Supplemental Appendix to "The Macroeconomic Effect of AI: Sizing the Software Engineering Channel"[1]

Alex Blumenfeld Jonathon Hazell Chen Lian Andreas Schaab

This document contains supplemental appendix for the article "The Macroeconomic Effect of AI: Sizing the Software Engineering Channel." Any references to equations or sections that are not preceded by a capitalized letter refer to the main article.

A Appendix Tables

Table A.1: O*NET codes in Revelio's "Software Engineer" category

O*NET Code	Occupation Title	Compensation Share (%)
15-1252.00	Software Developers	33.15
15-1299.08	Computer Systems Engineers/Architects	10.74
15-1254.00	Web Developers	5.57
17-2199.06	Microsystems Engineers	5.15
15-1253.00	Software Quality Assurance Analysts and Testers	3.91
11-9041.00	Architectural and Engineering Managers	3.38
15-1255.00	Web and Digital Interface Designers	3.34
15-2051.00	Data Scientists	3.01
27-1029.00	Designers, All Other	2.87
15-1241.00	Computer Network Architects	2.47
15-1243.00	Database Architects	2.34
15-1299.09	Information Technology Project Managers	2.10
15-1299.07	Blockchain Engineers	1.79
15-1221.00	Computer and Information Research Scientists	1.61
15-1251.00	Computer Programmers	1.56
41-9031.00	Sales Engineers	1.16
15-1299.05	Information Security Engineers	1.06
27-1014.00	Special Effects Artists and Animators	0.99
27-1011.00	Art Directors	0.96
17-2072.01	Radio Frequency Identification Device Specialists	0.90
Top 20 Total		88.06

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 Software Engineer workers in 2021, where the weighting uses Revelio's individual-level sampling weights. These twenty occupations collectively account for 88% of total software engineer compensation. Source: Revelio Labs.

[1] 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.

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

Filter	Firms	Firms with 2021 Sales Data	Percentiles of 2021 Sales				
			P10	P25	P50	P75	P90
Initial CRSP dataset (active GVKEY links)	5,070	4,040	0.3	29.3	340.6	2,026.1	8,455.3
Drop SIC 6000-6999	4,317	3,047	6.5	49.6	488.1	2,639.4	9,843.8
Revelio match in 2021	3,290	2,828	7.3	60.0	525.7	2,881.9	10,814.5
Average sales/assets filter	2,889	2,828	7.3	60.0	525.7	2,881.9	10,814.5
Balanced panel	2,547	2,498	11.5	104.3	656.3	3,346.3	12,156.9
Drop firms in THNQ/AIQ	2,470	2,421	10.6	97.5	607.3	3,125.4	11,141.0
Drop semiconductor-related firms	2,406	2,357	10.5	97.4	607.2	3,171.4	11,191.1
Drop NAICS 51/54	1,973	1,926	9.5	95.4	737.3	3,632.2	12,571.8
Revelio size trim	1,874	1,831	12.3	110.7	825.8	3,815.6	12,861.0

Note: Sales is Compustat `sale` (net sales/revenue), measured in USD millions for fiscal year 2021. Row 1 percentiles use the unrestricted latest 2021 Compustat filing; later rows use the positive-assets/positive-sales panel used for 2021 size controls.

Table A.3: Median characteristics of firms by software engineer payroll share in the final regression sample

Characteristic	Top 400	Bottom 400
Average sales, 2017–2021 (USD millions)	896 [400]	62 [400]
Average assets, 2017–2021 (USD millions)	1,029 [400]	175 [400]
End-of-2021 market capitalization (USD millions)	2,754 [400]	341 [400]
Tobin's Q	2.17 [392]	1.65 [373]
Return on assets (percent)	7.7 [397]	2.2 [374]
Capital expenditures/assets (percent)	1.7 [397]	1.7 [374]
Market leverage (percent)	8.6 [394]	11.6 [373]
Total employment	2,008 [400]	115 [400]
Average annual compensation (USD thousands)	108.5 [400]	129.8 [400]

Note: The sample is the balanced 2022–2025 final regression sample after all sample exclusions and the Revelio employment-size trim. Top 400 contains the 400 firms with the highest software engineer payroll shares. Bottom 400 contains the 400 firms with the lowest shares, including both zero and positive shares. Entries are unweighted medians. Bracketed values report the row-specific number of firms with nonmissing data.

Table A.4: 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	-0.05	0.00	-0.01	0.00	-0.10
	(0.05)	(0.06)	(0.06)	(0.06)	(0.09)
Quintile 2	-0.11**	-0.02	-0.02	-0.04	-0.10
	(0.05)	(0.06)	(0.06)	(0.06)	(0.09)
Quintile 3	-0.05	0.04	0.06	0.00	-0.07
	(0.05)	(0.06)	(0.06)	(0.06)	(0.09)
Quintile 4	0.00	0.09	0.12**	0.01	-0.06
	(0.05)	(0.06)	(0.06)	(0.06)	(0.09)
Quintile 5	0.19***	0.26***	0.28***	0.15**	0.09
	(0.05)	(0.06)	(0.06)	(0.06)	(0.09)
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	1,855	1,855	1,855	1,850	1,144
R^2	0.04	0.07	0.10	0.15	0.22
Within R^2		0.04	0.04	0.01	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 Table 1. Heteroskedasticity-robust standard errors are reported in parentheses. Observations are firms.

Table A.5: Sensitivity of the baseline coefficient to outlier treatment

Dependent Variable:	Beta on THNQ excess return					
Specification	Baseline (trim top 1%)	Drop firms with $\gamma_j = 0$	No outlier handling	Trim top 5%	Winsorize top 1%	Winsorize top 5%
Model:	(1)	(2)	(3)	(4)	(5)	(6)
Variables						
Software engineer payroll share	1.24***	1.25***	1.30***	1.20***	1.35***	1.55***
	(0.26)	(0.27)	(0.23)	(0.37)	(0.24)	(0.29)
Fixed-effects						
NAICS3	Yes	Yes	Yes	Yes	Yes	Yes
Quintile of average 2017-2021 assets	Yes	Yes	Yes	Yes	Yes	Yes
Quintile of average 2017-2021 sales	Yes	Yes	Yes	Yes	Yes	Yes
Fit statistics						
Observations	1,850	1,667	1,869	1,775	1,869	1,869
R^2	0.15	0.17	0.16	0.13	0.16	0.16
Within R^2	0.01	0.02	0.02	0.007	0.02	0.02

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 1; Column (2) drops firms with software engineer payroll shares of zero; Column (3) does not trim the software engineer payroll share at all; Column (4) trims at the 95th percentile; and Columns (5) and (6) winsorize at the 99th and 95th percentiles, respectively. All other controls, fixed effects, and the dependent variable are held fixed across columns. Heteroskedasticity-robust standard errors are reported in parentheses. Observations are firms.

Table A.6: 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	2.03	0.61	14.0	1.56
1.0	1.27	0.98	22.5	2.49
1.5	0.92	1.35	30.9	3.43
1.6 (baseline)	0.87	1.42	32.6	3.61
2.0	0.72	1.71	39.3	4.36

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: $\gamma \approx 0.111$, $\sigma_\gamma^2 \approx 0.0116$, $\alpha = 0.372$, and $\delta = 1.24$. 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, $\kappa = 1.42$ means that a residualized AI-index return of 1% predicts software engineering productivity news of 1.42%. 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 $\sum_{t=\text{'22 Nov}}^{\text{end '25}} \tilde{R}_t^{\text{AI}} = 22.9\%$. Each cell uses the no-IO formula from Section 2.

Table A.7: Robustness of market-implied software engineering productivity news

	Nov 2022–Dec 2025	
Specification	$\sum_t \tilde{R}_t^{\text{AI}}$	$\sum_t \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}_t^{\text{AI}}]$
FF3 (baseline), fixed	**22.9%**	**32.6%**
FF3, rolling 104-week	21.4%	30.5%
Market only, fixed	9.6%	13.7%
FF5, fixed	16.5%	23.4%
FF3, ex-Mag 7 market, fixed	38.3%	54.4%

Notes: In the fixed specifications, the factor loadings used to residualize the AI-index returns are estimated over the fixed window from January 2022 through December 2025 used in our main analysis. In the rolling specification, the factor loadings are estimated over the preceding 104 weeks, excluding the current week. The first outcome column reports cumulative factor-adjusted excess returns on the AI index. The second outcome column multiplies those returns by the corresponding productivity multiplier κ.

