.

We present some simple statistics about our main regressor, the firm-level payroll share of software engineers. Table 3 shows the top 20 firms with the highest software engineering intensity according to our measure. There seem to be two business models in common among the firms with a high share of software engineers, namely high-tech manufacturing (e.g. Sonos, Arlo) and e-commerce platforms (e.g. eBay, Expedia). We expand on this point in Figure B.2, which groups the top 100 firms according to our measure into several high-level "themes" based on descriptions of their business models from Compustat. Given that we explicitly exclude the tech sector (defined for our purposes as NAICS 51) from our analysis sample, it is of interest to summarize the business models of firms that do not produce software but rather use software engineers as an input into production: the figure shows that these include producers of high-tech hardware like computing equipment and Internet-of-Things connected devices, operators of e-commerce and logistics platforms, and precision health-care firms, among others.

Finally, Figure B.3 shows the full distribution of software engineer payroll shares; for reference, in our main sample, there are 196 firms with zero software engineers observed in our dataset in 2021.

3.3 Baseline Estimates

This section presents our main empirical results concerning the cross-sectional relationship between a firm's software engineering payroll share and the loading of its stock return on the AI index. We start with our baseline estimates of δ, estimated from equations (13) and (14). That is, in a first stage, we estimate the "beta" of each stock with respect to the AI index (after controlling for standard risk factors). In a second stage, we regress these betas on the firm's software engineering payroll share, as well as size and industry controls. The baseline regression is unweighted, with heteroskedasticity-robust standard errors. Our firm size controls are indicators for quintiles of average assets and sales between 2017 and 2021. The industry controls are either at the NAICS 2- or 3-digit level. We trim observations above the 99th percentile of the software engineering payroll share in our main analysis sample, which is 33.7%.

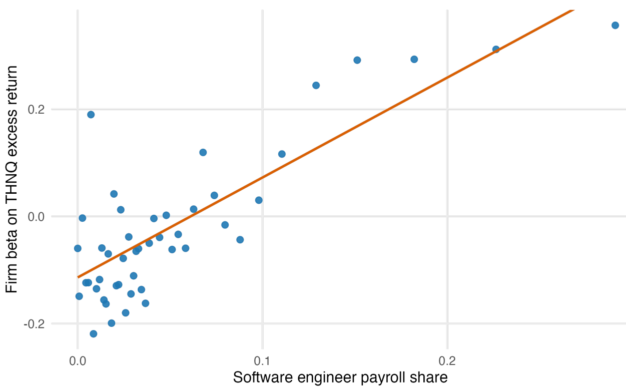
Figure B.4 shows the distribution of our first-stage betas for the specification with NAICS3 fixed effects, and Table A.5 shows summary statistics. This measure of firm-level loadings on our chosen AI index is centered near zero, and 18.6% of firms have AI betas that are significant at the 5% level.

Table 4 reports estimates of the second-stage regression. In the first column, we present estimates of the regression of the AI index beta on the software engineering payroll share without controls, which yields a coefficient of 1.86. The estimates remain similar as we add size controls in column 2 (2.02) and 2-digit industry fixed effects in column 3 (2.04). Column 4, with size controls and 3-digit industry fixed effects, is our baseline specification and yields $\delta = 1.52$. The coefficient means that, comparing two firms that differ by 1 pp in software engineering payroll share, a 1 pp AI index increase predicts a return that is 1.52 basis points higher for the more SWE-intensive firm. The estimates are statistically significant and precisely estimated in all specifications. In the final column, we add a control for workforce exposure to generative AI, measured as the employment-weighted share of a firm's occupational tasks that can be performed or assisted by large language models (Eisfeldt et al., 2026). The estimate is essentially unchanged at 1.69.

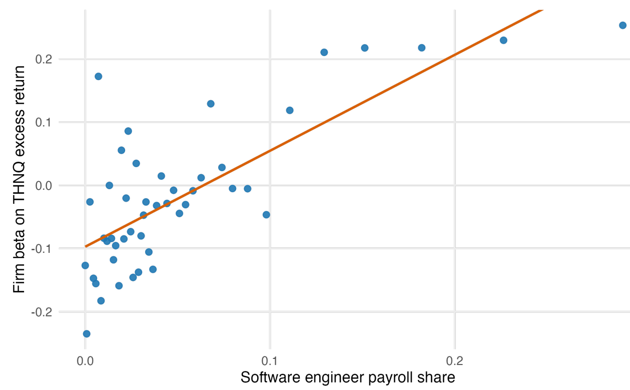
We visualize the result with binned scatter plots. Specifically, Figure 1 plots 50 bins of the software engineering payroll share on the x-axis, against the firm's sensitivity to the AI index on the y-axis. The top-left panel has no controls (i.e. column 1 of Table 4). The top-right panel controls for NAICS3 and size fixed effects (i.e. our baseline specification, in column 4 of the table). The bottom panel adds controls for workforce exposure to generative AI (i.e. column 5 of the table). In each case, there is a clear positive relationship between software engineering payroll share and sensitivity to the AI index.

The estimated coefficients remain similar as we consider an extensive range of robustness exercises, as Table 5 documents. The first row is the baseline specification, from column (4) of Table 4. The second row drops micro-cap firms, defined as firms with market capitalization below $1 billion at the end of 2021. The third row uses AIQ as the "AI index" as an alternative to our baseline choice of THNQ. As we discussed, AIQ has broader coverage, meaning it is more likely to include all AI stocks, but may also contain other technology stocks. The fourth and fifth rows add additional controls in the first-stage regression, which estimates the sensitivity of each firm to the AI index. These controls absorb other time-series confounders that might correlate with AI shocks and affect firm values. In particular, we control for the excess return of the ETF QQQ (which tracks the NASDAQ 100) in row 4, which absorbs the exposure of each shock to general technology shocks. In row 5 we control for two

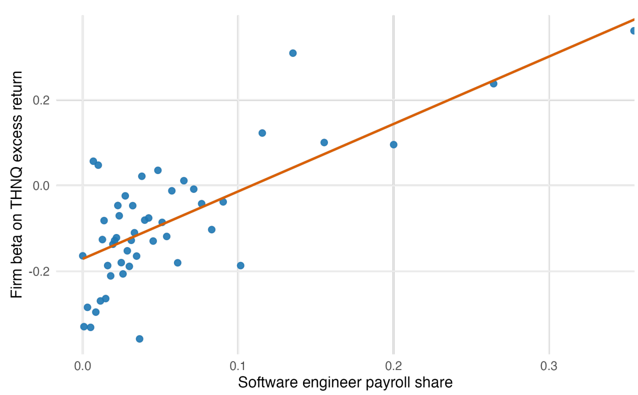
Figure 1: Software engineering payroll share vs. THNQ beta



(a) No controls



(b) NAICS3 and size fixed effects



(c) GenAI task exposure controls

Notes: Each panel plots 50 equal-sized bins of the software engineering payroll share (x-axis) against the firm's estimated beta on THNQ excess returns (y-axis). Panel (a) includes no controls. Panel (b) controls for NAICS3 industry and firm size quintile fixed effects. Panel (c) additionally controls for quintiles of the generative AI task exposure measure of Eisfeldt et al. (2026). The fitted line is the OLS slope from the corresponding column of Table 4.

more Fama-French factors: investment and profitability. Following Fama and French (2015), these factors absorb additional time-series shocks that affect firms heterogeneously. In row 6, we replace the standard Fama-French market factor in our first-stage regressions with an alternate version that excludes the "Magnificent 7" tech firms, to construct a market return less affected by AI-related news. Row 7 weights the regression by size (end of 2021 market capitalization) and row 8 controls for market capitalization. Row 9 weights by the software engineering payroll share, while row 10 instead uses raw employment shares of software engineers. Row 11 controls for additional characteristics that affect firms' sensitivity to aggregate shocks: book-to-market ratio, operating profitability, investment (capex-to-assets ratio), and book leverage. Row 12 controls for region, via the location of the firm's headquarters. Row 13 restricts the sample to US-incorporated firms only. Rows 14–15 control for the most granular possible industry fixed effects, at the 4- and 6-digit levels. Row 16 controls for three measures of firms' exposure to generative AI through the product market, exactly as constructed in Eisfeldt et al. (2026): (1) an indicator for whether GPT classified a firm as a likely beneficiary of GenAI based on the business description section of its 10-K, (2) the number of AI-related keywords in its 10-K, and (3) the share of its workers with AI-related skills as defined in Babina et al. (2024).[6] Row 17 controls for the payroll shares of several other occupations, namely Engineer, Finance, Project and IT Specialist, and Technician. Finally, row 18 addresses potential concerns about serial correlation in the first-stage time series and sampling uncertainty in the cross section by reporting bootstrap standard errors: calendar weeks are resampled using a moving-block bootstrap with block length 6 weeks, firms are resampled with replacement, and both stages are re-estimated in each iteration. The bootstrap standard error is larger than the baseline HC3 standard error, but the coefficient remains statistically significant.

It is reassuring that the coefficients are stable as we consider various permutations. As Oster (2019) points out, stability of the coefficients as we add observed heterogeneity to the controls is informative. It suggests that unobserved heterogeneity is unlikely to bias the estimates. Building on this idea, we implement the formal test of Oster (2019) in Appendix C.2 and find that there is limited scope for omitted variable bias to explain our results.

Several robustness exercises speak directly to the two identification assumptions of Section 2.4, which require that other aggregate shocks do not generate cross-sectional variation in AI betas correlated with the software engineering payroll share. The first threat is demand composition (Assumption 1). With industry fixed effects in the second stage, the direct demand channel (part 1 of Assumption 1) operates only through within-industry reallocation toward software-engineering-

[6] The decrease in the coefficient on the software engineer payroll share in this specification is only due to changing sample composition after merging with the product exposure measures, not to the inclusion of those measures as extra controls.

intensive firms; row 16 probes this channel by controlling for product-market exposure to generative AI, our empirical proxy for the demand-weight elasticity ω'_j. The coefficient remains stable, supporting part 1 of Assumption 1. The factor-price channel (part 2 of Assumption 1) is probed separately by the final-expenditure-weighted covariance test discussed in Section 2.4. The test implemented in Appendix C.1 cannot reject the null that the final-expenditure-weighted covariance is zero, supporting part 2 of Assumption 1. The second threat is discount-rate heterogeneity (Assumption 2), which biases δ only if the residualized AI index loads on discount-rate news not spanned by the first-stage factors and loadings covary with the SWE payroll share. Rows 4–5 probe the spanning condition (part 1 of Assumption 2) by enriching the first-stage factor span with QQQ and additional Fama-French factors. Row 11 probes the cross-sectional orthogonality condition (part 2 of Assumption 2) by controlling for standard equity-duration proxies, namely book-to-market, operating profitability, investment, and leverage, in the second stage (e.g., Lettau and Wachter, 2007; Campbell, Polk and Vuolteenaho, 2010). The stability of δ across these specifications supports both parts of Assumption 2.

Our baseline regression uses the software engineering payroll share as a continuous regressor. We now discretize firms into five quintiles of the payroll share. Table 6 reports the results. As the table shows, the AI sensitivity is higher in the top quintile and to some extent the fourth quintile. That is, the top tail of software engineering intensity drives the results. In Appendix Table A.7, we show this result is not driven by outliers (as is apparent from Figure 1). We also study the stability of our estimates, by estimating δ separately for every year. Figure 2 plots the coefficients. The estimates are similar in every year. Finally, in Appendix Figure B.5, we show results from estimating the regression separately on each NAICS3 sector with at least 25 firms in our main sample.

3.4 Quantifying the SWE Productivity Gain from AI

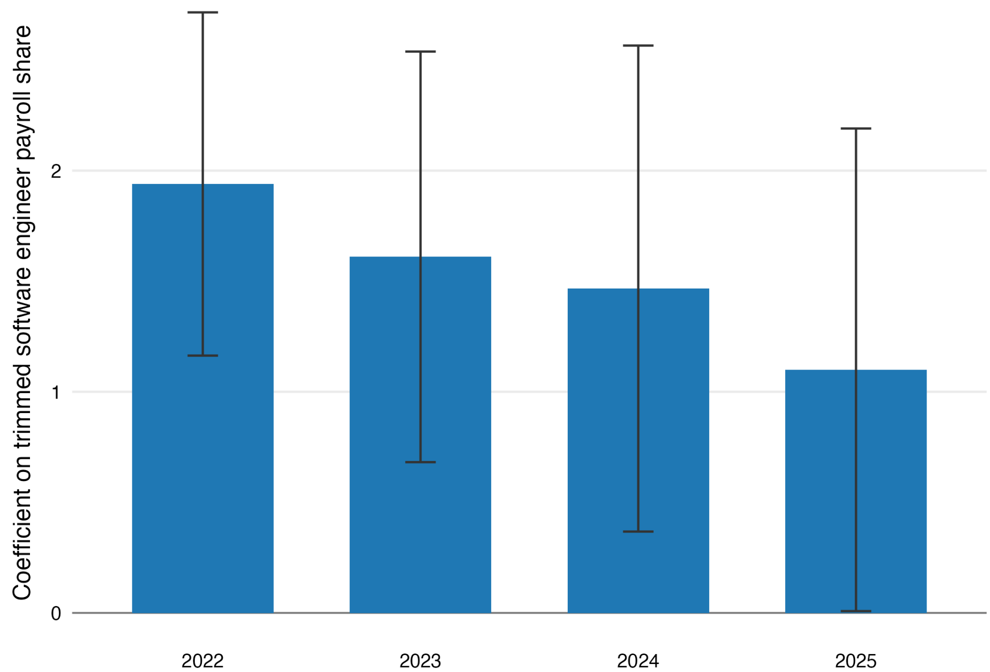
This subsection translates the cross-sectional slope δ into the news in software engineering productivity gain implied by the AI index, following the inversion of Section 2.

The formula. Proposition 1 uses δ to recover the market-implied AI-driven news in software engineering productivity. By equation (9), a movement $\tilde{R}_t^{\mathrm{AI}}$ in the residualized AI index implies a normalized present-value software engineering productivity change of $\kappa \cdot \tilde{R}_t^{\mathrm{AI}}$, where

$$\kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon-1)} \cdot \delta, \qquad \delta^S = \alpha\frac{\varepsilon-1}{\Sigma}, \qquad \Sigma = \sigma + \left[\alpha(\varepsilon-1)-(\sigma-1)\right]\frac{\sigma_s^2}{\gamma(1-\gamma)}.$$

That is, a residualized AI-index return of 1% predicts a normalized present-value software engineering productivity change of κ%. Recall from Section 2 that γ_j is firm j's software engineer payroll share, $\gamma = \sum_j f_j \gamma_j$ is the final-expenditure-weighted, or sales-weighted, average, and $\sigma_s^2 = \sum_j f_j(\gamma_j - \gamma)^2$ is the corresponding cross-sectional variance, where the weights f_j are normalized sales shares that sum

Figure 2: Year-by-year second-stage coefficient estimates



Notes: Each bar reports the estimated coefficient on the software engineering payroll share from the baseline NAICS3 fixed-effects second-stage specification, where the first-stage AI betas are estimated using data from a single calendar year only. Whiskers show 95% confidence intervals based on HC3 standard errors.

to one. The aggregate substitution elasticity Σ combines within-firm input-mix adjustment (σ) and between-firm expenditure reallocation (ε). A higher aggregate substitutability Σ means the economy substitutes more toward software engineering labor after a software-engineering productivity gain, bidding up its relative wage. This offsets part of the cost advantage of software engineering-intensive firms, so a larger productivity shock is needed to generate the same observed payroll-share slope δ.

The demand elasticity ε. The demand elasticity governs the product-market force: how strongly a relative cost advantage translates into relative profit gains. Standard values in the production-network literature are $\varepsilon \approx 4$–6, based on trade-elasticity estimates from Broda and Weinstein, 2006. Following Broda and Weinstein (2006), we set our baseline to $\varepsilon = 5$, which corresponds to a markup of $\mu = \varepsilon/(\varepsilon-1) = 1.25$.

The within-firm substitution elasticity σ. The parameter σ governs substitution between software engineering and other labor inputs within the firm. Since software engineers produce software, the most directly relevant evidence is the substitutability between software and other inputs in production. Using Korean firm-level data, Aum and Shin (2024) estimate an elasticity of substitution between software and other labor of 1.6. Our baseline is accordingly $\sigma = 1.6$. This value is also close to canonical estimates of substitution across education groups: Katz and Murphy (1992) estimate an elasticity of 1.4 between college and non-college labor, and Ciccone and Peri (2005) obtain 1.5 using long-run cross-state variation.

The elasticity σ is the most influential structural parameter: when cross-sectional dispersion in SWE payroll shares (σ_s^2) is small relative to $\gamma(1-\gamma)$, the aggregate substitution elasticity satisfies $\Sigma \approx \sigma$, so the productivity multiplier κ is approximately proportional to σ. Given its importance, we also explore $\sigma \in \{0.5, 1.0, 1.5, 2.0\}$, a range that deliberately spans the two economically distinct cases: gross complementarity between software engineering labor and non-SWE labor when $\sigma < 1$, and gross substitutability when $\sigma > 1$. The range also reflects the empirical ambiguity in AI's effect on non-SWE labor demand: AI may substitute for exposed tasks and reduce demand for some non-SWE labor, but it may also complement non-SWE workers by shifting effort toward tasks that remain difficult to automate (Acemoglu et al., 2022; Hampole et al., 2025; Agrawal, McHale and Oettl, 2026).

Data-based objects. We construct γ, the sales-weighted aggregate software engineering payroll share, directly from the all-sector Compustat-Revelio data, using data for 2021. Unlike in the regression sample, this object retains NAICS 51 and 54, semiconductor-related firms, and constituents of our chosen AI ETFs, because the object here is the sales-weighted aggregate software engineering payroll share for the full economy. We still exclude NAICS 52 and 53 because γ is a sales-weighted object, and sales for financial and real-estate firms are not directly comparable to sales for nonfinancial firms.

This requires one key assumption, which is that the sales-weighted average software engineer payroll share is the same among firms in our sample as in the universe of unobserved firms. With this assumption, we can compute

$$\gamma = \sum_{j \in \text{observed firms}} \frac{\text{sales}_j}{\text{total observed sales}} \gamma_j,$$

where γ_j is firm j's software engineer payroll share.

The above calculation yields $\gamma \approx 0.109$. We compute σ_s^2, the sales-weighted variance of software engineering payroll shares across firms, in a similar manner by assuming that the sales share-weighted variance of γ_j is the same for observed and unobserved firms. This gives us $\sigma_s^2 \approx 0.0119$. In practice, software engineering payroll shares are small and concentrated: most sectors have γ_j close to zero, while a handful of technology-intensive sectors have much higher values. This implies $\sigma_s^2 \ll \gamma(1-\gamma)$, so that the between-firm reallocation channel is quantitatively modest.

We construct the labor cost share α using the BEA/BLS KLEMS production accounts, which provide labor compensation and gross output for 61 private-sector industries (excluding federal and state-and-local government). Using this dataset, α is total labor compensation divided by total gross output across all private industries, giving $\alpha \approx 0.311$. Combining this estimate with $\gamma \approx 0.109$ implies a sales-weighted average cost share of software engineering ≈ 0.034.

Our second-stage regression yields $\delta = 1.52$ (column 4 of Table 4, the specification with NAICS3 fixed effects and size controls). From November 2022 (the launch of ChatGPT 3.5) to December 2025, the cumulative residualized AI-index return (after Fama-French projection) is $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} = 14.9\%$.

Quantifying AI-driven news in SWE productivity. Based on the calibrated elasticities $\sigma = 1.6$ and $\varepsilon = 5$, the labor cost share $\alpha = 0.311$, and the data-based objects $\gamma \approx 0.109$ and $\sigma_s^2 \approx 0.0119$, the aggregate substitution elasticity is approximately $\Sigma \approx 1.68$. The corresponding structural gradient is then

$$\delta^S = \alpha \frac{\varepsilon - 1}{\Sigma} = 0.311 \times \frac{4}{1.68} \approx 0.74.$$

Proposition 1 gives the productivity multiplier κ, the percent increase in normalized present-value software engineering productivity implied by a residualized AI-index return of 1%:

$$\kappa = \frac{\delta}{\delta^S} \approx 2.05.$$

This means that a residualized AI-index return of 1% implies a 2.05% increase in the normalized present value of software engineering productivity. Aggregating over the event window gives the software engineering productivity gain driven by AI news from November 2022 to December 2025:

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] = \kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} \approx \kappa \times 0.149 \approx 30.5\%,$$

where the cumulative residualized return over the window is 14.9%. If the software engineering productivity level follows a random walk (permanent innovations), each news term equals the corresponding innovation in the productivity level. The sum above therefore corresponds to a permanent 30.5% increase in software engineering productivity.

To benchmark these estimates against direct evidence, the experimental literature finds that AI coding assistants raise task-level productivity by 21–56%: Peng et al. (2023) find a 56% reduction in task completion time for a standardized programming task; Paradis et al. (2025) report a reduction of approximately 21% for a standardized enterprise task at Google; Cui et al. (2026) estimate a 26% increase in pull requests per developer per week, and Gambacorta et al. (2024) find a 55% increase in code output.

Two offsetting forces place our 30.5% estimate in the same range as these task-level figures. First, our estimate is forward-looking: $\mathscr{S}_t$ reflects not only today's AI-driven productivity gains but also anticipated future advances in AI models, workflows, and adoption. This forward-looking component pushes our estimate *above* the gains delivered by current tools alone. Second, weak-link complementarities push in the opposite direction: because a developer's workflow bundles many complementary tasks, total productivity is constrained by the least-improvable steps, so task-level speed-ups overstate overall gains (Jones and Tonetti, 2026; Demirer, Musolff and Yang, 2026). Indeed, recent firm-level evidence is more muted: Chen and Stratton (2026) find that adoption of GitHub Copilot or Cursor reduces software engineers' task completion time by 8.7%. These two forces roughly offset, leaving our estimate in the same range as the experimental evidence.

Real-time productivity news in 2026. One benefit of our approach is that it delivers real-time quantification of AI-driven news about software engineering productivity. Once the multiplier κ, which maps a residualized AI-index return of 1% into a κ% change in the normalized present value of software engineering productivity, is estimated, any subsequent residualized AI-index return translates into market-implied news about software engineering productivity. As an illustration, we extend the return window from the beginning of 2026 through May 29, 2026, the last date for which the official Fama-French factors are currently available. The cumulative residualized AI-index return over this period is $\sum_{t=\text{'26 Jan}}^{\text{'26 May}} \tilde{R}_t^{\text{AI}} = 22.7\%$. Applying the baseline multiplier $\kappa \approx 2.05$ gives a market-implied software engineering productivity gain driven by AI news of $\sum_{t=\text{'26 Jan}}^{\text{'26 May}} \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] \approx 46.5\%$ over the first five months of 2026. This acceleration is consistent with the rapid development and adoption of AI coding agents during this period.

Robustness. Table 8 reports the structural gradient δ^S, the productivity multiplier $\kappa = \delta/\delta^S$, and the event-window market-implied normalized present-value software engineering productivity change $\sum_t \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] = \kappa \cdot \sum_t \tilde{R}_t^{\text{AI}}$ across values of the within-firm substitution elasticity σ. A higher σ increases the inferred productivity gain.

The sensitivity comes from the effect of σ on aggregate substitutability Σ. A larger σ lets firms substitute more easily toward software engineering labor when software-engineering productivity rises, raising Σ; a smaller σ restricts this reallocation and lowers Σ. When Σ is higher, the economy shifts more demand toward software engineering labor, bidding up its relative wage and offsetting part of the cost advantage of SWE-intensive firms. This wage offset lowers the structural gradient δ^S. Hence the same observed payroll-share slope δ and the same residualized AI-index return imply a larger underlying productivity change. With lower substitutability, this wage offset is weaker, so the implied productivity change is smaller.

4 The GDP Impact of AI's Software Engineering Channel

This section translates the AI-induced software engineering productivity change $\mathscr{S}_t$ from Sections 2 and 3 into the aggregate GDP impact of AI via the software engineering channel. Section 4.1 derives this impact by applying Hulten's theorem in the simple environment of Section 2. Section 4.2 introduces an explicit upstream AI sector that supplies an AI input to a software supplier, which bundles a continuum of digital tasks performed by either software engineering labor or AI under comparative advantage. The resulting cost-side isomorphism leaves the inferred AI-induced software engineering productivity gain and GDP impact largely unchanged. Section 4.3 embeds the framework in the full U.S. input-output network. The inferred AI-induced software engineering productivity gain is nearly unchanged; the GDP impact declines moderately.

4.1 Hulten's Theorem for GDP Impact

The strategy above identifies AI-induced software engineering productivity news $\mathscr{S}_t$. An object of significant interest is the GDP change that this productivity change implies. We now establish, in the simple environment of Section 2, the map from $\mathscr{S}_t$ to the corresponding GDP news $\mathscr{Y}_t$ through the software engineering channel.

A software engineering productivity gain lowers firms' costs, and Hulten's theorem would translate this gain into GDP using a sales-based Domar weight (each firm's total sales divided by GDP). That benchmark holds only in an efficient economy and does not apply here: production carries constant markups, and firms buy materials from one another through a final-good roundabout, so a productivity gain at one firm lowers costs elsewhere and feeds back through the network. The relevant per-firm weight is therefore the cost-based Domar weight $\tilde{\lambda}_j$ of Baqaee and Farhi (2020), which in this model is exactly $\tilde{\lambda}_j = f_j/\alpha$: the firm's final-expenditure share f_j scaled by the roundabout multiplier $1/\alpha$. A unit rise in S_t lowers firm j's marginal cost by its software engineering *cost* share $\alpha\gamma_j$; weighting these cost reductions by $\tilde{\lambda}_j$ and summing, the $1/\alpha$ in the weight cancels the α in the cost share, so the component of GDP growth induced by S_t is

$$d\log Y_t\big|_S = \gamma\, d\log S_t, \qquad \gamma = \sum_j f_j\gamma_j,$$

where γ_j is firm j's software engineering payroll share and $\gamma = \sum_j f_j\gamma_j$ its final-expenditure-weighted average.[7] Because our cross-sectional strategy isolates the S_t component, combining this period-by-period map with the inversion of Proposition 1 delivers the headline GDP multiplier in closed form.

Proposition 3 (GDP Impact of AI through the Software Engineering Channel)**.** *Define the period-t news in normalized present-value GDP through the software engineering channel as*

$$\mathscr{Y}_t \equiv (1-\beta)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}\beta^k d\log Y_{t+k}\big|_S\Big].$$

[7] The resulting GDP elasticity γ exceeds the value the standard sales-Domar Hulten formula would assign to the same cost reductions, $\sum_j \lambda_j\alpha\gamma_j = \alpha\theta\gamma$, by the factor $1/(\alpha\theta)$: the joint markup-and-materials correction, which vanishes only without materials ($\alpha \to 1$) or without markups ($\varepsilon \to \infty$). One might expect from Baqaee and Farhi (2020) an additional reallocation term, since with markups a productivity gain that shifts resources across firms moves GDP even at first order. It is absent here because all firms have the same labor share and buy the same materials composite, so the markup wedge is uniform across firms and reallocating between them does not change aggregate efficiency; the input-output network of Section 4.3 breaks this symmetry and revives the term, where we find it to be small.

In our economy,

$$\mathscr{Y}_t = \gamma \mathscr{S}_t.$$

Hence, the best linear predictor of GDP news through the software engineering channel given the contemporaneous movement in the AI index is

$$\mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{AI}] = \gamma\kappa \cdot \tilde{R}_t^{AI}, \qquad \gamma\kappa = \frac{\gamma\Sigma}{\alpha(\varepsilon-1)} \cdot \delta. \tag{15}$$

Equation (15) provides a transparent sufficient-statistic formula for the GDP impact of AI through the software engineering productivity channel. The formula decomposes the multiplier into three components:

$$\gamma\kappa \quad = \quad \underbrace{\delta}_{\substack{\text{observed}\\ \text{cross-sectional slope}}} \quad \times \quad \underbrace{\frac{\Sigma}{\alpha(\varepsilon-1)}}_{\substack{\text{inversion:}\\ \text{index return}\to\mathscr{S}}} \quad \times \quad \underbrace{\gamma}_{\substack{\text{cost-based GDP weight:}\\ \mathscr{S}\to\mathscr{Y}}}$$

The first factor is the observed cross-sectional slope from the data. The second is the inverse of the structural gradient δ^S, converting the observed slope into the normalized present-value software engineering productivity change per unit movement in the residualized AI index. The third factor translates software engineering productivity into GDP: the final-expenditure-weighted payroll share γ governs how much a given productivity gain raises output.

As an illustrative special case, suppose the software engineering productivity level follows a random walk. Then each period-t GDP news term is the innovation in the GDP level, so

$$\mathscr{Y}_t = d\log Y_t\big|_S - \mathbb{E}_{t-1}[d\log Y_t\big|_S] = \gamma\big(d\log S_t - \mathbb{E}_{t-1}[d\log S_t]\big) = \gamma\mathscr{S}_t. \tag{16}$$

By analogy with the productivity multiplier κ, multiplying the GDP multiplier $\gamma\kappa$ by the total residualized AI-index return $\tilde{R}_t^{\text{AI}}$ over an event window yields the corresponding normalized cumulative GDP change through the software engineering channel implied by the AI news in that window. Substituting Σ into the GDP multiplier delivers the equivalent expansion

$$\gamma\kappa = \left[\frac{\sigma\gamma}{\alpha(\varepsilon-1)} + \frac{[\alpha(\varepsilon-1)-(\sigma-1)]\sigma_s^2}{\alpha(\varepsilon-1)(1-\gamma)}\right] \cdot \delta,$$

which separates the direct effect (first term) from the between-firm reallocation correction (second term).

We apply Proposition 3 to the productivity gain of 30.5% computed in Section 3.4, implied by the AI news from November 2022 (the launch of ChatGPT 3.5) to December 2025. The GDP weight is the final-expenditure-weighted software engineering payroll share $\gamma = \sum_j f_j\gamma_j \approx 0.109$. With multiplier $\kappa \approx 2.05$, the implied normalized present-value GDP change through the software engineering channel is

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{\text{AI}}] = \gamma\kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} \approx 0.1086 \times 2.046 \times 14.9\% \approx 3.31\%.$$

If software engineering productivity follows a random walk, the GDP news corresponds to a one-time, permanent increase of 3.31% in the level of GDP through the software engineering channel (not an increase in the GDP growth rate of 3.31% per year).

