The Effect of Software Adoption on Skill Demand and Wage Inequality*

Zara Contractor† Bledi Taska‡

Click here for latest version

Abstract

We study the impact of one type of technology—software—on the labor demand of firms and, consequently, equilibrium wages. At the firm level, we use job posting data to identify when a job adopts an additional type of software. Through a latent variable IV strategy we find causal impacts of software adoption on skill requirements: each software adoption makes a job 1.1pp more likely to require social skills and 0.8pp more likely to require analytical skills. The number of vacancies posted also rises post software adoption events. We embed these causal upskilling effects into an equilibrium model of software adoption and occupational sorting across white-collar occupations and simulate the effect of a 20% fall in software prices on inequality. We find the wage differential of high-wage STEM and professional occupations rises by 20pp, due to their complementarity with software. Within occupations, the wage premium of software-using jobs rises by 12pp, largely through the increase in job-level skill requirements that restrict some workers from moving to higher-paid jobs.

*__Acknowledgments:__ I am extremely grateful to my advisors Giuseppe Moscarini, Joseph Altonji and Ilse Lindenlaub for their advice and guidance. We also thank Noriko Amano, Barbara Biasi, Paula Calvo, Ran Gu, John Eric Humphries, Disa Hynsjo, Ryungha Oh, Cormac O'Dea, Rosie Li, Michael Peters, Alysha Shroff, Trevor Williams, and participants of the Yale Macroeconomics and Labor/Public prospectus workshops for productive discussions and constructive comments.

†Middlebury College, zara.contractor@yale.edu

‡Lightcast, bledi.taska@lightcast.io

1 Introduction

Skill-biased technological change has been a key driver of rising wage inequality in the Unites States. One such technology is software.[1] As of 2022, software comprises 17% of private nonresidential investment, up from 10% in 2000 (FRED). Understanding how software adoption affects labor demand helps identify which workers benefit or lose from its proliferation and can inform redistributive or retraining policies.

In this paper, we ask two questions. First, how does adopting software change a firm's demand for workers with different skill sets? For instance, say an accounting department switches from Microsoft Excel to a specialized accounting software like QuickBooks: How do the skills requirements of the accountants change? Does the new software complement or substitute for accountants—and for other workers in the firm? Second, how do software-induced changes to firms' skill demand impact wage inequality in equilibrium?

To answer the first question, we use job vacancy data from Lightcast (formerly Burning Glass Technologies) to identify when firms adopt additional software types, and estimate the impact of these adoption events on the firm's labor demand and skill requirements. Our identification comes from time variation in skill requirements within jobs—both adopting and non-adopting ones—within a firm. We find that *each* software adoption increases the likelihood of requiring analytic and social skills by 0.8pp and 1.1pp respectively. Firms' labor demand also rises: The number of job postings increases by 20% for jobs that adopted software and by 5% for non-adopting jobs within the firm.

Turning to the second question, we embed the estimated firm-level upskilling effects into an equilibrium model of firm software choice and occupational sorting. Firms choose the quantity of labor and, optionally, software inputs for each occupation. Workers match with jobs on two-dimensional skill requirements—social and analytic—where skill requirements depend on whether the job uses software. We estimate the model using Generalized Method of Moments (GMM). We find that a fall in the price of software increases wage inequality both within and across white-collar occupations.

[1] We define software broadly, encompassing widespread varieties like sales, accounting, and HR software, as well as specialized types like aviation and statistical software.

Our main counterfactual studies how software impacts wages and inequality if we shut down the effect of software on skill requirements, and finds this lowers the wage premium of working in a software job. Software-induced upskilling raises within-occupation inequality by restricting labor from moving to higher paid software jobs. Without the channel of upskilling, overall wage inequality (as measured by the Gini coefficient) would be lower, even though across-occupation inequality would rise slightly.

We now go into details of our analysis. Lightcast scrapes job ad data from over 40,000 online sources; our sample comprises U.S. online job ads from 2010 to 2019 that mention the firm's name. In addition to information like location, occupation, and industry, we observe skill requirements through disaggregated keywords that Lightcast extracts from the job description in the posting. We measure analytic and social skill requirements of jobs by assigning keywords to skill categories, building on the measures of Deming and Kahn (2018). An innovation of this paper is to identify software adoption events at the firm-occupation level based on software keywords in the text of job postings. In particular, we observe adoptions at the firm-occupation level over a broad range of software.

The timing of software adoption is an endogenous choice: Unobserved factors like shifts in product demand or local labor supply could cause an omitted variable bias when assessing the effect of software adoption on skill demand. For example, if a shift in product demand increased the returns to both software and skilled workers, this could lead to a spurious positive correlation between software and skill requirements. To the best of our knowledge, we are the first to identify the causal impact of software adoption on skill demand at the firm level, accounting for endogeneity in the timing of adoption. To do so, we adapt the latent variable strategy in Freyaldenhoven et al. (2019) to our setting.

The goal of this strategy is to 'control' for the unobserved confounders, which we model as a time-variant latent variable at the firm-occupation level. We construct a proxy for this latent variable from skill requirements in *non-adopting* occupations within the same firm. For instance, say we want to estimate the impact of software (e.g.: QuickBooks) on skills required for an accounting position. Our proxy could involve skill requirements of the firm's sales department, which does not adopt any software. We run an event study of skill requirements before and after the adoption event, controlling for the proxy.

Since the proxy is an imperfect measure of the true latent variable, we then instrument for the proxy with the lead of the observed software adoption event to deal with measurement error. We show that this strategy identifies the causal effects under the assumptions that 1) software adoption only affects skill requirements in the occupation using the new software[2]; and 2) the latent variable is correlated both over time and across occupations within a firm. Intuitively, skill requirements elsewhere in the firm and future software adoption are both correlated with the unobserved latent variable, but are not otherwise correlated with each other. So, leveraging both allows us to isolate the impact of the latent factor, and identify the causal impact of software on skills.

Using this strategy, we find that adopting *each* additional type of software increases analytic skill requirements by 0.8 percentage points and social skill requirements by 1.1 percentage points on average. The impact on social skills is quantitatively important, equaling 18% of the difference in social skill requirements between managerial and STEM jobs. Software types differ in their impact and the marginal effect of each software adoption is higher when the job does not already use multiple types of software. We also find sizable increases on the number of posted vacancies—our measure of labor demand. Firms post 30% more vacancies in jobs that have adopted software, with smaller spillover effects of 5% to non-adopting jobs in the firm.

Next, we investigate how these firm-level effects on skill demand aggregate across firms to impact equilibrium wages and inequality. To do so, we embed our causal estimates into an equilibrium model of software adoption, where workers sort across jobs based on wages and skill requirements. This structure allows us to estimate elasticities of substitution between labor and software for the white-collar sector, and to measure how falling software prices affect wage inequality. In addition, we can isolate software's effect on inequality through increasing skill requirements from its role as an input in production.

In our setting, perfectly competitive firms produce according to a nested CES production function with intermediary occupations. A firm chooses input quantities

[2] We do not require that software adoption affects total labor demand only in the occupation adopting software.

of labor and software for each occupation. Workers and occupations are characterized by two-dimensional (analytic and social) skills and skill requirements respectively, and a worker is qualified for an occupation if they meet the skill requirement in both dimensions. Workers derive utility from wages and idiosyncratic preferences for occupations, as in Choo and Siow (2006). They choose an occupation-software pair to maximize utility from amongst those for which they qualify. Conditional on being qualified, a worker's skill bundle does not affect their productivity at the occupation.

Firms choose software inputs at both the extensive margin (whether to use any software) and intensive margin (how much software to use) for each occupation. A key feature of the model is that using any software for an occupation raises its skill requirements, in line with our empirical findings, and thus disqualifies some workers from the job. From the firm's perspective, higher skill requirements imply a higher marginal cost of labor as the wage rate for jobs using software is higher due to fewer qualified workers. For each occupation, the choice of whether to adopt software is a trade-off between the additional output—and thus revenue—from using the software, against the total software cost and increase in the wage rate. The firm's software use and hiring decisions are complex, arising from interconnected discrete and continuous choices.

In equilibrium, total labor demand across firms for each occupation-software pair equals the total mass of workers choosing to work in that occupation-software pair. For example, the total demand for managers in non-software jobs across firms must equal the mass of workers who, at equilibrium wages, prefer to work in management without software.

We estimate the model with GMM. As a first step, we measure skill requirements and equilibrium wages for each occupation-software pair from our empirical results and Lightcast data. We estimate the labor demand side parameters to match a set of data moments, including equilibrium employment shares and the causal effect of software on labor demand. We find that software is a complement to workers, especially high-skilled workers. On the labor supply side, we calibrate the variance of worker preferences and the two-dimensional skill distribution to align with equilibrium wages and employment shares, accounting for workers' choice sets that depend on their skill bundles.

We use our model to compute how falling software prices impact inequality, as the real price index of software fell by around 20% over our sample period (FRED). We show that falling software prices raise software adoption, resulting in higher average wages and also higher inequality. As the proportion of firms employing software increases, labor demand within each occupation shifts toward software jobs, increasing the relative wage of software jobs. Software proliferation also increases inequality across occupations, as software-labor complementarity is stronger for high-paying managerial and STEM jobs.

In a counterfactual, we shut down the channel of increasing skill requirements after software adoption. We show that the within-occupation increase in inequality is largely driven by the upskilling effect of software. Skill requirements restrict labor supply from moving to software jobs, limiting total employment in software jobs and increasing the wage premium for software jobs within each occupation. These forces increase within-occupation inequality but lower inequality across occupations. On net, skill requirements increase total inequality by 5% compared to the counterfactual case, as measured by the Gini coefficient.

The rest of this paper is organized as follows. Section 2 describes the relevant literature and our contribution; Section 3 consists of the empirical analysis of the firm-level effects of software; Section 4 describes the model, estimation and counterfactuals. Section 5 concludes.

2 Related Literature

Our paper joins an extensive literature on how technological change affects workers. Technological advances like robots and computerization tend to be 'skill-biased,' increasing the demand for educated workers (Katz and Murphy (1992), Card and Lemieux (2001), David and Dorn (2013), Autor et al. (2003), Lindenlaub (2017)). Technology can automate the need for labor in a subset of tasks, raising the labor demand for labor in other, non-automatable tasks (Acemoglu and Autor (2011)), or even creating new tasks (Acemoglu and Restrepo (2019), Acemoglu and Restrepo (2020)). As the expanding non-automatable tasks tend to be high wage non-routine cognitive tasks, the consensus of this literature is that technologies have increased wage inequality and polarized the wage distribution. These task-based models have been applied to various technologies, ranging from robots (Acemoglu and Restrepo

(2020)) to ICT and, more recently, AI (Acemoglu et al. (2020)).

We focus the remainder of our review on the subset of this literature that studies software and closely related technologies like computerization and information & communications technology (ICT).[3] Several papers show that these technologies shift employment away from routine cognitive tasks that can be carried out by a computer algorithm (Autor et al. (2003), Atalay et al. (2018)). For software in particular, Webb (2019) finds lower employment growth in occupations with O*Net occupational descriptions most similar to software patents—and therefore at high risk of software automation. However, Aum (2017) finds that rising software innovation—relative to physical equipment innovation—can explain the reversal in demand for highly skilled workers at the beginning of the 21st century. Deming (2017) shows that the skill reversal was driven by lower demand for jobs requiring analytic skills alone, while occupations also requiring social skills were growing. He hypothesizes that ICT technologies increase social skill requirements through task trading between workers. Mariscal (2018) shows that IT technologies also lower the labor share through corporate hierarchies by increasing managers' span of control.