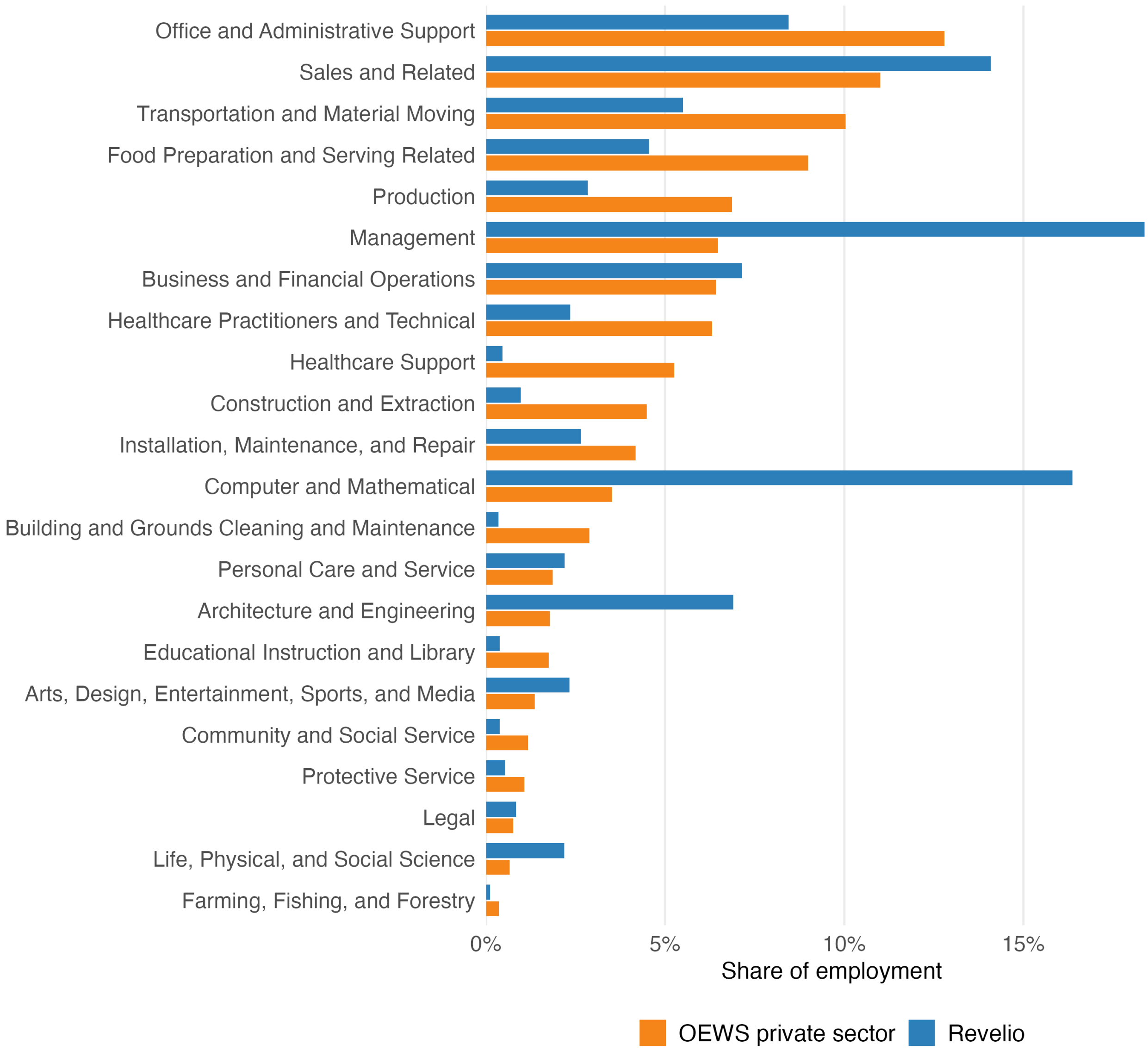
Table A.8: Robustness of market-implied GDP news to alternative calculations of γ

Row	Method	$\bar{\gamma}$	σ^2_γ	$\sum_t \mathbb{E}^*[\mathscr{S}_t \mid \tilde{R}^{\mathrm{AI}}_t]$ (%)	$\sum_t \mathbb{E}^*[\mathscr{Y}_t \mid \tilde{R}^{\mathrm{AI}}_t]$ (%)
1	Revelio: baseline	0.1110	0.0116	32.6%	3.61%
2	Revelio: reweight sectors using BEA gross output, exclude unmatched sectors	0.0929	0.0096	32.5%	3.02%
3	Revelio: reweight sectors using BEA gross output, fill unmatched sectors' values with zeros	0.0745	0.0090	32.8%	2.44%
4	OEWS: Computer Occupations	0.0528	0.0088	33.6%	1.77%
5	Revelio: all roles mapped to SOC codes for Computer Occupations	0.1542	0.0138	32.4%	4.99%

Notes: This table varies the measurement of the average software engineering payroll share $\bar{\gamma}$ and its cross-sectional variance σ^2_γ. All rows hold fixed the remaining baseline inversion inputs, including $\alpha = 0.372$, $\varepsilon = 5$, $\sigma = 1.6$, the empirical slope $\delta = 1.240$, and the cumulative residualized AI-index return of 22.9%. The last two columns show the implied software engineer productivity impact and GDP impact under each alternative methodology for calculating γ and σ^2_γ.
Row 1 uses our baseline Revelio methodology for computing γ: a sales-weighted average of firm payroll shares. In Rows 2 and 3, we take sales-weighted averages of firms' payroll shares within each industry, and then industry means are averaged using gross output weights for private industries from the BEA production accounts. Industries without any firms in our Revelio dataset are either excluded (with the weights then renormalized) or assigned zero payroll shares. Row 4 computes the economy-wide software engineering payroll share using OEWS data: each industry's numerator is its total wage bill for Computer Occupations (all SOC codes beginning with 15-12), the denominator is the industry's total wage bill, and industry payroll shares are averaged using BEA gross output weights. Row 5 again computes software engineer payroll shares in Revelio, but the numerator includes all Revelio employees whose job titles are mapped to SOC codes under Computer Occupations by Revelio's crosswalk, rather than using the Revelio-defined Software Engineer group.

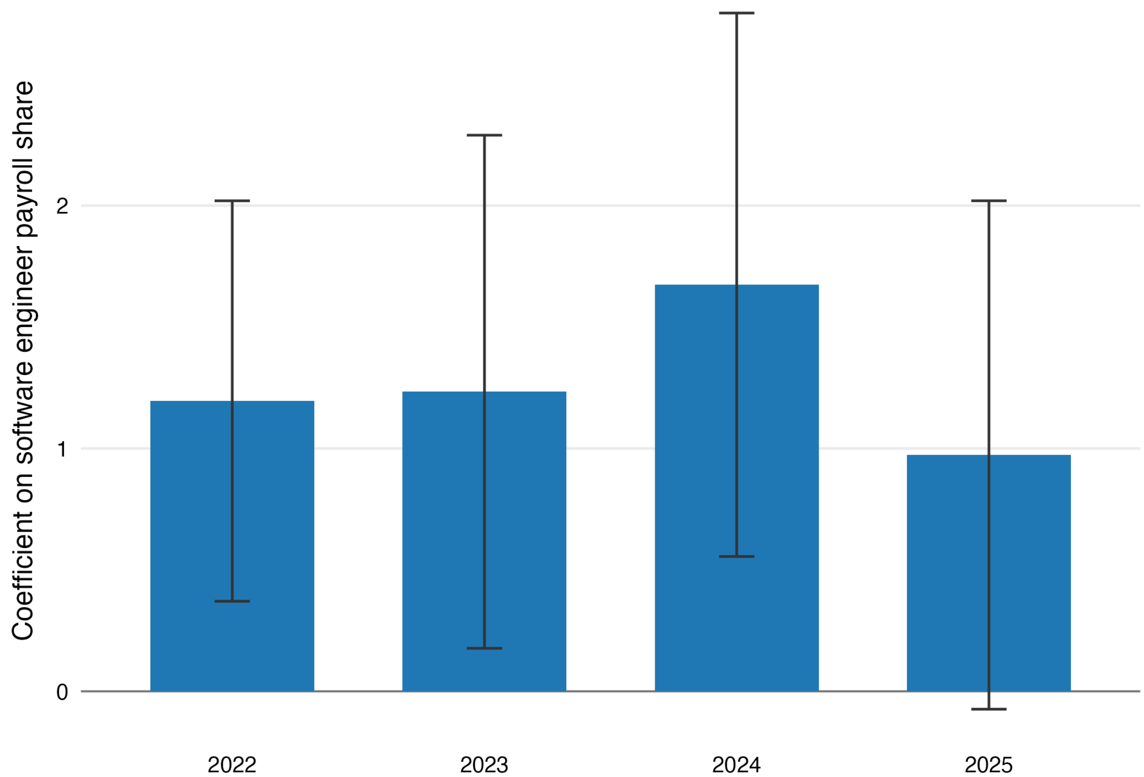
B Appendix Figures

Figure B.1: Employment shares of major occupation groups in Revelio data vs. OEWS