The final column of Table 8 reports the implied GDP change from the AI-induced software engineering productivity change for different values of σ. Two features stand out. First, the GDP impact is scaled by the final-expenditure-weighted software engineering payroll share γ, which limits the software engineering channel's contribution to GDP. Second, the GDP impact inherits the sensitivity of κ to σ noted in Section 3.4.

Finally, we consider AI-driven GDP news through the software engineering channel in real time. From January through May 2026, the corresponding GDP news is $\sum_{t=\text{'26 Jan}}^{\text{'26 May}} \mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{\text{AI}}] \approx 5.05\%$. This acceleration is consistent with the rapid development and adoption of AI coding agents.

4.2 AI Sector and Task-Based Production

The benchmark model of Section 2 studies AI's impact as a labor-augmenting productivity shock S_t that multiplies software engineering labor. One could argue that AI is not embodied in software engineering labor but is a separately produced commodity, supplied to firms by an upstream sector with its own productivity. One could also argue that the substitution between software engineering labor and AI is most naturally modelled at the task level, with an endogenous automation margin governing which tasks AI performs, as in Acemoglu and Restrepo (2019, 2022). This subsection addresses both concerns jointly. We introduce an explicit upstream AI sector that supplies an AI input to a software supplier that bundles a continuum of digital tasks, each performed by either software engineering labor or AI under comparative advantage. The differentiated-good producer, whose AI beta enters the empirical regression, then aggregates software inputs, materials, and non-SWE labor through a technology similar to that in Section 2. There is a cost-side isomorphism, i.e., firm marginal cost preserves the benchmark functional form. Hence, the inversion procedure mapping the empirical slope δ into the AI-induced software engineering productivity gain and its GDP impact carries through effectively unchanged.

Environment. Time, preferences, demand, and the outer production nest are exactly as in Section 2. We modify the supply side of the software engineering input in two steps. First, a perfectly competitive upstream AI sector supplies a homogeneous AI input. It is built from compute, a scarce primary factor in fixed supply K that households own and supply inelastically, earning a rental rate $p_{K,t}$. Under a linear, compute-augmenting technology, one unit of compute yields S_t^A units of the AI input, so competition prices the AI input at

$$p_t^A = \frac{p_{K,t}}{S_t^A}.$$

A rise in S_t^A lowers the effective price of the AI input for a given compute rental rate.

Second, competitive software suppliers now sell their output to each differentiated-good producer. Supplier j operates a continuum of digital tasks indexed by $z \in [0,1]$. Each task can be performed by software engineering labor or by AI. Let $s_{jt}(z)$ denote the SWE labor allocated to task z for supplier j and $X_{jt}^A(z)$ the corresponding AI input. The task is produced with the linear technology

$$y_{jt}(z) = \psi_S(z) s_{jt}(z) + \psi_A(z) X_{jt}^A(z),$$

where $\psi_S(z), \psi_A(z) > 0$ are task-specific factor productivities common across firms. We order tasks so that the comparative advantage of software engineering labor is increasing in z:

$$\frac{\psi_S(z)}{\psi_A(z)} \text{ is strictly increasing in } z.$$

Higher-z tasks are relatively more productive when performed by SWE labor; lower-z tasks are relatively more productive when performed by AI. The task productivities $\psi_S(z), \psi_A(z)$ and the cross-task elasticity η are uniform across firms.

Task outputs aggregate through a CES with elasticity $\eta > 0$ into a task composite that captures the software output of supplier j,

$$T_{jt} = \left[\int_0^1 y_{jt}(z)^{(\eta-1)/\eta} dz \right]^{\eta/(\eta-1)}.$$

The software composite T_{jt} enters the differentiated-good producer's technology in the role that effective software-engineering labor $S_t s_{jt}$ plays in Section 2: it combines with non-SWE labor through the CES aggregate

$$H_{jt} = \left[\phi_j^{1/\sigma} T_{jt}^{(\sigma-1)/\sigma} + (1-\phi_j)^{1/\sigma} L_{jt}^{(\sigma-1)/\sigma} \right]^{\sigma/(\sigma-1)},$$

which in turn enters the outer Cobb-Douglas nest over H_{jt} and final-good materials, with cost share α on H_{jt}, as in Section 2.

Market clearing is as follows. The final-good market clears with output absorbed by consumption and materials, exactly as in Section 2. The SWE and non-SWE labor markets clear with $\sum_j \int_0^1 s_{jt}(z) dz = s$ and $\sum_j L_{jt} = L$, and the compute market clears with the AI sector's compute demand equal to the fixed endowment, $\sum_j \int_0^{I_t} X_{jt}^A(z)/S_t^A dz = K$.

Equilibrium task allocation. Each task is performed by whichever factor delivers the lowest effective unit cost. The unit cost of task z is $w_t/\psi_S(z)$ if performed by SWE labor and $p_t^A/\psi_A(z)$ if performed by AI. Monotonicity of $\psi_S(z)/\psi_A(z)$ implies that AI performs all tasks below some threshold $I_t \in (0,1)$ and SWE labor performs all tasks above it, with the threshold pinned down by the indifference condition

$$\frac{w_t}{\psi_S(I_t)} = \frac{p_t^A}{\psi_A(I_t)}. \tag{17}$$

A rise in the AI sector's productivity S_t^A lowers p_t^A relative to w_t and shifts I_t to the right: AI takes over additional tasks at the margin. In this sense, the task model here captures the notion that AI advances displace software engineering labor.

Splitting the CES task-price integral at the equilibrium threshold I_t yields the closed-form software price

$$p_t^T = \left[B_A(I_t)(p_t^A)^{1-\eta} + B_S(I_t) w_t^{1-\eta} \right]^{1/(1-\eta)}, \tag{18}$$

with productivity indices $B_A(I_t) = \int_0^{I_t} \psi_A(z)^{\eta-1} dz$ and $B_S(I_t) = \int_{I_t}^1 \psi_S(z)^{\eta-1} dz$. That is, p_t^T is a CES aggregator over the two effective input prices w_t and p_t^A with elasticity η and endogenous weights $B_A(I_t), B_S(I_t)$. Appendix F.5 gives the derivation. Let q_{jt} denote firm j's unit cost of the CES aggregate H_{jt}, and let mc_{jt} denote firm j's marginal cost. Then

$$q_{jt} = \left[\phi_j (p_t^T)^{1-\sigma} + (1-\phi_j) W_t^{1-\sigma} \right]^{1/(1-\sigma)}, \qquad mc_{jt} = \frac{1}{A_t} q_{jt}^{\alpha}. \tag{19}$$

Equation (19) is identical to the benchmark unit cost of Section 2, materials nest and all, with the effective software engineering wage w_t/S_t replaced by the software price p_t^T. In this sense, there is a cost-side isomorphism.

Inversion for the AI-induced software productivity gain and its GDP impact. As in Section 2, we seek to infer the AI-induced normalized present-value productivity change from cross-sectional variation in firm AI-index betas. The task-based environment changes the source of the software productivity improvement: it now comes from upstream AI productivity S_t^A. Specifically, holding the SWE wage and the compute rental rate fixed, differentiating (18) gives $d \log p_t^T = -\rho d \log S_t^A$, with the threshold response dropping out by the indifference condition (17). The change in the total productivity of the software sector is hence given by

$$d \log S_t \equiv \rho d \log S_t^A,$$

where the AI share of software spending,

$$\rho = \frac{B_A(I)(p^A)^{1-\eta}}{(p^T)^{1-\eta}},$$

is common across firms because the task productivities and the task elasticity are uniform.

We now show that our inversion procedure from empirical estimates to the AI-induced software engineering productivity gain remains effectively unchanged with γ_j^T playing the role of γ_j, where $\gamma_j^T \equiv p^T T_j / (p^T T_j + W L_j) = \phi_j (p^T / q_j)^{1-\sigma}$ is the ratio of firm j's costs for software services to firm j's total costs for the value-added input bundle H_{jt}, the counterpart of the software-engineering payroll share γ_j in the main analysis (with total software costs $p^T T_j$ replacing the wage costs of software engineers in the main analysis). Specifically, equation (13), the first stage of our empirical strategy, remains the same, while the second stage now regresses firm betas on γ_j^T. We then invert the resulting estimate δ^T to obtain the normalized present-value software productivity gain driven by AI news, as in Proposition 1, replacing γ, $(\sigma_s)^2$, and Σ with $\gamma^T = \sum_j f_j \gamma_j^T$, $(\sigma_s^T)^2 = \sum_j f_j (\gamma_j^T - \gamma^T)^2$, and

$$\Sigma^T = \sigma + \left[\alpha(\varepsilon - 1) - (\sigma - 1)\right] \frac{(\sigma_s^T)^2}{\gamma^T (1 - \gamma^T)}, \tag{20}$$

where the final-expenditure shares $f_j = \omega_j (p_j / P)^{1-\varepsilon}$ are still defined as in Section 2. The GDP impact in Proposition 3 also follows with γ^T replacing γ.

Proposition 4 (Software Productivity Change and GDP Impact with Task-Based Production)**.** *In the task-based environment, the best linear predictor of the period-t news in normalized present-value software productivity given the contemporaneous movement in the residualized AI index is*

$$\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{AI}] = \kappa \tilde{R}_t^{AI}, \qquad \kappa = \frac{\delta^T}{\delta^{S,T}}, \qquad \delta^{S,T} = \alpha \frac{\varepsilon - 1}{\Sigma^T}, \tag{21}$$

and the corresponding best linear predictor of GDP news through the software channel is

$$\mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{AI}] = \gamma^T \kappa \tilde{R}_t^{AI}. \tag{22}$$

Quantifying the AI-induced software productivity gain and its GDP impact. To carry out the inversion procedure with task-based production, we need an empirical measure of γ_j^T, defined as firms' spending on the software composite T_{jt} (including both compensation of software engineers and costs of purchasing software services) as a share of total costs of the value-added input bundle H_{jt} (including both spending on the software composite and other labor costs). We construct this by augmenting our data on software engineer compensation from Revelio with additional data on firms' software budgets from the Ci Technology Database, produced by the marketing company Harte-Hanks (now owned by Spiceworks Ziff Davis) (Harte-Hanks, 2014).[8] The dataset is built from surveys of establishment-level spending on a variety of IT products, most importantly software, and is targeted at technology vendors seeking information on potential customers' IT needs and budgets. For our measure of firms' software expenditure (distinct from compensation of software engineers), we use the firm's software spending at its headquarters location; this is done in the interest of limiting measurement error concerns, because the total number of establishments surveyed for a particular firm can fluctuate substantially from year to year. To provide evidence that the Harte-Hanks dataset is giving us useful information on firms' software expenditures, we show in Appendix C.3 that Harte-Hanks's surveys capture a large share of total revenue and software expenditure among private businesses.

[8] The software budget data come from Bloom, Draca and Van Reenen (2016).

We measure γ_j^T as follows:

$$\gamma_j^T = \frac{\text{Software engineer compensation} + \text{HQ-location software budget}}{\text{Total labor compensation} + \text{HQ-location software budget}}. \tag{23}$$

We then repeat the second-stage regressions described in Table 4, with γ_j^T in place of γ_j. Here, we trim observations above the 99th percentile of γ_j^T within the subset of Compustat firms for which we are able to identify a match in the Harte-Hanks data. In this matched sample, the regression coefficient on the software composite payroll share variable in the regression corresponding to column (4) of Table 4 is $\delta^T = 1.36$.[9] Using γ_j^T calculated based on (23), we obtain $\gamma^T = 0.112$, $(\sigma_s^T)^2 = 0.0116$, and $\Sigma^T = 1.68$. With these numbers in hand, Proposition 4 gives

$$\kappa = \frac{\Sigma^T}{\alpha(\varepsilon - 1)} \cdot \delta^T = \frac{1.68}{0.311 \times 4} \times 1.36 \approx 1.83.$$

Thus a residualized AI-index return of 1% implies a 1.83% increase in the normalized present value of software productivity. Applying the cumulative residualized AI-index return over the window from November 2022 through the end of 2025 gives the implied software engineering productivity gain:

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] = \kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} \approx 1.83 \times 14.9\% \approx 27.2\%.$$

The corresponding GDP news driven by the software channel is:

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{\text{AI}}] = \gamma^T \kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} \approx 0.112 \times 1.83 \times 14.9\% \approx 3.06\%.$$

Finally, for real-time quantification over the January through May 2026 window, the same procedure implies market-implied software engineering productivity news of 41.5% and corresponding GDP news through the software engineering channel of 4.65%.

4.3 The Production Network

The model in Section 2 has no input-output structure. This subsection embeds the framework in the full U.S. production network: the economy consists of the 65 BEA Summary sectors, each populated by a continuum of firms that differ only in software engineering intensity, and each sector buys intermediate inputs from every other sector, treating capital as part of the intermediate block and keeping the demand elasticity, markup, and labor-substitution elasticity as in Section 2. Does the inversion from cross-sectional estimates to an AI-induced software engineering productivity gain survive input-output linkages, when a firm's marginal cost depends on prices in every other sector? Does the network amplify or dampen the GDP impact of a software engineering productivity gain,

[9]Note that when we re-run the baseline regression from Table 4 (which uses only the software engineer payroll share) in this subsample, the headline coefficient is $\delta = 1.47$, compared to 1.52 in the full analysis sample. So, some of the divergence from the baseline model is due to less-than-perfect merging with the Harte-Hanks data.

given that software is produced upstream and used throughout the economy? Appendix E sets up the economy and provides the results; here we report the answers.

First, the inversion from cross-sectional estimates to an AI-induced software engineering productivity gain remains essentially unchanged. Because firms are atomistic within sectors, a firm's cost advantage only shifts demand among its sector peers. The network effects on intermediate-input prices and sector-level demand are common within sector, so they are absorbed by the second-stage industry fixed effects (Proposition 8). The relative effective software engineering wage response, $d\log w_t - d\log S_t - d\log W_t$ in Section 2, barely moves (-0.58 when solved on the full network, against -0.60 without the network). The structural gradient δ^S_{in}, which we use to invert the empirical second-stage coefficient δ, is computed by replicating the empirical second stage inside the model: we use the same firms as in the empirical sample and drop the same industries as in the empirical specification, while keeping the full IO network active. This gives $\delta^S_{\text{in}} = 0.81$ in place of the 0.74 of Section 4.1. The multiplier becomes $\kappa = \delta/\delta^S_{\text{in}} = 1.52/0.81 \approx 1.88$, so the present-value software engineering productivity driven by AI news over the event window is $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}^{\text{AI}}_t] = \kappa \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}^{\text{AI}}_t = \kappa \times 14.9\% \approx 28\%$, close to the 30.5% of Section 3.4.

Second, the GDP impact through the software engineering channel is moderately lowered relative to the benchmark. Proposition 9 extends the cost-based Hulten logic of Section 4.1 to the network economy. The GDP elasticity γ^{IO} equals the sum of sector software engineering cost shares weighted by cost-based Domar weights, less the small markup-induced factor-reallocation term of Baqaee and Farhi (2020). In our calibration, $\gamma^{\text{IO}} \approx 0.075$. The gap between the network-economy γ^{IO} and the baseline $\gamma \approx 0.109$ mainly reflects the reweighting of sector SWE shares by BEA economy-wide sector sizes. The baseline is computed from Compustat-Revelio firms, whereas the network calculation reweights sector averages using the BEA accounts and includes every sector in the economy. γ^{IO} is smaller because roughly eleven percent of labor income in the full economy is earned in sectors with zero measured software engineering payroll share and that are not in the Compustat-Revelio sample.

Together, the present-value GDP change through the software engineering channel is

$$\sum_{t=\text{'22 Nov}}^{\text{end '25}} \mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}^{\text{AI}}_t] = \gamma^{\text{IO}}\kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}^{\text{AI}}_t \approx 0.075 \times 1.88 \times 14.9\% \approx 2.1\%,$$

smaller than the 3.31% benchmark in Section 4.1. Appendix Table E.1 further decomposes the difference in GDP impact between the benchmark and the network economy, and Appendix Table 9 reports the sensitivity to σ.

Finally, for real-time quantification over the January through May 2026 window, the same mapping implies market-implied software engineering productivity news of 42.7% and corresponding GDP news through the software engineering channel of 3.20%.

5 AI and Innovation

We now extend the framework of Section 2 to allow an AI-induced software engineering productivity gain to affect the macroeconomy through a second channel: innovation. Embedding software engineering labor in a semi-endogenous-growth-style R&D sector adds a new innovation channel: an AI-induced reduction in the cost of effective software engineering labor in R&D facilitates innovation, which in turn raises GDP. Under our tractable framework, the inversion of Section 2 for the AI-induced software engineering productivity gain is unchanged, but its GDP impact is now amplified by a closed-form innovation-augmented multiplier. The headline quantitative implication is that, at our baseline calibration, the implied GDP impact, measured in the same units as in our baseline analysis, as a permanent level effect, rises from 3.31% to roughly 5.98%, a 1.81-fold amplification because R&D is substantially more software-engineering-intensive than production.

5.1 Environment

Time is discrete, indexed by $t \in \{0, 1, \ldots\}$. Household preferences are as in Section 2, and each incumbent variety uses the production technology (1); what changes here is that the set of varieties is endogenous, $j \in [0, N_t]$ with N_t pinned by an endogenous-growth-style R&D sector. We use the same production labor endowments (s, L) and aggregate shocks (S_t, A_t) as in Section 2, drop time variation in the discount factor for clarity, and adopt the aggregate price index as numeraire ($P_t = 1$). Throughout, we drop t subscripts when referring to stationary steady-state values.

Variety producers. With an endogenous continuum of varieties, the household's CES bundle and price index become

$$c_t = \left[\int_0^{N_t} c_{jt}^{(\varepsilon-1)/\varepsilon} dj\right]^{\varepsilon/(\varepsilon-1)} \qquad \text{and} \qquad P_t = \left[\int_0^{N_t} p_{jt}^{1-\varepsilon} dj\right]^{1/(1-\varepsilon)}, \tag{24}$$

with the standard CES demand $c_{jt} = (p_{jt}/P_t)^{-\varepsilon} c_t$, exactly as in Section 2. Materials and the R&D sector final-good input are based on the same composite, so the total demand for variety j is $y_{jt} = (p_{jt}/P_t)^{-\varepsilon} X_t$, where $X_t \equiv c_t + M_t + R_t$ is the sum of consumption c_t, aggregate materials M_t, and the R&D final-good input R_t.[10] The final-expenditure-weighted aggregate software engineering payroll share is $\gamma = \int_0^{N^*} f_j \gamma_j dj$, where the SWE payroll share γ_j is as in Section 2, and $f_j = p_j y_j / X$ are the final-expenditure shares, which sum to one. Each variety survives to the next period with exogenous probability $1 - \eta$, where $\eta \in (0, 1)$ is the obsolescence rate, so the value of an active variety is

$$V_{jt} = \mathbb{E}_t\left[\sum_{\tau=0}^{\infty} (1-\eta)^\tau \Xi_{t,t+\tau} \pi_{j,t+\tau}\right], \tag{25}$$

[10] Specifically, $M_t = [\int_0^{N_t} m_{jt}^{(\varepsilon-1)/\varepsilon} dj]^{\varepsilon/(\varepsilon-1)}$ and $R_t = [\int_0^{N_t} r_{jt}^{(\varepsilon-1)/\varepsilon} dj]^{\varepsilon/(\varepsilon-1)}$

where $\Xi_{t,t+\tau}$ is the SDF from Section 2. This replaces the no-obsolescence value (2).[11]

R&D sector. A competitive R&D sector creates new varieties using three inputs: R&D-specific software engineering labor s_R, R&D-specific non-software-engineering labor L_R, and final-good output R_t. The labor endowments for R&D (s_R, L_R) are fixed and earn wages $(w_{R,t}, W_{R,t})$ that equal their marginal value products in R&D. The final-good output R_t adjusts endogenously. The flow of new varieties follows

$$N_{t+1} = (1-\eta)N_t + \delta_R G(S_t s_R, L_R, R_t)^{\lambda} N_t^{\phi}, \tag{26}$$

where G is a Cobb-Douglas aggregator with constant exponents α_s, α_L, α_R satisfying $\alpha_s+\alpha_L+\alpha_R = 1$,[12] the software engineering productivity S_t augments software engineering labor in R&D exactly as in production, $\delta_R > 0$ is an R&D productivity parameter, $\lambda \in (0,1]$ governs diminishing returns to R&D inputs, capturing duplication externalities, and $\phi < 1$ governs intertemporal knowledge spillovers, with lower values corresponding to stronger fishing-out effects (Jones, 1995; Kortum, 1997; Bloom et al., 2020). The case $\phi = 1$ corresponds to fully endogenous growth, while $\phi < 1$ corresponds to semi-endogenous growth. Following the empirical scale-effects argument in Jones (1995), we focus on the semi-endogenous case $\phi < 1$, in which a permanent increase $d\log S$ in software engineering productivity raises the steady-state level of GDP but does not permanently change the GDP growth rate.

New varieties are drawn from the same type distribution of software engineering intensity ϕ_j as existing varieties, so that the expected value of an entrant equals V^*, where $V^* = \pi^*/(1-\beta(1-\eta))$ and π^* is the average per-period steady-state profit, with the expectation taken over the distribution of ϕ_j. Hence, each additional unit of the final-goods input to R&D, R, creates $\delta_R \lambda G^{\lambda-1} G_R N^{\phi}$ new varieties with $G_R = \alpha_R G/R$, and each newly created variety has expected discounted value βV^*. The steady-state optimality condition on the R margin therefore equates the unit cost of final output to its marginal benefit in variety creation,

$$1 = \beta V^* \cdot \delta_R \lambda G^{*\lambda-1} \frac{\alpha_R G^*}{R^*} N^{*\phi}. \tag{27}$$

Markets, equilibrium, and decoupling. Goods markets clear variety by variety, $y_{jt} = c_{jt} + m_{jt} + r_{jt}$, where m_{jt} and r_{jt} are variety j's uses as materials and as an R&D input. Production labor-market clearing is $\int_0^{N_t} s_{jt}\,dj = s$ and $\int_0^{N_t} L_{jt}\,dj = L$ as in Section 2; R&D labor is sector-specific and clears

[11] In contrast to the fixed weights ω_j summing to one in Section 2, we deliberately do not normalize the aggregator (24) by the variety mass N_t. This missing normalization is precisely love of variety (Romer, 1990): it is the channel through which a larger variety stock raises real output.

[12] Treating software engineering as part of the R&D production function is consistent with evidence that software-intensive firms produce more patents per R&D dollar and that IT complements R&D in patent production (Branstetter, Drev and Kwon, 2019; Han and Ravichandran, 2006).

within the R&D sector. GDP is value added, gross final-good output net of intermediate materials,

$$Y_t = X_t - M_t = c_t + R_t. \tag{28}$$

A competitive equilibrium is a collection of quantities, prices, and firm values such that households and firms optimize, the R&D optimality condition (27) holds, R&D factor returns equal marginal value products, and all markets clear.

5.2 The Inversion for AI-Induced SWE Productivity Gain Is Unchanged

The inversion of Section 2 maps the observed cross-sectional slope δ from the regression of AI-index betas on software engineering payroll shares into AI-induced software engineering productivity change. The same inversion applies in our extension, with the implied present-value software engineering productivity change adjusted for obsolescence, because financial markets price existing varieties taking obsolescence into account.

Specifically, at the steady state, financial markets value each existing variety's future profit stream with discount factor $\beta(1-\eta)$, since the variety survives only with probability $1-\eta$ each period. We therefore define the obsolescence-discounted software engineering productivity news as

$$\mathscr{S}_t^{\eta} \equiv (1-\beta(1-\eta))\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}(\beta(1-\eta))^k d\log S_{t+k}\Big], \tag{29}$$

which replaces the pure β-discounted news object $\mathscr{S}_t$ of Section 2. If the software engineering productivity level follows a random walk (permanent innovations), the obsolescence-discounted news term equals the corresponding innovation in the productivity level. Hence a single unanticipated permanent innovation gives $\mathscr{S}_t^{\eta} = d\log S_t$ regardless of η.

Proposition 5 (Inversion under Endogenous Innovation)**.** *Consider the model of Section 5.1. To first order, each active variety j's unexpected return is linked to news in normalized present-value software engineering productivity by*

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j \mathscr{S}_t^{\eta} + \nu_t, \qquad \delta^S = \alpha\frac{\varepsilon-1}{\Sigma}, \qquad \Sigma = \sigma + \big[\alpha(\varepsilon-1)-(\sigma-1)\big]\frac{\sigma_s^2}{\gamma(1-\gamma)}, \tag{30}$$

where $\sigma_s^2 = \int_0^{N^*} f_j(\gamma_j-\gamma)^2 dj$ is the final-expenditure-weighted cross-sectional variance of software engineering payroll shares at the stationary steady state, and ν_t is firm-invariant. The best linear predictor of news in normalized present-value software engineering productivity given the contemporaneous movement in the residualized AI index is

$$\mathbb{E}^*[\mathscr{S}_t^{\eta} \mid \tilde{R}_t^{AI}] = \kappa \cdot \tilde{R}_t^{AI}, \qquad \textit{where} \quad \kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon-1)}\cdot\delta. \tag{31}$$

This is identical to the benchmark inversion of Proposition 1.

The structural gradient δ^S is the structural gradient from Section 2, with the same dependence on (σ, ε) and on the cost-share dispersion σ_s^2. Endogenous variety creation enters the firm-value response only through two level effects: a common shift in production wages induced by the changing mass of varieties, and a common shift in aggregate goods demand as R^* co-moves with Y^*. Both shifts are firm-invariant and absorbed into v_t; neither contaminates the payroll-share loading that the cross-sectional regression isolates.

5.3 The GDP Impact with Innovation

Because characterizing transitional dynamics is much more involved and analytically intractable in our environment with semi-endogenous growth, we focus on the first-order response of steady-state GDP to a permanent change $d\log S$, holding other aggregate shocks fixed. Following (29), define the obsolescence-discounted news in normalized present-value GDP through the software engineering channel as

$$\mathscr{Y}_t^{\eta} \equiv (1-\beta(1-\eta))\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}(\beta(1-\eta))^k d\log Y_{t+k}\big|_S\Big].$$

If the software engineering productivity level follows a random walk, a single unanticipated permanent innovation in S_t makes $\mathscr{Y}_t^{\eta}$ equal to the corresponding first-order innovation in the GDP level.

We first decompose the first-order response of steady-state GDP into within-variety and variety-margin effects:

$$d\log Y^* = \gamma\, d\log S + \frac{1}{\alpha(\varepsilon-1)} d\log N^*. \tag{32}$$

The first term is the within-variety production-side Hulten effect: cheaper effective software engineering labor lowers unit costs in every production firm, with aggregate impact equal to the final-expenditure-weighted software engineering payroll share γ, the cost-based Domar weight of Proposition 3. The second term is the variety-margin effect, combining love of variety with the dilution of per-variety output when the fixed production endowments (s, L) are spread across more firms. Both coefficients carry the roundabout multiplier $1/\alpha$, reflecting the feedback from final-good use as materials inputs. Setting $d\log N^* = 0$ recovers Proposition 3: with a fixed set of varieties, the GDP impact of a software engineering productivity shock is governed entirely by the cost-based Domar weight γ.

Because a positive $d\log S$ also augments software engineering labor in R&D, it raises effective R&D input G, which increases the flow of new varieties and hence the steady-state variety stock. Through the variety-margin term in (32), the larger variety stock raises GDP. The resulting GDP increase raises the steady-state final-goods input to R&D, R^*, which further raises G and generates a feedback loop. The main result characterizes the fixed point of this loop in closed form.

Proposition 6 (Hulten's Theorem with Innovation)**.** *The first-order response of steady-state GDP to a permanent change $d\log S$ is*

$$d\log Y^* = \frac{\gamma + \iota\alpha_s}{1-\iota\alpha_R} d\log S, \tag{33}$$

where $\iota \equiv \lambda/[\alpha(\varepsilon-1)(1-\phi)]$. The formula requires two regularity conditions: $\iota\alpha_R < 1$ for a stable, positive finite feedback multiplier, and $R^*/Y^* = \theta\beta\lambda\eta\alpha_R/[\varepsilon(1-\beta(1-\eta))] < 1$ for positive consumption.[13]

The innovation-augmented GDP coefficient in (33) combines three economically distinct objects: γ, $\iota\alpha_s$, and $\iota\alpha_R$. The first, γ, is the production channel in (32) as in the main analysis: cheaper effective software engineering labor lowers unit costs in every production firm, with aggregate impact equal to the final-expenditure-weighted software engineering payroll share, the cost-based Domar weight of Proposition 3. The second, $\iota\alpha_s$, is the direct innovation channel from S augmenting software engineering labor in R&D. More effective software engineering labor raises the effective R&D input G, which raises the steady-state variety count by the semi-endogenous factor $\lambda/(1-\phi)$. The larger variety stock then raises Y^* through love of variety by the additional factor $1/(\alpha(\varepsilon-1))$, where the $1/\alpha$ is the same materials-roundabout multiplier that appears in the production channel. The product of these two factors is ι (the semi-endogenous growth "increasing returns" term in the terminology of Jones (2022)), and weighting by the software engineering share of R&D inputs gives $\iota\alpha_s$. The third term, $\iota\alpha_R$, governs the GDP feedback loop in the denominator: higher GDP raises the final-goods input to R&D, R^*, and higher R^* raises effective R&D input G, the steady-state variety stock N^*, and hence GDP further.

Taking $\lambda \to 0$ shuts off the innovation channel entirely: $\iota \to 0$ and $d\log Y^* \to (\gamma)d\log S$, recovering the production-side Hulten formula of Proposition 3. The same limit obtains by setting both R&D-side exponents α_s and α_R to zero, in which case the direct innovation channel and the GDP feedback loop are both shut down even when ι is large.

5.4 Quantification

This subsection quantifies the GDP impact of AI's software engineering channel, taking the innovation channel into account. The exercise inherits $\gamma \approx 0.108$ from Section 4.1, the demand elasticity $\varepsilon = 5$, and the permanent software engineering productivity change $d\log S = 0.305$ from Section 3.4, because Proposition 5 preserves the inversion to AI-induced software engineering productivity gains.