One strand of this literature studies how technology adoption at the level of individual firms impacts the firms' own labor demand (Akerman et al. (2015), Acemoglu et al. (2022), Bessen et al. (2019), Almeida et al. (2020), Bessen et al. (2020)). The papers most related to ours use vacancy data to measure changes in skill demand *within* occupations (Bessen et al. (2022), Dillender and Forsythe (2022), Hershbein and Kahn (2018)). In contemporaneous work, Bessen et al. (2022) use the Lightcast data to study how in-house software production—measured by a spike in the share of IT vacancies[4]—affects the demand for skills. Using an event-study specification, they show that the number of skills mentioned in job ads[5] increases after a spike in IT hiring spike, as does the sorting of high skill workers to high-wage firms. They

[3]The software market has also been studied outside of its effect on labor demand, and software is generally found to differentially benefit larger firms. De Ridder (2019) shows that the rise of intangible inputs like software and ICT shift firm expenditure toward fixed costs, which can explain rising markups and slowing productivity growth. Lashkari et al. (2018) find that software (and hardware) have elastic demand and can rationalize increased firm size and concentration. Eckert et al. (2022) attribute higher wage growth in big cities to ICT driving up wages at business services firms more for larger firms that tend to be in cities.

[4]They define an 'IT spike' as events where vacancies for software developers as a share of all vacancies increased by at least 1% relative to the mean share over the previous four quarters.

[5]Across all occupations in the firm. They find higher effects on high skilled occupations.

address endogeneity in the timing of IT spikes by including controls related to labor market tightness and productivity. Our strategy is different as we identify when firms adopted new software types—developed in-house or purchased—through software requirements specified in vacancies. Our approach allows us to observe what type of software is being adopted, and to focus on skill requirements for the occupations that are using the software. We also differ in our identification strategy, as we follow a latent variable approach that accounts for unobserved confounds.

Another related paper using the Lightcast data is Dillender and Forsythe (2022), who focus on the upskilling effects of software for office and administrative support (OAS) workers. At the local labor market level, they find that improvements in OAS technology increase skill requirements and wages, with spillovers on non-adopting firms. Within firms, they show that OAS technologies correlate with higher skill requirements and with a broader set of requirements usually associated with higher skill office functions. The focus of their paper differs from ours as they concentrate on OAS occupations and their causal analysis is at the level of the local labor market rather than individual firm.

Our paper is also closely related to work on how occupational choice interacts with technology adoption. Dvorkin and Monge-Naranjo (2019) construct a dynamic Roy model of machine (computers or robot) penetration, where workers acquire human capital and choose occupations. They highlight the importance of occupation mobility in tempering the impact of technological innovations on inequality. Braxton and Taska (2023) use the share of vacancies requiring software or computer skills to measure occupation-level technological change. They show that displaced workers from occupations with high technological change receive lower future earnings, as they may not be qualified for new jobs in their previous occupation. Closely related to our paper, Atalay et al. (2018) also embed micro-level estimates of the effect of occupation-level ICT on skill demand into a structural model where ICT technology can change the task content of occupations. In contrast, we model software and labor choices of individual firms that are heterogeneous in their suitability to software, allowing for *within*-occupation inequality in skill requirements and wages. We require firms, not workers, to choose software, so software choices across occupations within a firm are interdependent. We also impose strict qualification requirements that drive occupational sorting. In our model workers match with jobs based on two-dimensional (analytic and social) skill bundles. In this sense, our paper is related to the literature on worker sorting on multidimensional skill bundles (Lindenlaub (2017), Lise and Postel-Vinay (2020)) and is an application of the model developed by Choo and Siow (2006).

3 Empirical Analysis

3.1 Data

Our data comes from online job vacancies compiled by Lightcast (formerly Burning Glass Technologies). The Lightcast database includes the near-universe of all US online job ads, collected from over 40,000 job portals and employer websites.[6] Our sample consists of all US non-internship vacancies posted from 2010 to 2019, for which we observe the firm's name. This eliminates approximately 40% of vacancies placed through recruiters that do not reveal the firm's identity. We identify a 'firm' as a unique combination of the employer's name and the MSA of the job to minimize the risk of conflating unrelated firms that happen to share a name.[7]

For each vacancy, we observe the year and month when the job ad was posted, the MSA where the job is located, the job title written in the vacancy, the 6-digit SOC occupational code it belongs to, as assigned by Lightcast and the qualifications required for the job, including educational attainment and years of relevant experience. The data also includes more disaggregate requirements, in the form of keywords parsed from the vacancy text. There are over $10,000$ unique keywords; they can be general, like 'problem solving' or 'teamwork', or describe a specific task, like 'telephone calls' or 'Stata.' Lightcast also assigns an indicator for which keywords

[6]For a detailed description of the Lightcast data, see Hershbein and Kahn (2018) and Deming and Kahn (2018).

[7]In effect, our measure of a 'firm' is between a firm and an establishment. We choose this approach because the firm name is our only identifier and wrongly conflating unrelated firms is more problematic for us than splitting establishments of a firm. For example, suppose that two firms in different MSAs share a name and one is more software intensive and hires higher skilled workers. If we conflated the two firms, we would wrongly attribute a switch between the two firms to an increase in both software and skill requirements. Suppose that we observe vacancies from the less software and skill intensive firm each year, and then the more software and skill intensive firm posts vacancies for the first time in 2015. If we did not differentiate the firms, it would appear like the original firm adopted software in 2015 and, at the same time, increased average skill requirements.

are related to software. We will use the keywords that are not related to software to construct measures of skill requirements, which will be our outcome variable, and the keywords related to software to identify software adoption events.

3.2 Measuring skill requirements

Our main variables of interest are analytic and social skill requirements at the firm-occupation-year level. We choose these skills as they are underlying skills relevant across occupations, and in line with previous literature; we also report results for administrative (routine cognitive), management and IT skill requirements in the appendix.[8] To simplify terminology, we will refer to a firm-occupation pair as a 'job'. Following Deming and Kahn (2018), we first choose a set of core keywords that can be classified into each skill category; for example, we classify 'research' under analytic and 'team' under social. Table 1 shows the keywords classified into each category; those in bold font have also been used in Deming and Kahn (2018)[9]. We then consider the full list of keywords not classified as software skills. From these, we classify a keyword as belonging to a skill category if it includes any of the core keywords of that category. For example, 'clinical research' includes the core analytic keyword 'research,' so we classify 'clinical research' as an analytic skill keyword.

We measure analytic (social) skill requirements at the job-year level as the proportion of vacancies mentioning at least one analytic (social) skill keyword.[10] For example, if a firm had twenty marketing vacancies in 2012, ten mentioning analytic skill keywords and five mentioning social skill keywords (possibly with overlap), our measures of analytic and social skill requirements would be 0.5 and 0.25, respectively.

Our skill measures are predictive of posted salaries, where available.[11] We compare our measures with occupation-level skill measures from O*Net, and find that

[8] We are not able to measure manual skills due to data limitations—we find job postings generally do not include manual skill keyword.

[9] For every skill category we share with Deming and Kahn (2018) we use all the keywords they chose in addition to expanding the lists to include other relevant keywords. Our resulting skill measures are highly correlated with theirs.

[10] We find these measures are strongly correlated with alternative measures of counting the average number of keywords per category. We choose this measure because it is in line with the literature and has a more intuitive definition as the proportion of vacancies requiring the skill.

[11] While the $\sim 20\%$ of vacancies posting wages may be a selected sample, we run regressions within the sample of non-missing wages, which should not be subject to this selection bias.

while O*Net measures are, unsurprisingly, better predictors of average occupational salaries, our measures can additionally explain within-occupation variation in salaries. Furthermore, Deming and Kahn (2018) show that their similarly constructed skill measures are predictive of wages even when aggregated to the local labor market level.

Table 1: Core keywords in each skill category

Skill Category	Core Keywords
Analytic	**problem solving, research, analytic, critical thinking, math, statistics**, solving, engineering, decision making, calculation, planning, estimating, algebra, geometry
Social	**communication, teamwork, collaboration, negotiation, presentation**, relationship building, leader, telephone, teach, listen, persuasion, social
Admin	billing, payroll, typing, scheduling, data entry, appointment setup, administration, office duties, mailing, filing, cash handling, copying, invoicing, secretarial, accounts payable, clerical, tax filing, telephoning
Management	**management, supervision, leadership, mentoring, staff**
IT	software, data, application support, troubleshooting, technical assistance, information technology

Notes: This table includes the core keywords for each skill category. A keyword that includes any of the core analytic keywords will be classified as an analytic keyword. Respectively for other skills. Keywords in bold are taken from Deming and Kahn (2018).

3.3 Software adoption events

A key challenge in using vacancy data to identify the impact of software is how to measure the software stock of a firm. We do not compare software skill requirements across firms because firms that choose to use more software may have entirely different business operations (Acemoglu et al. (2022)), and therefore skill demand. Even comparing software requirements within a firm across years, we cannot be sure of the software stock of a firm at any point in time. Say a firm mentions Salesforce in marketing vacancies in 2010 only. In 2013, no new vacancies may require sales software, but workers hired in 2010 may still be using Salesforce and perhaps teaching it to new hires. Furthermore, a firm may use software without having ever mentioned it in its job ads.

So, instead of attempting to measure the software stock accurately, we employ an event study strategy to estimate the marginal effect of adopting one more software type. For example, if marketing jobs at a firm that were previously using marketing software when we observe them, also adopt statistical software, then we will estimate the marginal effect of statistical software on their skill requirements.

We build our software adoption events in three steps. First, we identify which types of software are required for each vacancy. Second, we identify times when a job first requires a new type of software. In the last step, we impose criteria on potential software adoptions to separate actual adoptions from data artifacts.

First, to identify software types we make use of the fact that Lightcast classifies each keyword into clusters of related keywords: for example, 'Stata' and 'R' belong to the 'statistical software' cluster. For the first step, we start with all software-related clusters and further refine them by consolidating closely related ones. We keep those clusters that can be interpreted as a type of software (for example, we keep 'statistical software' but drop ' general programming skills,' which is not a software type.) and that have at least 50,000 mentions over our sample. We drop clusters related to Microsoft Office, as we expect it to be used ubiquitously even when not explicitly mentioned in vacancies.[12] This gives a final sample of 97 *software types*, listed in the appendix. Examples of software types include broadly used software like marketing software, human resources software, and sales software; specialized software like dental imaging software, legal software, and aviation software; and coding languages like C & C++, Java, and statistical software. We say a vacancy *requires* a type of software if it includes at least one keyword from that type of software. For example, a vacancy requiring Stata, R, and Salesforce requires statistical and marketing software.

[12]An exception is that we keep one cluster related to advanced Microsoft Excel skills.

Our second step is to identify potential software adoption events—the first time we observe a job mention each software type. For example, the first year we observe vacancies for accountants at Walmart in an MSA that mention specialized accounting software would be a potential software adoption for that firm-occupation pair. To separate genuine adoptions from misclassifications, we keep potential adoption events that meet four criteria:

1. Jobs must have posted a vacancy without mentioning the software type at least two years before the adoption event. For example, if vacancies for accountants at Walmart mention accounting software for the first time in July 2015, then we must first observe Walmart vacancies in accounting—not mentioning accounting software—before July 2013. This requirement eliminates jobs that have previously used the software type.

2. Jobs must have posted another vacancy (again, without the software type) within the three months preceding the potential adoption event. This requirement aims to narrowly pin down the timing of the observed software adoption.

3. Jobs must have posted another vacancy – with or without the adopted software — in the three months after the adoption. This requirement serves to balance the second requirement: we avoid selecting firms that are posting more vacancies just before (versus just after) the adoption, which would wrongly imply that firms post less vacancies after adopting software.

4. In the month of adoption, at least 10% of vacancies in the job must require the software. This requirement is targeted at very large jobs that post hundreds of vacancies, increasing the likelihood of posting a 'false positive' vacancy requiring the software. We check for robustness to eliminating this requirement, or increasing it to 20% or 50%.