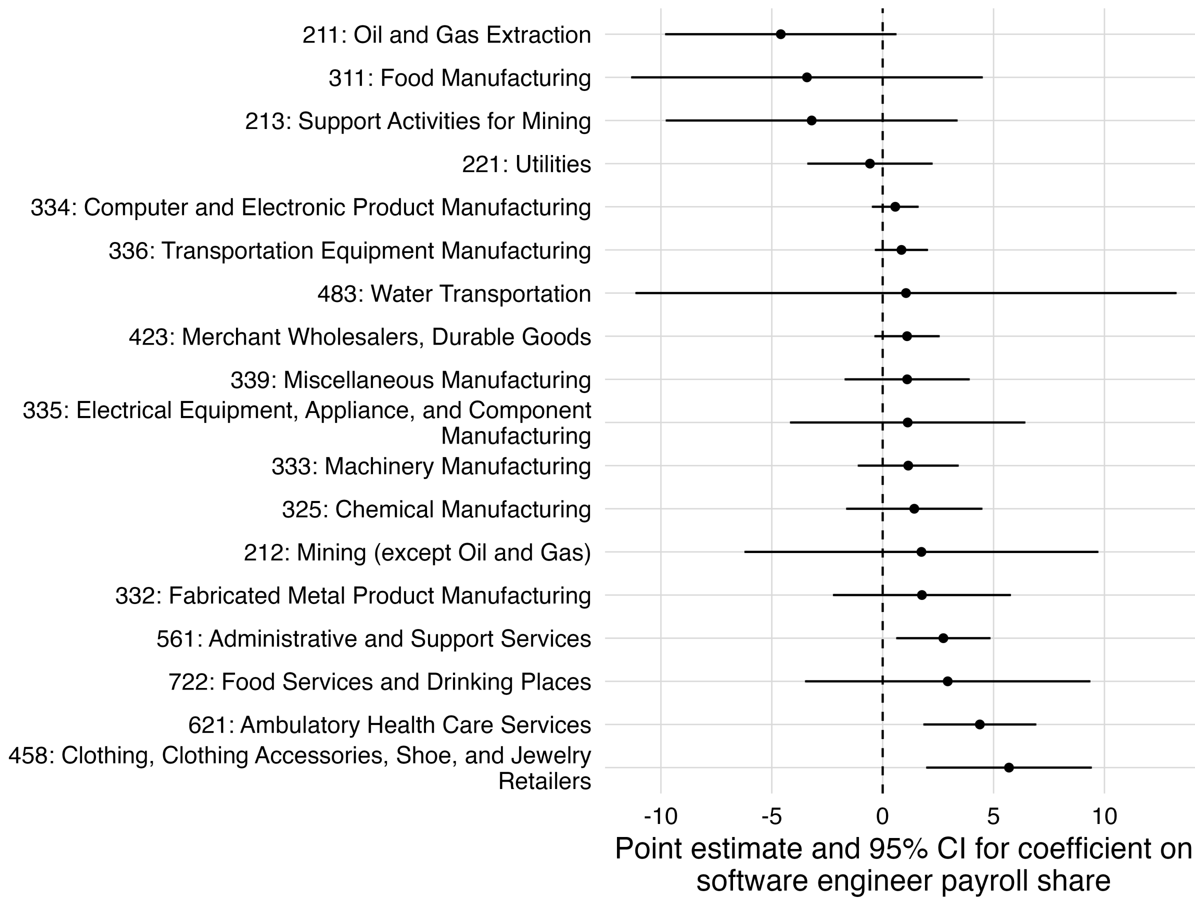
Notes: The blue bars show the 2021 employment shares of high-level SOC occupations in the set of firms retained at the "Balanced panel" stage of our sample selection process described in Table A.2, using Revelio's provided crosswalk to map from Revelio's own occupation classification to SOC codes. The orange bars show occupation shares at private establishments in OEWS data from May 2021. 98.8% of employees in our Revelio dataset are in occupation categories that map to a valid non-military SOC code.

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



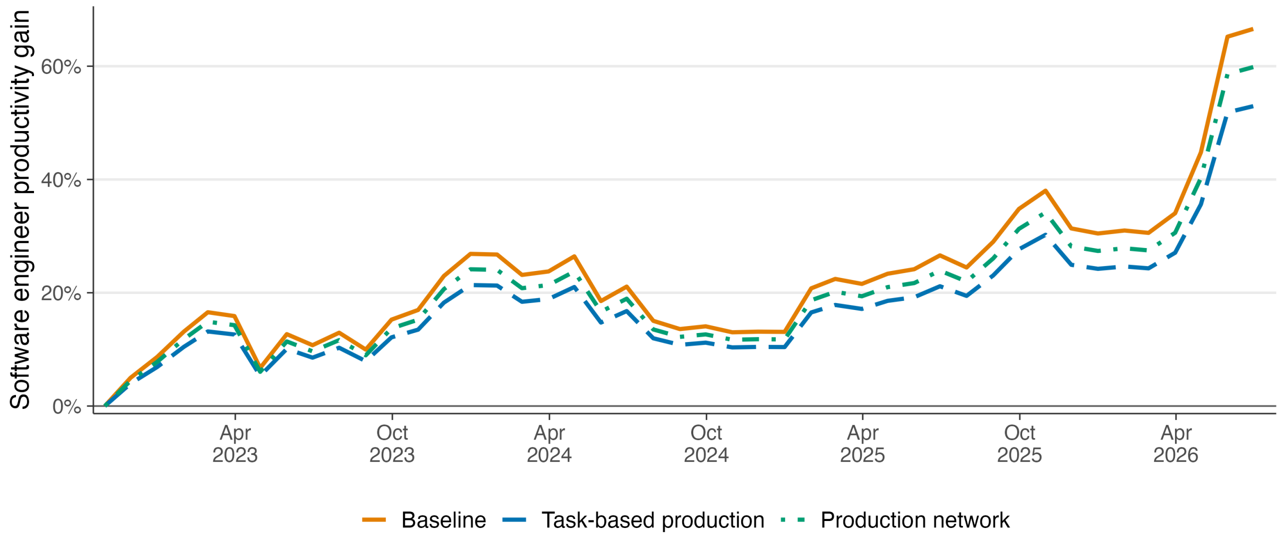
Notes: Each bar reports the estimated coefficient on the software engineering payroll share from the second-stage specification corresponding to Column (4) of Table 1, where the first-stage AI betas are estimated using data from a single calendar year only. Whiskers show 95% confidence intervals based on heteroskedasticity-robust standard errors.

Figure B.3: Second-stage regression results: industry heterogeneity

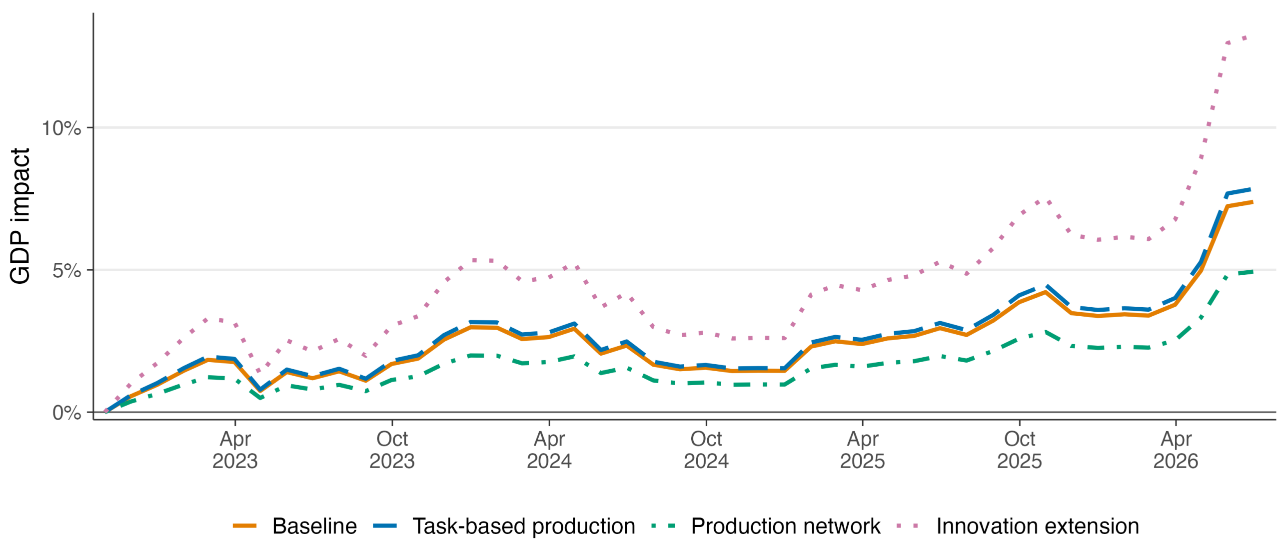


Notes: This figure shows point estimates and 95% confidence intervals from second-stage regressions run one NAICS3 sector at a time. The outcome is the firm-level beta on THNQ excess returns, and the main regressor is the software engineer payroll share, trimmed at the 99th percentile as in Table 1. 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 heteroskedasticity-robust standard errors.

Figure B.4: Market-implied software engineering productivity and GDP news in real time



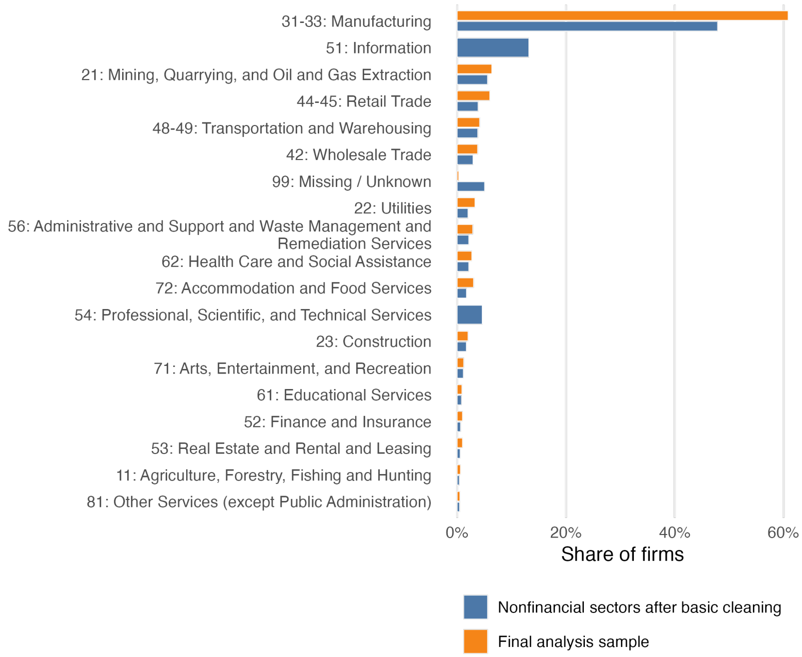
(a) Implied software engineer productivity gain