For the new parameters introduced by the innovation block, the innovation-augmented GDP coefficient in (33) depends on (λ, ϕ) only through the composite $\iota = \lambda/[\alpha(\varepsilon-1)(1-\phi)]$, so we calibrate

[13] Here Y^* denotes GDP, equal to consumption plus R&D investment: $Y^* = c^* + R^*$. Since the steady-state R&D spending share R^*/Y^* is constant, consumption has the same first-order log response as GDP: $d\log c^* = d\log Y^*$.

this composite directly. In the semi-endogenous accounting of Jones (2022), the corresponding object is the increasing returns parameter $\gamma_J = \lambda\sigma_J/\beta_J$, where σ_J maps ideas into output and β_J is the dynamic diminishing-returns parameter in idea production. In our notation, a larger idea (variety) stock raises real value-added output with elasticity $1/(\alpha(\varepsilon-1))$, the variety-margin coefficient in (32), so $\sigma_J = 1/(\alpha(\varepsilon-1))$; the fishing-out term is $\beta_J = 1-\phi$. Hence γ_J is exactly our $\iota = \lambda/[\alpha(\varepsilon-1)(1-\phi)]$. Following Jones (2022), we set $\iota = \gamma_J = 1/3$. This value is close to the corresponding value in Bloom et al. (2020): their baseline estimates imply $\iota = 1/3.1 \approx 0.323$, based on the aggregate estimate $\beta_J = 3.1$ combined with $\lambda = 1$.

The R&D non-labor share α_R is the cost share of final output used by the R&D sector. Using the NCSES Business Enterprise R&D Survey cost categories and classifying salaries, wages, fringe benefits, stock-based compensation, and temporary staffing as labor-like costs gives $\alpha_s + \alpha_L \approx 0.68$ and hence $\alpha_R \approx 0.32$. Given α_R, the relevant uncertainty is the split of R&D labor between software engineering and other labor. Our baseline sets $\alpha_s = 0.20$ and hence $\alpha_L = 0.48$, based on the Cobb-Douglas patent production function in Kleis et al. (2012), which estimates an IT-capital elasticity close to 0.20. We also report a case with a lower software engineering R&D share, $\alpha_s = 0.04$ and hence $\alpha_L = 0.64$, based on the IT-skills coefficient in the patent-value regressions of Brynjolfsson, Jin and Steffen (2024).

Back-of-the-envelope calculation. At the baseline calibration $(\iota, \alpha_s, \alpha_R) = (1/3, 0.20, 0.32)$, both stability conditions are satisfied by a comfortable margin. The direct innovation channel contributes $\iota\alpha_s \approx 0.0667$, and the GDP-feedback denominator is $1-\iota\alpha_R \approx 0.893$, corresponding to a feedback multiplier of $1/(1-\iota\alpha_R) \approx 1.12$. Combining these terms gives the innovation-augmented GDP coefficient:

$$\frac{\gamma + \iota\alpha_s}{1-\iota\alpha_R} = \frac{0.1086 + 0.0667}{0.893} \approx 0.196. \tag{34}$$

Compared to the production-only Hulten coefficient $\gamma \approx 0.109$, the macro impact of AI through the software engineering channel is roughly 1.81 times as large. The amplification decomposes as follows: the direct innovation channel $\iota\alpha_s$ raises the coefficient from 0.109 to 0.175 (a factor of 1.61), and the GDP feedback loop raises it from 0.175 to 0.196 (a further factor of 1.12). Using a permanent shock to software-engineering productivity $d\log S = 0.305$ from Section 3.4 and the residualized AI return based on news from November 2022 to December 2025, the implied AI-induced GDP change through the software engineering channel is

$$d\log Y^* = \frac{\gamma + \iota\alpha_s}{1-\iota\alpha_R} \cdot d\log S \approx 0.196 \times 0.305 \approx 5.98\%, \tag{35}$$

compared to the 3.31% production-only baseline of Section 4.

The lower software engineering R&D share case, with $\alpha_s = 0.04$ and $\alpha_L = 0.64$, gives a GDP impact $d\log Y^* = 4.16\%$. The difference comes from the direct innovation term $\iota\alpha_s$: holding α_R fixed, a

Table 1: Innovation-augmented multiplier (Proposition 6)

Specification	ι	$\iota\alpha_R$	$\iota\alpha_s$	Coefficient	$d\log Y^*$	Ratio
Standard Hulten ($\iota = 0$)	—	—	—	0.1086	3.31%	1.00×
Innovation Baseline	0.333	0.1067	0.0667	0.1962	5.98%	**1.81**×
Lower α_s	0.333	0.1067	0.0133	0.1365	4.16%	1.26×

Notes: Each row reports the innovation-augmented multiplier from Proposition 6. The production-side inputs are fixed at the cost-based Domar Hulten weight $\gamma \approx 0.109$ and $\varepsilon = 5$, inherited from Section 4.1. The innovation parameters are fixed at $\iota = 1/3$ and $\alpha_R = 0.32$. The baseline sets $(\alpha_s, \alpha_L) = (0.20, 0.48)$, while the lower software R&D case sets $(\alpha_s, \alpha_L) = (0.04, 0.64)$. The "Coefficient" column is $(\gamma + \iota\alpha_s)/(1 - \iota\alpha_R)$ from (33); the "$d\log Y^*$" column applies it to $d\log S = \mathscr{S}_t^\eta \approx 0.305$. The first row sets $\iota = 0$ and recovers the production-side Hulten coefficient γ.

smaller software engineering share in R&D means that AI-induced software engineering productivity gains matter less for innovation.

Finally, our real-time quantification for January through May 2026 implies an innovation-augmented GDP impact equivalent to a one-time permanent increase of 9.11% in the level of GDP through the software engineering channel.

6 Conclusion

This paper uses asset prices to infer the macroeconomic impact of AI through the software engineering channel. By exploiting the cross-sectional relationship between firms' AI-index betas and their software engineering payroll shares, and inverting this relationship through a structural model, we estimate that news about AI from November 2022 to December 2025 implies a present-value increase in software engineering productivity equivalent to a permanent increase of roughly 30.5 percent, and a present-value GDP increase through the software engineering channel equivalent to a permanent increase of approximately 3.31 percent. Accounting for the innovation channel can raise the permanent GDP impact to roughly 1.8 times the baseline estimate. A further benefit of our approach is that it quantifies AI-driven productivity news in real time. Applying our procedure to early 2026 implies large software engineering productivity and GDP gains from AI news during this period. These gains are consistent with the rapid development and adoption of AI coding agents during this period.

Our approach naturally extends to other occupation groups through which AI may affect the macroeconomy by applying the same cross-sectional identification strategy with occupation-specific cost shares. Future work can use this framework to quantify these additional channels and assess their contribution to the macroeconomic impact of AI.

Tables

Filter	Firms	Firms with 2021 Sales Data	Percentiles of 2021 Sales				
			P10	P25	P50	P75	P90
Initial CRSP dataset (active GVKEY links)	5,724	4,737	0.4	34.5	375.6	2,161.8	9,879.3
>=1 year in Compustat with AT>0 and SALE>0	5,140	4,355	9.3	69.9	498.0	2,655.7	11,272.1
Drop NAICS 52/53	4,101	3,404	6.8	55.1	514.5	2,930.2	12,090.3
Revelio match in 2021	2,739	2,573	7.8	73.6	578.5	3,064.7	11,648.6
Average sales/assets filter	2,621	2,573	7.8	73.6	578.5	3,064.7	11,648.6
Balanced panel	2,458	2,415	12.3	107.8	658.4	3,332.9	12,279.2
Drop firms in THNQ/AIQ	2,389	2,346	11.5	97.6	608.1	3,141.9	11,235.8
Drop semiconductor-related firms	2,318	2,275	11.0	97.5	607.2	3,198.0	11,528.9
Drop NAICS 51/54	1,893	1,852	9.7	93.8	732.9	3,647.2	12,778.9
Revelio size trim	1,798	1,761	11.0	103.9	812.6	3,886.8	13,327.6

Note: Sales is Compustat `sale` (net sales/revenue), measured in USD millions for fiscal year 2021. Row 1 percentiles use the latest 2021 Compustat filing before the `at`>0 and `sale`>0 screen; later rows use the cleaned Compustat panel.

Table 2: Sample size progression and 2021 sales percentiles across key cleaning filters.

Table 3: Firms with the highest and lowest software engineer payroll shares

Rank	Firm	NAICS2	Market Cap (2021)
Panel A: 20 firms with the highest software engineer payroll shares			
1	JUMIA TECHNOLOGIES AG	44-45	$0.9 B
2	INNOVIZ TECHNOLOGIES LTD	31-33	$0.9 B
3	FULL TRUCK ALLIANCE CO LTD	48-49	$0.7 B
4	UBIQUITI INC	31-33	$19.2 B
5	PDD HOLDINGS INC	44-45	$25.8 B
6	CIMPRESS PLC	31-33	$1.9 B
7	ENERGOUS CORP	31-33	$0.1 B
8	MASTECH DIGITAL INC	56	$0.2 B
9	MOGU INC -ADR	44-45	$0.0 B
10	CIENA CORP	31-33	$11.9 B
11	GARMIN LTD	31-33	$26.2 B
12	CALIX INC	31-33	$5.1 B
13	ARLO TECHNOLOGIES INC	31-33	$0.9 B
14	CAMBIUM NETWORKS CORP	31-33	$0.7 B
15	SONOS INC	31-33	$3.8 B
16	EXTREME NETWORKS INC	31-33	$2.0 B
17	COMTECH TELECOMMUN	31-33	$0.6 B
18	EBAY INC	44-45	$39.5 B
19	HARMONIC INC	31-33	$1.2 B
20	NETAPP INC	31-33	$20.4 B
Panel B: 20 largest firms with zero software engineer payroll share			
1	O'REILLY AUTOMOTIVE INC	44-45	$47.3 B
2	ICAHN ENTERPRISES LP	99	$14.5 B
3	WESTERN MIDSTRM PRTNRS LP	48-49	$9.1 B
4	AMN HEALTHCARE SERVICES INC	56	$5.8 B
5	SONOCO PRODUCTS CO	31-33	$5.7 B
6	CELSIUS HOLDINGS INC	31-33	$5.6 B
7	WINGSTOP INC	72	$5.2 B

Continued on next page

Table 3: Firms with the highest and lowest software engineer payroll shares (continued)

Rank	Firm	NAICS2	Market Cap (2021)
8	NATIONAL BEVERAGE CORP	31-33	$4.2 B
9	ACADEMY SPO AND OU INC	44-45	$3.9 B
10	AMERICAN STATES WATER CO	22	$3.8 B
11	INTERPARFUMS INC	31-33	$3.4 B
12	CANNAE HOLDINGS INC	72	$3.1 B
13	LATHAM GROUP INC	31-33	$3.0 B
14	CLEARWAY ENERGY INC	22	$2.9 B
15	KENON HOLDINGS LTD	22	$2.8 B
16	H2O AMERICA	22	$2.2 B
17	KNOWLES CORP	31-33	$2.2 B
18	PLAINS GP HOLDINGS LP	42	$2.0 B
19	XENON PHARMACEUTICALS INC	31-33	$1.6 B
20	GRIFFON CORP	31-33	$1.6 B

Notes: Panel A reports the 20 firms with the highest software engineer payroll shares in the sample. Panel B reports the 20 largest firms with zero software engineer payroll share in Revelio. Market capitalization is reported in billions of U.S. dollars.

Table 4: Regression of THNQ beta on software engineer payroll share (continuous)

Dependent Variable:	Beta on THNQ excess return				
Model:	(1)	(2)	(3)	(4)	(5)
Variables					
Software engineer payroll share	1.86***	2.02***	2.04***	1.52***	1.69***
	(0.23)	(0.22)	(0.23)	(0.26)	(0.33)
Fixed-effects					
Quintile of average 2017-2021 assets		Yes	Yes	Yes	Yes
Quintile of average 2017-2021 sales		Yes	Yes	Yes	Yes
NAICS2			Yes		
NAICS3				Yes	Yes
Quintile of GenAI task exposure (Eisfeldt et al., 2026)					Yes
Fit statistics					
Observations	1,780	1,780	1,780	1,773	1,093
R^2	0.04	0.09	0.11	0.15	0.23
Within R^2		0.05	0.05	0.02	0.04

HC3 standard-errors in parentheses

*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Notes: The dependent variable is the firm-level beta on THNQ excess returns, estimated in a first-stage weekly regression of firm excess returns on THNQ excess returns with FF3 controls over the 2022-2025 balanced panel. The key regressor is the software engineer payroll share, measured using Revelio and defined as the share of firm payroll attributable to software engineer occupations. Columns differ only in the controls and fixed effects indicated in the table. All columns trim observations above the 99th percentile of the software engineer payroll share in the main analysis sample. HC3 standard errors are reported in parentheses. Observations are firms.

		Coefficient	Std. Error	Observations
1.	Baseline	1.52	0.26	1773
2.	Drop micro-cap firms	1.80	0.30	988
3.	Alternate AI index: AIQ	1.41	0.35	1773
4.	NASDAQ control in first stage	1.50	0.26	1773
5.	FF5 factors in first stage	1.26	0.28	1773
6.	Ex-“Mag 7” market factor in first stage	1.47	0.25	1773
7.	Weight by end-of-2021 market cap	1.19	0.36	1773
8.	Control for log(end-of-2021 market cap)	1.72	0.27	1773
9.	Weight by software engineer payroll share	1.24	0.34	1577
10.	Use software engineer employment share	1.34	0.27	1773
11.	Control for common financial ratios	1.64	0.26	1726
12.	HQ state FE	1.40	0.29	1603
13.	US-incorporated firms only	1.36	0.28	1564
14.	NAICS4 FE	1.55	0.32	1687
15.	NAICS6 FE	1.48	0.40	1422
16.	Control for GenAI product exposure	1.53	0.37	804
17.	Control for adjacent occupation payroll shares	1.74	0.28	1756
18.	Bootstrap SE (block + firm resample)	1.52	0.44	1773

Notes: The Baseline row corresponds to the baseline NAICS3 fixed-effects specification in the main continuous table. Each subsequent row reports a one-at-a-time departure from that baseline. Unless otherwise noted in the row label, the dependent variable, software-intensity measure, and core controls are unchanged. Observation counts vary only when the robustness check changes the estimation sample; we apply the same trimming procedure as in Table 4 except in Row 10, where we trim at the 99th percentile of the software engineering employment share. Reported standard errors are HC3. Row 11 controls for book-to-market, operating profitability, capex to assets, and book leverage. Row 16 controls for three measures of firms' exposure to generative AI through the product market, as defined in Table VII of Eisfeldt et al. (2026); we do not use the “GS Product Exp” measure because it only takes values of 0 among our final set of firms, and we only retain firms with non-missing values for all of the three remaining measures. Bootstrap row: calendar weeks are resampled using a moving-block bootstrap with block length 6 weeks, firms are resampled with replacement, firm-level AI betas are re-estimated in the first stage, and the baseline second-stage regression is re-estimated in each iteration. The reported bootstrap standard error is the standard deviation of the bootstrap coefficient draws.

Table 5: Streamlined robustness checks relative to the NAICS3 FE anchor specification.

Table 6: Regression of THNQ beta on software engineer payroll share quintiles

Dependent Variable:	Beta on THNQ excess return				
Model:	(1)	(2)	(3)	(4)	(5)
Variables					
Quintile 1	-0.02	0.06	0.06	0.06	-0.01
	(0.06)	(0.06)	(0.06)	(0.06)	(0.08)
Quintile 2	-0.04	0.09	0.09	0.07	0.01
	(0.05)	(0.06)	(0.06)	(0.06)	(0.08)
Quintile 3	-0.04	0.09	0.10*	0.05	0.02
	(0.05)	(0.05)	(0.06)	(0.06)	(0.08)
Quintile 4	0.07	0.21***	0.23***	0.14**	0.08
	(0.05)	(0.06)	(0.06)	(0.06)	(0.08)
Quintile 5	0.24***	0.35***	0.36***	0.25***	0.20**
	(0.05)	(0.06)	(0.06)	(0.06)	(0.08)
Fixed-effects					
Quintile of average 2017-2021 assets		Yes	Yes	Yes	Yes
Quintile of average 2017-2021 sales		Yes	Yes	Yes	Yes
NAICS2			Yes		
NAICS3				Yes	Yes
Quintile of GenAI task exposure (Eisfeldt et al., 2026)					Yes
Fit statistics					
Observations	1,780	1,780	1,780	1,773	1,093
R^2	0.04	0.08	0.11	0.15	0.22
Within R^2		0.04	0.05	0.02	0.02

HC3 standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Notes: Quintile indicators are constructed from the same software engineer payroll-share measure used in the continuous specification. Firms with positive software engineer payroll share are partitioned into quintiles, and coefficients are reported relative to the omitted zero-share group. Columns retain the same sequence of controls, fixed effects, and trimming procedure as in 4. HC3 standard errors are reported in parentheses. Observations are firms.

Table 7: Calibrated parameter values

Parameter	**Description**	**Baseline**	**Range**	**Source**
Data-based objects				
γ	Sales-weighted mean software engineering payroll share	0.109	—	Our data
σ_s^2	Cross-sectional variance of γ_j	0.0119	—	Our data
δ	Cross-sectional slope (payroll-share basis)	1.52	—	Table 4, col. 4
δ^{cost}	Cross-sectional slope (cost-share basis)	4.87	—	δ/α; KLEMS $\alpha = 0.311$
Ω	IO cost-share matrix	—	—	BEA Make-Use
Calibrated parameters				
ε	Demand elasticity	5	—	Broda and Weinstein (2006)
σ	SWE–non-SWE substitution elast.	1.6	$\{0.5, 1, 1.5, 2\}$	Aum and Shin (2024)
β	Annual discount factor	0.96	—	Standard
Production network extension (Section 4.3)				
ζ	Across-sector demand elasticity	$\varepsilon = 5$	$\zeta \leq \varepsilon$	Nests Section 2

Notes: "Data-based objects" are directly computable from publicly available data. "Calibrated parameters" are taken from the literature or varied in sensitivity analysis. In the production network extension, ε is the within-sector (across-firm) demand elasticity, and $\zeta = \varepsilon$ is the benchmark under which the two-tier economy collapses to the single-tier model of Section 2 (with its roundabout intermediate block); lowering ζ leaves the headline essentially unchanged (from 2.09% at $\zeta = 5$ to 2.06% as $\zeta \to 1$).

Table 8: Productivity and GDP sensitivity to the within-firm substitution elasticity (simple model, no IO)

σ	Payroll gradient δ^S	Multiplier $\kappa = \delta/\delta^S$	Implied productivity change (%)	Implied GDP change (%)
0.5	1.74	0.87	13.0	1.41
1.0	1.08	1.40	20.9	2.27
1.5	0.78	1.94	28.9	3.14
1.6 (baseline)	0.74	2.05	30.5	3.31
2.0	0.61	2.47	36.9	4.00

Notes: The demand elasticity is fixed at the baseline value $\varepsilon = 5$. Data-based inputs, rounded for display, are the all-sector Compustat-Revelio objects from Section 3.4: $\gamma \approx 0.109$, $\sigma_s^2 \approx 0.0119$, $\alpha = 0.311$, and $\delta = 1.52$. δ^S is the payroll-share structural cross-sectional gradient from Proposition 1. $\kappa = \delta/\delta^S$ is the software engineering productivity news per unit of residualized AI-index return: for example, $\kappa = 2.05$ means that a residualized AI-index return of 1% predicts software engineering productivity news of 2.05%. The equivalent cost-share expression is $\kappa = \delta^{\text{cost}}/[(\varepsilon-1)/\Sigma]$, where $\delta^{\text{cost}} = \delta/\alpha$. The final two columns report the event-window fitted present-value productivity and GDP news, computed as $\kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}}$ and $\gamma\kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}}$, using $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} = 14.9\%$. Each cell uses the no-IO formula from Section 2.

Table 9: GDP impact sensitivity to the within-firm substitution elasticity (production network economy)

σ	δ^S_{in}	κ	Implied productivity change (%)	Implied GDP change (%)
0.5	1.87	0.81	12.1	0.86
1.0	1.17	1.30	19.3	1.42
1.5	0.85	1.78	26.5	1.98
1.6 (baseline)	0.81	1.88	27.9	2.09
2.0	0.67	2.26	33.7	2.54

Notes: Each row solves the production network economy of Appendix E at the indicated σ, holding $\varepsilon = \zeta = 5$, $\mu = \varepsilon/(\varepsilon - 1) = 1.25$, and the baseline slope $\delta = 1.52$ (payroll-share basis). δ^S_{in} is the in-sample structural gradient from Proposition 8 and $\kappa = \delta/\delta^S_{\text{in}}$ the software engineering productivity news per unit of AI-index return. The GDP news through the software engineering channel is $\gamma^{\text{IO}}\kappa \cdot \sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}^{\text{AI}}_t$, using $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}^{\text{AI}}_t = 14.9\%$ and the weight γ^{IO} of Proposition 9 recomputed at each σ (the reallocation term varies from 0.006 at $\sigma = 0.5$ to 0.002 at $\sigma = 2$).

References

Acemoglu, Daron. 2025. "The simple macroeconomics of AI." *Economic Policy,* 40(121): 13–58.

Acemoglu, Daron, and Pascual Restrepo. 2018. "The race between man and machine: implications of technology for growth, factor shares, and employment." *American Economic Review,* 108(6): 1488–1542.

Acemoglu, Daron, and Pascual Restrepo. 2019. "Automation and new tasks: how technology displaces and reinstates labor." *Journal of Economic Perspectives,* 33(2): 3–30.

Acemoglu, Daron, and Pascual Restrepo. 2022. "Tasks, automation, and the rise in U.S. wage inequality." *Econometrica,* 90(5): 1973–2016.

Acemoglu, Daron, David Autor, Jonathon Hazell, and Pascual Restrepo. 2022. "Artificial intelligence and jobs: evidence from online vacancies." *Journal of Labor Economics,* 40(S1): S293–S340.

Aghion, Philippe, and Simon Bunel. 2024. "AI and growth: where do we stand?" *Working Paper.*

Aghion, Philippe, Benjamin Jones, and Charles Jones. 2019. "Artificial intelligence and economic growth." In *The Economics of Artificial Intelligence: An Agenda.* 237–290. University of Chicago Press.

Agrawal, Ajay, John McHale, and Alexander Oettl. 2026. "Enhancing worker productivity without automating tasks: a different approach to AI and the task-based model." *NBER Working Paper.*

Althoff, Lukas, and Hugo Reichardt. 2026. "Task-specific technical change and comparative advantage." *Working Paper.*

Andrews, Isaiah, and Maryam Farboodi. 2025. "Do markets believe in transformative AI?" *NBER Working Paper.*

Aum, Sangmin, and Yongseok Shin. 2024. "Is software eating the world?" *NBER Working Paper.*

Babina, Tania, Anastassia Fedyk, Alex He, and James Hodson. 2024. "Artificial intelligence, firm growth, and product innovation." *Journal of Financial Economics,* 151: 103745.

Babina, Tania, Anastassia Fedyk, Alex He, and James Hodson. 2025. "Artificial intelligence and firms' systematic risk." *Working Paper.*

Baqaee, David Rezza, and Emmanuel Farhi. 2020. "Productivity and misallocation in general equilibrium." *The Quarterly Journal of Economics,* 135(1): 105–163.

Bertomeu, Jeremy, Yupeng Lin, Yibin Liu, and Zhenghui Ni. 2025. "The economic consequences of disrupted generative AI adoption." *Working Paper.*

Besiroglu, Tamay, Nicholas Emery-Xu, and Neil Thompson. 2024. "Economic impacts of AI-augmented R&D." *Research Policy,* 53(7): 105037.

Bick, Alexander, Adam Blandin, and David Deming. 2026. "The rapid adoption of generative AI." *Management Science.*

Bloom, Nicholas, Charles I. Jones, John Van Reenen, and Michael Webb. 2020. "Are ideas getting harder to find?" *American Economic Review*, 110(4): 1104–1144.

Bloom, Nicholas, Mirko Draca, and John Van Reenen. 2016. "Trade induced technical change? The impact of Chinese imports on innovation, IT and productivity." *The Review of Economic Studies*, 83(1): 87–117.

Branstetter, Lee, Matej Drev, and Namho Kwon. 2019. "Get with the program: software-driven innovation in traditional manufacturing." *Management Science*, 65(2): 541–558.

Briggs, Joseph, and Devesh Kodnani. 2023. "The potentially large effects of artificial intelligence on economic growth." Goldman Sachs Global Economics Analyst. Available at https://www.gspublishing.com/content/research/en/reports/2023/03/27/d64e052b-0f6e-45d7-967b-d7be35fabd16.html.

Broda, Christian, and David Weinstein. 2006. "Globalization and the gains from variety." *Quarterly Journal of Economics*, 121(2): 541–585.

Brynjolfsson, Erik, Danielle Li, and Lindsey Raymond. 2025. "Generative AI at work." *Quarterly Journal of Economics*, 140(2): 889–942.

Brynjolfsson, Erik, Daniel Rock, and Chad Syverson. 2019. "Artificial intelligence and the modern productivity paradox: a clash of expectations and statistics." In *The Economics of Artificial Intelligence: An Agenda*. 23–57. University of Chicago Press.

Brynjolfsson, Erik, Daniel Rock, and Chad Syverson. 2021. "The productivity J-curve: how intangibles complement general purpose technologies." *American Economic Journal: Macroeconomics*, 13(1): 333–372.

Brynjolfsson, Erik, Wang Jin, and Sebastian Steffen. 2024. "Do IT capabilities still drive productivity and innovation in the digital age?" *Working Paper*.

Campbell, John, Christopher Polk, and Tuomo Vuolteenaho. 2010. "Growth or glamour? Fundamentals and systematic risk in stock returns." *Review of Financial Studies*, 23(1): 305–344.

Center for Research in Security Prices. 2025. "CRSP US Stock Databases Guide." Documentation.

Chen, Fiona, and James Stratton. 2026. "Artificial intelligence in the firm." *Working Paper*.

Ciccone, Antonio, and Giovanni Peri. 2005. "Long-run substitutability between more and less educated workers: evidence from U.S. states, 1950–1990." *Review of Economics and Statistics*, 87(4): 652–663.

Cui, Kevin Zheyuan, Mert Demirer, Sonia Jaffe, Leon Musolff, Sida Peng, and Tobias Salz. 2026. "The effects of generative AI on high-skilled work: evidence from three field experiments with software developers." *Management Science*.

Davidson, Tom, Basil Halperin, Thomas Houlden, and Anton Korinek. 2026. "When does automating AI research produce explosive growth? Feedback loops in innovation networks." *NBER Working Paper*.

Dell'Acqua, Fabrizio, Edward McFowland III, Ethan Mollick, Hila Lifshitz, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim Lakhani. 2026. "Navigating the jagged technological frontier: field experimental evidence of the effects of artificial intelligence on knowledge worker productivity and quality." *Organization Science*, 37(2): 403–423.

Demirer, Mert, Leon Musolff, and Liyuan Yang. 2026. "Writing code vs. shipping code: productivity effects across generations of AI coding tools." *NBER Working Paper*.

De Ridder, Maarten. 2024. "Market power and innovation in the intangible economy." *American Economic Review*, 114(1): 199–251.

Eisfeldt, Andrea, Gregor Schubert, Bledi Taska, and Miao Ben Zhang. 2026. "Generative AI and firm values." *Journal of Finance*.

Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock. 2024. "GPTs are GPTs: labor market impact potential of large language models." *Science*, 384(6702): eadj0998.

Fama, Eugene, and James MacBeth. 1973. "Risk, return, and equilibrium: empirical tests." *Journal of Political Economy*, 81(3): 607–636.

Fama, Eugene, and Kenneth French. 1993. "Common risk factors in the returns on stocks and bonds." *Journal of Financial Economics*, 33(1): 3–56.

Fama, Eugene, and Kenneth French. 2015. "A five-factor asset pricing model." *Journal of Financial Economics*, 116(1): 1–22.

Felten, Edward, Manav Raj, and Robert Seamans. 2021. "Occupational, industry, and geographic exposure to artificial intelligence: a novel dataset and its potential uses." *Strategic Management Journal*, 42(12): 2195–2217.

Freund, Lukas, and Lukas Mann. 2026. "Job transformation, specialization, and the labor market effects of AI." *Working Paper*.

Gambacorta, Leonardo, Han Qiu, Shuo Shan, and Daniel Rees. 2024. "Generative AI and labour productivity: a field experiment on coding." *Working Paper*.

Hampole, Menaka, Dimitris Papanikolaou, Lawrence Schmidt, and Bryan Seegmiller. 2025. "Artificial intelligence and the labor market." *NBER Working Paper*.

Han, Shu, and T. Ravichandran. 2006. "Does IT impact firm innovativeness: an empirical examination of complementary and direct effects." *AMCIS 2006 Proceedings, Paper 91*.

Harte-Hanks. 2014. "Ci Technology Database Relational Database Format Data Dictionary." Revision 3.1 ed., Harte Hanks Market Intelligence, Last updated: 1Q14. A guide to understanding the scope and value of the 2014 database.

Jones, Charles, and Christopher Tonetti. 2026. "Past automation and future A.I.: how weak links tame the growth explosion." *Working Paper*.

Jones, Charles I. 1995. "R&D-based models of economic growth." *Journal of Political Economy*, 103(4): 759–784.

Jones, Charles I. 2022. "The past and future of economic growth: a semi-endogenous perspective." *Annual Review of Economics*, 14(1): 125–152.

Karger, Ezra, Otto Kuusela, Jason Abaluck, Kevin Bryan, Basil Halperin, Todd Jones, Connacher Murphy, Philip Trammell, Matt Reynolds, Dan Mayland, Ria Viswanathan, Ananaya Mittal, Rebecca Ceppas de Castro, Josh Rosenberg, and Philip Tetlock. 2026. "Forecasting the economic effects of AI." *NBER Working Paper*.

Katz, Lawrence, and Kevin Murphy. 1992. "Changes in relative wages, 1963–1987: supply and demand factors." *Quarterly Journal of Economics*, 107(1): 35–78.

Kleis, Landon, Paul Chwelos, Ronald Ramirez, and Iain Cockburn. 2012. "Information technology and intangible output: the impact of IT investment on innovation productivity." *Information Systems Research*, 23(1): 42–59.

Korinek, Anton, and Donghyun Suh. 2024. "Scenarios for the transition to AGI." *NBER Working Paper*.

Kortum, Samuel S. 1997. "Research, patenting, and technological change." *Econometrica*, 65(6): 1389–1419.

Lettau, Martin, and Jessica Wachter. 2007. "Why is long-horizon equity less risky? A duration-based explanation of the value premium." *Journal of Finance*, 62(1): 55–92.

Lindenlaub, Ilse, Ryungha Oh, María Alejandra Rodríguez, and Laura Veldkamp. 2026. "Beyond exposure: predicting AI adoption based on comparative advantage." *Working Paper*.