Approximately 10% of all potential adoptions meet all four criteria and we use these as our software adoption events.[13]

Our method of identifying software adoptions is conservative. Many real software adoptions may not meet our criteria, or might not appear in job ads at all. Since we

[13]Some jobs can have multiple software adoptions, for different types of software. Within a job, we use the first software adoption observed. Additional adoptions will have further effects.

cannot rule out software adoptions in jobs where we do not observe any, our identification comes primarily from time variation within jobs. Our resulting sample of firms and adopting jobs is biased toward high wage jobs and firms with more software mentions than average. This may reflect that high wage jobs use more software, but also that some firms choose to mention software in job ads more frequently. We also note that we are more likely to observe software types that require software-specific knowledge, as they are more likely to be mentioned in vacancy text.

A bigger concern would be a 'false positive'; that is, finding a software adoption event where in reality the software was previously present and no new adoption took place. Our first requirement—that we observe jobs at least two years before the claimed adoption event—partially mitigates this concern as we must observe a *change* in software requirements at each adoption event. But we could still identify 'false' events if the firm chooses to start mentioning a pre-existing software in job ads for some other reason. For example, a firm may start mentioning a software type if they want to stop training workers in the software or if they require a replacement hire after over two years. Such a 'false positive' error would mean the effects on skill requirements that we find are the average effect when including some untreated firms in the treatment group. Therefore our results could be considered a lower bound of the true effect of software adoption.

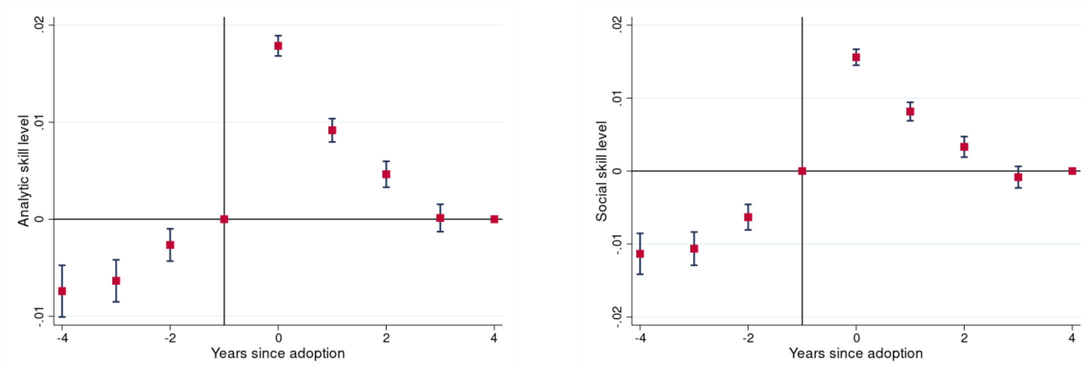
We find that software adoptions have limited persistence in vacancies: two years later, only 50% of jobs posting vacancies continue to mention the software. This is a lower bound on the persistence of software use in the firm: Once the firm has enough employees with knowledge of the software, they may not mention the software in vacancies. We further discuss the persistence of software requirements after adoption events in Appendix B.

3.4 Event study

We employ a staggered event study to show how skill requirements evolve around software adoption events. The event study exploits time variation only, and it is not meant to be interpreted causally. Let t denote calendar year and $\bar{t}_{fo}$ be the calendar year when an adoption is observed for job fo—namely in firm f and occupation o. Let $\tau = t - \bar{t}$ be the number of years since the adoption event. Define indicator

Figure 1: Event study estimates: impact of software adoption on skill requirements

(a) Analytic skill requirements (b) Social skill requirements



Notes: TWFE estimates of the evolution of skill requirements 5 years before and after a software adoption event. Skill requirements at year -1 are normalized to 0.

variables $D^{\tau}_{fot} \equiv \mathbb{I}_{t-\bar{t}_{fo}=\tau}$. That is, D^{τ}_{fot} equals 1 if the current calendar year t is τ years after (before, if $\tau < 0$) the year when the software was adopted.[14] The event study specification is as follows:

$$y_{fot} = \sum_{\tau=-5}^{5} \beta_{\tau} D^{\tau}_{fot} + \mu_{fo} + \nu_t + \epsilon_{fot}, \tag{1}$$

where the outcome variable y_{fot} is the analytic or social skill requirement measure of job fo in year t. The coefficients of interest are β_{τ}, which are associated with the years since (or before) adoption, D^{τ}_{fot}. We include μ_{fo} and ν_t as job fixed effects and year fixed effects respectively, and ϵ_{fot} is the residual. As events are staggered across jobs, we estimate Equation 1 with the Callaway and Sant'Anna (2021) estimator in addition to two-way fixed effects.

Figures 1a and 1b present the results using TWFE for analytic and social skills respectively. In the appendix, we show the results using the Callaway and Sant'Anna (2021) estimator, as well as the results for administrative (routine cognitive) skill requirements – the results are qualitatively the same.

We make three observations. First, skill requirements increase approximately 1.8pp in the calendar year of the software adoption for analytic skills and 1.5pp for

[14]For example: if job fo adopted software in 2014, $D^{1}_{fot} = 1$ for $t = 2015$ only and $D^{-3}_{fot} = 1$ for $t = 2011$ only.

social skills. Second, the observed increase is temporary and disappears within 5 years. Third, there are positive pre-trends before a software adoption.

Our goal is to identify how much of the observed jump in skill requirements is causal. Regarding the second observation, skill requirements could fall if the software is no longer being used for the job, or if the positive pre-trends are driven by temporary external factors that subside after a few years. The pre-trends present a threat to identifying the causal impact of software and can be interpreted in two ways. The first possibility is that firms are increasing skill requirements in anticipation of the software adoption. In this case, the upward pre-trend would form part of the causal effect of software. However, this interpretation is less plausible, as anticipatory hiring should not start five years in advance.[15]

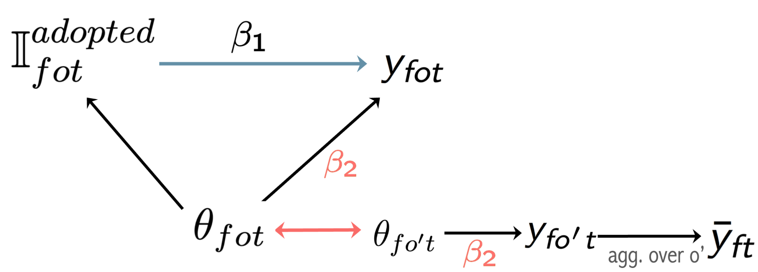
The more plausible explanation for the pre-trends is that the choice to adopt software is endogenous: software is adopted when skills requirements are rising. As we leverage variation from within jobs only, the event study estimates are not driven by cross-sectional differences across firms or occupations. The endogenous choice here is the *timing* of software adoption. For example, a shift in product demand or in the number of college-educated workers in the local labor force could prompt the firm to both adopt software and change skill requirements—causing a spurious correlation due to omitted variable bias. A discrete event like a management change right before the software is adopted would cause a similar bias, though it would not show up as a pre-trend.

3.5 Identification of causal effects

Our identification strategy, following Freyaldehoven et al (2020), supplements the within-job time variation used in the event study with information from skill requirements of other occupations in the same firm. The intuition here is as follows. Suppose we see skill requirements for sales and HR workers rise although only accountants adopted software; we conclude that some other factor affecting the firm caused sales and HR skill requirements to rise, and that factor should explain part of accountant's change in skill requirements too. We proceed in three steps: First,

[15] If part of the pre-trends is in fact driven by changing skill requirements in advance of adopting software, then our causal estimates of the next section would be a lower bound on the true causal effect of software on skill requirements.

Figure 2: Identification Argument Part 1



Notes: The blue arrow represents the effect of interest. Red arrows represent identification assumptions. The latent variable θ_{fot} causes omitted variable bias by affecting both the software adoption event and outcome. The latent variable is correlated with the proxy through the firm-level component of the latent variable.

we explicitly model the confounding factor as a latent variable; next we proxy for the latent variable with skill requirements in other occupations; then, we account for measurement error in our proxy using the lead of software adoption as an instrument.

For the remainder of this section, we focus on the two years directly before and after adoption, as our goal is to understand how much of the observed upskilling is causal. We re-write Equation 1 as a staggered difference-in-difference equation, pooling together years $\tau = -1$ and -2 as 'before adoption' and years $\tau = 0, 1$ and 2 as 'after adoption'.

$$y_{fot} = \beta_1 \mathbb{I}_{fot}^{adopted} + \mu_{fo} + \nu_t + u_{fot}, \tag{2}$$

where $\mathbb{I}_{fot}^{adopted}$ is an indicator for whether the adoption had taken place before or during year t. $E[u_{fot}\mathbb{I}_{fot}^{adopted}]$ may not equal zero due to the presence of time-varying confounding factors. Let θ_{fot} be a *latent variable* that captures all such unobserved confounds that are correlated with both skill requirements and software adoption. The first step of our identification strategy is to re-write u_{fot} as:

$$u_{fot} = \beta_2 \theta_{fot} + \varepsilon_{fot},$$

such that by construction all potential confounds are included in θ_{fot} so $E[\varepsilon_{fot}\mathbb{I}_{fot}^{adopted}] = 0$. As seen in Figure 2, the latent variable θ_{fot} can correlate with both $\mathbb{I}_{fot}^{adopted}$ and the outcome y_{fot}, leading to omitted variable bias. A possible solution to this bias would be to find an instrument for the software adoption event $\mathbb{I}_{fot}^{adopted}$. An ideal instrument would be firm-time level variation in the price of software, or availability of new software whose suitability differs across firms. However, this type of data is generally not available at the firm level. Instead, we will use a proxy-variable strategy to measure for the unobserved latent variable and control for it in equation (2).

Our second step is to proxy for θ_{fot} with skill requirements in other, non adopting occupations in the same firm. To construct our proxy measure, we start with all vacancies from firms where we observe at least one adoption, but from occupations where we never observe *any* software adoption in the 2-digit SOC occupation that the vacancy is from. For example, if a firm posts vacancies for accountants, sales and HR workers and we observe a software adoption for accountants only, we would use sales and HR vacancies to construct our proxy variable. We regress skill requirements of these vacancies on occupation fixed effects. Our proxy, $\bar{y}_{ft}$, is the average of the residuals of this regression for each firm-year cell. Note that the proxy is measured at the firm-year level[16] of within-firm skill requirements in non-adopting occupations.[17]

Our proxy $\bar{y}_{ft}$ equals the latent variable plus measurement error, η_{fot}:

$$\bar{y}_{ft} = \psi\theta_{fot} + \eta_{fot}, \tag{3}$$

where, ψ would be the best linear predictor of $\bar{y}_{ft}$ on θ_{fot}, and therefore θ_{fot} is uncorrelated with η_{fot} by construction.[18] In order for ψ to be positive, we require $\bar{y}_{ft}$ to be positively correlated with the latent variable θ_{ft}. This is true if θ_{fot} includes a firm-level component that affects all occupations.

Assumption 1

$$\theta_{fot} = \theta_{ft} + \tilde{\theta}_{fot}, \text{ s.t. } \mathrm{Corr}(\tilde{\theta}_{fot}, \tilde{\theta}_{fo't}) = 0.$$

We assume that θ_{fot} can be split into a firm-level component θ_{ft} and a job-level

[16]The set of non-adopting occupations is constant within a firm, so $\bar{y}_{ft}$ does not depend on the adopting occupation o.

[17]As a special case, if there was only 1 occupation o' which did not adopt software, then $\bar{y}_{ft}$ would equal $y_{fo't}$.