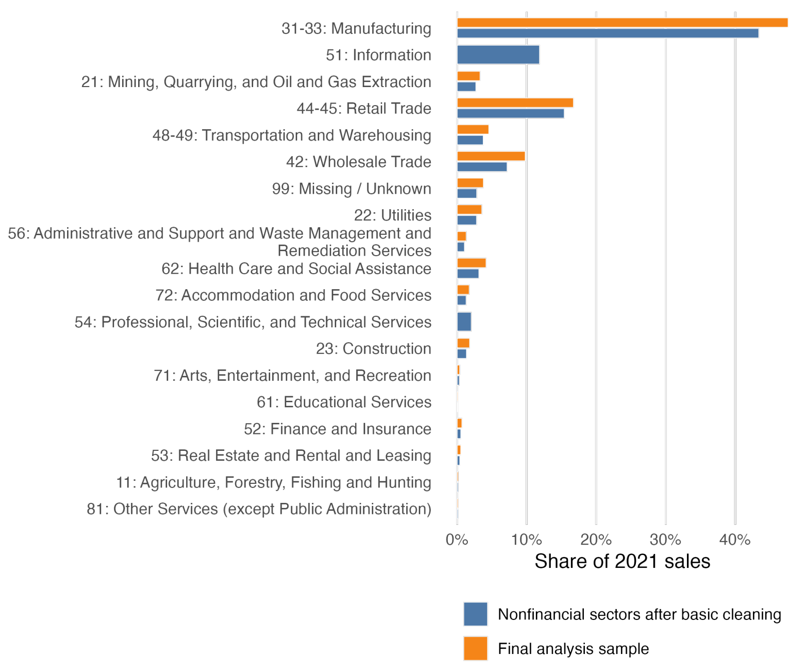
(b) Implied GDP impact

Notes: Panel (a) reports $\kappa \sum_t \tilde{R}_t^{\mathrm{AI}}$ at each month end, where κ is the productivity multiplier for each model. Panel (b) reports the corresponding cumulative GDP news. Weekly AI-index excess returns are residualized using Fama–French three-factor loadings estimated over the preceding 104 weeks, excluding the current week, and cumulated from November 2022. The innovation extension shares the baseline productivity multiplier and therefore appears only in panel (b).

Figure B.5: Sectoral composition of the analysis sample



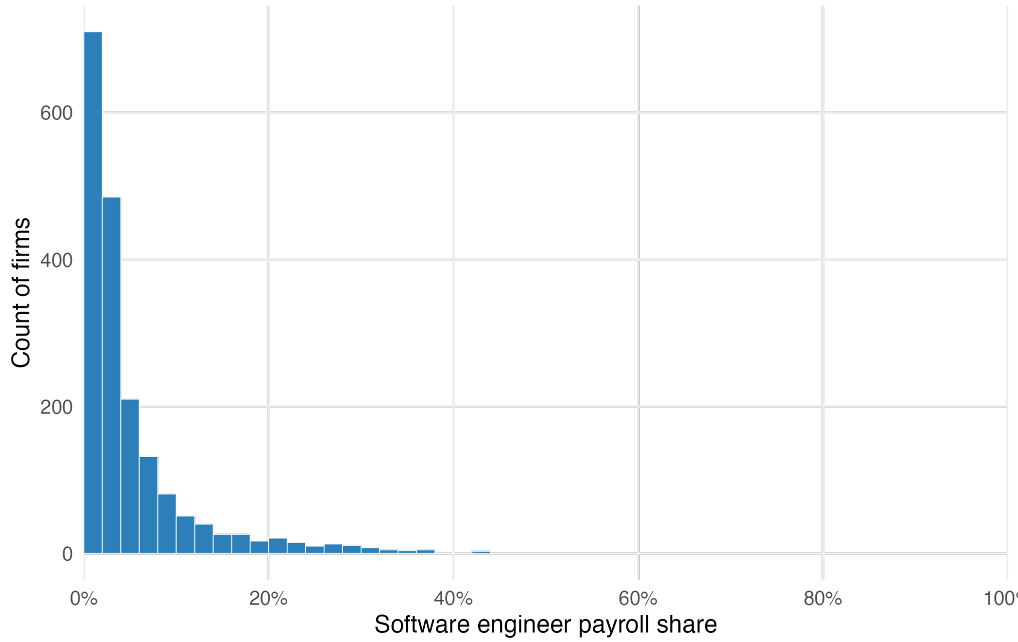
(a) Share of firms



(b) Share of 2021 sales

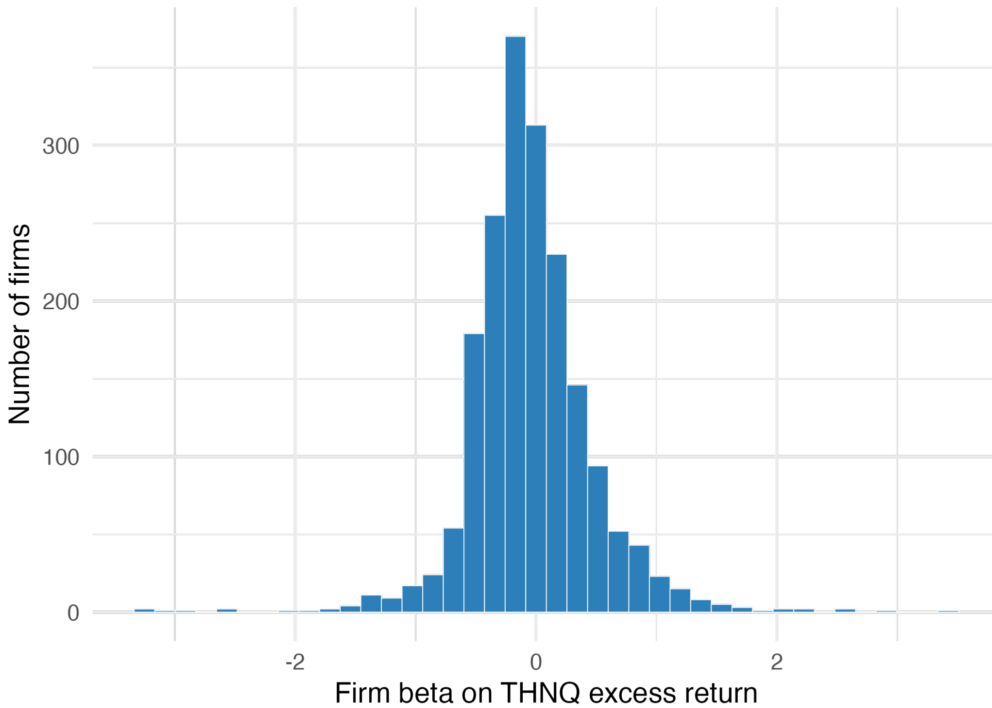
Notes: Panel (a) shows the share of firms in each NAICS2 sector; panel (b) shows the share of aggregate 2021 sales. The light bars show the distribution after the "Drop SIC 6000-6999" stage of sample selection in Table A.2. The dark bars show the distribution in the final analysis sample after the "Revelio size trim" row of Table A.2.

Figure B.6: Distribution of software engineer payroll shares



Notes: This figure shows the distribution 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 (185 firms) and a long right tail.

Figure B.7: Distribution of firm-level AI betas (baseline specification)



Notes: This figure shows the distribution 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.

C Data Appendix

C.1 Testing Assumption 1, Part 2

In this section, we explicitly test the second condition of Assumption 1: $\mathrm{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$, 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 toward 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 $\mathrm{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$ 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 $\hat{C}_f \equiv \widehat{\mathrm{Cov}}_{f,j}(\omega'_j, \gamma_j)$. In bootstrap iteration r, we draw a multiplier v_n^r for each NAICS2 sector n, then construct a simulated software engineer payroll share for each firm j under the null: $\gamma_j^r \equiv \gamma + v_{n(j)}^r \hat{u}_j$, where γ is the final-expenditure-weighted mean of γ_j and $\hat{u}_j = \gamma_j - \gamma$. Then, we compute $\hat{C}_f^r \equiv \widehat{\mathrm{Cov}}_{f,j}(\omega'_j, \gamma_j^r)$ for each iteration, and we label the standard deviation of this object across bootstrap iterations as $\widehat{\mathrm{se}}(\hat{C}_f)$. Finally, we form our test statistic as $\hat{C}_f / \widehat{\mathrm{se}}(\hat{C}_f)$ 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
$\text{Cov}_{f,j}(\omega'_j, \gamma_j)$	0.0159
Bootstrap s.e.	0.0112
95% confidence interval	[-0.0060, 0.0379]
p-value	0.155
Final-expenditure-weighted mean of ω'_j	0.0953
Final-expenditure-weighted mean of γ_j	0.1155
Number of firms	1,319
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 0.155, the bootstrap fails to reject the null that $\text{Cov}_{f,j}(\omega'_j, \gamma_j) = 0$. 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 2. 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 1, 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 1, as well as the following variables from Table 2: log(market cap) from row 9; four of the financial controls from row 8, namely book-to-market, operating profitability, investment, and book leverage; HQ state FE from row 10; the Generative AI product-market exposure measures in row 4; and the payroll shares of four key non-software-engineer occupations in row 11. 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	1.38	1.90	0.24	0.35	0.11	809	803
Within	1.38	1.90	0.03	0.10	0.07	809	803

Panel B. Sensitivity of Oster-implied coefficient

R^2 type	Assumed Oster's δ	$R_{\text{max}} = 1$	$R_{\text{max}} = 1.3R_1$	$R_{\text{max}} = 1.5R_1$
Overall	0.50	3.44	2.15	2.31
	1.00	4.97	2.40	2.73
	2.00	8.04	2.89	3.55
Within	0.50	5.33	2.02	2.09
	1.00	8.75	2.13	2.28
	2.00	15.60	2.36	2.67

Notes: Panel A shows results for the "baseline" regressions from column (4) of Table 1 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 1.

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, selection on unobservables would have to work in the opposite direction from selection on observables.