McElheran, Kristina, Frank Li, Erik Brynjolfsson, Zachary Kroff, Emin Dinlersoz, Lucia Foster, and Nikolas Zolas. 2024. "AI adoption in America: who, what, and where." *Journal of Economics & Management Strategy*, 33(2): 375–415.

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

Oster, Emily. 2019. "Unobservable selection and coefficient stability: Theory and evidence." *Journal of Business and Economic Statistics*, 37(2): 187–204.

Paradis, Elise, Kate Grey, Quinn Madison, Daye Nam, Andrew Macvean, Vahid Meimand, Nan Zhang, Ben Ferrari-Church, and Satish Chandra. 2025. "How much does AI impact development speed? An enterprise-based randomized controlled trial."

Peng, Sida, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. "The impact of AI on developer productivity: evidence from GitHub Copilot." *Working Paper*.

Revelio Labs. 2026. "Revelio Labs Data Dictionary." Documentation: `https://www.data-dictionary.reveliolabs.com/`.

Romer, Paul. 1990. "Endogenous technological change." *Journal of Political Economy*, 98(5): S71–S102.

S&P Global Market Intelligence. 2025. "Compustat North America: Data Guide." Documentation.

Trammell, Philip, and Anton Korinek. 2023. “Economic growth under transformative AI.” *NBER Working Paper.*

Webb, Michael. 2020. “The impact of artificial intelligence on the labor market.” *Working Paper.*

Supplemental Appendix

A Appendix Tables

Table A.1: Twenty largest holdings in the THNQ ETF

Rank	Company	Portfolio Weight (%)
1	Lumentum Holdings (LITE)	5.05
2	Palo Alto Networks (PANW)	3.95
3	Lam Research (LRCX)	3.12
4	Nebius Group (NBIS)	2.97
5	TSMC (2330.TW)	2.97
6	Teradyne (TER)	2.84
7	ASML Holding (ASML)	2.79
8	Analog Devices (ADI)	2.65
9	Cloudflare (NET)	2.64
10	Cognex (CGNX)	2.57
11	MediaTek (2454.TW)	2.50
12	Alphabet (GOOGL)	2.44
13	Global Unichip (3443.TW)	2.38
14	Arista Networks (ANET)	2.27
15	NVIDIA (NVDA)	2.25
16	Infineon Technologies (IFX)	2.25
17	Advanced Micro Devices (AMD)	2.24
18	Meta Platforms (META)	2.21
19	Amazon.com (AMZN)	2.21
20	Pure Storage (PSTG)	2.16
Top 20 Total		54.46

Notes: This table reports the twenty largest holdings in the THNQ ETF by portfolio weight as of March 21, 2026. The fund held 51 securities in total on this date. Source: ROBO Global ETFs (https://www.roboglobaletfs.com/thnq).

Table A.2: Twenty largest holdings in the AIQ ETF

Rank	Company	Portfolio Weight (%)
1	SK hynix (000660.KRX)	4.34
2	Samsung Electronics (005930.KRX)	4.26
3	Netflix (NFLX)	3.65
4	Micron Technology (MU)	3.54
5	Cisco Systems (CSCO)	3.47
6	TSMC (TSM)	3.34
7	Apple (AAPL)	3.32
8	Broadcom (AVGO)	3.16
9	NVIDIA (NVDA)	3.11
10	Meta Platforms (META)	3.05
11	Alphabet (GOOGL)	3.03
12	Palantir Technologies (PLTR)	3.02
13	Amazon.com (AMZN)	2.91
14	Microsoft (MSFT)	2.81
15	Tencent Holdings (0700.HK)	2.81
16	Oracle (ORCL)	2.79
17	Tesla (TSLA)	2.72
18	AMD (AMD)	2.64
19	IBM (IBM)	2.33
20	Alibaba Group (BABA)	2.30
Top 20 Total		62.60

Notes: This table reports the twenty largest holdings in the AIQ ETF by portfolio weight as of March 21, 2026. The fund held 85 securities in total on this date. Source: Global X ETFs (https://www.globalxetfs.com/funds/aiq).

Table A.3: O*NET codes in Revelio's "Software Engineer" category

O*NET Code	Occupation Title	Compensation Share (%)
15-1252.00	Software Developers	33.50
15-1299.08	Computer Systems Engineers/Architects	10.66
15-1254.00	Web Developers	5.78
17-2199.06	Microsystems Engineers	5.06
15-1253.00	Software Quality Assurance Analysts and Testers	3.92
15-1255.00	Web and Digital Interface Designers	3.44
15-2051.00	Data Scientists	3.34
11-9041.00	Architectural and Engineering Managers	3.04
27-1029.00	Designers, All Other	2.81
15-1243.00	Database Architects	2.49
15-1241.00	Computer Network Architects	2.37
15-1299.09	Information Technology Project Managers	2.21
15-1299.07	Blockchain Engineers	1.70
15-1221.00	Computer and Information Research Scientists	1.62
15-1251.00	Computer Programmers	1.43
41-9031.00	Sales Engineers	1.20
15-1299.05	Information Security Engineers	1.12
27-1014.00	Special Effects Artists and Animators	1.00
27-1011.00	Art Directors	0.91
17-2072.01	Radio Frequency Identification Device Specialists	0.89
Top 20 Total		87.39

Notes: This table lists the twenty largest O*NET occupation codes, ranked by compensation share, among the 125 codes that Revelio Labs classifies under the "Software Engineer" role category. The compensation share is each occupation's percentage of the total weighted compensation summed across all workers in all 125 software engineer occupation codes in 2021, where the weighting uses Revelio's individual-level sampling weights. These twenty occupations collectively account for 87.4% of total software engineer compensation. Source: Revelio Labs (Revelio Labs, 2026).

Table A.4: GPT-5.5-classified semiconductor-related firms

Rank	firm	Market Cap (2021)
Panel A: NAICS6 composition of GPT-5.5-classified semiconductor-related firms		
NAICS		Count of firms
334413: Semiconductor and Related Device Manufacturing		46
333242: Semiconductor Machinery Manufacturing		9
334515: Instrument Manufacturing for Measuring and Testing Electricity and Electrical Signals		3
513210: Software Publishers		3
325120: Industrial Gas Manufacturing		2
334513: Instruments and Related Products Manufacturing for Measuring, Displaying, and Controlling Industrial Process Variables		2
331410: Nonferrous Metal (except Aluminum) Smelting and Refining		1
333310: Commercial and Service Industry Machinery Manufacturing		1
333314: Missing / Unknown		1
334418: Printed Circuit Assembly (Electronic Assembly) Manufacturing		1
334511: Search, Detection, Navigation, Guidance, Aeronautical, and Nautical System and Instrument Manufacturing		1
Missing: Missing / Unknown		1
Panel B: 25 largest GPT-5.5-classified semiconductor-related firms		
Rank	firm	Market Cap (2021)
1	LINDE PLC	$177.6 B
2	TEXAS INSTRUMENTS INC	$174.1 B
3	APPLIED MATERIALS INC	$139.5 B
4	AIR PRODUCTS & CHEMICALS INC	$67.4 B
5	KLA CORP	$64.8 B
6	MICROCHIP TECHNOLOGY INC	$48.4 B
7	GLOBALFOUNDRIES INC	$34.6 B

Continued on next page

Table A.4: GPT-5.5-classified semiconductor-related firms (continued)

Rank	firm	Market Cap (2021)
8	ON SEMICONDUCTOR CORP	$29.4 B
9	SKYWORKS SOLUTIONS INC	$25.5 B
10	MONOLITHIC POWER SYSTEMS INC	$22.8 B
11	ENTEGRIS INC	$18.8 B
12	QORVO INC	$17.0 B
13	WOLFSPEED INC	$13.0 B
14	SYNAPTICS INC	$11.4 B
15	LATTICE SEMICONDUCTOR CORP	$10.6 B
16	MKS INC	$9.7 B
17	IPG PHOTONICS CORP	$9.1 B
18	SILICON LABORATORIES INC	$7.9 B
19	UNIVERSAL DISPLAY CORP	$7.9 B
20	SITIME CORP	$6.1 B
21	AMKOR TECHNOLOGY INC	$6.1 B
22	MAXLINEAR INC	$5.8 B
23	SEMTECH CORP	$5.7 B
24	POWER INTEGRATIONS INC	$5.6 B
25	MACOM TECHNLGY SOL HLDGS INC	$5.5 B

Notes: Panel A reports the NAICS6 composition of firms that GPT 5.5 classified as semiconductor-related. Panel B reports the 25 largest GPT-5.5-classified semiconductor-related firms ranked by end-of-2021 market capitalization. Panel A reports NAICS entries as code followed by description. Market capitalization is reported in billions of U.S. dollars.

Statistic	Value
N	1798
Mean	-0.019
SD	0.520
Min	-2.975
P10	-0.510
P25	-0.304
Median	-0.075
P75	0.202
P90	0.589
Max	2.920
Share > 0	0.420

Table A.5: Summary statistics for baseline first-stage firm betas on THNQ excess returns. The sample uses the balanced 2022–2025 panel and the baseline FF3 specification.

Note: Betas are firm-level coefficients on weekly THNQ excess returns from the baseline FF3 first-stage specification. Each observation is a unique firm (gvkey). *Share > 0* is the fraction of firms with a positive beta.

Table A.6: Software engineer payroll share rankings within NAICS2 sectors

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
11: Agriculture, Forestry, Fishing and Hunting					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	CORTEVA INC	$34.5 B	1	TEJON RANCH CO	$0.5 B
2	ALICO INC	$0.3 B	2	LIMONEIRA CO	$0.3 B
3	VILLAGE FARMS INTL INC	$0.6 B	3	ORIGIN AGRITECH LTD	$0.0 B
4	LOCAL BOUNTI CORP	$0.6 B	4	SADOT GROUP INC	$0.0 B
5	CAL-MAINE FOODS INC	$1.6 B			
21: Mining, Quarrying, and Oil and Gas Extraction					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	RPC INC	$1.0 B	1	HIGHPEAK ENERGY INC	$1.4 B
2	ALLIANCE RESOURCE PTNRS -LP	$1.6 B	2	ENERGY FUELS INC	$1.2 B
3	EOG RESOURCES INC	$52.0 B	3	NABORS INDUSTRIES LTD	$0.7 B
4	REALLOYS INC	$0.0 B	4	GOLD ROYALTY CORP	$0.7 B
5	SLB LTD	$42.0 B	5	RAMACO RESOURCES INC	$0.5 B
22: Utilities					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)

Continued on next page

Table A.6: Software engineer payroll share rankings within NAICS2 sectors (continued)

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	VISTRA CORP	$11.0 B	1	AMERICAN STATES WATER CO	$3.8 B
2	AMEREN CORP	$22.9 B	2	CLEARWAY ENERGY INC	$2.9 B
3	EVERGY INC	$15.7 B	3	KENON HOLDINGS LTD	$2.8 B
4	NEXTERA ENERGY INC	$183.2 B	4	H2O AMERICA	$2.2 B
5	AMERICAN ELECTRIC POWER CO	$44.9 B	5	CADIZ INC	$0.2 B

23: Construction

Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	DYCOM INDUSTRIES INC	$2.8 B	1	CONSTRUCTION PRTNR INC	$1.2 B
2	NVR INC	$20.6 B	2	FORESTAR GROUP INC	$1.1 B
3	ALSET INC	$0.0 B	3	HOVNANIAN ENTRPRS INC -CL A	$0.7 B
4	API GROUP CORP	$5.8 B			
5	M/I HOMES INC	$1.8 B			

31-33: Manufacturing

Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	INNOVIZ TECHNOLOGIES LTD	$0.9 B	1	SONOCO PRODUCTS CO	$5.7 B
2	UBIQUITI INC	$19.2 B	2	CELSIUS HOLDINGS INC	$5.6 B
3	CIMPRESS PLC	$1.9 B	3	NATIONAL BEVERAGE CORP	$4.2 B
4	ENERGOUS CORP	$0.1 B	4	INTERPARFUMS INC	$3.4 B
5	CIENA CORP	$11.9 B	5	LATHAM GROUP INC	$3.0 B

Continued on next page

Table A.6: Software engineer payroll share rankings within NAICS2 sectors (continued)

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
42: Wholesale Trade					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	RESIDEO TECHNOLOGIES	$3.8 B	1	PLAINS GP HOLDINGS LP	$2.0 B
2	EPLUS INC	$1.5 B	2	HF FOODS GROUP INC	$0.5 B
3	CDW CORP	$27.6 B	3	EVI INDUSTRIES INC	$0.4 B
4	INSIGHT ENTERPRISES INC	$3.7 B	4	G WILLI-FOOD INTERNATIONAL	$0.3 B
5	COPART INC	$36.0 B	5	MARTIN MIDSTREAM PARTNERS LP	$0.1 B
44-45: Retail Trade					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	JUMIA TECHNOLOGIES AG	$0.9 B	1	O'REILLY AUTOMOTIVE INC	$47.3 B
2	PDD HOLDINGS INC	$25.8 B	2	ACADEMY SPO AND OU INC	$3.9 B
3	MOGU INC -ADR	$0.0 B	3	ENVELA CORP	$0.1 B
4	EBAY INC	$39.5 B	4	NAAS TECHNOLOGY INC	$0.0 B
5	WAYFAIR INC	$14.8 B			
48-49: Transportation and Warehousing					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	FULL TRUCK ALLIANCE CO LTD	$0.7 B	1	WESTERN MIDSTRM PRTNRS LP	$9.1 B
2	GRAB HOLDINGS LIMITED	$25.8 B	2	FLEX LNG LTD	$1.3 B

Continued on next page

Table A.6: Software engineer payroll share rankings within NAICS2 sectors (continued)

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
3	LYFT INC	$14.4 B	3	KINETIK HOLDINGS INC	$1.2 B
4	HUNT (JB) TRANSPRT SVCS INC	$21.5 B	4	SCORPIO TANKERS INC	$0.7 B
5	KNIGHT-SWIFT TRPTN HLDGS INC	$10.1 B	5	GENCO SHIPPING & TRADING	$0.7 B
56: Administrative and Support and Waste Management and Remediation Services					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	MASTECH DIGITAL INC	$0.2 B	1	AMN HEALTHCARE SERVICES INC	$5.8 B
2	EXPEDIA GROUP INC	$27.1 B	2	AQUA METALS INC	$0.1 B
3	QUHUO LIMITED	$0.0 B	3	GREENPRO CAPITAL CORP	$0.0 B
4	UPWORK INC	$4.4 B	4	AVALON HOLDINGS CORP	$0.0 B
5	FIVERR INTERNATIONAL LTD	$4.2 B			
61: Educational Services					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	DUOLINGO INC	$1.8 B			
2	COURSERA INC	$3.4 B			
3	YOUDAO INC -ADS	$0.1 B			
4	VISIONSYS AI INC	$0.0 B			
5	ATA CREATIVITY GLOBAL -ADS	$0.0 B			
62: Health Care and Social Assistance					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		

Continued on next page

Table A.6: Software engineer payroll share rankings within NAICS2 sectors (continued)

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	DARIOHEALTH CORP	$0.2 B	1	AIRSCULPT TECHNOLO INC	$1.0 B
2	TELADOC HEALTH INC	$14.7 B	2	AMERICAN SHARED HSPTL SERV	$0.0 B
3	GUARDANT HEALTH INC	$10.2 B			
4	NATERA INC	$8.9 B			
5	HUMANA INC	$59.6 B			
71: Arts, Entertainment, and Recreation					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	DRAFTKINGS INC	$11.2 B			
2	RUSH STREET INTERACTIVE INC	$1.0 B			
3	BRIGHTSTAR LOTTERY PLC	$5.9 B			
4	CHURCHILL DOWNS INC	$9.2 B			
5	XWELL INC	$0.2 B			
72: Accommodation and Food Services					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	AIRBNB INC	$60.7 B	1	WINGSTOP INC	$5.2 B
2	YUM BRANDS INC	$40.7 B	2	CANNAE HOLDINGS INC	$3.1 B
3	YUM CHINA HOLDINGS INC	$21.3 B	3	FIRST WATCH RESTAURA GRO INC	$1.0 B
4	MONARCH CASINO & RESORT INC	$1.4 B	4	KURA SUSHI USA INC	$0.7 B

Continued on next page

Table A.6: Software engineer payroll share rankings within NAICS2 sectors (continued)

Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
5	LAS VEGAS SANDS CORP	$28.8 B	5	ONE GP HOSPITALITY (THE)	$0.4 B
81: Other Services (except Public Administration)					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	FRONTDOOR INC	$3.1 B			
2	REGIS CORP/MN	$0.1 B			
3	CINTAS CORP	$46.0 B			
4	SERVICE CORP INTERNATIONAL	$11.7 B			
5	EUROPEAN WAX CENTER INC	$1.1 B			
99					
Top 5 by software engineer payroll share			*Largest 5 zero-software-engineer firms*		
Rank	Firm	Market Cap (2021)	Rank	Firm	Market Cap (2021)
1	HONEYWELL INTERNATIONAL INC	$134.8 B	1	ICAHN ENTERPRISES LP	$14.5 B
2	3M CO	$86.5 B	2	HYPERSCALE DATA INC	$0.1 B
3	BERKSHIRE HATHAWAY	$389.7 B			
4	SEABOARD CORP	$4.6 B			
5	ICAHN ENTERPRISES LP	$14.5 B			

Notes: Within each NAICS2 sector, the left panel reports the 5 firms with the highest software engineer payroll shares. The right panel reports the 5 largest firms with zero software engineer payroll share in Revelio. Market capitalization is reported in billions of U.S. dollars.

Table A.7: Sensitivity of the baseline coefficient to outlier treatment

Dependent Variable:	Beta on THNQ excess return				
Specification	Baseline (trim top 1%)	No outlier handling	Trim top 5%	Winsorize top 1%	Winsorize top 5%
Model:	(1)	(2)	(3)	(4)	(5)
Variables					
Software engineer payroll share	1.52***	1.53***	1.91***	1.59***	2.03***
	(0.26)	(0.23)	(0.38)	(0.25)	(0.30)
Fixed-effects					
NAICS3	Yes	Yes	Yes	Yes	Yes
Quintile of average 2017-2021 assets	Yes	Yes	Yes	Yes	Yes
Quintile of average 2017-2021 sales	Yes	Yes	Yes	Yes	Yes
Fit statistics					
Observations	1,773	1,791	1,703	1,791	1,791
R^2	0.15	0.16	0.14	0.16	0.16
Within R^2	0.02	0.03	0.02	0.03	0.03

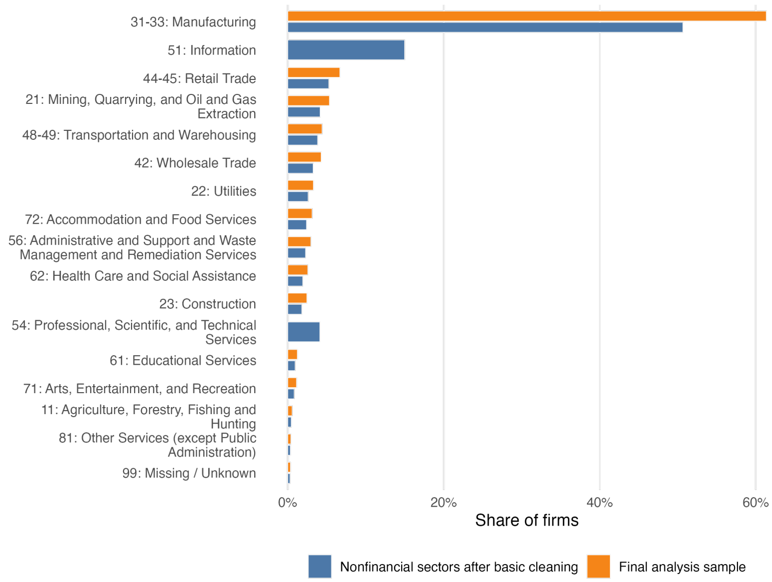
HC3 standard-errors in parentheses

*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

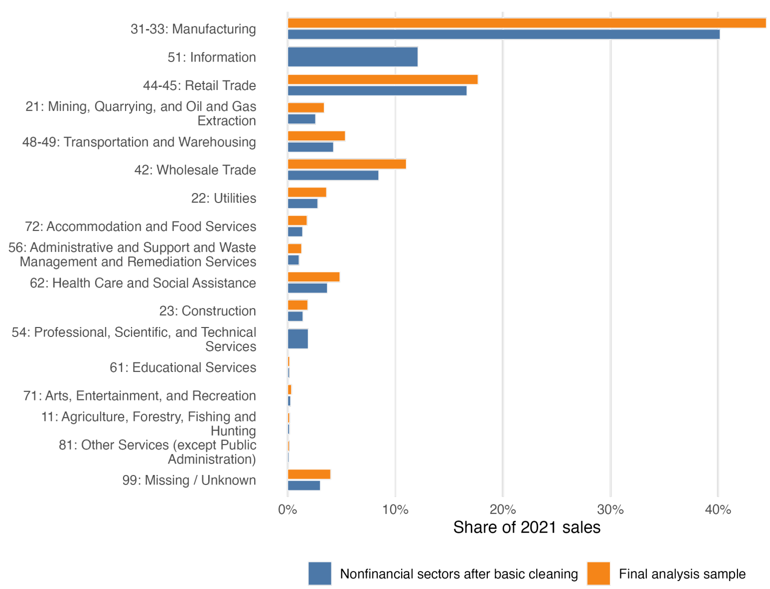
Notes: This table starts from the baseline NAICS3 fixed-effects specification and varies only the treatment of the software engineer payroll-share regressor. Column (1) is identical to Column (4) of Table 4; Column (2) does not trim the software engineer payroll share at all; Column (3) trims at the 95th percentile; and Columns (4) and (5) winsorize at the 99th and 95th percentiles, respectively. All other controls, fixed effects, and the dependent variable are held fixed across columns. HC3 standard errors are reported in parentheses. Observations are firms.

B Appendix Figures

Figure B.1: Sectoral composition of the analysis sample



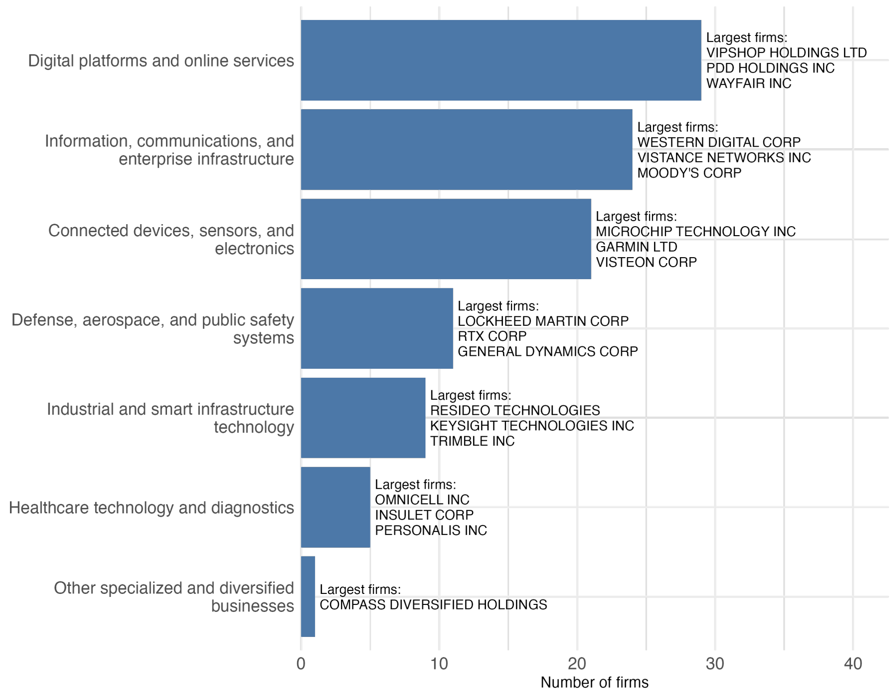
(a) Share of firms



(b) Share of 2021 sales

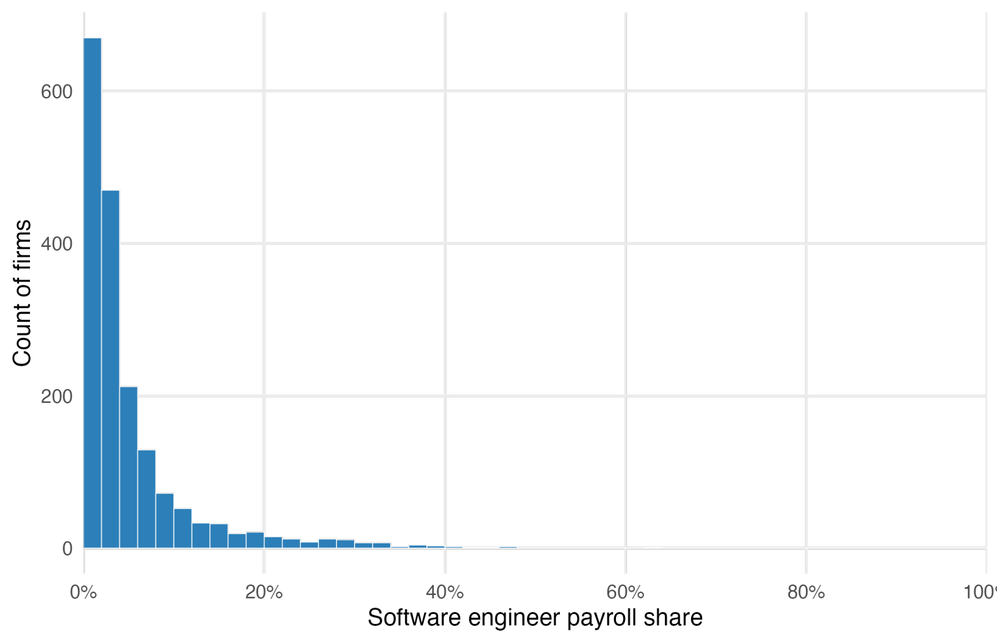
Notes: Panel (a) shows the share of firms in each NAICS2 sector; panel (b) shows the share of aggregate 2021 sales. The light bars show the distribution after dropping NAICS sectors 51, 52, 53, and 54. The dark bars show the distribution in the final analysis sample after all remaining filters (semiconductor exclusion, Revelio match, balanced panel, and size trim).

Figure B.2: Business models of firms with high software engineering payroll shares



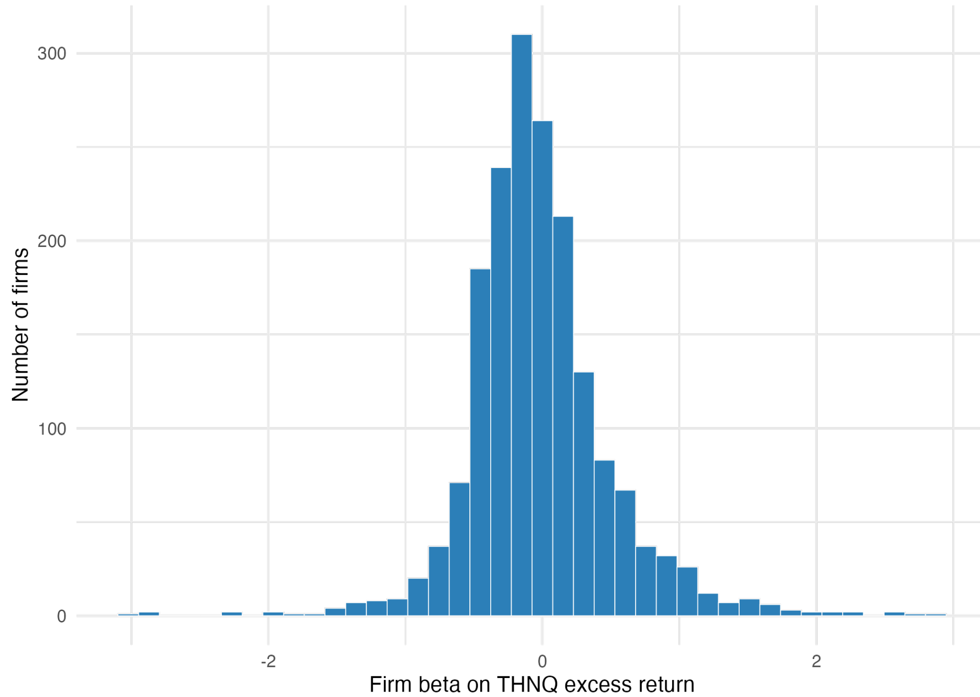
Notes: This figure shows the high-level business models of the top 100 firms in our dataset ranked by their software engineer payroll shares, as classified by GPT 5.4 based on the `busdesc` field in Compustat.

Figure B.3: Distribution of software engineer payroll shares



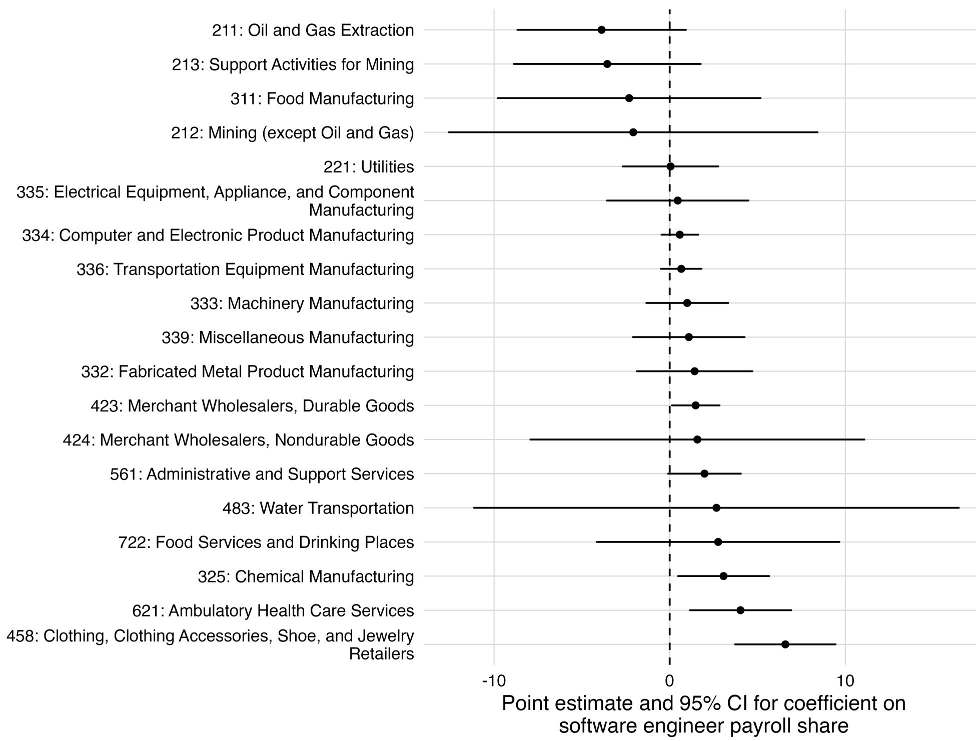
Notes: This figure shows the histogram of the software engineer payroll share across firms in the final analysis sample, measured in 2021. The payroll share is the fraction of total firm compensation attributable to workers classified under Revelio's "Software Engineer" role category. The distribution has a mass point at zero (196 firms) and a long right tail.