[18]While θ_{fot} is uncorrelated with the measurement error term η_{fot} by construction, θ_{ft} and $\tilde{\theta}_{fot}$ can individually correlate with η_{fot}.

term $\tilde{\theta}_{fot}$, uncorrelated across occupations. We require that there is some variation in the θ_{ft} component across firm-years. For example, firms may face management or labor supply changes that affect all occupations. As seen in Figure 2, this assumption implies that θ_{fot} is correlated with the latent variable of other, non-adopting occupations $\theta_{fo't}$ through the common component θ_{ft}. In turn, $\theta_{fo't}$ affects skill requirements of occupation o' by definition of the latent variable. We require that the effect of the latent variable on skill requirements, β_2, does not vary by occupation.[19]

Under Assumption 1, $\psi \neq 0$. Inserting the proxy into Equation (2) we obtain

$$y_{fot} = \beta_1 \mathbb{I}^{adopted}_{fot} + \mu_{fo} + \nu_t + \frac{\beta_2}{\psi}(\bar{y}_{ft} - \eta_{fot}) + \varepsilon_{fot}, \tag{4}$$

The source of bias is now the measurement error η_{fot}. If we were to estimate specification Equation (4) using OLS, measurement error in $\bar{y}_{ft}$ would underestimate β_2 due to attenuation bias, and thus ascribe part of the variation in the latent variable to the software adoption event instead. This would cause us to overestimate the parameter of interest β_1.

This brings us to the third step. Following Freyaldehoven et al (2020), we instrument for the proxy variable $\bar{y}_{ft}$ using the lead of software adoption event. To avoid confusing our instrument $\mathbb{I}^{adopted}_{fj,t+1}$ with $\mathbb{I}^{adopted}_{fj,t}$, we denote the lead of software adoption by z_{fot}:

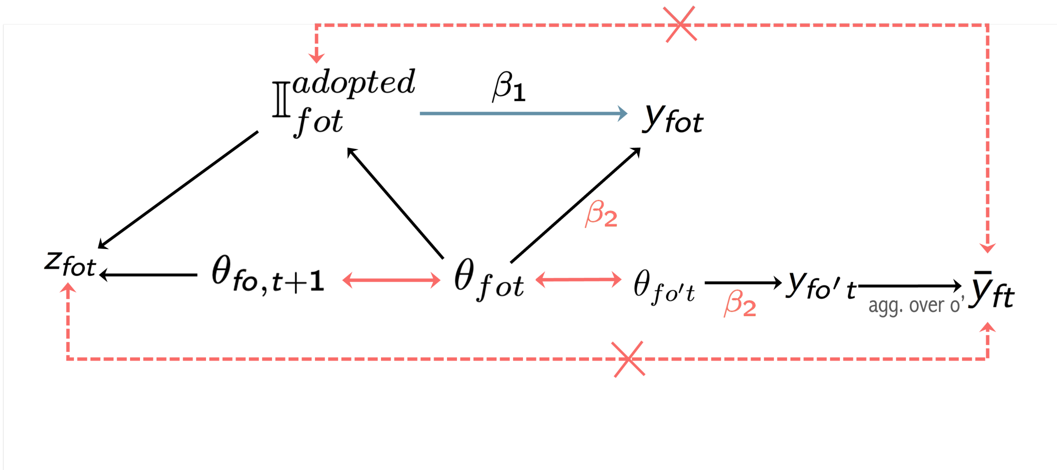
$$z_{fot} \equiv \mathbb{I}^{adopted}_{fj,t+1}.$$

The intuition is as follows. Suppose, for example, the software is adopted the first time the latent variable crosses a threshold θ^*.[20] We observe a firm that has not adopted software by time $t+1$, but does adopt software at time $t+1$. We know that $\theta_{fo,t+1} > \theta^*$. If we believe the latent variable is autocorrelated over time, this implies

[19] This is not a particularly strong assumption because $\tilde{\theta}_{fot}$ is allowed to vary across occupation. For example, a management change could affect sales occupations more than administrative; this would lead to an increase in both θ_{ft} and the idiosyncratic part for sales $\tilde{\theta}_{fot}$.

[20] This is a special case as software adoption *only* depends on θ_{fot}. In reality, the latent variable will not contain all the factors related to software adoption—only those also associated with skill requirements—so the latent variable may be correlated but not perfectly correlated with the timing of software adoption, but this argument still holds.

Figure 3: Identification Argument Part 2



Notes: Blue line indicates effect of interest; red lines indicate identification assumptions. Refer to Figure 2 for Part 1 of the identification argument. The IV (lead of software adoption) is correlated with the latent variable (and consequently the proxy) through autocorrelation of the latent variable. Exclusion restrictions are marked in dotted red lines: the proxy cannot be correlated with current or future software adoption status except through the latent variable

θ_{fot} must already be close to the threshold θ^* (as compared to a similar firm that did not adopt software by $t+1$). Therefore, conditional on not having adopted software yet ($\mathbb{I}^{adopted}_{fot} = 0$), whether a firm adopts software at time $t+1$ ($z_{fot} \equiv \mathbb{I}^{adopted}_{fo,t+1} = 1$) is correlated with time t latent variable θ_{fot}. Under Assumption 1, the instrument z_{fot} is correlated with the proxy $\bar{y}_{ft}$ (the endogenous regressor) through θ_{fot}.

We now formally state the relevance assumptions and exclusion restrictions. For z_{fot} to be a relevant instrument, it must be correlated with $\bar{y}_{ft}$ - and therefore with the latent variable θ_{fot}. We assume that θ_{fot} is positively autocorrelated:

Assumption 2

$$Corr(\theta_{fot}, \theta_{fo,t+1}) = Corr(\theta_{ft}, \theta_{f,t+1}) = Corr(\tilde{\theta}_{fot}, \tilde{\theta}_{fo,t+1}) > 0$$

For example, the latent variable will be autocorrelated if the underlying productivity of the firm is trending upward, or college educated workers are moving into the local labor market over time. Assumption 2 would be satisfied if the latent variable was, for example, an AR(1) process, but also allows for more general formulations like jumps in the latent variable. The pre-trends we observed in our event studies support this assumption: We observe skill requirements steadily rising before the software adoption, suggesting some driver of skill requirements—the latent variable—is steadily increasing. Assumption 2 also requires that the firm-specific and job-specific components of the latent variable have the *same* autocorrelation. This part of the assumption is required to ensure that not only the current latent variable, but also the next year's latent variable, is uncorrelated with the measurement error η_{fot}. This assumption would be violated if, for example, management changes that affect individual occupations occur more frequently than those that affect the whole firm.[21]

As seen in Figure 3, under Assumption 2 the current latent variable is correlated with the latent variable next year $\theta_{fo,t+1}$, which is correlated with whether the job has adopted software by time t+1, z_{fot}.[22] Therefore, the instrument z_{fot} is correlated with the latent variable θ_{fot}, and consequently with the proxy $\bar{y}_{ft}$.

For our instrumental variable strategy to causally identify the impact of software adoption, z_{fot} must be uncorrelated with the error terms η_{fot} and ε_{fot}. $E[z_{fot}\varepsilon_{fot}] = 0$ by construction: any components of the original error term u_{fot} that influence software adoption are part of θ_{fot}, not ε_{fot}[23]. z_{fot} will be uncorrelated with η_{fot} if z_{fot} is only related to the proxy $\bar{y}_{ft}$ through the latent variable pathway, as seen in Figure 3.

Assumption 3: Exclusion restriction

$$E[\bar{y}_{ft}|z_{fot}, \theta_{fot}] = 0$$

$$E[\bar{y}_{ft}|\mathbb{I}^{adopted}_{fot}, \theta_{fot}] = 0$$

[21] As noted before, θ_{fot} is uncorrelated with the measurement error term η_{fot} by construction but θ_{ft} and $\tilde{\theta}_{fot}$ can individually correlate with η_{fot}. If $\theta_{fo,t+1}$ depends directly on θ_{fot} rather than on its components separately then $E[\theta_{fo,t}\eta_{fot}] = 0 \Rightarrow E[\theta_{fo,t+1}\eta_{fot}] = 0$. For example, in the AR(1) case: if $Corr(\theta_{ft}, \theta_{f,t+1}) = Corr(\tilde{\theta}_{fot}, \tilde{\theta}_{fo,t+1})$ then θ_{ft} and $\tilde{\theta}_{fo,t+1}$ have the same autocorrelation parameter ρ_θ so $\theta_{fo,t+1} = \rho_\theta theta_{fot} + u_\theta$

[22] If the last part of this argument does not hold—that is, the latent variable at time t+1 does not affect software adoption at t+1, then the same would be true for time t and there would be no endogeneity.

[23] It may remain a concern that ε_{fot} may be correlated with software adoption next period even if it is not correlated with software adoption this period: for example if a firm chooses to adopt software in time t+1 based on skills of workers hired at time t. But θ_{fot} can be constructed to incorporate all these factors as well: in the extreme case, we can have $\theta_{fot} = u_{fot}$, so $\varepsilon_{fot} = 0$.

In particular, Assumption 3 requires that software adoption does not causally impact skill requirements in other occupations. For example, a team of accountants adopting Quickbooks should not directly affect the skill requirements of sales workers. The exclusion restriction allows for software adoption by the accountants to change the total labor demand for sales workers, or the demand for sales workers relative to HR workers; we only require that the skill requirements within each 6-digit SOC sales occupation do not change.

Task-trading between occupations would violate Assumption 3: for example, if accounting workers take over some tasks of sales workers after adopting software. As task-trading should be easier within related occupations, when constructing the proxy we left out occupations in the same 2-digit SOC category as adopting occupations. Assumption 3 would also be violated if the 'non-adopting' occupations did, in fact, adopt software although we do not observe the adoption event. This is possible if the software adopted was not mentioned in vacancies, or the adoption event did not meet our criteria. In this case, we may wrongly ascribe variation in skill requirements to a latent variable when in fact they were due to a software adoption event.[24] If the exclusion restriction was violated, we expect it to be because software adoption *increases* skill requirements in other occupations.[25] If so, we would overstate the role of the latent variable and *understate* the causal impact of the software by attributing all variation in skill requirements of non-adopters to θ_{fot}. So, our estimates would be a lower bound on the true magnitude of software impacts.

3.6 Results

Table 2 reports the estimated effect of software on skill requirements, β_1, for analytic and social skills. We also report OLS estimates for comparison. The reported results use a proxy variable $\bar{y}_{ft}$ constructed from social skill levels in non-adopting occupations since the first stage is weak when the proxy is constructed with analytic skills; we report results using alternative forms of the proxy variables in Appendix C.4.

[24] Technically speaking, for this scenario to violate causality it must also be the case that software adoptions decisions within a firm are not independent.

[25] We observe increases skill requiremnts in other occupations in OLS regressions around the time of adoption (which we attribute to the latent variable). Papers like Bessen, Denk and Meng (2022) also argue for a skill increase.

Table 2: Causal (IV) v.s.OLS Estimates of Software Adoption on Skill Requirements

	Analytic OLS	Analytic IV	Social OLS	Social IV
Post adoption indicator	0.0139***	0.00817***	0.0157***	0.0109***
	(0.000905)	(0.00213)	(0.000977)	(0.00167)
Latent variable proxy		0.887***		0.673***
		(0.290)		(0.222)
N	1391518	1344263	1391518	1344263

Standard errors in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Notes: Causal estimates through latent variable IV strategy, compared to naïve OLS results. Results can be interpreted as change in the probability of mentioning the skill requirement.

We find that a software adoption event increases the likelihood of requiring analytic skills by 0.8pp and increases the likelihood of requiring social skills by 1.1pp. These effects are significant at the 5% level.

The OLS coefficients are slightly larger at 1.3pp and 1.5pp respectively; the difference between the OLS and IV estimates can be attributed to endogeneity in the time of software adoption.