C.3 Coverage of Employment in Revelio Data

In this section, we use sector-by-occupation employment data from OEWS to account for possible selection in which kinds of workers have profiles observed in the Revelio data. This addresses concerns about Revelio's coverage raised by Table A.8: in OEWS data, the national payroll share of Computer Occupations is 5.3%, compared to our Revelio sales-weighted payroll share for Software Engineers of 11.1% (note that this is not a perfect comparison due to different weights and exact occupation definitions). Although Revelio does provide sampling weights at the occupation-by-country level that aim to correct for the fact that workers in some occupations are more likely to have online profiles than others, we still find that our sample of Compustat firms over-represents workers in Computer and Mathematical Occupations, as shown by Figure B.1. To deal with this, we consider an extra correction in the construction of firms' software engineer payroll shares that attempts to mitigate these remaining coverage concerns.

The idea is to compute a scaling factor a_{no} for each sector n and occupation o, which captures the extent to which occupation o is over- or under-represented in sector n's total employment in Revelio relative to OEWS. Then, we will multiply each worker's Revelio-provided weight by their corresponding a_{no} when building firm-level software engineer payroll shares: if Revelio says, for instance, that Software Engineers have a higher employment share in sector n than OEWS does, then we will have $a_{no} < 1$ for Software Engineers. This means that we will down-weight any Software Engineers we observe in sector n, because we are concerned that this occupation is over-represented in Revelio data for firms in that sector. The construction is as follows:

$$a_{no} = \frac{s_{no}^{OEWS}}{s_{no}^{R}}$$

where s_{no}^{OEWS} is the employment share of o in n from OEWS data, and s_{no}^{R} is the employment share of o in n in Revelio data, after applying Revelio's provided weights.

For each high-level Revelio occupation and NAICS sector, we compute s_{no}^{R} using all US employment from 2021, based on firms' NAICS codes recorded in Revelio. We build s_{no}^{OEWS} using May 2021 OEWS data for private industries. Each six-digit SOC code in the OEWS data has its OEWS employment allocated proportionally across the corresponding Revelio occupations identified by Revelio's provided crosswalk, using the observed employment shares of Revelio occupations among all Revelio employment linked to that SOC code in the crosswalk.

We consider three options for this reweighting procedure: one where we estimate the "scaling factor" a_{no} separately for each NAICS2 sector n and Revelio occupation o, one where we use NAICS3 sectors instead, and one where we pool across industries to generate a single factor a_o for each occupation. The NAICS3-based version is more flexible because it corrects for patterns of selection into having an online profile that may vary across industries, such as if Software Engineers in the Telecommunications sector are more likely to have online profiles than Software Engineers in the Apparel Manufacturing sector. In contrast, the version that pools across sectors is less sensitive to noisy employment data in small NAICS3-by-occupation cells. As a middle ground, we report results for the NAICS2-based version in Table 2.

In the specification where we pool across industries, the estimated scaling factor for Software Engineers is $a_o \approx 0.522$: even after applying Revelio's provided sampling weights, the national employment share of Software Engineers is almost twice as high in Revelio as in the OEWS data, so we down-weight each Software Engineer observed in Revelio by almost 50% when computing firms' software engineer payroll shares. On the other hand, the scaling factor for Healthcare Providers is $a_o \approx 1.28$, meaning that those workers are under-represented in the Revelio data.

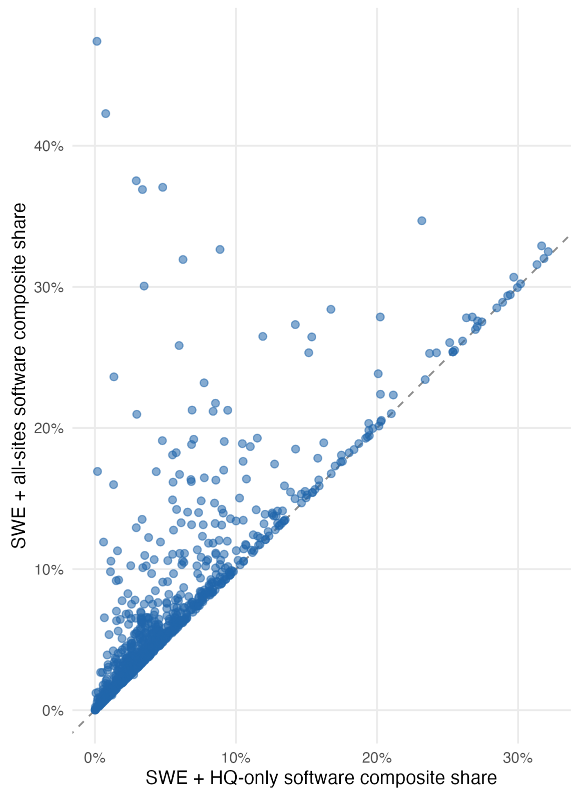
As shown in Table 2, our second-stage regression coefficient actually increases from 1.24 to 1.38 when we apply the NAICS2-based scaling factors to each firm's payroll data. The estimates with the NAICS3-based and pooling-industries versions of a_{no} are 1.15 and 1.55, respectively, and are both still highly significant. So, although applying these corrections implies a somewhat lower economy-wide software engineer payroll share γ compared to the value used in the main text (as suggested by Figure B.1), the cross-sectional relationship between firms' AI betas and their software engineer payroll shares is highly robust to this change in measurement.

C.4 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. In all results reported in Section 4.2, when we combine the Harte-Hanks software budget data with our main dataset of firms' software engineering payroll shares (which are from 2021), we multiply each year's software budgets by the growth of CPI from the corresponding year to 2021, to ensure that software expenditure and software engineer payroll have consistent units.

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 budgets sums site-level budgets across all of a firm's establishments, averaged over all years between 2012 and 2015 for which the firm has non-missing software budget information. As discussed in Section 4.2, in our regressions, potential concerns about measurement error lead us to instrument for γ_j^T with an alternate version that includes only HQ-location software budgets, for the same reasons that De Ridder (2024) focuses on HQ locations when measuring firms' software budgets using this data. Figure C.2 plots the "all-establishments software" version of γ_j^T against the "headquarters-only software" version (the instrument), demonstrating that the relationship is reasonably linear.

Figure C.2: Scatterplot of γ_j^T values computed using all-establishments vs. headquarters-only software budgets



Notes: The y-axis plots values of γ_j^T as defined in Section 4.2, and the x-axis plots values of the corresponding variable produced by replacing "Total software budget" in the numerator and denominator with the software budget of the firm's headquarters location. Each variable is trimmed at the 99th percentile within the subset of firms in our final analysis sample that have non-zero values for both. This figure is conceptually similar to the first stage of the IV procedure used to estimate δ^T in Section 4.2, although for simplicity, it plots the raw values of the LHS and RHS variables, without residualizing on the controls used in the regression; the corresponding residualized scatterplot looks similar.

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

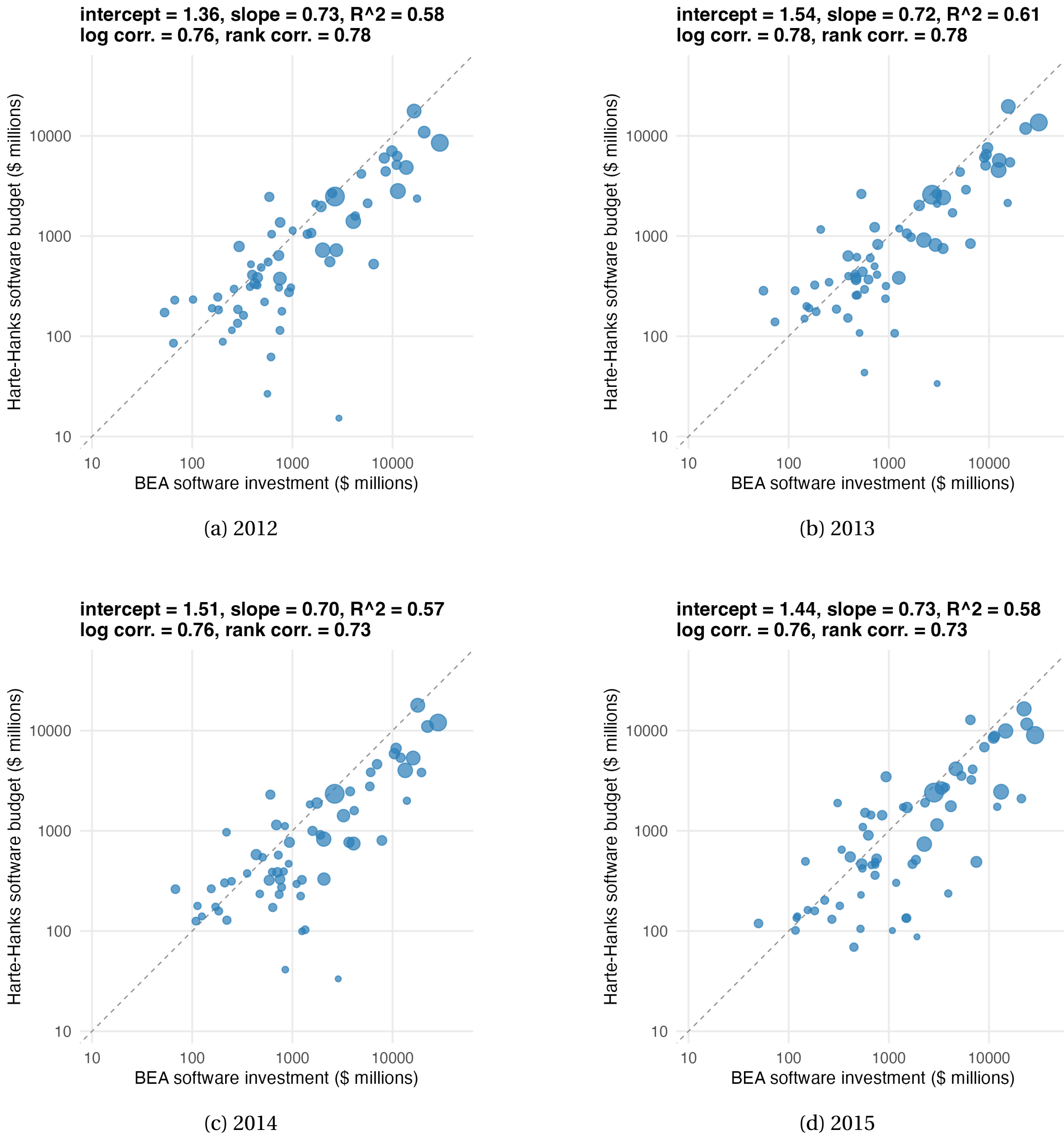
Year	H-H Software Budget ($bn)	BEA Software Investment ($bn)	Ratio
2012	114.4	221.8	0.52
2013	131.1	233.5	0.56
2014	127.5	248.0	0.51
2015	152.9	257.3	0.59

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.3. 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.3: Cross-Industry Comparison of Harte-Hanks and BEA Software Spending



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.