Figure B.4: Distribution of firm-level AI betas (baseline specification)



Notes: This figure shows the histogram of estimated firm-level AI betas from the first-stage regression of weekly firm excess returns on THNQ excess returns, controlling for the Fama-French three factors (MKT, SMB, HML), over the balanced 2022–2025 panel. Each observation is a unique firm.

Figure B.5: Second-stage regression results: industry heterogeneity



Notes: This figure shows point estimates and 95% confidence intervals from second-stage regressions run one NAICS3 sector at a time. The outcome is the firm-level beta on THNQ excess returns, and the main regressor is the software engineer payroll share, trimmed at the 99th percentile as in Table 4. We drop NAICS3 sectors with fewer than 25 firms. To avoid singleton fixed-effect cells in these smaller samples, we replace fixed effects for quintiles of assets and sales with continuous controls for log assets and log sales. Confidence intervals are based on HC3 standard errors.

C Data Appendix

C.1 Testing Assumption 1, Part 2

In this section, we explicitly test the second condition of Assumption 1: $Cov_{f,j}(\omega'_j, \gamma_j) = 0$, where ω'_j is the elasticity of a firm's final-demand weight with respect to a demand shifter, γ_j is its software engineering payroll share, and f_j is its final expenditure weight. Conceptually, we are asking whether aggregate shocks could be reallocating demand towards software-engineering-intensive firms in a final-expenditure-weighted sense. If this were the case, then our estimate of the cross-sectional slope δ could be contaminated by movements in firm values that were caused by these shifts in demand, not by news about software engineers' productivity. As our empirical proxy for ω'_j, we use a measure of firms' exposure to Generative AI through product market channels from Eisfeldt et al. (2026). Specifically, they build an indicator called *GPT10K Product Exposure* by prompting GPT 3.5 Turbo with text from the business description section of a firm's 10-K report and asking it whether the firm could become a short-term beneficiary of improvements in Generative AI by adding AI features to its products. This captures the extent to which consumers' demand for a firm's output could move in response to a change in AI capabilities, which is exactly what ω'_j is meant to capture. For clarity, we show in Figure C.1 the full prompt used by Eisfeldt et al. (2026) to construct this indicator; their Internet Appendix contains further details.

We test the hypothesis $\text{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$ with a wild bootstrap clustered at the level of NAICS2 sectors. We cluster in order to allow residual dependence between software engineer payroll shares and product-market exposure to Generative AI within a sector, and we choose a wild bootstrap because of the small number of clusters. Unlike in our baseline empirical analysis in Section 3, we include both members of our chosen AI ETFs and other firms in NAICS sectors 51 and 54, since we are concerned with economy-wide effects of AI innovations on firms' final-demand weights. To implement the test, we first compute the empirical covariance $\hat{C}_f \equiv \widehat{\text{Cov}}_{f,j}(\omega'_j, \gamma_j)$. In bootstrap iteration r, we draw a multiplier v^r_s for each NAICS2 sector s, then construct a simulated software engineer payroll share for each firm j under the null: $\gamma^r_j \equiv \gamma + v^r_{s(j)}\hat{u}_j$, where γ is the final-expenditure-weighted mean of γ_j and $\hat{u}_j = \gamma_j - \gamma$. Then, we compute $\hat{C}^r_f \equiv \widehat{\text{Cov}}_{f,j}(\omega'_j, \gamma^r_j)$ for each iteration, and we label the standard deviation of this object across bootstrap iterations as $\widehat{\text{se}}(\hat{C}_f)$. Finally, we form our test statistic as $\hat{C}_f / \widehat{\text{se}}(\hat{C}_f)$ and compute a p-value based on a normal approximation using our bootstrap standard error. We report results for the test in Table C.1.

Figure C.1: Full prompt for building *GPT10K Product Exposure* measure from Eisfeldt et al. (2026)

```
systemprompt = "You are a financial analyst who is an expert in evaluating company filings for their business
implications. \
    You are evaluating whether a given company's business description in their annual report suggests that the
company's products will be in higher demand as a result of a recent boom in generative artificial intelligence
applications.\
     You are only interested in companies whose product sales might be directly impacted by the Generative AI
boom, not companies that might benefit from labor productivity gains due to incorporating Generative AI into
their processes.\
Among the companies that are likely to see their product market positively affected by the Generative AI boom,
ALL of the following might be reasons why they have positive direct exposure:\
    1) Empowered users. Technology companies leveraging Generative AI technology to amplify their product
functionality  will be directly positively exposed.\n \
    2) Enablers: Select makers of semiconductors and related equipment or other infrastructure needed to build
or deploy Generative AI technology will be directly positively exposed.\n \
    3) Hyperscalers:  Large companies using their extensive cloud computing infrastructures to commercialize
Generative AI on a large scale will be directly positively exposed. \n \
    You are reasoning through how a company's product as described in the business description section of the
annual report might be affected by a boom in demand for Generative artificial intelligence applications, taking
into account the possible impact channels just described.\
    Then,  you use that reasoning to decide if a company is: \n \
        P1: directly positively affected in its product market, which requires that it either supplies products
(e.g. equipment, chips, infrastructure) that will be needed to support the Generative AI boom, or has immediate
benefits in terms of adding AI capabilities into its product, e.g. because of access to proprietary data, or
providing solutions to user problems for which Generative AI is highly complementary. The company should fall
into one of the categories of Empowered User, Enabler, or Hyperscaler to get a P1 exposure score. \n \
        P0: no direct positive impact in its product market, as it does not sell inputs into the Generative AI
activities of other firms and does not have immediate ability to incorporate new AI capabilities into its
products. This includes more remote indirect impacts from generative AI leading to broader increases in
innovation or demand for the industry but not necessarily for this company in particular. \n \
         Note that you should assign P1 only when there is immediate product market impact (e.g. the company
product involves to a large degree chatbots or text generation that Generative AI can easily improve, or
supplies key inputs), but should assign P0 (not exposed) when the AI impact on the firm's product capabilities
is more remote, or only somewhat probable, such as requiring future AI-based tools to be built by other
companies that might then complement the firm's product. Also note that companies that provide critical inputs
into Generative AI usage, such as infrastructure or hardware can also be directly affected and should get a
score of P1, as those components are essential for using AI and will see higher demand in a boom."
```

Prompt preceding the "Business" text submission":

```
prefix = "Read the following business description from a company's annual report. Then do two things. \
1: Reason step by step to apply a score of P0 (not directly positively exposed) or P1 (directly positively
exposed) to rate its product market exposure. Use the rubric and generative AI product market exposure
categories provided to you before (considering whether the company might be an Empowered User, Enabler, or
Hyperscaler, all of which positively exposed) to decide whether the company's product market is going to be
positively affected by the boom in Generative artifical intelligence. \
Report the label that you think fits best and provide up to two sentences that summarize your reasoning for the
product market exposure score that you assigned. Do not say N/A.\n \
    2: Report only the product market exposure score that you determined for the company, which should match
the score in step 1. Do not reply N/A. \n\
   The business description follows:\n  "
```

Overall prompt structure (here, *fulltext* is the "Business" description to be scored):

```
prompt = [{"role": "system", "content": systemprompt},
      {"role": "user", "content": user_prompts[0]},
        {"role": "assistant", "content": assistant_prompts[0]},
       {"role": "user", "content": user_prompts[1]},
       {"role": "assistant", "content": assistant_prompts[1]}]
```

Table C.1: Wild cluster bootstrap results for testing part 2 of Assumption 1

	Estimate
$\mathrm{Cov}_{f,j}(\omega'_j, \gamma_j)$	0.0163
Bootstrap s.e.	0.0115
95% confidence interval	[-0.0061, 0.0388]
p-value	0.154
Final-expenditure-weighted mean of ω'_j	0.0937
Final-expenditure-weighted mean of γ_j	0.1122
Number of firms	1,299
Number of NAICS2 clusters	17

Notes: The test uses 9,999 bootstrap draws. We use Rademacher weights (−1 or 1 with equal probability) for scaling the residuals in the wild bootstrap procedure.

With a p-value of 0.154, the bootstrap fails to reject the null that $\mathrm{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$. We therefore conclude that potential correlation between firms' software engineer payroll shares and their exposure to Generative AI through demand channels is not a major threat to identification.

C.2 Oster (2019) Coefficient Stability Test

In Section 3.3, we showed that our second-stage estimate of the cross-sectional slope δ is robust to adding a wide variety of controls, as illustrated by Table 5. Following the logic of Oster (2019), if we believe that selection on unobservables would move our regression coefficient in the same direction as selection on observables, then we can use this observation to argue that selection on unobservables is unlikely to fundamentally change our results. We formalize that idea in this section by implementing the test described by Oster (2019). The concept is straightforward: if we know how much our coefficient of interest changes when we add a variety of observable controls to the regression, and we believe that selection on observables is somewhat informative about selection on unobservables, then we can directly scale the difference between the coefficients from the "baseline" and "controlled" regressions to back out the implied coefficient that would correspond to any possible magnitude of selection on unobservables. The test is only useful if the observable controls have explanatory power, in terms of increasing the R^2 of the regression relative to the baseline specification, because otherwise they would offer no useful information about either kind of selection.

We take as our baseline regression the specification from column (4) of Table 4, which already includes NAICS3 fixed effects as well as fixed effects for quintiles of firm assets and sales. For the "controlled" specification that will be informative about the extent of selection on observables, we additionally control for the Generative AI task exposure measure from column (5) of Table 4, as well as the following variables from Table 5: log(market cap) from row 8, the four financial ratios from row 11, HQ state FE from row 12, the Generative AI product-market exposure measures in row 16, and the payroll shares of four key non-software-engineer occupations in row 17. To ensure an apples-to-apples comparison, we run both the baseline and controlled specifications on the subsample of firms for which none of the extra controls are missing. The results are shown in Table C.2; because the baseline regression already includes multiple fixed effects, we run versions of the test using both the overall R^2 and within R^2.

Table C.2: Results of Oster (2019) test for second-stage regression

Panel A. Regression results for baseline and controlled specifications

R^2 type	δ (initial)	δ (augmented)	R^2 (initial)	R^2 (augmented)	ΔR^2	N (initial)	N (augmented)
Overall	1.58	2.07	0.26	0.40	0.14	759	751
Within	1.58	2.07	0.05	0.15	0.10	759	751

Panel B. Sensitivity of Oster-implied coefficient

R^2 type	Assumed Oster's δ	$R_{\max} = 1$	$R_{\max} = 1.3R_1$	$R_{\max} = 1.5R_1$
Overall	0.50	3.13	2.28	2.42
	1.00	4.19	2.49	2.76
	2.00	6.31	2.90	3.46
Within	0.50	4.12	2.18	2.25
	1.00	6.17	2.28	2.43
	2.00	10.27	2.50	2.79

Notes: Panel A shows results for the "baseline" regressions from column (4) of Table 4 and for the "controlled" specification described above. Panel B plugs these coefficients into the Oster test's formula for the adjusted δ under various assumptions about two parameters: (1) the maximum possible R^2 from including all possible covariates, R_{max}, and (2) the degree of selection on unobservables relative to selection on observables, referred to as "Oster's δ" (to distinguish it from our second-stage cross-sectional slope, which is also labeled δ).
Each comparison uses the same estimation sample in the baseline and augmented specifications; the final sample sizes are different only because there are slightly more singleton FE observations in the augmented specification. The table reports results for both the overall R^2 and within R^2. Because the augmented specification includes controls for task-based and product-market-based exposure to Generative AI from Eisfeldt et al. (2026), which are missing for many firms, the sample size is smaller than in Table 4.

Panel A of Table C.2 shows the regression results from the "baseline" and "extra controls" specifications, and Panel B gives the implied second-stage coefficient based on various assumptions about the test's two parameters. The rows correspond to the degree of selection on unobservables relative to selection on observables, and the columns correspond to the maximum achievable R^2 from including all possible covariates in the regression, as a function of the "controlled" regression's R^2 (referred to as R_1). The entry at the center of the table corresponds to the parameter values suggested by Oster (2019). In this benchmark, selection on unobservables is believed to have the same magnitude as selection on observables, and the maximum achievable R^2 is 1.3x the R^2 from the "controlled" specification.

Our results clearly pass the Oster test: adding extra controls to our second-stage regression actually increases our estimate of δ. This is the opposite of what we would expect if we believed that failing to include omitted variables was causing us to arrive at a spurious positive estimate $\hat{\delta}$ instead of a true zero δ. So, in order to drive the true δ to zero relative to our baseline estimates that lie roughly in the range of 1.3 – 1.6, selection on unobservables would have to work in the opposite direction from selection on observables.

C.3 Harte-Hanks Software Budget Data

As described in Section 4.2, we use data on firm-level software budgets from the marketing company Harte-Hanks to estimate firms' total spending on inputs that correspond to the model's software composite object. These numbers are based on establishment-level surveys asking about the IT products that firms have installed, such as laptops, servers, and software, and they involve some imputation in addition to the raw survey results. The data cover 2010 and 2012-2016, but we use only 2012-2015 because these are the years with the most stable coverage of IT budgets.

We match the Harte-Hanks dataset to Compustat based on cleaned firm names, then link firms to individual "sites" in the Harte-Hanks data. Our baseline measure of firms' software spending is based on the average software budget of the site labeled as the firm's "Ultimate HQ" in Harte-Hanks, following the approach of De Ridder (2024), averaged over 2012 through 2015. Although this means we will necessarily miss each firm's total software expenditure that is occurring at non-HQ establishments, we choose the HQ-only measure in order to limit measurement error coming from variation in coverage of a firm's sites over time, since the number of sites surveyed for a particular firm can fluctuate meaningfully from year to year.

To validate the scale of the Harte-Hanks data, we first benchmark total private revenue reported in Harte-Hanks against private gross output from the Bureau of Economic Analysis (BEA). The results are shown in Table C.3: in terms of total revenue of the establishments surveyed, the Harte-Hanks dataset has reasonably good coverage of the universe of US private businesses.

Table C.3: Harte-Hanks Site-Level Revenue vs. BEA Gross Output

Year	H-H Revenue ($bn)	BEA Gross Output ($bn)	Ratio
2012	18,989.8	24,319.8	0.78
2013	19,098.7	25,368.1	0.75
2014	26,741.1	26,670.3	1.00
2015	25,334.8	26,984.8	0.94

Notes: Private Harte-Hanks revenue excludes establishments tagged as government-related, such as military bases and schools. Private BEA gross output excludes the federal government, state and local government, and housing sectors.

We now compare total private software spending in Harte-Hanks to a measure of software investment from the BEA. Specifically, we consider private investment in prepackaged and custom software (fields `ENS1` and `ENS2`) from the BEA's fixed asset tables. We do not include `ENS3` (own-account software) because this would likely not be captured by Harte-Hanks's surveys, which are focused on purchases of software rather than in-house spending on the development of new software. The comparison is shown in Table C.4.

Table C.4: Harte-Hanks Software Budgets vs. BEA Software Investment

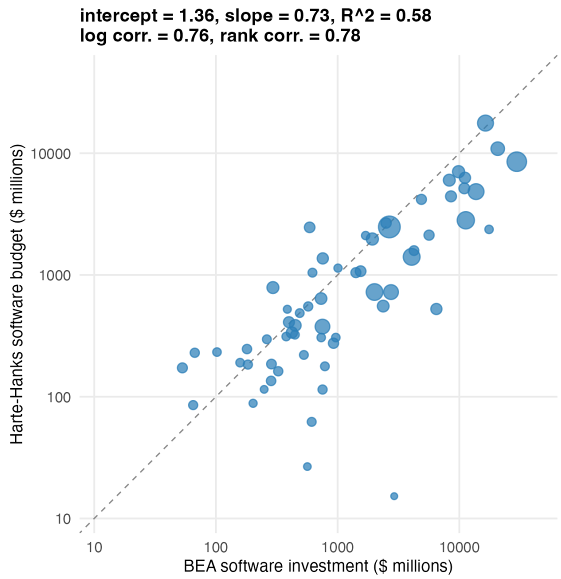
Year	H-H Software Budget ($bn)	BEA Software Investment ($bn)	Ratio
2012	114.4	221.8	0.52
2013	131.1	233.5	0.56
2014	127.5	248.0	0.51
2015	152.9	257.3	0.59

Notes: Private BEA software investment is defined as the sum of prepackaged and custom software investment, over all sectors excluding federal government, state and local government, and housing.

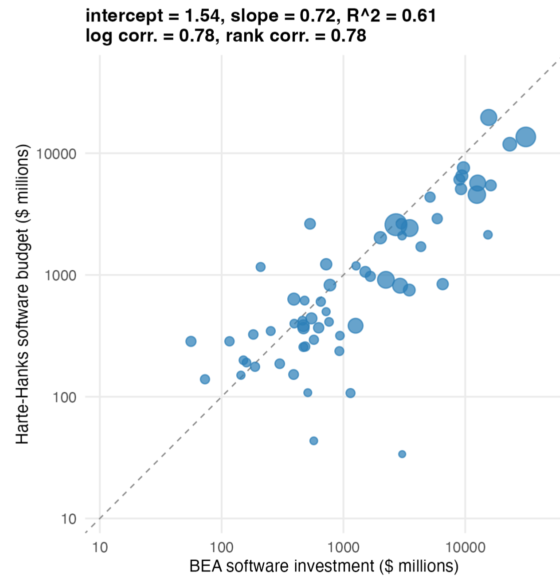
Although we are unable to perfectly match total software investment in the national accounts, it is still the case that the Harte-Hanks data cover a significant portion of software investment by US private businesses. It is worth noting that because Harte-Hanks covers somewhat less than 100% of private gross output over 2012-2015, as seen in Table C.3, its coverage of true software budgets among the subset of firms that were surveyed is likely slightly better than the 50-60% figures shown in Table C.4, where the denominator is based on the full universe of firms.

We also present cross-industry comparisons in Figure C.2. In each year of data that we use (2012-2015), the cross-industry correlation between log(Harte-Hanks software budget) and log(BEA software investment) is above 0.7, as is the corresponding rank correlation. This is further evidence that Harte-Hanks's surveys are picking up meaningful variation in software usage across firms.

Figure C.2: Cross-Industry Comparison of Harte-Hanks and BEA Software Spending

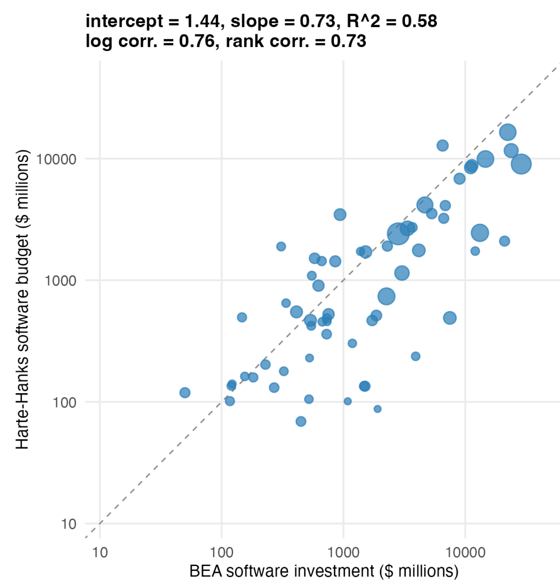


(a) 2012



(b) 2013

(c) 2014



(d) 2015

Notes: Each panel plots BEA software investment on the x-axis against Harte-Hanks software budgets on the y-axis, across private industries. BEA software investment is defined as the sum of prepackaged and custom software investment. Harte-Hanks software budgets are summed across all private sites in the industry. Both axes are in millions of dollars and use log scales. Panel titles report coefficients from a regression of log Harte-Hanks software budget on log BEA software investment, along with the log correlation and rank correlation. We match NAICS3 codes from the Harte-Hanks dataset to BEA industry codes in the fixed asset tables.

D Franchise Duration and Heterogeneous SDF Loading

This appendix microfounds the extended firm-value response (10) of Section 2.4 by endowing each firm with a finite franchise duration on the common stochastic discount factor. The benchmark of Section 2 is recovered exactly when every firm is an infinite perpetuity.

D.1 Setup

Each firm j is a single legal entity that holds a proprietary franchise over its good. If firm j is alive at the start of period t, it survives to the start of $t+1$ with probability $\varrho_j \in (0,1]$, independently across firms and independently of aggregate shocks. Upon death, the incumbent's equity is extinguished and a replacement entity with identical technology (ϕ_j, ϱ_j) is immediately installed in the same goods-market position, facing the same CES demand curve. The replacement's equity is endowed to the representative household at birth. The parameter ϱ_j is a structural primitive of firm j's franchise (a patent length, imitation lag, or charter renewal horizon), not an asset-pricing parameter. The benchmark of Section 2 corresponds to $\varrho_j = 1$.

For the discount-rate component, write the one-period discount factor as

$$\beta_t = \beta e^{v_t^{\Xi}}, \qquad \Xi_{t,t+1} = \beta e^{v_t^{\Xi}} \frac{u_c(c_{t+1})}{u_c(c_t)}.$$

The steady state has $v_t^{\Xi} = 0$, so the deterministic SDF is $\bar{\Xi}_{t,t+k} = \beta^k$.

Because the replacement firm operates the same technology and faces the same demand curve as the incumbent it succeeds, the cross-sectional distribution of active firms is time-invariant: at each date exactly one firm of each index j is alive and produces using technology ϕ_j. The static allocation, equilibrium factor prices (w_t, W_t), unit costs mc_{jt}, Domar weights λ_{jt}, and aggregate consumption c_t are identical to those of the benchmark economy. Mortality changes only the mapping from firm j's profits into equity claims on those profits, not the profits themselves. In particular, the period-profit response of Section 2 carries through verbatim.

The replacement at $t+1$ has the same technology, faces the same aggregate state, and is itself alive with probability ϱ_j going forward, so its equity at birth has the same conditional value as the incumbent would have had at $t+1$ had it survived. Firm-level death is therefore idiosyncratic with respect to the household's marginal utility, and the representative-household SDF $\Xi_{t,t+1}$ depends only on aggregate shocks. The same SDF prices every firm's equity. Firm j's equity, the claim on the operating profits of the current incarnation, satisfies the recursive pricing equation

$$V_{jt} = \pi_{jt} + \varrho_j \mathbb{E}_t[\Xi_{t,t+1} V_{j,t+1}],$$

together with the transversality condition $\lim_{k\to\infty} \mathbb{E}_t[(\beta\varrho_j)^k V_{j,t+k}] = 0$, which holds under $\beta\varrho_j < 1$.

Iterating forward and using the independence of survival from aggregate shocks,

$$V_{jt} = \mathbb{E}_t\Big[\sum_{k=0}^{\infty} \varrho_j^k \Xi_{t,t+k}\pi_{j,t+k}\Big]. \tag{D.1}$$

Setting $\varrho_j = 1$ recovers (2) exactly.

D.2 Firm-Value Response

Define the firm-specific news operator

$$\mathscr{X}_t^{(j)} \equiv (1-\tilde{\beta}_j)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty} \tilde{\beta}_j^k d\log X_{t+k}\Big], \qquad \tilde{\beta}_j \equiv \beta\varrho_j, \tag{D.2}$$

for any stochastic process X. For a single permanent innovation, $\mathscr{X}_t^{(j)}$ reduces to the contemporaneous value $d\log X_t$. When $\varrho_j = 1$, $\mathscr{X}_t^{(j)}$ reduces to the benchmark news operator $\mathscr{X}_t$.

Lemma 2 (Firm-Value Response with Franchise Duration)**.** *To first order, firm j's unexpected return is*

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S(\gamma_j - \gamma)\mathscr{S}_t^{(j)} + \mathscr{C}_t^{(j)} + \mathscr{D}_t^{(j)}, \tag{D.3}$$

where $\mathscr{S}_t^{(j)}$ and $\mathscr{C}_t^{(j)}$ are the firm-specific news in $\log S$ and $\log c$ defined via (D.2), and the firm-specific SDF news is

$$\mathscr{D}_t^{(j)} \equiv (1-\tilde{\beta}_j)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty} \tilde{\beta}_j^k d\log \Xi_{t,t+k}\Big].$$

Proof. The franchise-duration block leaves the static allocation unchanged, so the period-profit response of Section 2 applies horizon by horizon: $d\log\pi_{j,t+k} = \delta^S(\gamma_j - \gamma)d\log S_{t+k} + d\log c_{t+k}$. Let bars denote deterministic steady-state values. From (D.1), $\bar{\Xi}_{t,t+k} = \beta^k$ and $\bar{\pi}_{j,t+k} = \bar{\pi}_j$ give

$$\bar{V}_j = \sum_{k=0}^{\infty} (\beta\varrho_j)^k \bar{\pi}_j = \frac{\bar{\pi}_j}{1-\tilde{\beta}_j}.$$

First-order perturbation of (D.1) around this steady state yields, for each horizon k,

$$d(\Xi_{t,t+k}\pi_{j,t+k}) = \beta^k \bar{\pi}_j (d\log\Xi_{t,t+k} + d\log\pi_{j,t+k}),$$

so

$$dV_{jt} = \mathbb{E}_t\Big[\sum_{k=0}^{\infty} \varrho_j^k \beta^k \bar{\pi}_j (d\log\Xi_{t,t+k} + d\log\pi_{j,t+k})\Big].$$

Dividing by $\bar{V}_j$ and substituting the period-profit response gives the level response in terms of the present-value operators $(1-\tilde{\beta}_j)\mathbb{E}_t[\cdot]$. Subtracting the $t-1$ conditional expectation from the return $R_{jt} = d\log V_{jt} - d\log V_{j,t-1}$ applies the revision $(\mathbb{E}_t - \mathbb{E}_{t-1})$ to each operator, giving (D.3). □

Cross-sectional heterogeneity in loadings on aggregate news arises through the effective discount factor $\tilde{\beta}_j$: when an innovation has a non-flat expected horizon profile, longer-duration firms (higher ϱ_j) place more weight on distant horizons and load differently than shorter-duration firms.

D.3 Closed-Form Loading

To collapse the firm-specific operators into the benchmark-shaped scalar SDF channel, we impose two conditions.

Assumption 3. *(Permanent real cash-flow news) At the event horizon,* $\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)[d\log S_{t+k}] = d\log S_t$ *and* $\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)[d\log c_{t+k}] = d\log c_t$ *for all* $k \geq 0$.

Assumption 4. *(One-factor SDF term structure) The revision in the expected cumulative log-SDF response to the discount-rate shock admits the representation*

$$\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)[d\log \Xi_{t,t+k}]|_{\Xi} = a_k dv_t^{\Xi}, \qquad a_0 = 0,$$

for a deterministic sequence $\{a_k\}_{k\geq 0}$ *common across firms, with* $(1-\beta)\sum_{k=0}^{\infty}\beta^k a_k \neq 0$.

Assumption 3 is the standard asset-pricing convention that news at the event horizon is a step-function revision: the event shifts the expected paths of $\log S$ and $\log c$ uniformly at all horizons, with the $k=0$ case stating that the event-window change itself is unanticipated; dynamics already anticipated at $t-1$ are left unrestricted. Under this assumption, the consumption-driven component of cumulative log-SDF news vanishes from the expectation revision to first order, so the cumulative SDF news is driven entirely by dv_t^{Ξ}. Assumption 4 requires that the horizon profile of SDF news is generated by a single factor with a common term-structure loading $\{a_k\}$, which holds whenever v_t^{Ξ} follows a linear stationary process. Define the benchmark discount-rate news as the operator (D.2) evaluated at the benchmark duration $\varrho_j = 1$:

$$\mathscr{D}_t \equiv (1-\beta)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}\beta^k d\log \Xi_{t,t+k}|_{\Xi}\Big]. \tag{D.4}$$

Proposition 7 (Heterogeneous SDF Loading)**.** *Under Assumptions 3 and 4, the unexpected firm return admits the representation*

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j d\log S_t + \eta_j \mathscr{D}_t + \mu_t, \tag{D.5}$$

where the firm-invariant component is $\mu_t \equiv d\log c_t - \delta^S \gamma d\log S_t$*, and the loading is*

$$\eta_j = \frac{(1-\tilde{\beta}_j)\sum_{k=0}^{\infty}\tilde{\beta}_j^k a_k}{(1-\beta)\sum_{k=0}^{\infty}\beta^k a_k}, \qquad \tilde{\beta}_j = \beta\varrho_j, \tag{D.6}$$

depending only on ϱ_j *and the common term structure* $\{a_k\}$*. When* v_t^{Ξ} *follows an AR(1) with persistence* $\rho_{\Xi} \in [0,1)$,

$$\eta_j = \frac{\varrho_j(1-\beta\rho_{\Xi})}{1-\beta\varrho_j\rho_{\Xi}}, \tag{D.7}$$

which is strictly increasing in ϱ_j*. The benchmark* $\varrho_j = 1$ *gives* $\eta_j = 1$.

Proof. Under Assumption 3, the revision inside (D.2) equals $d\log S_t$ (resp. $d\log c_t$) at every horizon, and the duration weights $(1-\tilde{\beta}_j)\tilde{\beta}_j^k$ sum to one, so $\mathscr{S}_t^{(j)} = d\log S_t$ and $\mathscr{C}_t^{(j)} = d\log c_t$ independent of j. Substituting into (D.3) and collecting firm-invariant terms gives

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j d\log S_t + \mathscr{D}_t^{(j)} + \underbrace{(d\log c_t - \delta^S \gamma d\log S_t)}_{=\mu_t}.$$

Under Assumption 4, $\mathscr{D}_t^{(j)} = (1-\tilde{\beta}_j)\sum_{k=0}^{\infty}\tilde{\beta}_j^k a_k \cdot dv_t^{\Xi}$, and (D.4) gives $\mathscr{D}_t = (1-\beta)\sum_{k=0}^{\infty}\beta^k a_k \cdot dv_t^{\Xi}$. Taking the ratio defines $\eta_j \equiv \mathscr{D}_t^{(j)}/\mathscr{D}_t$, which is (D.6) and depends on ϱ_j but not on dv_t^{Ξ}.