The magnitude of the effect of the proxy variable $\bar{y}_{ft}$ does not have an interpretation but we note it is positive as expected, confirming that the latent variable is positively associated with skill requirements. To put our results into perspective, each software adoption event increases social skill requirements by to 7% of the standard deviation in the 2-digit SOC occupation social skill distribution. The impact of a software adoption is approximately 18% of the difference between average social skill requirements for management and STEM jobs. The effect on analytic skills is smaller, at 4% of a standard deviation. We show results for other requirements (management and IT skills, education, experience) in Appendix C.2

To test how these results depends on how much software of different types the job was previously using, we approximate the previous software level of the firm by the number of software types ever mentioned, and find that the impact of software adoption is the strongest when the job previously uses less than two other types of software. We show these results in Appendix C.3. The first two software types used by a firm increase social skill requirements by 1.8pp and analytic skill requirements by 1.2pp, while the effects of later adoptions are generally not significant.

In Appendix D, we estimate (2) separately by software type with OLS and find suggestive evidence that the impact of software on analytic and social skills is heterogeneous by the type of software adopted.[26]

3.7 Effects on labor demand

We estimate the impact of software adoption on the quantity of labor demanded – proxied by the number of vacancies posted - for the software-adopting occupation as well as non-adopting occupations in the firm. To do so, we follow the same identification strategy, estimating Equation 2 using the log of the number of vacancies as the outcome variable in place of skill requirements. The interpretation of the latent variable changes: the latent variable now includes unobserved factors like an increase in total product demand, that may not affect skill requirements but do affect labor demand. We use the same proxy variable as before, based on skill requirements from non-adopting occupations, (rather than the number of vacancies), to allow for software to causally affect the number of vacancies in non-adopting occupations as well. We find that a software adoption event increases the log of the desired number of vacancies by 0.26 on average, which corresponds to a $\sim 30\%$ increase in hiring. We also find evidence of smaller increases of 5% on other, non-adopting occupations in the firm.

4 Equilibrium effects of software

We have shown that, for a single firm, adopting software increases skill requirements and the quantity of labor demanded. We now turn to the equilibrium impacts of software on wages. How has the declining price of software and the associated increase in software investment affected the wage distribution and inequality?

[26]We use OLS because the smaller sample sizes do not have enough power for the causal identification strategy.

Table 3: Causal estimates of software adoption on the number of vacancies

	Log vacancies: Adopters	Log vacancies: Non-adopters
Post adoption	0.257***	0.0520***
	(0.0240)	(0.0114)
Latent variable proxy	13.98***	5.600***
	(3.046)	(1.504)
N	1350004	1349205

Standard errors in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Notes: Causal estimates of software adoption on the log of the number of vacancies posted annually, in the software adopting occupation and non-software adopting occupations. Effects translate to a $\sim 30\%$ increase for adopters and $\sim 5\%$ increase for non-adopters.

The answer to this question depends on which occupations adopt software, and how software impacts relative labor demand across occupations. Furthermore, higher skill requirements induced by software can create wage differentials within occupations and can force workers to move to lower wage occupations. For example, a worker may be qualified for a sales job without software but not with software; if most firms adopt software for sales then the demand for workers in non-software sales jobs will fall, and she would choose between receiving a lower wage in sales or choosing a different occupation.

To estimate the net effect of software on the wage distribution, we now embed our partial equilibrium results of the previous section into a static equilibrium model where workers sort into jobs based on skill requirements and firms' software use.

4.1 Theory

4.1.1 Environment

The economy consists of a unit measure of perfectly competitive firms (indexed by $f \in [0,1]$) of and a continuum of workers (indexed by i) of mass n_w. Firm output q_f is an aggregate of four (broad) occupations $J \equiv \{$Managerial, STEM, Sales, Administrative$\}$ These are chosen to be white-collar occupations prevalent across industries.[27] We focus on white-collar occupations because our data are not suited to measuring changes in manual skills, which would be an important skill dimension for other occupations. We index these occupations by j rather than o to highlight that these are broad occupation groups, not the 6-digit SOC occupations used in the previous sections. Each occupation $j \in J$ in firm f uses labor l_{fj} and optionally software s_{fj} to produce t_{fj} units of occupation output.

Each occupation is characterized by a vector of *skill requirements* $\boldsymbol{y}_{fj} \in \mathbb{R}^2$ needed to perform its associated tasks. We assume $\boldsymbol{y}_{fj}$ is two-dimensional, specifying analytic and social skills respectively. For example, the STEM occupation may require high analytic skills, while the management occupation may require relatively high analytic and social skills levels.

Worker i is endowed with skill bundle $\boldsymbol{x}_i \in \mathbb{R}^2$. Analogous to the skill requirements, skill bundles are two-dimensional, specifying the worker's analytic and social skill levels. The joint distribution of analytic and social skills over n_w is given by distribution $X(\boldsymbol{x})$.

4.1.2 Skill Requirements and Software

A worker with skills $\boldsymbol{x}_i$ is *qualified* for occupation j in firm f if and only if $\boldsymbol{x}_i \geq \boldsymbol{y}_{fj}$ in a vector sense. We assume an underqualified worker—in either dimension—cannot perform the occupation. For example, a worker with skill bundle $\boldsymbol{x}_i = (1, 2)$ would be qualified to perform occupations $\boldsymbol{y}_{fj} = (1, 1)$ and $\boldsymbol{y}_{fj} = (1, 2)$, but not $\boldsymbol{y}_{fj} = (2, 1)$. We assume that if a worker is overqualified in any or all dimensions of the occupation, they may choose to work in the occupation but are no more productive than an exactly qualified worker. For example, a worker with skill bundle $\boldsymbol{x}_i = (2, 2)$ and a worker with skill bundle $\boldsymbol{x}_i = (1, 1)$ would be equally productive at occupation $\boldsymbol{y}_{fj} = (1, 1)$. We make this assumption because our data does not contain allow us to observe how much a mismatched worker earns at a job.

While our framework allows occupation-specific skill requirements to impact occupation productivity, skill requirements will not directly enter the production function. Instead, skill requirements impact the economy through labor supply, as workers can only choose a job they are qualified for. Software adoption increases the share of soft-

[27] More specifically, we do not include medical, legal, or education occupations that are concentrated in specific industries

ware jobs with relatively higher skill levels $\boldsymbol{y}_j^s$, which shifts labor demand toward the smaller set of workers qualified for these jobs. In equilibrium, this impacts wages for each occupation-software pair and consequently labor demand and software adoption decisions.

A key feature of the model is that software use of job fj affects its skill requirement $\boldsymbol{y}_{fj}$. Let $\mathbb{I}_{fj}^{sw}$ be an indicator for whether firm f chooses to use any software for occupation j.

Skill requirements $\boldsymbol{y}_{fj}$ depend on software use $\mathbb{I}_{fj}^{sw}$:

$$\boldsymbol{y}_{fj} = \begin{cases} \boldsymbol{y}_j^s & \text{if } \mathbb{I}_{fj}^{sw} = 1 \\ \boldsymbol{y}_j^{ns} & \text{if } \mathbb{I}_{fj}^{sw} = 0 \end{cases} \tag{5}$$

While equation (5) allows software to increase or decrease the skill requirements in both analytic and social dimensions, we know from our empirical results that skill requirements increase; that is, $\boldsymbol{y}_j^{sw} \geq \boldsymbol{y}_j^{ns}$. Note that $\boldsymbol{y}_{fj}$ only depends on f through the software choice. For a given occupation j, there is one set of skill requirements common across all firms that use software, and another skill requirement for all firms that do not use software. Therefore, from here on, we will write $\boldsymbol{y}_{fj}$ as $\boldsymbol{y}_{\mathbb{I}_{fj}^{sw},j}$ to highlight that conditional on a firm's software use in occupation j, skill requirements are not firm specific.

4.1.3 Nested CES production

Each occupation within a firm produces output t_{fj} according to

$$t_{fj} = \left(\alpha_j (s_{fj} + b_j)^{\frac{\gamma_j - 1}{\gamma_j}} + l_{fj}^{\frac{\gamma_j - 1}{\gamma_j}} \right)^{\frac{\gamma_j}{\gamma_j - 1}}, \tag{6}$$

where l_{fj} and s_{fj} are the quantity of labor and software, respectively, employed by firm f in occupation j. Occupation-specific elasticity of substitution between labor and software is denoted by γ_j. This specification nests specific cases: for $\gamma \to \infty$, software and labor are perfect substitutes within occupations, so software could fully automate the occupation; for $\gamma \to 0$ software and labor are perfect complements, as if each worker required her own software license; for $\gamma \to 1$, labor and software are inputs into a Cobb-Douglas technology and changes in software prices do not directly affect labor demand. A baseline technology level, b_j, can be interpreted as computer hardware and generates non-homotheticity in the demand for software. We introduce b_j to ensure producing without software is feasible even when software and labor are complements.

Firm output q_f is a CES aggregate of occupation output t_{fj}.

$$q_f = A\left(\sum_{j\in J}\omega_j t_{fj}^{\frac{\sigma-1}{\sigma}}\right)^{\frac{\sigma\delta}{\sigma-1}}, \tag{7}$$

where σ is the elasticity of substitution across occupations and ω_j denote occupation weights. The firm has factor neutral TFP A and is characterized by returns to scale $\delta < 1$.

Skill requirements $\boldsymbol{y}_{fj}$ do not directly enter production, but occupations with higher skill requirements may be more productive—this would be captured by ω_j.[28] The occupation weight ω_j incorporates occupation productivity, the importance of the occupation to production, and also scale effects from changing the elasticity of substitution.[29]

4.1.4 Profit maximization

Firms choose the quantity of labor l_{fj} and software s_{fj} for each occupation, taking prices as given. For software, the firm makes both an extensive margin decision—which occupations, if any, to use software for—and, for occupations using software, an intensive margin decision of how much software to use. Firms pay wages $w_{fj}(\boldsymbol{x}_i)$ per unit of labor of type $\boldsymbol{x}_i$ in occupation j. The cost of software for each occupation has a fixed component p^{fix} that must be paid if any software is used and a variable component p^{var} per unit of software.

Firms are heterogeneous in their profitability of software for each occupation. Each firm receives a software profitability shock $\kappa_{fj} \sim \text{EV}(1)$, with variance $\varkappa$, for

[28]Between ω_j and α_j and A, the production function is flexible enough to incorporate differences in software and labor productivity across occupations.

[29]Changing γ_j affects the elasticity of substitution between workers and software but also the scale of occupational production. By allowing the occupation weights to subsume this scale effect, we ensure the γ_j parameters are estimated to match the correct elasticity of substitution and not distorted by scale. The same holds for σ and A.

each occupation. The software profitability shock κ_{fj} is the only source of heterogeneity between firms, and can be interpreted as a firm's suitability to software or heterogeneity in the price of acquiring software.

The firm chooses software and labor for each occupation j to maximize profits according to

$$\max_{(s_{fj},l_{fj}(\cdot))} q_f(s_{fj}, l_{fj}(\boldsymbol{x}_i)) - \sum_{j\in J}\sum_{\boldsymbol{x}_i\in X} w_{fj}(\boldsymbol{x}_i) l_{fj}(\boldsymbol{x}_i) - \sum_{j\in J}(p_j^{fixed}\mathbb{I}_{fj}^{sw}) - \sum_{j\in J} p_j^{var} s_{fj} + \kappa_{\mathbb{I}^{sw}}, \tag{8}$$

where the firm chooses the number of workers of type $\boldsymbol{x}_i$ subject to their wage in that occupation $w_{fj}(\boldsymbol{x}_i)$; it also makes an extensive and intensive margin software decision for each occupation subject to (7).

Proposition: Conditional on a worker being qualified for a job, the equilibrium wage w_{fj} paid by firm f for occupation j does not depend on the worker's type $\boldsymbol{x}_i$

Proof: By assumption, overqualified workers are not more productive, so all qualified workers are perfect substitutes within a job. If there were two qualified worker types $\boldsymbol{x}'$ and $\boldsymbol{x}''$, such that $w_{fj}(\boldsymbol{x}') < w_{fj}(\boldsymbol{x}'')$, the firm would strictly prefer to hire $\boldsymbol{x}'$ at $w_{fj}(\boldsymbol{x}')$, as they can hire any quantity of labor at the lower equilibrium price.