C.5 LLM Prompt for Classifying Firms as Semiconductor-Related

I have attached `firms_for_semiconductor_classification.csv`, a CSV list of firms with their Compustat GVKEY, ticker, company name, NAICS codes/names, end-of-2021 market cap, and Compustat business description.

Please return a CSV named `firms_with_semiconductor_classification_gpt56_20260823.csv` that preserves every original row and column, and adds:

1. `semiconductor_related`: 1 if the firm is directly part of the semiconductor production supply chain, 0 otherwise.
2. `semiconductor_category`: short label such as chip_design, chip_manufacturing, fabless_semiconductor, semiconductor_equipment, semiconductor_materials, EDA_IP, assembly_testing_packaging, or none.
3. `semiconductor_reason`: one or two sentences explaining the classification.

Use `semiconductor_related = 1` for firms whose business description or industry classification indicates that they:
- design, manufacture, fabricate, package, test, or sell semiconductor devices;
- are fabless semiconductor companies;
- make semiconductor manufacturing equipment;
- produce specialized materials, wafers, photomasks, gases, chemicals, or other direct inputs for semiconductor fabrication;
- provide EDA software, semiconductor IP, or design tools used primarily in chip design.

Use `semiconductor_related = 0` for firms that are only downstream users of semiconductors; do not classify a firm as semiconductor-related solely because it uses chips, sells AI products, operates data centers, sells electronic products, or is in a broad computer/electronics NAICS sector. For solar/PV companies, use 0 unless the description clearly indicates semiconductor-device production beyond solar panels/modules.

When evidence is ambiguous, lean towards `semiconductor_related = 1` only if the firm appears to be a direct supplier to semiconductor production rather than a downstream technology user. Keep the output row order unchanged.

D Proofs

D.1 TFP through the Software Engineering Channel

Step 1: relate TFP to GDP. Proposition 1 of Baqaee and Farhi (2020) defines aggregate TFP changes as GDP changes net of raw-factor changes weighted by cost-based factor Domar weights:

$$d\log \mathrm{TFP}_{t+k}\big|_S = d\log Y_{t+k}\big|_S - \tilde{\Lambda}_s d\log s - \tilde{\Lambda}_L d\log L.$$

Both raw primary-factor endowments are fixed, so $d\log s = d\log L = 0$. Using (45),

$$d\log \mathrm{TFP}_{t+k}\big|_S = d\log Y_{t+k}\big|_S = \gamma d\log S_{t+k}.$$

Step 2: pass from flow effects to news. Applying the normalized present-value news operator period by period gives

$$\begin{aligned}
\mathscr{T}_t &= (1-\beta)(\mathbb{E}_t - \mathbb{E}_{t-1})\Big[\sum_{k=0}^{\infty} \beta^k d\log \mathrm{TFP}_{t+k}\big|_S\Big] \\
&= (1-\beta)(\mathbb{E}_t - \mathbb{E}_{t-1})\Big[\sum_{k=0}^{\infty} \beta^k d\log Y_{t+k}\big|_S\Big] \\
&= \mathscr{Y}_t = \gamma \mathscr{S}_t.
\end{aligned}$$

D.2 Endogenous CES Representation of the Task Price Index

This appendix derives the closed form of the software price index (19) stated in Section 4.2, repeated here for convenience:

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

with productivity indices $B_A(I_t) = \int_0^{I_t} \psi_A(z)^{\eta-1} dz$ and $B_S(I_t) = \int_{I_t}^1 \psi_S(z)^{\eta-1} dz$.

Derivation of (19). The CES task aggregator $T_{jt} = [\int_0^1 y_{jt}^T(z)^{(\eta-1)/\eta} dz]^{\eta/(\eta-1)}$ 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 (18). 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 (19). 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. □

D.3 Proof of Proposition 4

Proof. We work to first order around the steady state and drop time subscripts on steady-state objects. We assume that the steady-state price ratio w/p^A lies strictly inside the range of $\psi_S(I)/\psi_A(I)$ for $I \in [0,1]$, so the equilibrium threshold satisfies $I \in (0,1)$. We restrict attention to sufficiently small perturbations that preserve interiority.

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 use of the final-good composite (consumption plus materials) and C_t aggregate consumption, which equals real GDP in this economy. 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 total use, $y_{jt} = \omega_j(p_{jt}/P_t)^{-\varepsilon}Q_t$.

As in Section 2, the time-t final-expenditure shares are $f_{jt} = \omega_j(p_{jt}/P_t)^{1-\varepsilon} = p_{jt}c_{jt}/C_t$, which integrate to one; the sales-Domar weights $\lambda_{jt} = p_{jt}y_{jt}/C_t$ equal θf_{jt}.[2] Write $\gamma_t^T \equiv \int_0^1 f_{jt}\gamma_{jt}^T dj$ for the time-t final-expenditure-weighted software-composite cost share, and let γ^T denote its steady-state value. Constant markups make each producer's cost the fraction $\frac{\varepsilon-1}{\varepsilon}$ of its revenue, with the input bundle H_{jt} accounting for the fraction α of cost and materials the fraction $1-\alpha$; the software supplier and the AI sector are competitive. Integrating input bills across producers at time t and splitting the software bill into its SWE and compute parts, in proportions $1-\rho_t$ and ρ_t, gives the factor-income identities

$$w_t s = \tfrac{\varepsilon-1}{\varepsilon}(1-\rho_t)\alpha\gamma_t^T Q_t, \qquad p_{K,t}K = \tfrac{\varepsilon-1}{\varepsilon}\rho_t\alpha\gamma_t^T Q_t, \qquad W_t L = \tfrac{\varepsilon-1}{\varepsilon}\alpha(1-\gamma_t^T)Q_t. \tag{D.1}$$

Adding materials spending $\frac{\varepsilon-1}{\varepsilon}(1-\alpha)Q_t$ and profit $\frac{1}{\varepsilon}Q_t$ recovers gross output Q_t; netting out materials

[2] Unlike in Section 2, the f_{jt} are no longer the firms' value-added shares: firm value added here also nets out purchased software, and the SWE payroll and the compute rents are the value added of the software supplier and the AI sector. The proof uses these shares only as final-expenditure weights.

leaves nominal GDP $w_t s + p_{K,t}K + W_t L + \frac{1}{\varepsilon}Q_t = C_t$, of which the compute rent $p_{K,t}K$ is a component.

Step 1: the software price (envelope argument). Differentiating (19) with respect to (w_t, p_t^A, I_t),[3] 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 $\left[(p_t^A)^{1-\eta}\psi_A(I_t)^{\eta-1} - w_t^{1-\eta}\psi_S(I_t)^{\eta-1}\right]dI_t$, which the indifference condition (18) 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

$$d\log p_t^T = (1-\rho)d\log w_t + \rho d\log p_t^A, \qquad p_t^A = \frac{p_{K,t}}{S_t^A}, \tag{D.2}$$

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 (20) and (D.2) gives

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

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, $0 = \int_0^1 f_j d\log mc_{jt} dj$, hence

$$\alpha d\log W_t - d\log A_t = \alpha\gamma^T x_t^T. \tag{D.3}$$

Subtracting,

$$d\log\pi_{jt} = (\varepsilon-1)\alpha(\gamma_j^T - \gamma^T)x_t^T + d\log C_t. \tag{D.4}$$

The Hicks-neutral shock A_t cancels from x_t^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 engineering productivity is the sole source of cross-sectional variation.

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

(i) Numeraire. Because $d\log A_t = 0$, equation (D.3) reduces to $\alpha d\log W_t = \alpha\gamma^T x_t^T$. Dividing by α yields $d\log W_t = \gamma^T x_t^T$.

(ii) The displacement elasticity. Differentiating the time-t compute share $\rho_t = B_A(I_t)(p_t^A)^{1-\eta}/(p_t^T)^{1-\eta}$, with the threshold moving through the indifference condition (18), and evaluating at the steady state gives

$$\frac{d\rho_t}{\rho(1-\rho)} = (\eta^\star - 1)(d\log w_t - d\log p_t^A), \qquad \eta^\star = \eta + \frac{\nu}{\vartheta\rho(1-\rho)} > \eta, \tag{D.5}$$

where $\vartheta = \frac{d}{dI}\log\frac{\psi_S(I)}{\psi_A(I)}$ is the slope of comparative advantage at the margin and $\nu = (w/\psi_S(I))^{1-\eta}/(p^T)^{1-\eta}$

[3] Throughout this appendix, we assume that ψ_A and ψ_S are positive and continuous on $[0,1]$ and differentiable in a neighborhood of any equilibrium threshold, with $0 < \frac{d}{dI}\log\frac{\psi_S(I)}{\psi_A(I)} < \infty$ at the threshold.