For the AR(1) specialization, the conditional forecast revision satisfies

$$\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)[dv_{t+\ell}^{\Xi}] = \rho_{\Xi}^{\ell} dv_t^{\Xi}.$$

Therefore,

$$\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)[d\log \Xi_{t,t+k}]|_{\Xi} = \sum_{\ell=0}^{k-1} \rho_{\Xi}^{\ell} dv_t^{\Xi} = \frac{1-\rho_{\Xi}^k}{1-\rho_{\Xi}} dv_t^{\Xi},$$

which is Assumption 4 with $a_k = (1-\rho_{\Xi}^k)/(1-\rho_{\Xi})$. Evaluating the numerator of (D.6),

$$(1-\tilde{\beta}_j)\sum_{k=0}^{\infty}\tilde{\beta}_j^k a_k = \frac{1-\tilde{\beta}_j}{1-\rho_{\Xi}}\left(\frac{1}{1-\tilde{\beta}_j} - \frac{1}{1-\tilde{\beta}_j\rho_{\Xi}}\right) = \frac{\tilde{\beta}_j}{1-\tilde{\beta}_j\rho_{\Xi}}.$$

The denominator evaluates analogously with $\tilde{\beta}_j$ replaced by β, giving $\beta/(1-\beta\rho_{\Xi})$. Taking the ratio and substituting $\tilde{\beta}_j = \beta\varrho_j$,

$$\eta_j = \frac{\tilde{\beta}_j/(1-\tilde{\beta}_j\rho_{\Xi})}{\beta/(1-\beta\rho_{\Xi})} = \frac{\varrho_j(1-\beta\rho_{\Xi})}{1-\beta\varrho_j\rho_{\Xi}},$$

which is (D.7). Differentiating with respect to ϱ_j,

$$\frac{\partial\eta_j}{\partial\varrho_j} = \frac{1-\beta\rho_{\Xi}}{(1-\beta\varrho_j\rho_{\Xi})^2} > 0$$

for all $\beta\in(0,1)$, $\rho_{\Xi}\in[0,1)$, and $\varrho_j\in(0,1]$. □

Under the AR(1) specialization, the loading η_j is monotone in ϱ_j: longer-duration firms load more strongly on common discount-rate news. Two limiting cases are illuminating. With $\rho_{\Xi} = 0$ (iid news), $\eta_j = \varrho_j$, so the loading equals the survival probability directly. With $\rho_{\Xi} \to 1$ (permanent news), $\eta_j = \varrho_j(1-\beta)/(1-\beta\varrho_j)$, the familiar duration-of-a-perpetuity formula adjusted for franchise mortality. Equation (D.5) is the appendix counterpart of the body's extended firm-value response (10).

E Production Network Economy

This appendix sets up the production network economy of Section 4.3 and proves the results used there. The economy places a continuum of firms inside each of N sectors: firms are the objects the empirical analysis observes, sectors are the objects the input-output accounts measure, and the two tiers are connected by an exact aggregation result. We state the environment, the aggregation lemma, the first-order responses, the identification result, and Hulten's theorem for the GDP impact. Proofs are in Appendix F.

E.1 Environment

Time is discrete, indexed by $t \in \{0,1,\dots\}$. The economy comprises N sectors, indexed by n, each populated by a unit continuum of firms indexed by $j \in [0,1]$. The representative household is as in Section 2: it owns all firms, supplies software engineers in quantity s and non-SWE labor in quantity L inelastically, and financial markets are complete. The economy is driven by software engineering productivity S_t, which augments SWE labor, and by a vector $Z_t = (Z_{1t},\dots,Z_{Kt})$ of other aggregate shocks, which may include Hicks-neutral productivity, discount-factor shocks, and shifters of the demand composition across sectors; all shocks may be arbitrarily correlated.

Demand. Consumption is a two-tier CES aggregate. The upper tier combines sector composites C_{nt} with elasticity ζ and demand weights $\omega_n(Z_t)$ summing to one; the lower tier combines the varieties of the firms in a sector with elasticity ε,

$$c_t = \Big[\sum_{n=1}^{N} \omega_n(Z_t)^{1/\zeta} C_{nt}^{(\zeta-1)/\zeta}\Big]^{\zeta/(\zeta-1)}, \qquad C_{nt} = \Big[\int_0^1 c_{nt}(j)^{(\varepsilon-1)/\varepsilon} dj\Big]^{\varepsilon/(\varepsilon-1)},$$

with sector price indices P_{nt} and consumer price index P_t, which serves as numeraire, $P_t = 1$. We take $\varepsilon \geq \zeta$: two firms producing close substitutes inside a narrowly defined sector are at least as substitutable as two composites drawn from different sectors. The single-tier benchmark of Section 2, in which a firm and a sector are the same object, is the special case $\zeta = \varepsilon$.

The sector composite C_{nt} is the only channel through which sector n's output reaches the economy: the household consumes it, and every downstream firm uses it as an intermediate input, allocating across the constituent firm varieties with the same elasticity ε. This is what the nesting of Section 2 already implies, and it is the substantive aggregation assumption of the model: a firm that cuts its relative price gains demand from its sector peers at the rate ε whether the buyer is the household or a downstream producer. Writing $f_{nt} = P_{nt}C_{nt}/\sum_m P_{mt}C_{mt}$ for sector n's final-expenditure share, $\sum_n f_{nt} = 1$.

Production. Firm j in sector n produces with the technology of Section 2, with sector composites in place of final-good materials,

$$y_{nt}(j) = A_t z_n(j)\Big(\frac{H_{nt}(j)}{\alpha_n}\Big)^{\alpha_n} \prod_{m:\Omega_{nm}>0}\Big(\frac{X_{nm,t}(j)}{\Omega_{nm}}\Big)^{\Omega_{nm}}, \qquad \alpha_n + \sum_{m=1}^{N}\Omega_{nm} = 1,$$

where $X_{nm,t}(j)$ is the firm's use of the sector-m composite, Ω_{nm} the corresponding cost share, and $H_{nt}(j)$ the CES labor composite of Section 2,

$$H_{nt}(j) = \left[\phi_n(j)^{1/\sigma}(S_t s_{nt}(j))^{(\sigma-1)/\sigma} + (1-\phi_n(j))^{1/\sigma} L_{nt}(j)^{(\sigma-1)/\sigma}\right]^{\sigma/(\sigma-1)}.$$

Capital is a reproducible input purchased through the intermediate block, extending Section 2's final-good materials to the multi-sector setting: α_n is the labor share of sector n's costs, the row $\Omega_{n\cdot}$ absorbs materials and capital services alike, and GDP is value added. As in Section 2, we measure α_n by sector labor compensation over gross output (Appendix E.6). Firms in a sector share $(\alpha_n, \Omega_{n\cdot}, \sigma)$ and differ only in the software engineering intensity $\phi_n(j)$, which generates the firm's payroll share $\gamma_{nt}(j) = w_t s_{nt}(j)/(w_t s_{nt}(j) + W_t L_{nt}(j))$, and in the idiosyncratic productivity $z_n(j)$, which is orthogonal to $\phi_n(j)$ and becomes the regression residual. With $q_{nt}(j)$ the unit cost of the labor composite, marginal cost is

$$mc_{nt}(j) = \frac{1}{A_t z_n(j)} q_{nt}(j)^{\alpha_n} \prod_{m=1}^{N} P_{mt}^{\Omega_{nm}}.$$

Each firm is atomistic within its sector: it takes sector composites and aggregate prices as given, faces the within-sector CES demand with elasticity ε, and charges the constant markup $\mu = \varepsilon/(\varepsilon-1)$ of Section 2 (a parameter, not to be confused with the firm-invariant return component μ_t), so $p_{nt}(j) = \mu \cdot mc_{nt}(j)$ and per-period profits are $\pi_{nt}(j) = \frac{1}{\varepsilon} p_{nt}(j) y_{nt}(j)$.

Sector aggregates. Let $\mathscr{R}_{nt}(j) = p_{nt}(j) y_{nt}(j)$ denote firm revenue, $\mathscr{R}_{nt} = \int_0^1 \mathscr{R}_{nt}(j)dj$ sector revenue, and $\varsigma_{nt}(j) = \mathscr{R}_{nt}(j)/\mathscr{R}_{nt}$ the firm's share of sector sales; CES demand implies $\varsigma_{nt}(j) = (p_{nt}(j)/P_{nt})^{1-\varepsilon}$. We summarize the within-sector distribution of payroll shares by its sales-weighted mean and variance,

$$\bar{\gamma}_{nt} = \int_0^1 \varsigma_{nt}(j)\gamma_{nt}(j)dj, \qquad \sigma^2_{s,n} = \int_0^1 \varsigma_{nt}(j)\big(\gamma_{nt}(j) - \bar{\gamma}_{nt}\big)^2 dj,$$

the sector-level counterparts of γ and σ_s^2 in Section 2. Sector n's software engineering and non-SWE labor cost shares are $\alpha_n\bar{\gamma}_n$ and $\bar{\theta}_n = \alpha_n(1-\bar{\gamma}_n)$. Stacking the intermediate shares into the $N \times N$ matrix Ω, the Leontief inverse is $\Lambda = (I-\Omega)^{-1}$, and the cost-share identity gives $\Lambda\alpha = \mathbf{1}$. The cost-based Domar weight propagates final expenditure through the un-marked-up Leontief inverse,

$$\tilde{\lambda}_n = \sum_{m=1}^{N} f_m \Lambda_{mn}, \qquad \text{equivalently} \qquad \tilde{\lambda} = \Lambda^\top f, \tag{E.1}$$

and coincides with the sales share $\mathscr{R}_n/(P_t c_t)$ when $\mu = 1$ or $\Omega = 0$. Finally, we record the equilibrium factor-price responses to a unit software engineering productivity shock in the two scalars

$$u = \frac{d\log w_t - d\log S_t}{d\log S_t}, \qquad v = \frac{d\log W_t}{d\log S_t},$$

so that u is the response of the effective SWE wage, v that of the non-SWE wage, and $u-v$ the response of the effective relative wage that drives the entire cross-section.

E.2 Exact Aggregation

The first result states that the sector-level objects the input-output accounts measure are exactly the sales-weighted aggregates of their firm-level counterparts, with no residual dependence on the shape of the within-sector distribution.

Lemma 3 (Exact Aggregation)**.** *Suppose firms in sector n share α_n and the common markup μ. Then:*

(i) The sector software engineering cost share, sector SWE compensation divided by sector total cost, equals the sales-weighted mean of the firm cost shares,

$$\frac{\textit{SWE compensation in } n}{\textit{total cost of } n} = \int_0^1 \varsigma_n(j)\alpha_n\gamma_n(j)\,dj = \alpha_n\bar{\gamma}_n.$$

(ii) Each firm's cost-based Domar weight factorizes as $\tilde{\lambda}_n(j) = \tilde{\lambda}_n\varsigma_n(j)$, so the economy-wide cost-Domar-weighted SWE cost share aggregates exactly to the sector level,

$$\sum_{n=1}^{N}\int_0^1 \tilde{\lambda}_n(j)\alpha_n\gamma_n(j)\,dj = \sum_{n=1}^{N}\tilde{\lambda}_n\alpha_n\bar{\gamma}_n.$$

Lemma 3 is what licenses measuring sector shares and firm slopes from different sources. The sales weighting is not a choice but a consequence of the common composite: a firm's weight in any sector aggregate is its share of sector sales, because that is the share of every transaction the sector supplies. And because a sales-weighted mean of ratios collapses to a ratio of totals, the sector share in part (i) is the sector SWE wage bill over sector cost, an object measured directly from industry totals with the correct weighting already built in. Within a sector, moreover, a firm's sales share equals its wage-bill share, since the wage bill is the common fraction α_n/μ of revenue for every firm; the sales-weighted mean $\bar{\gamma}_n$ is therefore the ratio of the sector's SWE wage bill to its total wage bill.

E.3 First-Order Responses

We characterize the first-order response of firm profits to a software engineering productivity shock at steady state; we drop t subscripts on steady-state values. Let $\nu_n = \alpha_n\bar{\gamma}_n\mathscr{R}_n/\sum_k \alpha_k\bar{\gamma}_k\mathscr{R}_k$ denote sector n's share of economy-wide SWE compensation and $\nu_n^L = \bar{\theta}_n\mathscr{R}_n/\sum_k\bar{\theta}_k\mathscr{R}_k$ its share of non-SWE compensation.

Lemma 4 (Equilibrium Responses)**.** *To first order in $d\log S_t$, with $d\log S_t = 1$:*

(i) A firm's price relative to its sector composite moves only with its own software engineering intensity,

$$d\log p_{nt}(j) - d\log P_{nt} = \alpha_n\big(\gamma_n(j) - \bar{\gamma}_n\big)(u - v).$$

(ii) *The firm's profit response is a sector-common term plus a firm-specific deviation carrying the within-sector demand elasticity,*

$$d\log\pi_{nt}(j) = b_n + (1-\varepsilon)\alpha_n\big(\gamma_n(j) - \bar{\gamma}_n\big)(u-v), \tag{E.2}$$

where b_n is common to all firms in sector n, is measured net of the economy-wide expenditure response (a firm-invariant level annihilated by the centering matrix below), and carries the entire input-output and across-sector propagation of the shock: stacking, $b = \mathscr{B}(\alpha\bar{\gamma}u + \bar{\theta}v)$ with $\mathscr{B} = \Psi\mathrm{diag}(f_n E_t/\mathscr{R}_n)(1-\zeta)(I - \mathbf{1}f^\top)\Lambda$, where E_t is nominal final expenditure, $\Psi = (I-\mathscr{T})^{-1}$, and $\mathscr{T}_{nm} = \Omega_{mn}\mathscr{R}_m/(\mu\mathscr{R}_n)$ is the share of sector n's revenue coming from sector m's intermediate purchases.

(iii) *The pair (u,v) solves the two-equation factor-market-clearing system*

$$u+1 = \sum_n \nu_n\Big[b_n + (1-\bar{\gamma}_n)(1-\sigma)(u-v)\Big] + \sum_n \nu_n\big[(1-\varepsilon)\alpha_n - (1-\sigma)\big]\frac{\sigma^2_{s,n}}{\bar{\gamma}_n}(u-v),$$

$$v = \sum_n \nu^L_n\Big[b_n - \bar{\gamma}_n(1-\sigma)(u-v)\Big] - \sum_n \nu^L_n\big[(1-\varepsilon)\alpha_n - (1-\sigma)\big]\frac{\sigma^2_{s,n}}{1-\bar{\gamma}_n}(u-v),$$

which is linear in (u,v); the within-sector distributions enter only through the moments $(\bar{\gamma}_n, \sigma^2_{s,n})$. With b net of the common expenditure level, the system determines (u,v) in the normalization $d\log E_t = 0$; the numeraire $P_t = 1$ shifts both by a common constant, and every object used below, $u-v$, δ^S_{in}, and the GDP impact, is invariant to this normalization.

Part (i) is the engine of the aggregation: all firms in a sector buy the same composites at the same shares, so their marginal costs differ only through the labor composite, and within it only through the payroll share; the entire input-output block cancels when a firm's price is measured against its sector. Part (ii) turns the price deviation into a profit deviation through the within-sector elasticity $(1-\varepsilon)$: a firm whose relative price falls gains revenue at the rate $\varepsilon - 1$, because that is the elasticity every buyer applies inside the composite. The across-sector elasticity ζ and the network matrices (Ψ, Λ) appear only in the sector-common term b_n, which the sector fixed effects of the second stage absorb.

A remark on the firm coefficient is in order, because it is the single modeling choice that moves the quantitative answer. The deviation in (E.2) carries the within-sector $(1-\varepsilon)$ and no factor reflecting the firm's position in the production network: an atomistic firm's price cut reallocates demand only among its own sector peers, and every buyer of those peers substitutes through the same composite at ε. Two alternatives that may seem natural are wrong for the firm. Replacing $(1-\varepsilon)$ with the diagonal of the sector revenue-response matrix $\mathscr{B}$ embeds the demand-network propagation that a measure-zero firm does not trigger. Replacing it with $\alpha^f_n(1-\zeta)$, the sector's final-sales share times the across-sector reallocation rate, treats the firm's business-to-business sales as non-reallocating; the final-sales share correctly appears in the sector response b_n, where only final demand reallocates across sectors, but it does not govern reallocation within a sector. Both conventions import a sector-level object to the firm level, understate the firm gradient, and overstate the implied productivity gain.

E.4 Identification

Capitalizing Lemma 4 into firm values exactly as in Section 2, firm (n, j)'s period-t return is

$$R_{nt}(j) = b_n \mathscr{S}_t + \sum_k b_n^{Z_k} \mathscr{Z}_{kt} + (1-\varepsilon)\alpha_n\big(\gamma_n(j) - \bar{\gamma}_n\big)\Big[(u-v)\mathscr{S}_t + \sum_k (u_k - v_k)\mathscr{Z}_{kt}\Big] + \mu_t, \tag{E.3}$$

where $\mathscr{Z}_{kt}$ is the news in normalized present-value Z_k defined analogously to $\mathscr{S}_t$, the scalars (u_k, v_k) are the factor-price responses to a unit Z_k shock, $b_n^{Z_k}$ collects the sector-common response to Z_k, and μ_t is firm-invariant. Every within-sector deviation term is proportional to the same gradient $(1-\varepsilon)\alpha_n(\gamma_n(j) - \bar{\gamma}_n)$: within a sector, an aggregate shock can separate firms by software engineering intensity only by moving the effective relative wage. Substituting (E.3) into the two-stage regression and projecting on the payroll share conditional on the controls X_j, which include the sector fixed effects, yields

$$\delta = \kappa\delta_{\text{in}}^S + \sum_k \kappa_k^Z \delta_k^Z, \qquad \delta_{\text{in}}^S = (1-\varepsilon)\bar{\alpha}(u-v), \qquad \delta_k^Z = (1-\varepsilon)\bar{\alpha}(u_k - v_k), \tag{E.4}$$

where κ and κ_k^Z are the projection coefficients of $(\mathscr{S}_t, \mathscr{Z}_{kt})$ onto the residualized AI-index return, and $\bar{\alpha} = \sum_n \varpi_n \alpha_n$ averages sector labor shares with the regression's identifying weights $\varpi_n \propto$ (sample firms in n)$\times$ $\text{Var}_n(\gamma \mid X)$. The sector fixed effects absorb b_n and $b_n^{Z_k}$ entirely, so the network propagation never contaminates the slope. The empirical fixed effects are NAICS3 cells, which are, with minor exceptions, nested within the BEA Summary sectors; any partition at least as fine as the model's sectors absorbs b_n, and because the within-sector response is affine in $\gamma_n(j)$, finer cells leave the within-cell slope unchanged. Identification then requires only that the other aggregate shocks not move the effective relative wage.

Assumption 5. *(Balanced incidence in the network.) For each aggregate shock Z_k, the sector revenue responses are balanced across software-engineering-intensive and other sectors,*

$$\sum_{n=1}^{N} \big(\nu_n - \nu_n^L\big) b_n^{Z_k} = 0.$$

Assumption 5 is the network analogue of Assumption 1. It holds exactly for any shock whose sector-level incidence is uniform: discount-factor shocks, which move all values through the common μ_t, and Hicks-neutral productivity under a common labor share, as in Section 2; with heterogeneous α_n a Hicks-neutral shock propagates slightly differently across sectors through their roundabout intensity, and the assumption requires that this residual incidence not line up with sector software engineering intensity. Its main content is to restrict demand-composition shocks: expenditure shifts across sectors must not systematically favor sectors that employ software engineers intensively. Under it, the relative wage does not respond to Z_k, and the contamination terms vanish. Unlike Assumption 1, which restricts primitives, Assumption 5 is stated on the equilibrium incidence $b_n^{Z_k}$, because that is exactly what identification requires; uniform sector incidence and demand-weight shifts orthogonal to sector software engineering intensity are primitive sufficient conditions.

Proposition 8 (Identification in the Network Economy)**.** *Under Assumption 5, $u_k = v_k$ for every k, so $\delta_k^Z = 0$, the master equation (E.4) collapses to $\delta = \kappa\delta_{in}^S$, and*

$$\kappa = \frac{\delta}{\delta_{in}^S}, \qquad \delta_{in}^S = (1-\varepsilon)\bar{\alpha}(u-v). \tag{E.5}$$

In the one-sector roundabout economy of Section 2 ($N = 1$ and $\Omega_{11} = 1-\alpha$, so that $b = 0$), the system of Lemma 4(iii) gives $u - v = -1/\Sigma$ and $\delta_{in}^S = \alpha(\varepsilon-1)/\Sigma$, the closed form of Section 2.

The two-stage procedure therefore identifies κ exactly as in the simple model, with the structural gradient now solved on the network: the closed form of the gradient survives, and only the scalar $u-v$ changes, from $-1/\Sigma$ to the solution of the clearing system on the full network.

E.5 Hulten's Theorem and the GDP Impact

Real GDP is the consumption aggregate $Y_t = c_t$. Define the cost-based factor shares

$$\tilde{s}_S = \sum_{n=1}^{N} \tilde{\lambda}_n \alpha_n \bar{\gamma}_n, \qquad \tilde{s}_L = \sum_{n=1}^{N} \tilde{\lambda}_n \bar{\theta}_n, \qquad \tilde{s}_S + \tilde{s}_L = 1,$$

which partition GDP because $\Lambda\alpha = \mathbf{1}$, and let $s_{S,t} = w_t s/Y_t$ and $s_{L,t} = W_t L/Y_t$ denote the two factors' income shares.

Proposition 9 (Hulten in the Network Economy)**.** *To first order, the GDP impact of the software engineering channel is*

$$d\log Y_t\big|_S = \gamma^{IO} d\log S_t, \qquad \gamma^{IO} = \underbrace{\sum_{n=1}^{N} \tilde{\lambda}_n \alpha_n \bar{\gamma}_n}_{\textit{cost-Domar technology}} - \underbrace{\sum_{\tau\in\{S,L\}} \tilde{s}_\tau \frac{d\log s_{\tau,t}}{d\log S_t}}_{\textit{factor reallocation } \Delta}, \tag{E.6}$$

a cost-based-Domar-weighted sum of sector software engineering cost shares, minus the Baqaee and Farhi (2020) factor-reallocation term. The reallocation term vanishes when $\mu = 1$ or $\Omega = 0$. At $\mu = 1$ the technology term equals the economy-wide ratio of software engineering compensation to labor compensation, independent of the network. Consequently $\mathscr{Y}_t = \gamma^{IO}\mathscr{S}_t$, and the best linear predictor of GDP news given the residualized AI-index return is

$$\mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{AI}] = \gamma^{IO}\kappa \cdot \tilde{R}_t^{AI}. \tag{E.7}$$

The proposition separates measurement from mechanism. The technology term depends on the within-sector distributions only through the sector means $\bar{\gamma}_n$, by Lemma 3, and is independent of ε, ζ, and σ; at the efficient benchmark it is an accounting ratio, SWE compensation over labor compensation, so the network's shape does essentially no quantitative work in it. The reallocation term is first order only because markups keep factor allocations away from the efficient frontier; more precisely, it requires the compounded markup wedge to differ across sectors. In the roundabout economy of Section 2 the wedge is uniform, so the term vanishes exactly despite $\mu > 1$, consistent with the discussion in Section 4.1. On the empirical network the wedge varies with sectors' intermediate intensity, and the term is computed from the solved clearing system of Lemma 4(iii); at our calibration it equals $\Delta \approx 0.003$, about four percent of the technology term. The weight γ^{IO} sums over all N sectors of the economy, including those excluded from the regression sample; Section 4.3 uses this distinction to implement the full-economy aggregation.

E.6 Numerical Procedure and Results

The quantities reported in Section 4.3 are computed as follows.

1. Assemble Ω and $\Lambda = (I - \Omega)^{-1}$ from the BEA Make-Use tables at the Summary level ($N = 65$ endogenous private sectors), with each row rescaled to sum to $1 - \alpha_n$, where α_n is the sector's labor compensation share of gross output; this implements the reproducible-capital convention, with the intermediate block absorbing materials and capital services alike, extending Section 2's materials treatment and $\Lambda\alpha = \mathbf{1}$ holds exactly. Read the final-expenditure shares f_n off the use table. Measuring the cost share α_n by the revenue-based compensation ratio follows Section 2's treatment of α; grossing it up by the markup, the cost-basis mapping of Baqaee and Farhi (2020), would steepen the gradient and lower the headline from 2.09% to 1.71%.

2. Construct the sector moments $(\bar{\gamma}_n, \sigma^2_{s,n})$ as sales-weighted means and variances of the Revelio firm payroll shares within each BEA sector, using the all-sector firm panel; sectors without listed firms enter with $\bar{\gamma}_n = 0$.

3. Solve the two-equation clearing system of Lemma 4(iii) for (u, v) at the baseline calibration $(\sigma, \varepsilon, \zeta, \mu) = (1.6, 5, 5, 1.25)$.

4. Compute each sample firm's profit elasticity from Lemma 4(ii), using the firm's own payroll share and its sector's $(\alpha_n, \bar{\gamma}_n)$, and project it on the payroll share with the empirical controls and fixed effects X_j, restricted to the firms of the empirical estimation sample that are matched to BEA sectors. This is the same specification as the empirical second stage, the same regressor

Table E.1: The GDP impact of the software engineering channel: from the one-sector benchmark to the economy-wide production network

	δ^S_{in}	GDP weight	GDP impact	Step
Benchmark: one sector, listed-firm weight (Section 4.1)	0.74	0.109	3.31%	—
Identification: gradient solved on the full network	0.81	0.109	3.03%	×0.92
Aggregation: economy-wide SWE share of labor compensation	0.81	0.078	2.17%	×0.72
Distortion: markup-induced factor reallocation	0.81	0.075	2.09%	×0.96
Economy-wide impact (network appendix)	0.81	0.075	2.09%	

Notes: Each row replaces one ingredient of the Section 4.1 calculation with its production-network counterpart, holding the baseline slope $\delta = 1.52$ and the cumulative residualized AI-index return $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}^{\text{AI}}_t = 14.9\%$ fixed; the GDP impact is weight $\times\ (\delta/\delta^S_{\text{in}}) \times 14.9\%$. Row 2 replaces the closed-form gradient with the in-sample gradient solved on the full network. Row 3 extends the GDP weight from the listed-firm universe to the whole economy: the cost-Domar technology term equals the software engineering share of national labor compensation, combining the panel's sector-level payroll shares with the sector sizes of the BEA accounts. Row 4 subtracts the markup-induced factor-reallocation term of Proposition 9. On the matched subsample of firms assigned to BEA sectors the estimated slope is $\delta = 1.30$ and the bottom-row headline is 1.79%. Appendix Table 9 reports the sensitivity to σ.

and the same controls, restricted to the matched firms, on which the estimated slope is 1.30 against the full-sample 1.52; it delivers δ^S_{in}, and the GDP weight follows from (E.6).

Results. Table E.1 traces the answer from the benchmark of Section 4.1 to the network economy in Section 4.3 one ingredient at a time. The two model channels, identification and markup-induced reallocation, move the headline by eight and four percent; the aggregation step from listed firms to the economy carries the rest. The production network thus leaves the economics of the preceding sections intact, the same inversion logic and nearly the same multiplier, while extending the GDP impact from the listed-firm universe to the economy at large.

The headline inherits the sensitivity of the multiplier κ to the within-firm substitution elasticity σ documented in Section 3.4: across $\sigma \in [0.5, 2]$, the implied GDP change ranges from 0.9% to 2.5% (Appendix Table 9). It is essentially invariant to the across-sector demand elasticity, and it falls if firms within narrowly defined sectors are more substitutable than our benchmark, so the calibration sits at the upper end in that dimension.

Verification. Every closed form above is verified against a nonlinear general-equilibrium solver of the same economy. On an exactly calibrated version of the network, the linearized system reproduces the nonlinear finite-difference responses of the relative wage $u - v$ and of real GDP to machine precision, at $\mu = 1$ and at the baseline $\mu = 1.25$; the reallocation term in (E.6) is exactly zero at $\mu = 1$; and in the one-sector roundabout limit the solver reproduces the closed forms $u - v = -1/\Sigma$ and $\delta^S = \alpha(\varepsilon - 1)/\Sigma$ of Section 2 to machine precision. On the empirical network, the cost-Domar technology term equals the accounting ratio of SWE compensation to labor compensation to within 2×10^{-4}.

Appendix Table 9 reports the sensitivity of the resulting GDP impact to the within-firm substitution elasticity σ.

F Proofs

F.1 Proof of Lemma 1

Proof. The proof has three steps: relative profits given factor prices, the general-equilibrium factor-price response to the software engineering shock, and valuation.

Step 1: relative profits. Since firms charge a constant markup, profits are a constant fraction of revenue. Under CES demand and the numeraire $P_t = 1$,

$$\pi_{jt} = \frac{1}{\varepsilon} p_{jt} y_{jt} = \frac{1}{\varepsilon} \omega_j p_{jt}^{1-\varepsilon} c_t,$$

so

$$d\log \pi_{jt} = (1-\varepsilon) d\log mc_{jt} + d\log c_t.$$

By Shephard's lemma applied to the unit-cost function, and using that materials are priced in units of the numeraire,

$$d\log mc_{jt} = -d\log A_t + \alpha\gamma_j (d\log w_t - d\log S_t) + \alpha(1-\gamma_j) d\log W_t.$$

Differentiating the CES price index gives the final-expenditure-weighted average cost change,

$$d\log P_t = \sum_j f_j d\log p_{jt} = -d\log A_t + \alpha\gamma(d\log w_t - d\log S_t) + \alpha(1-\gamma) d\log W_t.$$

The numeraire $P_t = 1$ sets the left side to zero. Subtracting this zero from $d\log mc_{jt}$ isolates the relative cost change and substituting into the profit equation yields

$$d\log \pi_{jt} = (\varepsilon - 1)\alpha(\gamma_j - \gamma)(d\log S_t - (d\log w_t - d\log W_t)) + d\log c_t. \tag{F.1}$$

Step 2: factor prices. Define

$$x_t \equiv d\log S_t - (d\log w_t - d\log W_t),$$

the change in the effective software engineering cost advantage relative to non-SWE labor. We derive x_t from labor-market clearing.