Since, conditional on working in firm f occupation j, both wages and worker productivity do not depend on worker type $\boldsymbol{x}$, the firm is indifferent when hiring workers of all qualified skill types. Therefore we can simplify the firm's problem to

$$\max_{(s_{fj},l_{fj}),j\in J} q_f(s_{fj}, l_{fj}) - \sum_{j\in J} w_{fj} l_{fj} - \sum_{j\in J}(p_j^{fixed}(s_{fj} > 0)) - \sum_{j\in J} p_j^{var} s_{fj} + \kappa_{\mathbb{I}^{sw}}, \tag{9}$$

where l_{fj} is the number of *qualified* workers the firm chooses to hire for occupation j and w_{fj} is the unique[30] market wage for the occupation-software group.

[30] As all qualified workers are equally productive and there are no search frictions, the market wage must be unique by occupation-software group.

4.1.5 Workers

Workers derive utility u_i from wages and idiosyncratic preferences $\zeta_{ij\mathbb{I}_j^{sw}} \sim \text{EV}(1)$, scale ρ, over occupations and software use (not individual firms).

$$u_{ifj} = w_{fj} + \zeta_{ij\mathbb{I}_j^{sw}}. \tag{10}$$

Proposition: Wages only depend on the occupation j and software use of the occupation $\mathbb{I}_j^{sw}$, and not directly on the firm.

Proof: Firms cannot set different wages for the same occupation-software pair because workers do not have preferences over individual firms, so conditional on the type of job, they will always choose the highest wage. Firms have no incentive to raise wages as they can hire any number of workers at the equilibrium wage rate, and no incentive to unilaterally undercut because no worker would choose their firm.

Workers choose the occupation j they will work in, as well as whether they to work in a firm with or without software for occupation j, according to:

$$\max_{\mathbb{I}_j^{sw},j} w_{\mathbb{I}_j^{sw},j} + \zeta_{i,\mathbb{I}_j^{sw},j} \tag{11}$$

$$\text{s.t. } \boldsymbol{x}_i \geq \boldsymbol{y}_{j\mathbb{I}_j^{sw}},$$

where $w_{j\mathbb{I}_j^{sw}}$is the wage of the worker. The values of $\zeta_{ij\mathbb{I}_j^{sw}}$ are independent for each worker. The worker can choose any occupation-software combination subject to the restriction that she must be qualified for the occupation—that is $\boldsymbol{x}_i \geq \boldsymbol{y}_{j,\mathbb{I}^{sw}}$. A worker may choose an occupation she is overqualified for if she has a strong preference for the occupation.

4.1.6 Equilibrium

In equilibrium, a firm f chooses s_{fj} and qualified labor l_{fj} for each occupation j to maximize profits, given wages and software prices; workers choose an occupation-software pair to maximize their utility, and wages adjust such that labor demand equals labor supply for each occupation-software pair.

Equilibrium wages equalize supply and demand for each occupation-software pair. As shown earlier, there is a unique wage for each combination of occupation and software use. For example, there is a wage attributed to management workers in jobs not using software, and to STEM workers in jobs using software. On average, workers with higher skills in either dimension will receive weakly higher utility because they are qualified for more occupations[31].

Let l^*_{fj} denote the optimal demand for qualified workers by firm f, for occupation j. Then the total quantity of labor demanded for each occupation-software pair $l_{j,\mathbb{I}^{sw}}$ is given by:

$$l_{j,\mathbb{I}^{sw}_j} = \int_{[0,1]} l^*_{fj} \mathbb{I}^{sw}_{fj} df \tag{12}$$

That is, the total labor demand in the economy for an occupation -software pair $(j, \mathbb{I}^{sw})$ is total labor demand for occupation j over the continuum of firms, separately by the firms' software choice for occupation j $\mathbb{I}^{sw}_{fj}$.

To calculate labor supply to each occupation-software pair, we first calculate the labor supply separately by worker type $\boldsymbol{x}_i$. By the properties of the Gumbel distribution, the proportion of workers of type $\boldsymbol{x}_i$ in each occupation is given by:

$$l^{supply}_{j\mathbb{I}^{sw}_j}(\boldsymbol{x}_i) = \frac{exp(w_{j\mathbb{I}^{sw}_j}/\rho)\mathbb{I}(\boldsymbol{x}_i \geq \boldsymbol{y}_{j\mathbb{I}^{sw}_j})}{\sum_j exp(w_{j\mathbb{I}^{sw}_j}/\rho)\mathbb{I}(\boldsymbol{x}_i \geq \boldsymbol{y}_{j\mathbb{I}^{sw}_j})}, \tag{13}$$

which is the extreme value choice outcome with the added stipulation that the worker's choice set is determined by her skills.

The total labor supply to each occupation-software pair $l^{supply}_{j\mathbb{I}^{sw}_j}$ is the sum of the skill-specific labor supplies $l^{supply}_{j\mathbb{I}^{sw}_j}(\boldsymbol{x}_i)$ over all skill types $\boldsymbol{x}_i$:

[31] This is true because the choice set of a worker can only grow with skills. We say 'weakly increasing' because differences in skill levels do not matter unless they cross the skill requirements threshold of an occupation. We say 'on average' because each worker's total utility also depends on the magnitude of her idiosyncratic preference shock. We do not say that *wages* (weakly) increase with skills, because depending on the set of occupations this may not be true: if there is an occupation with very low labor demand and high qualification requirements: some high skilled workers with a strong preference for that occupation will work there for lower wages. In our empirical application with broad occupational groups, wages will rise with skills

$$l^{supply}_{j\mathbb{I}_s} = n_w \sum_{\boldsymbol{x}_i} \left(l^{supply}_{j\mathbb{I}^{sw}_j}(\boldsymbol{x}_i) \times X(\boldsymbol{x}_i) \right) \tag{14}$$

In equilibrium, $l^{supply}_{j\mathbb{I}^{sw}_j} = l^{demand}_{j\mathbb{I}^{sw}_j}$ for all occupation-software groups $(j, \mathbb{I}^{sw}_j)$

We can numerically compute the equilibrium as follows. Given an initial guess for wages, we solve the firm's problem in equation (9) into two steps. First, for every possible extensive margin choice of software across occupations, we compute the optimal labor and, where relevant, the quantity of software inputs for each occupation.[32] For example, if we are calculating the optimal profit in the case where the firm chooses to use software for occupation 1 and not occupation 2, we would require the firm to pay the fixed cost for occupation 1 software and use the higher wage rate associated with software jobs, and then choose inputs quantities freely given wages and variable costs; for occupation 2 the occupation would not need to pay the fixed software cost and could choose labor according to the non-software wage rate, but would not be allowed to use any software. Knowing the optimal inputs for each extensive margin choice, we can compute the related firm profit, in the absence of the software profitability shock κ_{fj}. As κ_{fj} only impacts the extensive margin choice of using software, it will not affect optimal inputs at this stage.

The second step is to choose the profit-maximizing extensive margin software choice for each firm. As κ is distributed Extreme Value Type 1 (Gumbel), we can calculate the proportion of firms choosing each software combination using the extreme value choice formula. This gives us the total labor and software demand by each firm.

We aggregate software and labor demand across firms for each occupation-software pair. For example: suppose 20% of firms use software for occupation 1 only and hire 10 workers each, and 40% of firms use software for both occupation 1 and occupation 2 and hire 20 workers each. The total labor demand across the unit mass of firms for occupation 1 -software using jobs will be $0.2 \times 10 + 0.4 \times 20 = 10$.

To compute labor supply, we discretize the skill distribution on a two-dimensional grid. For each skill-type, we find the set of occupation-software pairs the worker

[32]For example, in the 2-occupation case there are 4 possibilities: could be software for both occupations, software for the first occupation only, software for the second occupation only, and software for both occupations. For 4 occupations there are $4^2 = 16$ possibilities.

qualities for. Within this set, we again employ the properties of the Gumbel distribution to compute the proportion of workers who choose each occupation-software pair under the current wage schedule. We compute total labor supply for each occupation-software pair by aggregating over the skill distribution, given ex-ante.

We iterate on wages separately for each occupation-software pair until labor demand and labor supply coincide for each pair, indicating the labor market is at equilibrium.

4.2 Estimation

We estimate the model in three steps. First, we use the Lightcast data, in conjunction with the Occupational Employment and Wage Statistics (OEWS), to measure skill requirements, equilibrium wages, equilibrium employment shares and the returns to scale parameter δ. Second, we estimate the remaining demand-side parameters with GMM using a global optimization algorithm, under the assumption that the economy is in equilibrium. The last step is to back up the underlying worker skill distribution $X(\boldsymbol{x})$ and the variance of the preference shock ζ that are consistent with the equilibrium allocation and qualification constraints.

We are currently estimating the model with two occupations, grouping managerial and STEM jobs (occupation 1) and sales and administrative jobs (occupation 2). We first calibrate analytic and social skill requirements for each occupation with and without software. We choose $\boldsymbol{y}_j^{ns}$ for each occupation to be the mean skill level in the Lightcast data for all jobs-years in the occupation that do not use software. We compute the skill levels with software $\boldsymbol{y}_j^s$ by adding our causal estimates of software adoption on skill requirements to each non-software skill measure.

We measure equilibrium wages and labor demand using both Lightcast and external data sources. We set non-software wages for each occupation to be mean wages from the OEWS.[33] From the Lightcast data, we estimate the ratio of software to non-software full-time salaries for each occupation.[34] We obtain wages for software jobs by multiplying the OEWS wages by the corresponding occupation wage ratio.[35]

[33]Taking the employment-weighted mean of all 2-digit occupations that comprise each broad occupational category.

[34]If salaries are given as a range, we take the mean.

[35]This procedure results in slightly more wage variation across occupations than if we had used

We normalize the wages for occupation 1 with no software to equal 1.[36]

We obtain relative occupational employment shares from OEWS and the share of software-using firms for each occupation from Lightcast. Taken together, these give the equilibrium employment share of each occupation-software group.

We calibrate the returns to scale parameter δ to 0.8 and normalize the base technology level b to equal 1,[37] so the quantity of software s_{fj} can then be interpreted as relative to the base level of technology. We also normalize the first occupational weight ω_1 and software productivity α_1 to 1, without loss of generality.[38]

The remaining parameters to estimate by GMM are: occupation-specific elasticity of substitution between software and labor γ_j; elasticity of substitution across occupations σ; TFP A; occupational weight ω_2; software costs p^{fix} and p^{var}; relative software productivity α_2; and variance of the software profitability shock $\varkappa$.

For each set of potential parameter values, we solve the firm's profit maximization problem as described in in Section 4.1.6. We then compute the following moments for which we have empirical counterparts: relative labor demand for software vs. non-software jobs by occupation; relative labor demand over occupations, the share of employment and firms in each occupation using software, and the relative intensive margin software use. We use the genetic algorithm to find the set of parameter values that minimize the distance between the model and the data moments.

It is not possible to formally derive an identification argument as the parameters are all intricately related through the interconnected discrete choices in the model. We present below a heuristic argument of how our parameters are identified.

The elasticity of substitution between workers and software γ_j for each occupation j is pinned down by the increase in vacancy postings for jobs that use software, before and after the software adoption event. As γ_j can differ by occupation j, we estimate the causal effect of vacancies separately by occupation, and use each of these as

Lightcast only. We consider these measures more accurate as less than 20% of vacancies post wage rates.

[36]This normalization is convenient to compare software costs to wages, but means we cannot normalize the price of output to 1. The interpretation of A changes to include final good price.

[37]Under an interpretation of δ as driven by monopolistic competition instead of returns to scale, $\delta = 0.8$ corresponds to elasticity of substitution of 4 between firms.