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 (D.1) imply

$$\begin{aligned} d\log w_t &= -\frac{d\rho_t}{1-\rho} + d\log\gamma_t^T + d\log C_t, \\ d\log p_{K,t} &= \frac{d\rho_t}{\rho} + d\log\gamma_t^T + d\log C_t, \\ d\log W_t &= -\frac{d\gamma_t^T}{1-\gamma^T} + d\log C_t. \end{aligned} \tag{D.6}$$

Subtracting the first two relations in (D.6) gives the incidence identity $d\log w_t - d\log p_{K,t} = -d\rho_t/[\rho(1-\rho)]$: with both factors fixed, the relative price absorbs the share movement one-for-one. Combining with (D.5) 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,

$$d\rho_t = \rho(1-\rho)\frac{\eta^\star - 1}{\eta^\star} d\log S_t^A, \qquad d\log\frac{w_t}{p_{K,t}} = -\frac{\eta^\star - 1}{\eta^\star} d\log S_t^A. \tag{D.7}$$

(iv) Reallocation and solving for x_t^T. CES demand and (D.4) give $d\log f_{jt} = (\varepsilon-1)\alpha(\gamma_j^T - \gamma^T)x_t^T$, and the CES payroll-share formula gives $d\gamma_{jt}^T = (\sigma-1)\gamma_j^T(1-\gamma_j^T)x_t^T$; these combine exactly as in the benchmark in the proof of Lemma 1 to

$$d\log\gamma_t^T = (1-\gamma^T)(\Sigma^T - 1)x_t^T, \qquad \Sigma^T = \sigma + \big[\alpha(\varepsilon-1) - (\sigma-1)\big]\frac{(\sigma_\gamma^T)^2}{\gamma^T(1-\gamma^T)}. \tag{D.8}$$

Combining (D.6) with the envelope (D.2) and $d\log p_t^A = d\log p_{K,t} - d\log S_t^A$ collapses the share movement: $d\log p_t^T = d\log\gamma_t^T + d\log C_t - \rho\, d\log S_t^A$, in which $d\rho_t$, and hence $\eta^\star$, cancels. The non-SWE identity in (D.6), together with (D.3) at $d\log A_t = 0$ and (D.8), gives $d\log C_t = \gamma^T\Sigma^T x_t^T$. Substituting both into $x_t^T = d\log W_t - d\log p_t^T$, the coefficients collect to Σ^T and

$$x_t^T = \frac{1}{\Sigma^T} d\log S_t. \tag{D.9}$$

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

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

The same linearization holds at every horizon. Aggregating with $V_{jt} = \mathbb{E}_t\Big[\sum_{\tau\ge 0}\Xi_{t,t+\tau}\pi_{j,t+\tau}\Big]$ 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 $\beta_j^{\text{AI}} = \bar{\beta}^{\text{AI}} + \alpha\frac{\varepsilon-1}{\Sigma^T}\kappa\gamma_j^T$, where $\bar{\beta}^{\text{AI}}$ is firm-invariant and κ is the projection of $\mathscr{S}_t$ on $\tilde{R}_t^{\text{AI}}$ in (8). 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

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

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

Step 5: GDP impact. Real GDP is consumption, $d\log Y_t = d\log C_t$. The non-SWE income identity in (D.1), with L fixed, gives $d\log C_t = d\log W_t + \frac{d\gamma_t^T}{1-\gamma^T}$. Using (D.3) at $d\log A_t = 0$ together with (D.8),

$$d\log Y_t = \gamma^T\left[1 + (\Sigma^T - 1)\right]x_t^T = \gamma^T\,\Sigma^T\,x_t^T = \gamma^T\,d\log S_t,$$

the last step using (D.9). Equivalently, the fictitious-producer construction in Baqaee and Farhi (2020) gives the software-composite supplier the cost-based Domar weight $\int_0^1 (f_j/\alpha)\alpha\gamma_j^T\,dj = \gamma^T$. The factor-reallocation term in (43) again vanishes for the same reason as in Step 3 of the proof of Proposition 3. Although the shock changes the division of factor income among SWE labor, non-SWE labor, and compute, sales remain proportional to f_j, the markup μ and the cost share α are uniform across producers, and the competitive software suppliers and the AI sector pass revenue through to factor payments one for one. Aggregate factor income $w_t s + W_t L + p_{K,t}K$ therefore remains the fraction $\alpha\theta/\mu$ of GDP at all dates, so the weighted changes in the three factor-income shares cancel. At the steady state, the relevant cost-based shares are $(1-\rho)\gamma^T$, $1-\gamma^T$, and $\rho\gamma^T$. The movement in ρ_t merely redistributes the software bill between SWE wages and compute rents within this constant total. 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). Substituting (D.9) into $d\log Y_t = \gamma^T\Sigma^T x_t^T$ gives $d\log Y_t = \gamma^T d\log S_t$: a larger aggregate substitution elasticity Σ^T dampens the relative-price response but proportionally raises the GDP response to that relative-price change. The cost-based Domar interpretation is on a GDP basis: the downstream weight f_j/α offsets the direct software cost share $\alpha\gamma_j^T$, leaving the aggregate weight γ^T. As $\rho \to 0$ the composite collapses to software-engineering labor and γ^T, Σ^T reduce to their Section 2 counterparts.

$\square$

D.4 R&D Optimality and Market Clearing in Section 5

This appendix states the R&D optimality condition, the market-clearing conditions, and the definition of competitive equilibrium for the innovation economy of Section 5.

R&D optimality. 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 $V^* =$

$\pi^*/(1-\beta\varrho)$ 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, X_R, creates $\delta_R\lambda_R G^{\lambda_R-1}(\partial G/\partial X_R)N^{\phi_R}$ new varieties with $\partial G/\partial X_R = \alpha_R G/X_R$, and each newly created variety has expected discounted value βV^*. The steady-state optimality condition on the X_R margin therefore equates the unit cost of final output to its marginal benefit in variety creation,

$$1 = \beta V^* \cdot \delta_R \lambda_R G^{*\lambda_R - 1} \frac{\alpha_R G^*}{X_R^*} N^{*\phi_R}. \tag{D.10}$$

Markets and equilibrium. Let m_{jt}^M denote economy-wide materials demand for variety j, and let $x_{R,jt}$ denote R&D demand for variety j. The corresponding aggregate composites are

$$M_t = \left[\int_0^{N_t} (m_{jt}^M)^{(\varepsilon-1)/\varepsilon} dj\right]^{\varepsilon/(\varepsilon-1)},$$
$$X_{R,t} = \left[\int_0^{N_t} x_{R,jt}^{(\varepsilon-1)/\varepsilon} dj\right]^{\varepsilon/(\varepsilon-1)}.$$

Goods markets clear variety by variety, $y_{jt} = c_{jt} + m_{jt}^M + x_{R,jt}$. 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 within the R&D sector. GDP is value added, gross final-good output net of intermediate materials,

$$Y_t = Q_t - M_t = C_t + X_{R,t}. \tag{D.11}$$

A competitive equilibrium is a collection of quantities, prices, and firm values such that households and firms optimize, the R&D optimality condition (D.10) holds, R&D factor returns equal marginal value products, and all markets clear.

D.5 Inversion under Endogenous Innovation

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

$$\mathscr{S}_t^{\varrho} \equiv (1-\beta\varrho)\big(\mathbb{E}_t - \mathbb{E}_{t-1}\big)\Big[\sum_{k=0}^{\infty} (\beta\varrho)^k d\log S_{t+k}\Big], \tag{D.12}$$

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

Proposition 6 (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*

$$R_{jt} - \mathbb{E}_{t-1}[R_{jt}] = \delta^S \gamma_j \mathscr{S}_t^{\varrho} + \mu_t, \qquad \delta^S = \alpha \frac{\varepsilon - 1}{\Sigma}, \qquad \Sigma = \sigma + \left[\alpha(\varepsilon-1) - (\sigma-1)\right] \frac{\sigma_\gamma^2}{\gamma(1-\gamma)}, \tag{D.13}$$

where $\sigma_\gamma^2 = \int_0^{N^*} f_j(\gamma_j - \gamma)^2 dj$ *is the final-expenditure-weighted cross-sectional variance of software engineering payroll shares at the stationary steady state, and* μ_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*

$$\mathbb{E}^*[\mathscr{S}_t^{\varrho} \mid \tilde{R}_t^{AI}] = \kappa \cdot \tilde{R}_t^{AI}, \qquad \text{where} \quad \kappa = \frac{\delta}{\delta^S} = \frac{\Sigma}{\alpha(\varepsilon - 1)} \cdot \delta. \tag{D.14}$$

This is identical to the benchmark inversion of Proposition 1.

The structural gradient δ^S is the same as the structural gradient from Section 2, with the same dependence on (σ, ε) and on the payroll-share dispersion σ_γ^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 X_R^* co-moves with Y^*. Both shifts are firm-invariant and absorbed into μ_t; neither contaminates the payroll-share loading that the cross-sectional regression isolates.

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 ϱ. 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.