The Cobb-Douglas materials nest fixes the aggregate labor cost share at α. Since markups are constant, each firm's total cost is the constant fraction $\frac{\varepsilon-1}{\varepsilon}$ of its revenue, and the cost shares $\alpha\gamma_j$ and

$\alpha(1-\gamma_j)$ of that cost go to SWE and non-SWE labor. Summing across firms,

$$w_t s = \frac{\varepsilon-1}{\varepsilon} c_t \sum_j \lambda_{jt} \alpha \gamma_{jt} = \frac{\varepsilon-1}{\varepsilon} c_t \theta \alpha \gamma_t \qquad \text{and} \qquad W_t L = \frac{\varepsilon-1}{\varepsilon} c_t \theta \alpha (1-\gamma_t),$$

where $\gamma_t = \sum_j f_{jt}\gamma_{jt}$ and $\theta = \sum_j \lambda_{jt} = Q_t/c_t$ is the constant gross-output-to-GDP ratio. Because θ is constant, fixed factor supplies give

$$d\log w_t = d\log c_t + \frac{d\gamma_t}{\gamma} \qquad \text{and} \qquad d\log W_t = d\log c_t - \frac{d\gamma_t}{1-\gamma}.$$

The aggregate demand term $d\log c_t$ cancels in the relative wage:

$$d\log w_t - d\log W_t = \frac{d\gamma_t}{\gamma(1-\gamma)}.$$

It remains to compute $d\gamma_t$. First, CES demand implies

$$d\log f_{jt} = (1-\varepsilon)(d\log p_{jt} - d\log P_t) = (\varepsilon-1)\alpha(\gamma_j - \gamma)x_t,$$

where the second equality uses (F.1). Second, the CES payroll-share formula implies

$$\log\left(\frac{\gamma_j}{1-\gamma_j}\right) = \log\left(\frac{\phi_j}{1-\phi_j}\right) + (1-\sigma)(\log w_t - \log S_t - \log W_t),$$

so

$$d\gamma_{jt} = (\sigma-1)\gamma_j(1-\gamma_j)x_t.$$

Therefore, using $df_{jt} = f_j d\log f_{jt}$ to convert the log differential to the absolute differential,

$$d\gamma_t = \sum_j \gamma_j df_{jt} + \sum_j f_j d\gamma_{jt} = (\varepsilon-1)\alpha \sum_j f_j\gamma_j(\gamma_j-\gamma)x_t + (\sigma-1)\sum_j f_j\gamma_j(1-\gamma_j)x_t.$$

Using

$$\sum_j f_j\gamma_j(\gamma_j-\gamma) = \sigma_s^2 \qquad \text{and} \qquad \sum_j f_j\gamma_j(1-\gamma_j) = \gamma(1-\gamma) - \sigma_s^2,$$

we obtain

$$d\gamma_t = \Big[(\sigma-1)\gamma(1-\gamma) + \big(\alpha(\varepsilon-1)-(\sigma-1)\big)\sigma_s^2\Big]x_t.$$

Substituting into the relative wage expression gives

$$d\log w_t - d\log W_t = \left[(\sigma-1) + \big(\alpha(\varepsilon-1)-(\sigma-1)\big)\frac{\sigma_s^2}{\gamma(1-\gamma)}\right]x_t.$$

Using the definition of x_t,

$$\begin{aligned} x_t &= d\log S_t - (d\log w_t - d\log W_t) \\ &= d\log S_t - \left[(\sigma-1) + \big(\alpha(\varepsilon-1)-(\sigma-1)\big)\frac{\sigma_s^2}{\gamma(1-\gamma)}\right]x_t. \end{aligned}$$

Hence

$$x_t = \frac{1}{\Sigma} d\log S_t, \qquad \text{where} \qquad \Sigma = \sigma + \left[\alpha(\varepsilon-1) - (\sigma-1)\right]\frac{\sigma_s^2}{\gamma(1-\gamma)}.$$

Substituting back into (F.1) yields the period-by-period profit response

$$d\log \pi_{jt} = \delta^S(\gamma_j - \gamma) d\log S_t + d\log c_t, \qquad \text{where} \qquad \delta^S \equiv \alpha\frac{\varepsilon-1}{\Sigma}. \tag{F.2}$$

Step 3: valuation and the NPV bridge. The same linearization applies at every future horizon. Therefore, for each $k \geq 0$,

$$d\log \pi_{j,t+k} = \delta^S(\gamma_j - \gamma) d\log S_{t+k} + d\log c_{t+k}.$$

The Hicks-neutral shock A_t cancels from relative unit costs entirely, and the discount-rate wedge v_t^{Ξ} does not enter profits directly; the firm-invariant level shift $d\log c_{t+k}$ captures the aggregate demand effect. Log-linearizing firm value to first order gives

$$d\log V_{jt} = (1-\beta)\mathbb{E}_t\Big[\sum_{k=0}^{\infty}\beta^k(d\log \pi_{j,t+k} + d\log \Xi_{t,t+k})\Big].$$

Substituting the expression above and combining the firm-invariant terms yields the firm-value level response

$$d\log V_{jt} = \delta^S\gamma_j(1-\beta)\mathbb{E}_t\Big[\sum_{k=0}^{\infty}\beta^k d\log S_{t+k}\Big] + \mu_t,$$

where μ_t collects all firm-invariant terms: the present-value aggregate demand effect, the stochastic discount factor effect, and the cross-sectional mean profit response. The empirical first stage uses returns; subtracting the $t-1$ conditional expectation, $R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = d\log V_{jt} - \mathbb{E}_{t-1}[d\log V_{jt}]$ applies the revision $(\mathbb{E}_t - \mathbb{E}_{t-1})$ to each term, giving

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S\gamma_j\mathscr{S}_t + \mu_t, \tag{F.3}$$

where $\mathscr{S}_t = (1-\beta)(\mathbb{E}_t - \mathbb{E}_{t-1})[\sum_{k=0}^{\infty}\beta^k d\log S_{t+k}]$ is the software engineering productivity news. This is (6). $\square$

F.2 Proof of Proposition 1

Proof. From the valuation step in the proof of Lemma 1, the unexpected firm return admits the representation

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S\gamma_j\mathscr{S}_t + \mu_t,$$

where μ_t is common across firms. Because $\mathbb{E}_{t-1}[R_{jt}]$ and R_t^f are known at $t-1$ while $\tilde{R}_t^{\mathrm{AI}}$ is unforecastable, $\mathrm{Cov}_t(R_{jt} - R_t^f, \tilde{R}_t^{\mathrm{AI}}) = \mathrm{Cov}_t(R_{jt} - \mathbb{E}_{t-1}[R_{jt}], \tilde{R}_t^{\mathrm{AI}})$. Projecting the excess return $R_{jt} - R_t^f$ on the residualized AI-index return gives

$$\beta_j = \frac{\mathrm{Cov}_t(R_{jt} - R_t^f, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})} = \bar{\beta} + \delta^S \kappa \gamma_j,$$

where

$$\bar{\beta} \equiv \frac{\mathrm{Cov}_t(\mu_t - R_t^f, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})}$$

is firm-invariant and

$$\kappa \equiv \frac{\mathrm{Cov}_t(\mathscr{S}_t, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})}$$

is the slope of the population linear projection of $\mathscr{S}_t$ on $\tilde{R}_t^{\mathrm{AI}}$. Taking the cross-sectional partial regression coefficient on the payroll share γ_j conditional on controls X_j,

$$\delta = \frac{\mathrm{Cov}_j(\beta_j, \gamma_j \mid X_j)}{\mathrm{Var}_j(\gamma_j \mid X_j)} = \delta^S \kappa,$$

where the second equality uses the fact that $\beta_j = \bar{\beta} + \delta^S \kappa \gamma_j$ is affine in γ_j alone, so partialling out X_j does not affect the slope. Rearranging,

$$\kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon - 1)} \cdot \delta.$$

The residualized AI return is mean-zero by construction. The variables $d\log S_t$ and $\mathscr{S}_t$ are first-order deviations from the no-shock steady-state path. Therefore, the best linear predictor has no constant term:

$$\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\mathrm{AI}}] = \kappa \cdot \tilde{R}_t^{\mathrm{AI}},$$

which is (9). □

F.3 Proof of Proposition 2

Proof. With state-dependent demand weights, per-period profit satisfies $d\log \pi_{jt} = \omega_j' d\log Z_t + (1 - \varepsilon) d\log mc_{jt} + d\log c_t$, where $\omega_j' = \partial \log \omega_j / \partial \log Z|_{\mathrm{ss}}$. The unit cost response is unchanged from the benchmark:

$$d\log mc_{jt} = -d\log A_t + \alpha\gamma_j(d\log w_t - d\log S_t) + \alpha(1-\gamma_j) d\log W_t.$$

Without loss of generality, normalize $d\log Z_t = 1$; all expressions below are per unit of the demand shock.

Relative wage response to the demand shifter. Define $\tilde{w} \equiv (d\log w_t - d\log W_t)|_Z$. With inelastic factor supplies, the factor-market-clearing identities

$$ws = \frac{\varepsilon - 1}{\varepsilon}\theta\alpha\gamma Y \qquad \text{and} \qquad WL = \frac{\varepsilon - 1}{\varepsilon}\theta\alpha(1-\gamma)Y$$

with $\gamma = \sum_j f_j \gamma_j$ and $\theta = Q/Y$ denoting the constant gross-output-to-GDP ratio (which drops from the differential), yield

$$\tilde{w} = \frac{d\gamma|_Z}{\gamma(1-\gamma)}.$$

Differentiating $\gamma = \sum_j f_j \gamma_j$ and using

$$d\log f_j = \omega'_j + (1-\varepsilon)\alpha(\gamma_j - \gamma)\,\tilde{w} + c_0$$

where c_0 is firm-invariant, together with

$$d\gamma_j = (1-\sigma)\gamma_j(1-\gamma_j)\,\tilde{w},$$

and eliminating c_0 via $\sum_j f_j \, d\log f_j = 0$, gives

$$\begin{aligned} d\gamma = &\sum_j f_j(\gamma_j - \gamma)\omega'_j \\ &+ \Big[(1-\sigma)\gamma(1-\gamma) + \big(\alpha(1-\varepsilon) - (1-\sigma)\big)\sigma_s^2\Big]\,\tilde{w}. \end{aligned}$$

Substituting $\tilde{w} = d\gamma/[\gamma(1-\gamma)]$ and using

$$\Sigma\gamma(1-\gamma) = \sigma\gamma(1-\gamma) + \big[\alpha(\varepsilon-1) - (\sigma-1)\big]\sigma_s^2$$

yields

$$\tilde{w} = \frac{\sum_j f_j(\gamma_j - \gamma)\omega'_j}{\Sigma\gamma(1-\gamma)}. \tag{F.4}$$

Slope decomposition. Let $\mathscr{Z}_t$ be the demand-shifter news, defined analogously to $\mathscr{S}_t$. Combining the firm-specific demand elasticity derived above with the valuation decomposition in (10), the first-order unexpected firm return can be written as

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j \mathscr{S}_t + [\omega'_j + (1-\varepsilon)\alpha\gamma_j\,\tilde{w}]\mathscr{Z}_t + \eta_j \mathscr{D}_t + \mu_t,$$

where $\delta^S = \alpha(\varepsilon-1)/\Sigma$ is the structural gradient on the payroll share and μ_t collects all firm-invariant terms. Let $\tilde{R}_t^{\text{AI}}$ denote the AI index return residualized on the first-stage factor span. The Frisch–Waugh–Lovell theorem implies that the population first-stage coefficient is

$$\beta_j = \frac{\text{Cov}_t(R_{jt} - R_t^f, \tilde{R}_t^{\text{AI}})}{\text{Var}_t(\tilde{R}_t^{\text{AI}})}.$$

Because the first stage partials out the span of the priced factors, μ_t and the risk-free rate R_t^f contribute only a firm-invariant constant, and the predictable component $\mathbb{E}_{t-1}[R_{jt}]$ is orthogonal to the unforecastable $\tilde{R}_t^{\text{AI}}$ and drops out. Therefore

$$\beta_j = \frac{\text{Cov}_t(\mu_t - R_t^f, \tilde{R}_t^{\text{AI}})}{\text{Var}_t(\tilde{R}_t^{\text{AI}})} + \kappa\delta^S\gamma_j + \kappa^Z[\omega'_j + (1-\varepsilon)\alpha\gamma_j\,\tilde{w}] + \kappa^\Xi\eta_j,$$

where $\kappa = \text{Cov}_t(\mathscr{S}_t, \tilde{R}_t^{\text{AI}})/\text{Var}_t(\tilde{R}_t^{\text{AI}})$ is the projection coefficient of interest, $\kappa^Z = \text{Cov}_t(\mathscr{Z}_t, \tilde{R}_t^{\text{AI}})/\text{Var}_t(\tilde{R}_t^{\text{AI}})$ measures the AI index's loading on the demand shifter, and $\kappa^\Xi = \text{Cov}_t(\mathscr{D}_t, \tilde{R}_t^{\text{AI}})/\text{Var}_t(\tilde{R}_t^{\text{AI}})$ measures its loading on common discount-rate news. Projecting β_j on γ_j conditional on X_j then gives

$$\delta = \kappa\delta^S + \kappa^Z\delta^Z + \kappa^\Xi\delta^\Xi, \tag{F.5}$$

where

$$\delta^\Xi \equiv \frac{\text{Cov}_j(\eta_j, \gamma_j \mid X_j)}{\text{Var}_j(\gamma_j \mid X_j)}.$$

Contamination gradient. The firm-specific component of the demand elasticity is $\omega'_j + (1-\varepsilon)\alpha\gamma_j\tilde{w}$. Projecting it onto γ_j conditional on X_j gives

$$\delta^Z = \frac{\text{Cov}_j(\omega'_j, \gamma_j \mid X_j)}{\text{Var}_j(\gamma_j \mid X_j)} + (1-\varepsilon)\alpha\tilde{w}. \tag{F.6}$$

The first condition of Assumption 1 zeros the first term directly. The second condition zeros the second: it sets the final-expenditure-weighted covariance $\text{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$, and hence $\tilde{w} = 0$ by (F.4). Hence $\delta^Z = 0$.

Discount-rate contamination. The remaining contamination term in (F.5) is $\kappa^\Xi\delta^\Xi$, where $\delta^\Xi = \text{Cov}_j(\eta_j, \gamma_j \mid X_j)/\text{Var}_j(\gamma_j \mid X_j)$ by construction. Assumption 2 eliminates it directly: factor spanning gives $\text{Cov}_t(\mathscr{D}_t, \tilde{R}_t^{\text{AI}}) = 0$, hence $\kappa^\Xi = 0$; cross-sectional orthogonality gives $\text{Cov}_j(\eta_j, \gamma_j \mid X_j) = 0$, hence $\delta^\Xi = 0$. In either case,

$$\kappa^\Xi\delta^\Xi = 0.$$

A sufficient primitive condition for cross-sectional orthogonality is that franchise duration ϱ_j be conditionally independent of γ_j given X_j: since η_j is a deterministic function of ϱ_j by Proposition 7, conditional independence transfers to η_j and yields $\text{Cov}_j(\eta_j, \gamma_j \mid X_j) = 0$. Combining this with $\delta^Z = 0$ from Assumption 1, (F.5) reduces to

$$\delta = \kappa\delta^S.$$

The SWE productivity shock S_t does not enter $\omega_j(Z_t)$, so the supply-side derivation of $\delta^S = \alpha(\varepsilon - 1)/\Sigma$ from Section 2 is unaffected by the demand generalization. Rearranging gives

$$\kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon-1)} \cdot \delta.$$

□

F.4 Proof of Proposition 3

Proof. We prove the present-value statement first and then derive the flow special case under a permanent software engineering productivity shock.

Annualized present value of GDP growth. In the model of Section 2, final-good materials enter the outer Cobb-Douglas nest and are priced in units of the numeraire. Thus the direct effect of S_t on firm j's unit cost operates only through the effective software engineering wage w_t/S_t. At each horizon k,

$$\frac{\partial \log mc_j}{\partial \log S} = -\alpha\gamma_j,$$

where $\alpha\gamma_j$ is the software engineering wage bill as a share of total cost, including materials. Thus, to first order, $d\log S_{t+k}$ is equivalent to a firm-specific Hicks-neutral TFP shock of size $\alpha\gamma_j d\log S_{t+k}$. The economy is distorted: firms charge a constant markup $\mu = \varepsilon/(\varepsilon-1)$, and materials are bought at the marked-up final-good price, so the materials roundabout compounds the markup wedge. Hulten's theorem, which assigns each firm its sales-Domar weight λ_j, is the efficient-economy ($\mu = 1$) special case and does not apply here. In a distorted economy a firm-specific Hicks-neutral shock raises real GDP by the firm's cost-based Domar weight $\tilde{\lambda}_j$ (Baqaee and Farhi, 2020), which coincides with the sales-Domar weight λ_j when $\mu = 1$ or when $\alpha = 1$. With the cost-based Leontief inverse $\Lambda = (I-\Omega)^{-1}$ and the rank-one materials roundabout $\Omega_{jk} = (1-\alpha)f_k$ (every firm spends a share $1-\alpha$ of cost on the final-good composite, whose composition is the final-expenditure shares f), the cost-based Domar weight $\tilde{\lambda} = \Lambda^\top f$ is

$$\tilde{\lambda}_j = (\Lambda^\top f)_j = f_j\big(1 + (1-\alpha) + (1-\alpha)^2 + \cdots\big) = \frac{f_j}{\alpha}, \qquad \sum_j \tilde{\lambda}_j = \frac{1}{\alpha}$$

(a dollar of final demand for j recursively draws in materials, the geometric sum being $1/\alpha$). Therefore, applying the cost-based first-order GDP accounting period by period,

$$d\log Y_{t+k}\big|_S = \sum_j \tilde{\lambda}_j \alpha\gamma_j d\log S_{t+k} = \sum_j f_j\gamma_j d\log S_{t+k} = \gamma\, d\log S_{t+k}, \qquad \gamma = \sum_j f_j\gamma_j.$$

Multiplying by $(1-\beta)\beta^k$, summing over k, and applying the revision operator $(\mathbb{E}_t - \mathbb{E}_{t-1})$ gives the GDP news

$$\mathscr{Y}_t = (1-\beta)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty} \beta^k d\log Y_{t+k}\big|_S\Big] = \gamma\mathscr{S}_t,$$

which is the expression in Proposition 3.

Conditional prediction. Using Proposition 1,

$$\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\mathrm{AI}}] = \kappa \cdot \tilde{R}_t^{\mathrm{AI}}.$$

Since $\mathscr{Y}_t = \gamma\mathscr{S}_t$, and since $\mathscr{Y}_t$ is therefore also a first-order deviation from the no-shock steady-state path, linearity of the linear projection gives

$$\mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{\mathrm{AI}}] = \gamma\kappa \cdot \tilde{R}_t^{\mathrm{AI}},$$

with slope $\gamma\kappa = \frac{\gamma\Sigma}{\alpha(\varepsilon-1)} \cdot \delta$, which is (15).

Flow special case under a random walk. If the software engineering productivity level follows a random walk, the news operator extracts the level innovation:

$$\mathscr{S}_t = d\log S_t - \mathbb{E}_{t-1}[d\log S_t].$$

Applying the same period-by-period GDP map gives

$$\mathscr{Y}_t = d\log Y_t\big|_S - \mathbb{E}_{t-1}[d\log Y_t\big|_S] = \gamma\big(d\log S_t - \mathbb{E}_{t-1}[d\log S_t]\big) = \gamma\mathscr{S}_t,$$

which is (16). Combining this with the random-walk specialization of (9) yields (15) with slope $\gamma\kappa = \frac{\gamma\Sigma}{\alpha(\varepsilon-1)}\cdot\delta$. $\square$

F.5 Endogenous CES Representation of the Task Price Index

Derivation of (18). The CES task aggregator $T_{jt} = [\int_0^1 y_{jt}(z)^{(\eta-1)/\eta}dz]^{\eta/(\eta-1)}$ has dual unit cost

$$p_t^T = \Big[\int_0^1 q_t(z)^{1-\eta}dz\Big]^{1/(1-\eta)},$$

where $q_t(z)$ is the unit cost of task z. Cost minimization within each task assigns task z to whichever factor has lower effective unit cost: $q_t(z) = \min\{w_t/\psi_S(z), p_t^A/\psi_A(z)\}$. By the monotonicity of $\psi_S(z)/\psi_A(z)$ in z, this minimum is attained by AI for $z < I_t$ and by SWE labor for $z > I_t$, with the threshold I_t defined by the indifference condition (17). Splitting the integral at I_t,

$$\begin{aligned}\int_0^1 q_t(z)^{1-\eta}dz &= \int_0^{I_t}\Big(\frac{p_t^A}{\psi_A(z)}\Big)^{1-\eta}dz + \int_{I_t}^1\Big(\frac{w_t}{\psi_S(z)}\Big)^{1-\eta}dz\\ &= (p_t^A)^{1-\eta}\int_0^{I_t}\psi_A(z)^{\eta-1}dz + w_t^{1-\eta}\int_{I_t}^1\psi_S(z)^{\eta-1}dz\\ &= B_A(I_t)(p_t^A)^{1-\eta} + B_S(I_t)w_t^{1-\eta}.\end{aligned}$$

Raising to the power $1/(1-\eta)$ gives (18). The expression has the form of a two-factor CES aggregator over w_t and p_t^A with elasticity η and weights $B_S(I_t), B_A(I_t)$ that depend endogenously on the equilibrium threshold. $\square$

F.6 Proof of Proposition 4

Proof. We work to first order around the steady state and drop time subscripts on steady-state objects. The proof has five steps: equilibrium accounting with scarce compute; the envelope characterization of the software price; the cross-sectional profit response; the general-equilibrium factor prices, where the automation margin appears; and the aggregation to firm value, which delivers the two best-linear-predictor statements (21) and (22). Throughout, ρ is held at its steady-state value; the closing remark shows that the endogenous movement $d\rho_t$ is absorbed into the definition of $\mathscr{S}_t$ and changes none of the results.

Step 0: accounting with scarce compute and materials. Because the AI input is built from compute rather than from the final good, the only roundabout use of output is materials, exactly as in Section 2. Let Q_t denote total final-good absorption (consumption plus materials) and c_t real GDP. With materials the fraction $1-\alpha$ of each producer's cost, $c_t = Q_t/\theta$ with $\theta = 1/[1-(1-\alpha)\frac{\varepsilon-1}{\varepsilon}]$, so $d\log c_t = d\log Q_t$. Each good's demand scales with absorption, $y_{jt} = \omega_j(p_{jt}/P_t)^{-\varepsilon}Q_t$. As in Section 2, the value-added shares, each firm's value added over GDP, are the final-expenditure shares $f_j = \omega_j(p_j/P)^{1-\varepsilon} = p_j c_j/c$, which sum to one; the sales-Domar weights $\lambda_j = p_{jt}y_{jt}/c_t$ equal θf_j. Write $\gamma^T \equiv \sum_j f_j\gamma_j^T$ for the value-added software-composite payroll share. Constant markups make each producer's cost the fraction $\frac{\varepsilon-1}{\varepsilon}$ of its revenue, with the labor aggregate the fraction α of cost and materials the fraction $1-\alpha$; the software supplier and the AI sector are competitive. Summing input bills across producers and splitting the software bill into its SWE and compute parts, in proportions $1-\rho$ and ρ, gives the factor-income identities

$$ws = \tfrac{\varepsilon-1}{\varepsilon}(1-\rho)\alpha\gamma^T Q, \qquad p_K K = \tfrac{\varepsilon-1}{\varepsilon}\rho\alpha\gamma^T Q, \qquad WL = \tfrac{\varepsilon-1}{\varepsilon}\alpha(1-\gamma^T)Q. \tag{F.7}$$

Adding materials spending $\frac{\varepsilon-1}{\varepsilon}(1-\alpha)Q$ and profit $\frac{1}{\varepsilon}Q$ recovers gross output Q; netting out materials leaves nominal GDP $ws + p_K K + WL + \frac{1}{\varepsilon}Q = c$, of which the compute rent $p_K K$ is a component.

Step 1: the software price (envelope argument). Differentiating (18) with respect to (w_t, p_t^A, I_t), the threshold terms enter through $B_A'(I_t) = \psi_A(I_t)^{\eta-1}$ and $B_S'(I_t) = -\psi_S(I_t)^{\eta-1}$ and sum to $\big[(p_t^A)^{1-\eta}\psi_A(I_t)^{\eta-1} - w_t^{1-\eta}\psi_S(I_t)^{\eta-1}\big]dI_t$, which the indifference condition (17) sets to zero. Dividing the remaining terms by $(1-\eta)(p_t^T)^{1-\eta}$ and using $\rho_t = B_A(I_t)(p_t^A)^{1-\eta}/(p_t^T)^{1-\eta}$ gives

$$d\log p_t^T = (1-\rho)d\log w_t + \rho\, d\log p_t^A, \qquad p_t^A = \frac{p_{K,t}}{S_t^A}, \tag{F.8}$$

with the coefficients evaluated at the steady-state ρ. The cross-task elasticity η does not appear: by the envelope theorem, reassigning the indifferent marginal task is first-order costless, so p_t^T moves as if the automation threshold were fixed. Holding w_t and $p_{K,t}$ fixed, this delivers the software-sector productivity gain $d\log S_t \equiv -d\log p_t^T|_{w,p_K} = \rho\, d\log S_t^A$.

Step 2: cross-sectional profit response. With $y_{jt} = \omega_j p_{jt}^{-\varepsilon}Q_t$ and constant markups, $d\log\pi_{jt} = (1-\varepsilon)d\log mc_{jt} + d\log c_t$. Shephard's lemma applied to (19) and (F.8) gives

$$d\log mc_{jt} = -d\log A_t + \alpha\, d\log W_t - \alpha\gamma_j^T\chi_t, \qquad \chi_t \equiv d\log W_t - d\log p_t^T,$$

so the relative price driving substitution is that of the software composite against non-SWE labor. The numeraire $P_t = 1$ weights by final-expenditure shares, $0 = \sum_j f_j d\log mc_{jt}$, hence $\alpha\, d\log W_t - d\log A_t =$

$\alpha\gamma^T\chi_t$. Subtracting,

$$d\log\pi_{jt} = (\varepsilon-1)\alpha(\gamma_j^T - \gamma^T)\chi_t + d\log c_t. \tag{F.9}$$

The Hicks-neutral shock A_t cancels from χ_t and the time-varying discount factor β_t does not enter production; both move only the firm-invariant terms $d\log W_t$ and $d\log c_t$. Software productivity is the sole source of cross-sectional variation.

Step 3: general-equilibrium factor prices and the automation margin. We determine χ_t from factor-market clearing, setting $d\log A_t = 0$.

(i) Numeraire. The condition of Step 2 is $\alpha\, d\log W_t = \alpha\gamma^T\chi_t$, i.e. $d\log W_t = \gamma^T\chi_t$.

(ii) The displacement elasticity. Differentiating the compute share $\rho = B_A(I)(p^A)^{1-\eta}/(p^T)^{1-\eta}$, with the threshold moving through the indifference condition (17), gives

$$d\log\frac{\rho}{1-\rho} = (\eta^\star - 1)(d\log w_t - d\log p_t^A), \qquad \eta^\star = \eta + \frac{\nu}{\vartheta\rho(1-\rho)} > \eta, \tag{F.10}$$

where $\vartheta = \frac{d}{dI}\log\frac{\psi_S(I)}{\psi_A(I)}$ is the slope of comparative advantage at the margin and $\nu = (w/\psi_S(I))^{1-\eta}/(p^T)^{1-\eta}$ the cost weight of the marginal task. The derivation parallels Step 1 but retains the share term that drops out of the price derivation: $\eta^\star$ combines an intensive part η (substitution at a fixed threshold) and an extensive part $\nu/[\vartheta\rho(1-\rho)]$ (the threshold reassigning tasks).

(iii) Factor markets and incidence. With (s, L, K) fixed, the income identities (F.7) read in log changes

$$d\log w_t = -\frac{d\rho}{1-\rho} + d\log\gamma^T + d\log c_t, \qquad d\log p_{K,t} = \frac{d\rho}{\rho} + d\log\gamma^T + d\log c_t,$$

and $d\log W_t = -\frac{d\gamma^T}{1-\gamma^T} + d\log c_t$. Subtracting the first two gives the incidence identity $d\log w_t - d\log p_{K,t} = -d\rho/[\rho(1-\rho)]$: with both factors fixed, the relative price absorbs the share movement one-for-one. Combining with (F.10) and $d\log p_t^A = d\log p_{K,t} - d\log S_t^A$ delivers the change in the share and the SWE-compute displacement,

$$d\rho = \rho(1-\rho)\frac{\eta^\star - 1}{\eta^\star} d\log S_t^A, \qquad d\log\frac{w_t}{p_{K,t}} = -\frac{\eta^\star - 1}{\eta^\star} d\log S_t^A.$$

(iv) Reallocation and solving. CES demand and (F.9) give $d\log f_{jt} = (\varepsilon-1)\alpha(\gamma_j^T - \gamma^T)\chi_t$, and the CES payroll-share formula gives $d\gamma_{jt}^T = (\sigma-1)\gamma_j^T(1-\gamma_j^T)\chi_t$; these combine exactly as in the benchmark of Appendix F.1 to

$$d\log\gamma^T = (1-\gamma^T)(\Sigma^T - 1)\chi_t, \qquad \Sigma^T = \sigma + \left[\alpha(\varepsilon-1) - (\sigma-1)\right]\frac{(\sigma_s^T)^2}{\gamma^T(1-\gamma^T)}. \tag{F.11}$$

Combining the three factor-price equations of (iii) with the envelope (F.8) and $d\log p_t^A = d\log p_{K,t} - d\log S_t^A$ collapses the share movement: $d\log p_t^T = d\log\gamma^T + d\log c_t - \rho\, d\log S_t^A$, in which $d\rho$, and hence $\eta^\star$, cancels. The non-SWE identity with $d\log W_t = \gamma^T\chi_t$ and (F.11) gives $d\log c_t = \gamma^T\Sigma^T\chi_t$.