[38]These normalizations are wlog through adjustments in A and p^{var} respectively. This can affect the interpretation of A and p^{var}; in particular, p^{var} should be considered a quality-weighted price index.

moments. As we observe an increase in the number of vacancies, we expect to find γ_j is less than one across occupations.

Given values of γ_j's, σ comes from changes in the labor demand of other occupations relative to the adopting occupations. While we observe increases in labor demand here too, the increase is smaller than for the adopting occupations; therefore, we expect $\sigma > 1$, to indicate substitution across occupations.

The occupation weight of the second occupation ω_2 is mainly determined by occupation employment shares in equilibrium.

We previously estimated wage differentials between occupations, normalizing the wage of non-software management workers to 1. Keeping these normalizations, we have three price parameters: p^{fix}, p^{var}, and A, which now incorporates both TFP and the price of the final good. The proportion of firms choosing to use software for each occupation informs the relative value of the software price parameters up^{fix} and p^{var} relative to wages and A.

Since the software cost parameters are constant across occupations, the weight of non-labor inputs α_j is pinned down by the intensive margin software choices across occupations. We approximate the intensive margin software choice by counting the number of software types used by each occupation, conditional on using any software. As we do not want to impose a scale on s_{fj}, we normalize this intensive margin choice to 1 for occupation 1 and only use information on the relative number of software types across occupations.

Given the other parameters, we can calculate a measure of firm profitability without the shock κ, for each possible extensive margin software decision. Comparing these profits with realized software use determines the variance of κ. Lastly, we set the total mass of workers n_w equal to the total labor demand in equilibrium.

On the labor supply side, we parameterize the skill distribution as bivariate normal:

$$X(\boldsymbol{x}) = N(\mu^{ls}, \sigma^{ls}),$$

where we assume no correlation between analytic and social skills and normalize the variance of analytic skills to 1. Given wages, if we choose parameters for the skill distribution and the variance of the preference shock, we can compute the total employment share for each occupation-software pair using Equations 13 and 14.

We choose these parameters to match the equilibrium employment shares for each occupation-software pair at equilibrium wages.

4.2.1 Results and model fit

Our main parameters of interest are the elasticity of substitution between labor and software for each occupation, γ_1 and γ_2. We find that software and labor are complements for managerial/ STEM workers, with $\gamma_1 = 0.75$. Sales/ administrative workers are weaker complements to software, with $\gamma_2 = 0.92$. We find evidence that occupations are substitutes, with elasticity of substitution $\sigma = 2.96$ We report our parameter estimates in Table 5.

As seen in Figure 4, our model fits almost exactly the effect of software on labor demand by occupation, though we slightly underestimate the effect of software adoption in occupation 2 on labor demand for occupation 1. We also closely match the share of firms using software in each occupation, at 55% and 38% respectively. However, we overestimate the employment share of occupation 2 (0.66 vs 0.55) and the relative software intensity s_{fj} of occupation 2 (0.38 vs 0.32).

4.3 Effect of a Fall in Software Price

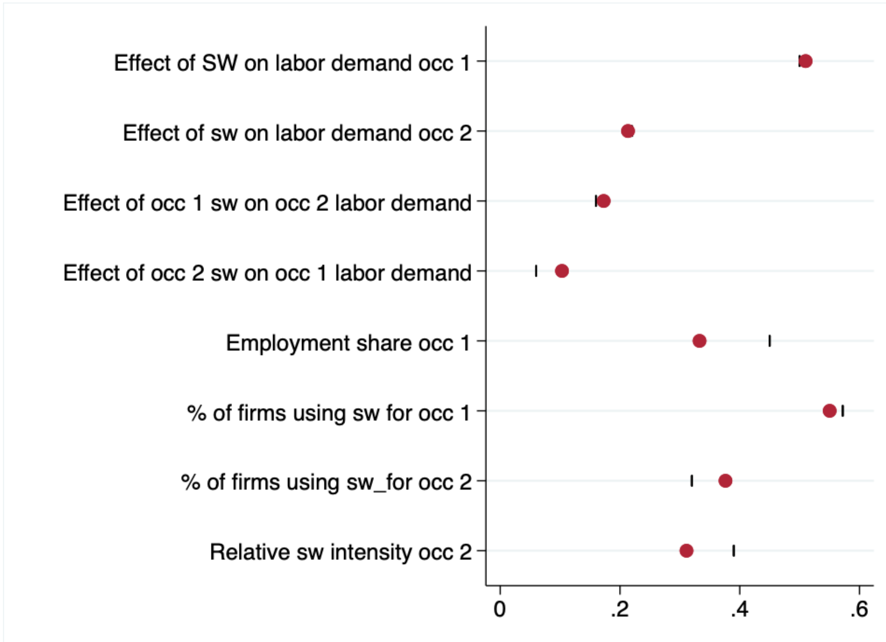
The price index of software fell by 20% in real terms between 2010 and 2019 (FRED).[39] We use our model to assess the effect of a price decrease on equilibrium outcomes. As the price of software decreases, the proportion of firms adopting software increases in each occupation. This drives up labor demand, and therefore wages, for jobs that use software versus jobs in the same firm that do not. As a result, the average skill requirements of each occupation increase, which further boosts wages for high-skilled workers because they become increasingly scarce.

We vary the variable price of software from 0.5 to 1.5 times its equilibrium level.[40] Keeping all other parameters constant, we compute the equilibrium at each price level, as described in Section 4.1.6. While we observe equilibrium wage inequality over a broader range in prices, we interpret the difference between outcomes at 1.1

[39] This statistic is for the real price index.

[40] Our estimates show the software variable cost is quantitatively more important. We can also vary both costs proportionally.

Figure 4: Model Fit



Notes: Comparison of model (red) and data (blue) targeted moments.

and 0.9 times the equilibrium price as the effect of a 20% fall in software prices. A caveat here is that we do not observe the effect of falling software prices on the price level of the final good, or on the total labor demand for white-collar labor. These caveats do not influence our results on inequality within white-collar occupations but may affect the average level of wages and the total quantity of labor demanded across both occupations.

A fall in the price of software increases the proportion of firms adopting software for each occupation (Figure 5a), as well as the average quantity of software used s_{fj} (Figure 5b). Within occupations, software-using firms absorb an increasing share of employment (Figure 5c).[41] The increase in labor demand is reflected in wages, as seen in Figure 5d. For workers using software, a 20% fall in the price of software increases wages 12.5% in occupation 2 and 20% in occupation 1 (managerial/ STEM). Wages slightly increase for non-software users in occupation 2 (sales/admin), while

[41]Total occupational shares do not change much across broad occupations, in part due to the wide wage gap between managerial/ STEM and sales/administrative workers.

the effects on non-software users in software 1 are small and non-monotonic.[42]

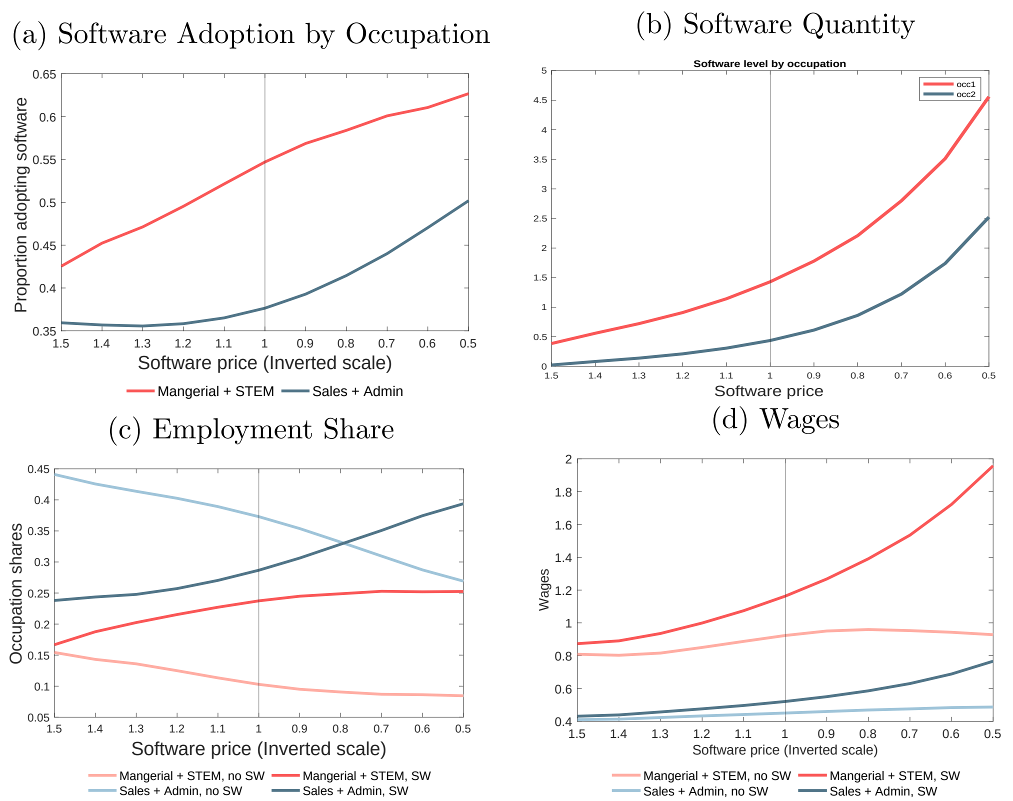
We next turn to the effect of falling software prices on inequality, both across occupations and within each occupation. We measure within-occupation inequality as the wage premium of software jobs compared to non-software jobs. As seen in Figure 6a, within-occupation inequality rises in both occupations as the price of software falls, more so for the Managerial & STEM occupation.

We measure inequality between occupations as the wage differential of occupation 1 compared with occupation 2, whereby each occupation's wage is an employment-weighted average of software and non-software wages. We can also compute the wage premium of occupation 1 within software and non-software jobs separately. Figure 6b shows overall between-occupation wage inequality as well as the software and non-software wage premiums for comparison. We see that the average wage differential, as well as the wage differential for software jobs, rises as the price of software falls. If the price of software fell by 20%, the wage premium $\frac{w_1}{w_2}$ averaged over all firms would increase from 1.17 to 1.24, as seen in Figure 6b.

The increase in between-occupation inequality is the net result of two opposing effects. On one hand, as seen in Figure 5d wages rise more for software jobs in the higher paid occupation, occupation 1, due to the stronger complementarity between software and labor—this increases between-occupation inequality. On the other hand, close to the equilibrium software price, the share of occupation 1 jobs using software is already high, so a fall in the price of software affects software adoption more in occupation 2 (5a). As software jobs have higher wages in each occupation, this lowers between-occupation inequality. On net, Figure 6b shows the first effect dominates as falling software prices *increase* the wage premium of the higher wage occupation.