We first show that the extensive margin drops out of the distribution of expenditure shares. To formalize, let $h(\phi)$ denote the fixed probability density of software engineering intensity. Because survival is type-independent and entrants draw their type from $h(\phi)$, the measure of active varieties of type ϕ is $N_t h(\phi) d\phi$ at every date. Writing $p_t(\phi)$ and $\gamma_t(\phi)$ for the price and payroll share of a type-ϕ variety, the final-expenditure-share density over types is

$$f_t(\phi) = \frac{N_t h(\phi) p_t(\phi)^{1-\varepsilon}}{P_t^{1-\varepsilon}}, \qquad P_t^{1-\varepsilon} = N_t \int h(\phi) p_t(\phi)^{1-\varepsilon} d\phi,$$

with $\int f_t(\phi) d\phi = 1$ and $\gamma_t = \int f_t(\phi)\gamma_t(\phi) d\phi$. Differentiating the price index and imposing the numeraire $P_t = 1$ gives $\int f(\phi) d\log p_t(\phi) d\phi = \frac{1}{\varepsilon - 1} d\log N_t$: a change in the variety mass must be offset by a common change in prices. Because $h(\phi)$ is fixed, differentiating the expenditure-share density then gives

$$d\log f_t(\phi) = d\log N_t + (1-\varepsilon) d\log p_t(\phi) = (1-\varepsilon)\left[d\log p_t(\phi) - \int f(\tilde{\phi}) d\log p_t(\tilde{\phi}) d\tilde{\phi}\right],$$

so the direct $d\log N_t$ term is exactly offset by the common price change required to keep $P_t = 1$, and only price changes relative to the expenditure-weighted mean move the distribution of expenditure shares.

The production technology, markup, and payroll shares (with cost share $\alpha\gamma(\phi)$ for a type-ϕ variety) are the same as in Section 2, so the constant markup and Shephard's lemma deliver the same demeaned price response as (31):

$$d\log p_t(\phi) - \int f(\tilde{\phi})\, d\log p_t(\tilde{\phi})\, d\tilde{\phi} = -\alpha[\gamma(\phi) - \gamma]x_t.$$

Hence $d\log f_t(\phi) = \alpha(\varepsilon-1)[\gamma(\phi)-\gamma]x_t$ and the CES payroll-share response $d\gamma_t(\phi) = (\sigma-1)\gamma(\phi)[1-\gamma(\phi)]x_t$ coincide with their fixed-variety counterparts, so the computation of $d\gamma_t$ in Step 2 of the proof of Lemma 1 applies verbatim and gives $d\gamma_t = \left[(\sigma-1)\gamma(1-\gamma) + (\alpha(\varepsilon-1)-(\sigma-1))\sigma_\gamma^2\right]x_t$. Production labor-market clearing,

$$w_t s = \frac{\varepsilon-1}{\varepsilon}\alpha\gamma_t Q_t \qquad \text{and} \qquad W_t L = \frac{\varepsilon-1}{\varepsilon}\alpha(1-\gamma_t)Q_t,$$

with fixed factor supplies again implies $d\log w_t - d\log W_t = d\gamma_t/[\gamma(1-\gamma)]$, because total final-good use Q_t cancels in the ratio of the two conditions. Substituting the expression for $d\gamma_t$ into the definition of x_t gives, exactly as in the proof of Lemma 1,

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

Per-period profit for an active variety j of type ϕ_j is $\pi_{jt} = p_t(\phi_j)^{1-\varepsilon}Q_t/\varepsilon$. Splitting its price change into the demeaned response above and the common mean $\frac{1}{\varepsilon-1}d\log N_t$, and substituting $x_t = d\log S_t/\Sigma$,

$$d\log \pi_{jt} = \alpha\frac{\varepsilon-1}{\Sigma}\gamma_j d\log S_t + (\text{firm-invariant}),$$

where the second term collects everything common across active varieties: the demand change $d\log Q_t$, the variety-mass term $-d\log N_t$, and the term proportional to the mean payroll share γ. The same linearization applies at every horizon. Substituting this profit response into (25), using the steady-state relation $\Xi_{t,t+k} = \beta^k$, and applying the expectation revision in (D.12) gives

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

where $\delta^S = \alpha(\varepsilon-1)/\Sigma$ and μ_t collects all firm-invariant terms, including SDF news. This establishes (D.13).

Projecting the excess return $R_{jt} - R_t^f$ (equivalently its unexpected component by condition (ii) in

Section 2.3) on the residualized AI-index return $\tilde{R}_t^{\mathrm{AI}}$ gives

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

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

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

Condition (i) in Section 2.3 gives $\mathbb{E}[\tilde{R}_t^{\mathrm{AI}}] = 0$. The law of iterated expectations also gives $\mathbb{E}[\mathscr{S}_t^{\varrho}] = 0$ because $\mathscr{S}_t^{\varrho}$ is a conditional-expectation revision. Hence the best-linear-predictor intercept in (D.14) is $\mathbb{E}[\mathscr{S}_t^{\varrho}] - \kappa \mathbb{E}[\tilde{R}_t^{\mathrm{AI}}] = 0$. This proves (D.14). □

D.6 Proof of Proposition 5

Proof. We first derive the production-side decomposition in (27), holding A fixed, and then combine it with the innovation block.

Output. Constant markups imply $Q^* = \int p_j y_j dj = \frac{\varepsilon}{\varepsilon-1}\mathscr{C}$, where Q^* is total use of the final-good composite 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, $Y^* = Q^* - M^*$; eliminating $\mathscr{C} = (w^* s + W^* L)/\alpha$ gives

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

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

$$d\log Y^* = \gamma\, d\log w^* + (1-\gamma)\, d\log W^*, \qquad \gamma = \frac{w^* s}{w^* s + W^* L}. \tag{D.15}$$

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 unit-cost function dual to (1), with final-good materials priced at the numeraire,

$$d\log mc_j = \alpha \gamma_j\, d\log(w^*/S) + \alpha(1-\gamma_j)\, d\log W^*, \tag{D.16}$$

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 (D.16) 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,

$$\gamma \, d\log(w^*/S) + (1-\gamma) d\log W^* = \frac{d\log N^*}{\alpha(\varepsilon-1)}. \tag{D.17}$$

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 (D.17), 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 (27).

We now solve for the innovation-augmented response. The proof proceeds in three steps: the R&D optimality condition pins X_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 (27).

Step 1: the optimality condition pins $d\log X_R^* = d\log Y^*$. Multiplying both sides of (D.10) by X_R^* and using the Cobb-Douglas property $(\partial G/\partial X_R) X_R = \alpha_R G$ gives

$$X_R^* = \beta V^* \lambda_R \alpha_R \delta_R G^{*\lambda_R} N^{*\phi_R}. \tag{D.18}$$

The stationary-variety condition requires new variety creation to exactly replace obsolescence. Setting $N_{t+1} = N_t = N^*$ in (26),

$$(1-\varrho) N^* = \delta_R G^{*\lambda_R} N^{*\phi_R}. \tag{D.19}$$

Substituting (D.19) into (D.18) yields $X_R^* = \beta V^* \lambda_R \alpha_R (1-\varrho) N^*$. Using the steady-state variety value $V^* = \pi^*/(1-\beta\varrho)$ from the SDF-weighted survival value (25), and the fact that, since entrants are drawn from the same type distribution as existing varieties, $\pi^* N^* = \int_0^{N^*} \pi_j^* dj \equiv \Pi^*$,

$$X_R^* = \frac{\beta\lambda_R(1-\varrho)\alpha_R}{1-\beta\varrho}\Pi^*.$$

Under constant markups, total profit is a constant fraction $1/\varepsilon$ of total revenue, so $\Pi^* = Q^*/\varepsilon$. Materials are the fraction $1-\alpha$ of each variety's cost, and cost is the fraction $\frac{\varepsilon-1}{\varepsilon}$ of revenue, so aggregate materials are $M^* = (1-\alpha)\frac{\varepsilon-1}{\varepsilon} Q^*$. The resource constraint (D.11) therefore gives $Y^* = Q^* - M^* = Q^*/\theta$, where $\theta = 1/[1-(1-\alpha)\frac{\varepsilon-1}{\varepsilon}]$ is the ratio of gross final-good use to GDP. Hence $\Pi^* = \theta Y^*/\varepsilon$, and

$$X_R^* = \frac{\beta\lambda_R(1-\varrho)\alpha_R\theta}{\varepsilon\big(1-\beta\varrho\big)}Y^*. \tag{D.20}$$

Every coefficient on the right is a structural parameter that does not vary across steady states. Log-differentiating (D.20) gives

$$d\log X_R^* = d\log Y^*. \tag{D.21}$$

Step 2: the stationary-variety condition gives $d\log N^*$. Log-differentiating (D.19) (with ϱ and δ_R constants),

$$d\log N^* = \lambda_R d\log G^* + \phi_R d\log N^*,$$

which rearranges to $d\log N^* = \frac{\lambda_R}{1-\phi_R}d\log G^*$. Because G is Cobb-Douglas with constant exponents and the sector-specific R&D endowments (s_R, L_R) are fixed,

$$d\log G^* = \alpha_s d\log(Ss_R) + \alpha_L d\log L_R + \alpha_R d\log X_R^* = \alpha_s d\log S + \alpha_R d\log X_R^*.$$

Substituting (D.21) gives

$$d\log N^* = \frac{\lambda_R}{1-\phi_R}\big(\alpha_s d\log S + \alpha_R d\log Y^*\big). \tag{D.22}$$

Step 3: solving the system. Equations (27) and (D.22) form a linear system in $(d\log Y^*, d\log N^*)$ given $d\log S$. Substituting (D.22) into (27),

$$\begin{aligned} d\log Y^* &= \gamma\, d\log S + \frac{1}{\alpha(\varepsilon-1)}\cdot\frac{\lambda_R}{1-\phi_R}\big(\alpha_s d\log S + \alpha_R d\log Y^*\big) \\ &= \gamma\, d\log S + \iota\alpha_s d\log S + \iota\alpha_R d\log Y^*, \end{aligned}$$

where $\iota \equiv \lambda_R/[\alpha(\varepsilon-1)(1-\phi_R)]$. 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 footnote 30. Dividing gives (28). □

References

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

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*.

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