Substituting both into $\chi_t = d\log W_t - d\log p_t^T$, the coefficients collect to Σ^T and

$$\chi_t = \frac{1}{\Sigma^T} d\log S_t. \tag{F.12}$$

Step 4: firm value and inversion. Substituting (F.12) into (F.9),

$$\begin{aligned} d\log \pi_{jt} &= \alpha \frac{\varepsilon - 1}{\Sigma^T}(\gamma_j^T - \gamma^T) d\log S_t + d\log c_t \\ &= \alpha \frac{\varepsilon - 1}{\Sigma^T}\gamma_j^T d\log S_t + (\text{firm-invariant}). \end{aligned}$$

The same linearization holds at every horizon. Aggregating with $V_{jt} = \mathbb{E}_t \sum_{\tau \ge 0} \Xi_{t,t+\tau} \pi_{j,t+\tau}$ and taking the unexpected period-t return, exactly as in the proof of Lemma 1, yields the first-order firm-return response

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \alpha \frac{\varepsilon - 1}{\Sigma^T}\gamma_j^T \mathscr{S}_t + \mu_t, \qquad \mathscr{S}_t = (1-\beta)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k \ge 0} \beta^k d\log S_{t+k}\Big],$$

where μ_t is firm-invariant. Projecting on the residualized AI-index return gives $\beta_j = \bar{\beta} + \alpha \frac{\varepsilon-1}{\Sigma^T}\kappa\gamma_j^T$, where $\bar{\beta}$ is firm-invariant and κ is the projection of $\mathscr{S}_t$ on $\tilde{R}_t^{\text{AI}}$ in (7). The firm beta is affine in the composite payroll share γ_j^T, so the second-stage partial regression on γ_j^T has slope

$$\delta^T = \frac{\text{Cov}_j(\beta_j, \gamma_j^T \mid X_j)}{\text{Var}_j(\gamma_j^T \mid X_j)} = \kappa\, \alpha \frac{\varepsilon - 1}{\Sigma^T} = \kappa\, \delta^{S,T}, \qquad \delta^{S,T} = \alpha \frac{\varepsilon - 1}{\Sigma^T}.$$

Rearranging gives $\kappa = \delta^T / \delta^{S,T}$, which is (21), with the structural gradient $\delta^{S,T} = \alpha(\varepsilon - 1)/\Sigma^T$ the Section 2 gradient read on composite payroll shares.

Step 5: GDP impact. Real GDP is consumption, $d\log Y_t = d\log c_t$. The non-SWE income identity in (F.7), with L fixed, gives $d\log c_t = d\log W_t + \frac{d\gamma^T}{1-\gamma^T}$, and using the numeraire $d\log W_t = \gamma^T \chi_t$ together with (F.11),

$$d\log Y_t = \gamma^T\big[1 + (\Sigma^T - 1)\big]\chi_t = \gamma^T \Sigma^T \chi_t = \gamma^T d\log S_t,$$

the last step using (F.12). Equivalently, the same cost-based Domar accounting as in the benchmark applies: a firm-specific Hicks-neutral shock of size $\alpha\gamma_j^T d\log S_t$ raises real GDP by the firm's cost-based Domar weight f_j/α (Baqaee and Farhi, 2020), and $\sum_j (f_j/\alpha)\alpha\gamma_j^T = \gamma^T$. Aggregating over horizons exactly as in the proof of Proposition 3 gives $\mathscr{Y}_t = \gamma^T \mathscr{S}_t$, and linearity of the projection delivers $\mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}_t^{\text{AI}}] = \gamma^T \kappa \tilde{R}_t^{\text{AI}}$, which is (22). The reallocation elasticity Σ^T governs both the relative-price response (F.12) and the response of output to it, and the two appearances cancel, leaving Hulten's theorem on a value-added basis with the cost-based Domar weight γ^T; the factor $1/\alpha$ is the roundabout value-added multiplier, the inverse of the labor (cost-based value-added) share. As $\rho \to 0$ the composite collapses to software-engineering labor and γ^T, Σ^T reduce to their Section 2 counterparts.

Endogenous AI cost share. If ρ_t responds to the shock, (F.8) still uses the steady-state ρ because the envelope evaluates the first-order pass-through at the reference allocation. Thus $d\log S_t \equiv -d\log p_t^T|_{w,p_K} = \rho d\log S_t^A$ remains the relevant total software-sector productivity change to first order. The movement $d\rho_t$ derived in Step 3 is the distributional margin: by (F.10) it changes how the software bill splits between SWE wages and compute rents, lowering $w_t/p_{K,t}$, but, as Steps 4 and 5 show, it leaves the structural gradient and the GDP impact untouched. Recovering the pure upstream productivity change $\mathscr{S}_t^A$ separately would require the comparative-advantage schedule that governs ρ_t; the empirical strategy never needs it. □

F.7 Proof of Proposition 5

Proof. The proof differs from the proof of Lemma 1 in two ways: the mass of varieties can move, and active varieties survive with probability $1-\eta$. Neither change affects the cross-sectional production-side gradient. Let

$$x_t \equiv d\log S_t - (d\log w_t - d\log W_t)$$

denote the reduction in the effective software engineering cost relative to non-SWE labor. The production technology and the payroll share γ_j (with cost share $\alpha\gamma_j$) are the same as in Section 2. Repeating the unit-cost, CES-demand, and production labor-market clearing calculation in the proof of Lemma 1 gives

$$x_t = \frac{1}{\Sigma} d\log S_t, \qquad \text{where} \qquad \Sigma = \sigma + \left[\alpha(\varepsilon-1)-(\sigma-1)\right]\frac{\sigma_s^2}{\gamma(1-\gamma)}.$$

Endogenous variety creation adds the same extensive-margin correction as in (32). It also changes total final-good absorption through $X_t = c_t + M_t + R_t$. Because new varieties are drawn from the same type distribution as incumbents, these terms shift all active varieties uniformly, or cancel in the final-expenditure-weighted mean when relative costs are formed. They therefore enter only the firm-invariant term below.

The per-period profit response of an active variety can be written as

$$d\log\pi_{jt} = \alpha\frac{\varepsilon-1}{\Sigma}\gamma_j d\log S_t + \tilde{v}_t,$$

where $\tilde{v}_t$ is firm-invariant. The same linearization applies at every horizon. Using (25), the steady-state relation $\Xi_{t,t+k} = \beta^k$, and subtracting the $t-1$ conditional expectation from the return $R_{jt} = d\log V_{jt} - d\log V_{j,t-1}$, which applies the revision $(\mathbb{E}_t - \mathbb{E}_{t-1})$ in the definition of $\mathscr{S}_t^\eta$,

$$\begin{aligned} R_{jt} - \mathbb{E}_{t-1}[R_{jt}] &= (1-\beta(1-\eta))\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty}(\beta(1-\eta))^k(d\log\pi_{j,t+k} + d\log\Xi_{t,t+k})\Big] \\ &= \delta^S\gamma_j\mathscr{S}_t^\eta + v_t, \end{aligned}$$

where $\delta^S = \alpha(\varepsilon - 1)/\Sigma$ and ν_t collects all firm-invariant terms, including SDF news. This establishes (30).

Projecting R_{jt} (equivalently its unexpected component, since $\tilde{R}_t^{\mathrm{AI}}$ is unforecastable) on the residualized AI-index return $\tilde{R}_t^{\mathrm{AI}}$ gives

$$\beta_j = \bar{\beta} + \delta^S \kappa \gamma_j, \qquad \bar{\beta} = \frac{\mathrm{Cov}_t(\nu_t, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})}, \qquad \kappa = \frac{\mathrm{Cov}_t(\mathscr{S}_t^{\eta}, \tilde{R}_t^{\mathrm{AI}})}{\mathrm{Var}_t(\tilde{R}_t^{\mathrm{AI}})}.$$

Taking the cross-sectional partial regression coefficient on the payroll share γ_j conditional on controls X_j gives $\delta = \delta^S \kappa$. Rearranging yields

$$\kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon - 1)} \cdot \delta,$$

which is (31). □

F.8 Proof of Proposition 6

Proof. We first derive the production-side decomposition in (32), holding A fixed, and then combine it with the innovation block.

Output. With constant markups, aggregate revenue equals $\frac{\varepsilon}{\varepsilon-1}$ times aggregate cost: summing $p_j y_j = \frac{\varepsilon}{\varepsilon-1}\mathscr{C}_j$ over j gives $X^* = \frac{\varepsilon}{\varepsilon-1}\mathscr{C}$, where $X^* = \int p_j y_j \, dj$ is total final-good absorption and $\mathscr{C} = \int \mathscr{C}_j \, dj$ is total cost. Labor is the cost share α and materials the share $1-\alpha$, so $w^* s + W^* L = \alpha \mathscr{C}$ and aggregate materials are $M^* = (1-\alpha)\mathscr{C}$. GDP is value added, $Y^* = X^* - M^*$; eliminating $\mathscr{C} = (w^* s + W^* L)/\alpha$ gives

$$Y^* = \frac{w^* s + W^* L}{\alpha}\left[\frac{\varepsilon}{\varepsilon - 1} - (1-\alpha)\right],$$

where (w^*, W^*) are steady-state production wages. The endowments (s, L) are fixed and the bracket is constant, so log-differentiating gives

$$d\log Y^* = \gamma \, d\log w^* + (1-\gamma)\, d\log W^*, \qquad \gamma = \frac{w^* s}{w^* s + W^* L}. \tag{F.13}$$

The weight γ is the aggregate software engineering *payroll* share. Writing the SWE wage bill $w^* s = \int \alpha \gamma_j \mathscr{C}_j \, dj$ and the total labor bill $w^* s + W^* L = \alpha \mathscr{C}$, and using the value-added share identity $\mathscr{C}_j / \mathscr{C} = f_j$,

$$\gamma = \frac{w^* s}{w^* s + W^* L} = \frac{\int \alpha \gamma_j \mathscr{C}_j \, dj}{\alpha \mathscr{C}} = \int f_j \gamma_j \, dj,$$

the final-expenditure-weighted payroll share of Section 2.

Numeraire. Under constant markups, $d\log p_j = d\log mc_j$. By Shephard's lemma applied to the CES unit-cost function dual to (1), with final-good materials priced at the numeraire,

$$d\log mc_j = \alpha\gamma_j d\log(w^*/S) + \alpha(1-\gamma_j)d\log W^*, \tag{F.14}$$

where the labor coefficients $\alpha\gamma_j$ and $\alpha(1-\gamma_j)$ sum to α, the labor cost share, because materials take the cost share $1-\alpha$ and carry no price movement, and we hold A fixed. When N^* changes simultaneously with S, the numeraire condition must account for the extensive margin. The total differential of $P^{1-\varepsilon} = \int_0^{N^*} p_j^{1-\varepsilon} dj = 1$ is

$$0 = (1-\varepsilon)\int_0^{N^*} f_j d\log p_j dj + \frac{\int_0^{N^*} p_j^{1-\varepsilon} dj}{N^*} dN^*,$$

where the second term is the average contribution of a marginal entrant mass drawn from the same type distribution as incumbents. Since $\int_0^{N^*} p_j^{1-\varepsilon} dj = 1$, substituting and rearranging,

$$\int_0^{N^*} f_j d\log p_j dj = \frac{d\log N^*}{\varepsilon - 1}.$$

Applying (F.14) gives $\alpha\gamma\, d\log(w^*/S) + \alpha(1-\gamma)d\log W^* = d\log N^*/(\varepsilon-1)$; dividing by α gives the numeraire condition with extensive margin,

$$\gamma\, d\log(w^*/S) + (1-\gamma)d\log W^* = \frac{d\log N^*}{\alpha(\varepsilon-1)}. \tag{F.15}$$

Combining. Adding and subtracting $\gamma\, d\log S$ in the output expression above,

$$d\log Y^* = \gamma\, d\log S + \gamma\, d\log(w^*/S) + (1-\gamma)d\log W^*.$$

By (F.15), the last two terms equal $d\log N^*/(\alpha(\varepsilon-1))$. Hence $d\log Y^* = \gamma\, d\log S + d\log N^*/(\alpha(\varepsilon-1))$, which is (32).

We now solve for the innovation-augmented response. The proof proceeds in three steps: the R&D optimality condition pins R^* as a constant fraction of GDP, the stationary-variety condition determines $d\log N^*$ in terms of $(d\log S, d\log Y^*)$, and the resulting system is solved jointly with (32).

Step 1: the optimality condition pins $d\log R^* = d\log Y^*$. Multiplying both sides of (27) by R^* and using the Cobb-Douglas property $G_R R = \alpha_R G$ gives

$$R^* = \beta V^* \lambda \alpha_R \delta_R G^{*\lambda} N^{*\phi}. \tag{F.16}$$

The stationary-variety condition requires new variety creation to exactly replace obsolescence. Setting $N_{t+1} = N_t = N^*$ in (26),

$$\eta N^* = \delta_R G^{*\lambda} N^{*\phi}. \tag{F.17}$$

Substituting (F.17) into (F.16) yields $R^* = \beta V^* \lambda \alpha_R \eta N^*$. Using the steady-state variety value $V^* = \pi^*/(1-\beta(1-\eta))$ from the SDF-weighted survival value (25), and the fact that, since entrants are drawn from the same type distribution as existing varieties, $\pi^* N^* = \int_0^{N^*} \pi_j^* dj \equiv \Pi^*$,

$$R^* = \frac{\beta\lambda\eta\alpha_R}{1-\beta(1-\eta)}\Pi^*.$$

Under constant markups, total profit is a constant fraction $1/\varepsilon$ of total revenue, so $\Pi^* = X^*/\varepsilon = \theta Y^*/\varepsilon$, using $X^* = \theta Y^*$ from (28). Therefore

$$R^* = \frac{\Theta\alpha_R\theta}{\varepsilon}Y^*, \qquad \text{where} \qquad \Theta \equiv \frac{\beta\lambda\eta}{1-\beta(1-\eta)}. \tag{F.18}$$

The coefficients Θ, α_R, and ε are all structural parameters that do not vary across steady states. Log-differentiating (F.18) gives

$$d\log R^* = d\log Y^*. \tag{F.19}$$

Step 2: the stationary-variety condition gives $d\log N^*$. Log-differentiating (F.17) (with η and δ_R constants),

$$d\log N^* = \lambda d\log G^* + \phi d\log N^*,$$

which rearranges to $d\log N^* = \frac{\lambda}{1-\phi}d\log G^*$. Because G is Cobb-Douglas with constant exponents and the sector-specific R&D endowments (s_R, L_R) are fixed,

$$d\log G^* = \alpha_s d\log(Ss_R) + \alpha_L d\log L_R + \alpha_R d\log R^* = \alpha_s d\log S + \alpha_R d\log R^*.$$

Substituting (F.19) gives

$$d\log N^* = \frac{\lambda}{1-\phi}\big(\alpha_s d\log S + \alpha_R d\log Y^*\big). \tag{F.20}$$

Step 3: solving the system. Equations (32) and (F.20) form a linear system in $(d\log Y^*, d\log N^*)$ given $d\log S$. Substituting (F.20) into (32),

$$\begin{aligned} d\log Y^* &= \gamma\, d\log S + \frac{1}{\alpha(\varepsilon-1)}\cdot\frac{\lambda}{1-\phi}\big(\alpha_s d\log S + \alpha_R d\log Y^*\big) \\ &= \gamma\, d\log S + \iota\alpha_s d\log S + \iota\alpha_R d\log Y^*, \end{aligned}$$

where $\iota \equiv \lambda/[\alpha(\varepsilon-1)(1-\phi)]$. Collecting $d\log Y^*$ terms,

$$(1-\iota\alpha_R)d\log Y^* = (\gamma + \iota\alpha_s)d\log S.$$

The coefficient $1-\iota\alpha_R > 0$ by the stability condition stated in Proposition 6. Dividing gives (33). □

F.9 Proof of Lemma 3

Part (i). Under constant returns and the constant markup, firm (n,j)'s total cost is $\mathscr{R}_n(j)/\mu$, of which the fraction $\alpha_n\gamma_n(j)$ is software engineering compensation. Sector SWE compensation is therefore $\frac{1}{\mu}\int_0^1 \alpha_n\gamma_n(j)\mathscr{R}_n(j)dj$ and sector total cost is $\frac{1}{\mu}\int_0^1 \mathscr{R}_n(j)dj = \mathscr{R}_n/\mu$. Their ratio is

$$\frac{\int_0^1 \alpha_n\gamma_n(j)\mathscr{R}_n(j)dj}{\mathscr{R}_n} = \alpha_n\int_0^1 \varsigma_n(j)\gamma_n(j)dj = \alpha_n\bar{\gamma}_n,$$

using that α_n is common within the sector. The markup cancels, so the sector share is the accounting ratio of SWE compensation to cost for any μ.

Part (ii). Firm (n,j)'s output reaches the economy only as a component of the sector-n composite, of which it supplies the fraction $\varsigma_n(j)$, the same fraction in every transaction, since all buyers purchase the composite. Its total direct-plus-indirect requirement per unit of final expenditure is therefore the sector requirement scaled by its share, $\tilde{\lambda}_n(j) = \tilde{\lambda}_n\varsigma_n(j)$. Hence

$$\int_0^1 \tilde{\lambda}_n(j)\alpha_n\gamma_n(j)dj = \tilde{\lambda}_n\alpha_n\int_0^1 \varsigma_n(j)\gamma_n(j)dj = \tilde{\lambda}_n\alpha_n\bar{\gamma}_n,$$

and summing over n gives part (ii). ■

F.10 Proof of Lemma 4

Throughout, $\tilde{w}_t = w_t/S_t$ is the effective SWE wage; we normalize $d\log S_t = 1$, so $u = d\log\tilde{w}_t$ and $v = d\log W_t$.

Part (i): relative price. By Shephard's lemma on the labor composite, $d\log q_n(j) = \gamma_n(j)u + (1-\gamma_n(j))v$. With the Cobb-Douglas outer nest and the constant markup,

$$d\log p_n(j) = \alpha_n\big[\gamma_n(j)u + (1-\gamma_n(j))v\big] + \sum_m \Omega_{nm}d\log P_m.$$

The composite price moves with the sales-weighted average of member prices, $d\log P_n = \int_0^1 \varsigma_n(j)d\log p_n(j)dj$, which takes the same form with $\bar{\gamma}_n$ in place of $\gamma_n(j)$. Subtracting, the intermediate-price terms cancel, and the two labor terms combine through $\alpha_n[(1-\gamma_n(j))-(1-\bar{\gamma}_n)] = -\alpha_n(\gamma_n(j)-\bar{\gamma}_n)$, giving part (i).

Part (ii): profit deviation. Because every buyer allocates across sector-n varieties with elasticity ε, firm revenue is $\mathscr{R}_n(j) = (p_n(j)/P_n)^{1-\varepsilon}\mathscr{R}_n^{\text{tot}}$, where $\mathscr{R}_n^{\text{tot}}$ is total spending on sector n. Hence

$$d\log\mathscr{R}_n(j) = (1-\varepsilon)\big(d\log p_n(j) - d\log P_n\big) + d\log\mathscr{R}_n^{\text{tot}},$$

and the second term is common to all firms in the sector. Profits are the constant fraction $1/\varepsilon$ of revenue, so $d\log\pi_n(j) = d\log\mathscr{R}_n(j)$; substituting part (i) and setting b_n equal to $d\log\mathscr{R}_n^{\text{tot}}$ net of the common expenditure level $d\log E_t$, which joins μ_t, gives (E.2). For the stacked form of b, total spending on sector n is final expenditure plus intermediate purchases, $\mathscr{R}_n = f_n E_t + \sum_m \Omega_{mn}\mathscr{R}_m/\mu$, where E_t is nominal final expenditure and intermediate purchases equal cost shares times cost $\mathscr{R}_m/\mu$. Differentiating with constant (Ω,μ) and using the upper-tier demand $d\log(f_n E_t) = (1-\zeta)(d\log P_n - \sum_m f_m d\log P_m) + d\log E_t$,

$$d\log\mathscr{R}_n = \frac{f_n E_t}{\mathscr{R}_n}\Big[(1-\zeta)\big(d\log P_n - \textstyle\sum_m f_m d\log P_m\big)\Big] + \textstyle\sum_m \mathscr{T}_{nm} d\log\mathscr{R}_m,$$

up to the common level term, with $\mathscr{T}_{nm} = \Omega_{mn}\mathscr{R}_m/(\mu\mathscr{R}_n)$. Stacking sectors and substituting the price response $d\log P = \Lambda(\alpha\bar{\gamma}u + \bar{\theta}v)$ from part (i) yields $b = \mathscr{B}(\alpha\bar{\gamma}u + \bar{\theta}v)$ with $\mathscr{B} = \Psi\mathrm{diag}(f_n E_t/\mathscr{R}_n)(1-\zeta)(I - \mathbf{1}f^\top)\Lambda$ and $\Psi = (I-\mathscr{T})^{-1}$, as stated; the common level term joins μ_t.

Part (iii): the clearing system. SWE labor clears when $w_t s = \frac{1}{\mu}\sum_n \int_0^1 \alpha_n\gamma_n(j)\mathscr{R}_n(j)dj$. Differentiating, two within-sector channels operate: the cost-share response of the labor composite, $d\log(\alpha_n\gamma_n(j)) = (1-\gamma_n(j))(1-\sigma)(u-v)$ by Shephard's lemma on the CES nest, and the revenue response of part (ii), which splits into the common b_n and the deviation $(1-\varepsilon)\alpha_n(\gamma_n(j)-\bar{\gamma}_n)(u-v)$. Aggregating each channel against SWE compensation uses the two moment identities

$$\int_0^1 \varsigma_n(j)\gamma_n(j)(1-\gamma_n(j))dj = \bar{\gamma}_n(1-\bar{\gamma}_n) - \sigma^2_{s,n}, \qquad \int_0^1 \varsigma_n(j)\gamma_n(j)\big(\gamma_n(j)-\bar{\gamma}_n\big)dj = \sigma^2_{s,n}.$$

Dividing by $w_t s$ and using $d\log s = 0$ delivers the first equation of the system, with the SWE-compensation weights ν_n; the non-SWE condition is symmetric with weights ν^L_n, the complementary share responses, and the deviation aggregated with weight $-\sigma^2_{s,n}/(1-\bar{\gamma}_n)$. Both equations are linear in (u,v), and the distributions enter only through $(\bar{\gamma}_n, \sigma^2_{s,n})$.

One-sector roundabout closed form. With $N=1$ and $\Omega_{11} = 1-\alpha$, the centering matrix $(I-\mathbf{1}f^\top)$ annihilates the single sector's price response, so $b=0$; differencing the two equations, the cost-share difference contributes $(1-\sigma)[1-\sigma^2_s/(\bar{\gamma}(1-\bar{\gamma}))](u-v)$, and the revenue deviations contribute $\alpha(1-\varepsilon)[\sigma^2_s/(\bar{\gamma}(1-\bar{\gamma}))](u-v)$. Collecting terms and using $u-v = (d\log w_t - d\log W_t) - d\log S_t$,

$$u - v = -\frac{1}{\Sigma}, \qquad \Sigma = \sigma + \big[\alpha(\varepsilon-1) - (\sigma-1)\big]\frac{\sigma^2_s}{\bar{\gamma}(1-\bar{\gamma})},$$

which is the closed form of Section 2. ■

F.11 Proof of Proposition 8

Step 1: returns. For each aggregate shock, the argument of Lemma 4 applies verbatim: the only firm-level heterogeneity inside a sector is the payroll share, and demand weights $\omega_n(Z_t)$ are sector-level objects, so a shock Z_k generates the profit response $b_n^{Z_k} + (1-\varepsilon)\alpha_n(\gamma_n(j)-\bar{\gamma}_n)(u_k - v_k)$, where (u_k, v_k) are the wage responses to a unit Z_k shock. Capitalizing profits into values and taking period-t revisions exactly as in Lemma 1 yields (E.3).

Step 2: the relative wage does not respond to Z_k. Conjecture $u_k = v_k = c$. Then every labor-composite price moves by $d\log q_n(j)=c$ uniformly, so by part (i) of Lemma 4 all within-sector relative prices are unchanged and every share response in the clearing system vanishes, as they are proportional to u_k-v_k. The wage-driven component of b also vanishes: $\mathscr{B}(\alpha\bar{\gamma}c+\bar{\theta}c)=\mathscr{B}(\alpha c)$, and $(I-\mathbf{1}f^\top)\Lambda\alpha c=(I-\mathbf{1}f^\top)\mathbf{1}c=0$. The clearing conditions therefore reduce to

$$u_k=\sum_n \nu_n b_n^{Z_k}, \qquad v_k=\sum_n \nu_n^L b_n^{Z_k},$$

where $b_n^{Z_k}$ collects the direct (non-wage) incidence of the shock. The conjecture is consistent if and only if the two right-hand sides agree, which is Assumption 5: $\sum_n(\nu_n-\nu_n^L)b_n^{Z_k}=0$. Since the system is linear with a generically unique solution, the conjectured solution is the solution, so $u_k=v_k$ and every contamination gradient $\delta_k^Z=(1-\varepsilon)\bar{\alpha}(u_k-v_k)$ is zero.

Step 3: projection. Within sector n, the return (E.3) is affine in $\gamma_n(j)$ with slope $\kappa(1-\varepsilon)\alpha_n(u-v)$ after the first-stage projection on the residualized AI-index return, the idiosyncratic component $z_n(j)$ being orthogonal to $\gamma_n(j)$. Partialling out the controls and fixed effects X_j removes the sector-common loadings b_n and $b_n^{Z_k}$. Pooled least squares then averages the within-sector slopes with the identifying weights $\varpi_n\propto$ (sample firms in n) $\times\,\mathrm{Var}_n(\gamma\mid X)$,

$$\delta=\kappa(1-\varepsilon)(u-v)\sum_n\varpi_n\alpha_n=\kappa(1-\varepsilon)\bar{\alpha}(u-v)=\kappa\delta_{\text{in}}^S,$$

which is (E.5). The no-network closed form follows from the last part of the proof of Lemma 4. ■

F.12 Proof of Proposition 9

Step 1: prices. Because S_t enters costs only through the effective SWE wage, the sector price responses propagate through the Leontief inverse: using the sector cost shares of Lemma 3,

$$d\log P_n=\sum_m\Lambda_{nm}\left[\alpha_m\bar{\gamma}_m d\log\tilde{w}_t+\bar{\theta}_m d\log W_t\right].$$

Define $\bar{\Gamma}_n=\sum_m\Lambda_{nm}\alpha_m\bar{\gamma}_m$. Since $\Lambda\alpha=\mathbf{1}$, the coefficient on $d\log W_t$ is $1-\bar{\Gamma}_n$, so $d\log P_n=\bar{\Gamma}_n d\log\tilde{w}_t+(1-\bar{\Gamma}_n)d\log W_t$. The numeraire $P_t=1$ imposes $\sum_n f_n d\log P_n=0$; with $\sum_n f_n\bar{\Gamma}_n=\tilde{s}_S$ and $d\log\tilde{w}_t=d\log w_t-d\log S_t$, this is

$$\tilde{s}_S d\log w_t+\tilde{s}_L d\log W_t=\tilde{s}_S d\log S_t. \tag{F.21}$$

Step 2: GDP accounting. Both factor supplies are fixed and $P_t=1$, so the income shares $s_{\tau,t}$ satisfy $d\log s_{S,t}=d\log w_t-d\log Y_t$ and $d\log s_{L,t}=d\log W_t-d\log Y_t$, that is, $d\log Y_t=d\log w_t-d\log s_{S,t}=d\log W_t-d\log s_{L,t}$. Taking the $\tilde{s}$-weighted average of these two identities, whose weights sum to one, and substituting (F.21),

$$d\log Y_t=\left(\tilde{s}_S d\log w_t+\tilde{s}_L d\log W_t\right)-\sum_\tau\tilde{s}_\tau d\log s_{\tau,t}=\tilde{s}_S d\log S_t-\sum_\tau\tilde{s}_\tau d\log s_{\tau,t},$$

which is (E.6) with $\tilde{s}_S = \sum_n \tilde{\lambda}_n \alpha_n \bar{\gamma}_n$.

Step 3: when the reallocation term vanishes. At $\mu = 1$, cost equals revenue, so the cost-based Domar weights solve the same fixed point as sales shares: goods-market clearing gives $\mathscr{R}_n = f_n Y_t + \sum_m \Omega_{mn} \mathscr{R}_m$, hence $\mathscr{R}_n / Y_t = \tilde{\lambda}_n$, and nominal GDP equals factor income, $Y_t = w_t s + W_t L$. Then $\tilde{s}_\tau = s_{\tau,t}$ and

$$\sum_\tau \tilde{s}_\tau d\log s_{\tau,t} = \left(s_S d\log w_t + s_L d\log W_t\right) - d\log Y_t = d\log(w_t s + W_t L) - d\log Y_t = 0,$$

using $P_t = 1$. More generally, suppose sector sales shares are proportional to the cost-based weights, $\mathscr{R}_n / Y_t = \bar{\varrho}\tilde{\lambda}_n$ for a common wedge $\bar{\varrho}$; the roundabout economy of Section 2 is of this kind, and $\Omega = 0$ with a uniform markup is the case with no intermediates. Then $s_{\tau,t} = (\bar{\varrho}/\mu)\tilde{s}_\tau$, so $\tilde{s}_\tau = s_{\tau,t}/(s_{S,t} + s_{L,t})$ and $\sum_\tau \tilde{s}_\tau d\log s_{\tau,t} = d\log[(w_t s + W_t L)/Y_t]$, which vanishes because labor income is then a constant share of GDP: $w_t s + W_t L = \sum_n \alpha_n \mathscr{R}_n / \mu = (\bar{\varrho}/\mu) Y_t \sum_n \tilde{\lambda}_n \alpha_n = (\bar{\varrho}/\mu) Y_t$. The reallocation term is therefore nonzero only insofar as the wedge varies across sectors. Away from these cases it is computed from the solved system of Lemma 4(iii), with $d\log Y_t$ obtained from the income side, including the profit response.

Step 4: the accounting identity at $\mu = 1$. With $\tilde{\lambda}_n = \mathscr{R}_n / Y_t$ and Y_t equal to labor income,

$$\sum_n \tilde{\lambda}_n \alpha_n \bar{\gamma}_n = \sum_n \frac{\mathscr{R}_n}{Y_t} \cdot \frac{\text{SWE compensation}_n}{\mathscr{R}_n} = \frac{\text{SWE compensation}}{\text{labor income}},$$

using Lemma 3(i) with cost equal to revenue. This ratio is an economy-wide accounting object, independent of the arrangement of the network.

Step 5: best linear predictor. Multiplying (E.6) by $(1-\beta)\beta^k$, summing over horizons, and applying the revision operator $(\mathbb{E}_t - \mathbb{E}_{t-1})$ gives $\mathscr{Y}_t = \gamma^{\text{IO}} \mathscr{S}_t$. By Proposition 8, $\mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}] = \kappa \tilde{R}_t^{\text{AI}}$, and linearity of the projection gives (E.7). ■