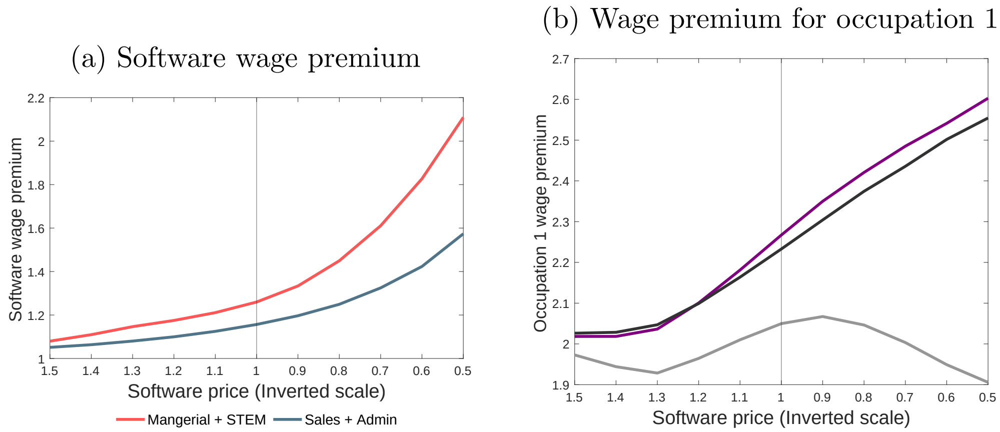
[42] These wage changes can be due to changes in the demand or supply of non-software labor. On the demand side, less non-software labor is demanded as firms shift toward software jobs but for firms using software in only the other occupation, overall production rises, increasing labor demand for the other occupation as well. On the labor supply side, qualified workers will increasingly leave the non-software job as the software wage increases, lowering labor supply and increasing wages

Figure 5: Effect of falling software prices on each outcome



Notes: Simulations of software prices falling from 1.5 times equilibrium value to 0.5 times equilibrium value on four outcomes: extensive margin software adoption, total software adoption, employment shares by occupation-software pair and wages by occupation-software pair. Red indicates occupation 1 (STEM + Managerial); Blue occupation 2 (Sales + Admin). Where applicable, light and dark lines indicate software and non-software respectively.

Figure 6: Effect of falling software prices on inequality within and across occupations



Notes: Simulations of software prices falling from 1.5 times equilibrium value to 0.5 times equilibrium value on within-occupation (a) and between-occupation (b) inequality. Figure (a) plots the wage premium for software (over non-software) jobs for each occupation. Figure (b) plots the average wage premium (purple) of occupation 1 (Managerial+ STEM) over occupation 2 (Sales + Admin). The grey lines indicate this occupation wage premium separately for software and non-software jobs.

4.4 Counterfactual: Shutting down the upskilling channel

We next ask how these effects depend on upskilling versus software entering the production function as an input. We set the skill levels in each occupation to their non-software levels and repeat the above analysis of falling software prices, holding other parameters constant. In this counterfactual, software no longer affects skill requirements within jobs but is effectively just a form of capital. Without the within-job skill channel, out model is similar to models of skill biased technological change and capital deepening, such as Krusell et al. (2000). As skill requirements are now the same for software and non-software jobs within each occupation, the set of workers qualified for each is also the same. We compare the effect of a fall in the price of software in this counterfactual case with the baseline case (allowing for upskilling).

In this counterfactual case, the proportion of firms choosing to adopt software rises at a faster rate as software prices fall (Figure 7a). This is because the proportion of employment in software jobs is no longer restricted by the qualified labor supply. Accordingly, we see in Figure 7b that employment in software jobs is higher in this case, across both occupations.

Turning to the effect on wages (Figure 7c), we find that wages for software jobs are very similar to their baseline counterpart. But wages for non-software jobs now rise with software as well, and the within-occupation wage premium of software jobs (Figure 7d) is lower than in the baseline case. At the estimated software price, revoking the upskilling channel lowers the software wage premium by 13.4% in occupation 1 and 10.4% in occupation 2; a fall in software prices of 20% would increase these differences to 27.5% and 13.3% respectively.

The mechanism behind this effect is as follows: the same set of workers is now qualified for software and non-software jobs within each occupation, so as wages rise for software jobs, workers increasingly choose to work for firms with software. As some jobs still prefer not to use software, they need to increase the wage of the non-software job to attract workers. In the baseline case, a subset of workers in each occupation was qualified for the non-software job only, so they remained in non-software jobs even if the wage gap between software and non-software jobs grew. This is no longer the case.

While within-occupation inequality is lower than the baseline, the between- occupation inequality as measured by the wage differential of the Managerial & STEM job is 0.7% higher (Figure 7e), since software levels are slightly higher. The mechanism behind this increase is as follows: without the labor supply restriction from the upskilling effect more firms choose to adopt software for each occupation, as seen in Figure 7a. Software proliferation differentially benefits the high wage occupation due to stronger software- labor complementarity, thus increasing the occupational wage differential.

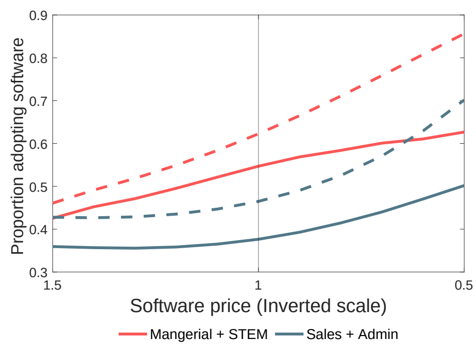
We compare the overall inequality of the counterfactual with the baseline using the Gini coefficient. In both cases, the Gini coefficient increases as the price of software falls, representing an increase in inequality (Figure 7f). We find that on net, overall inequality within the white collar sector is 5.32% *lower* in the counterfactual case without upskilling.

5 Conclusion

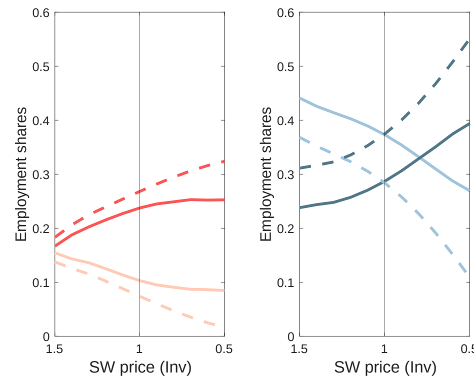
As of 2022, software comprises 17% of non-residential private investment, and this share has been trending upward since the 1960s. Given the prominent role of software in the economy, it is important to ask how software interacts with the labor market.

Figure 7: Effect of falling software prices: Baseline (—) vs counterfactual (- - -)

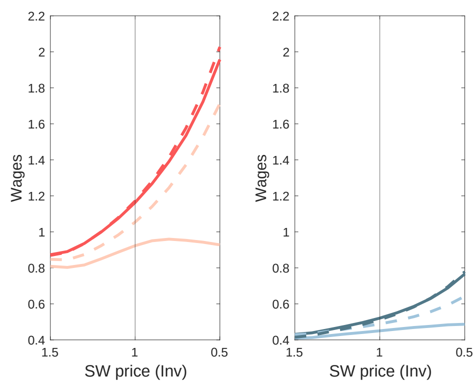
(a) Extensive margin software use



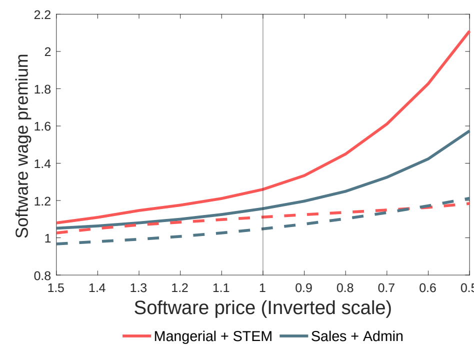
(b) Employment Shares



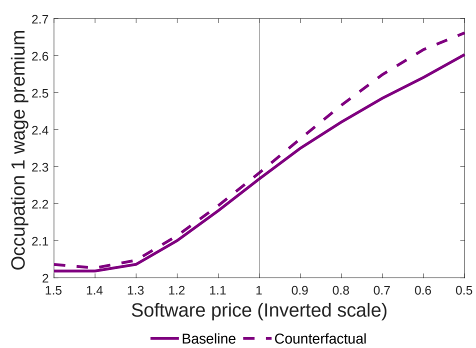
(c) Wages



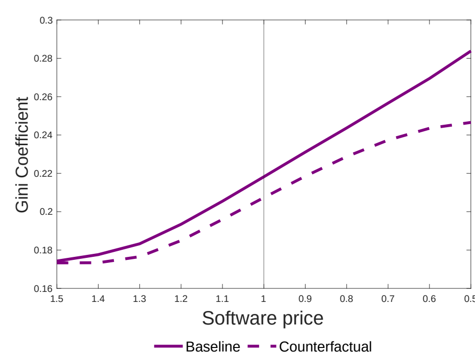
(d) Software wage premium



(e) Wage premium of occupation 1



(f) Gini Coefficient



Notes: Simulations of software prices falling from 1.5 times equilibrium value to 0.5 times equilibrium value on outcomes in Figures 7 and 8; baseline (—) compared with counterfactual without rising skill channel (- - -). Outcome in Figure (f) is total inequality as measured by the Gini Coefficient.

In this paper, we show that software adoption can raise both firms' labor demand and skill requirements. We develop a strategy to identify job-level software adoption events from job posting data and compare skill requirements around the time of adoption. We account for endogeneity in adoption timing using a latent variable strategy, under the identifying assumption that software adoption does not affect skill requirements in other, non-adopting, occupations within the firm. We find that each type of software adopted increases social and analytic skill requirements by 1.1pp and 0.8pp respectively, with effects strongest for the first few adoptions. The effect on social skills is more robust and quantitatively more important relative to differences in skill levels between occupations. Apart from increasing skill requirements, software also increases the number of vacancies, a proxy for labor demand.

We then embed our casual estimates into a model with nested CES production and workers matching with jobs on two-dimensional skills. A key feature of our model is that firms make interdependent software choices across occupations, which consequently affect each occupation's analytic and social skill requirements. In turn, the skill requirements determine the set of workers qualified for each job. Software adoption raises skill requirements, in line with our empirical results, which means firms using software can match with a smaller set of qualified workers. Jobs using software must pay a premium over other jobs in the same occupation to attract a greater share of the qualified workforce; this drives a wedge between software and non-software wages within an occupation.

We estimate the model using GMM and find that labor is complementary to software, especially in high-skill STEM and managerial jobs. As the price of software falls the proportion of firms using software for each occupation rises, driving up within-occupation inequality as described above. At the same time, the stronger complementarity of high-skill labor with software leads to an increase in inequality across occupations. We show that in the absence of skill requirements the same set of workers choose between software and non-software jobs within each occupation, so non-software jobs must also raise wages to attract labor when software-using firms do so. This lowers within-occupation inequality. Total inequality, as measured by the Gini coefficient, also falls, although across-occupation inequality widens slightly.

References

Acemoglu, Daron, Gary W. Anderson, David N. Beede, Catherine Buffington, Eric E. Childress, Emin Dinlersoz, Lucia S. Foster, Nathan Goldschlag, John C. Haltiwanger, Zachary Kroff, Pascual Restrepo, and Nikolas Zolas (2022) *Automation and the Workforce: A Firm-Level View from the 2019 Annual Business Survey*, University of Chicago Press.

Acemoglu, Daron and David Autor (2011) "Skills, tasks and technologies: Implications for employment and earnings," in *Handbook of labor economics*, 4, Elsevier, 1043–1171.

Acemoglu, Daron, David Autor, Jonathon Hazell, and Pascual Restrepo (2020) "AI and jobs: Evidence from online vacancies,"Technical report, National Bureau of Economic Research.

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

Acemoglu, Daron and Pascual Restrepo (2020) "The wrong kind of AI? Artificial intelligence and the future of labour demand," *Cambridge Journal of Regions, Economy and Society*, 13, 25–35.

Akerman, Anders, Ingvil Gaarder, and Magne Mogstad (2015) "The skill complementarity of broadband internet," *The Quarterly Journal of Economics*, 130, 1781–1824.

Almeida, Rita K, Ana M Fernandes, and Mariana Viollaz (2020) "Software adoption, employment composition, and the skill content of occupations in chilean firms," *The Journal of Development Studies*, 56, 169–185.

Atalay, Enghin, Phai Phongthiengtham, Sebastian Sotelo, and Daniel Tannenbaum (2018) "New technologies and the labor market," *Journal of Monetary Economics*, 97, 48–67.

Aum, Sangmin (2017) "The rise of software and skill demand reversal," *Unpublished manuscript*.

Autor, David H, Frank Levy, and Richard J Murnane (2003) "The skill content of recent technological change: An empirical exploration," *The Quarterly journal of economics*, 118, 1279–1333.

Bessen, James E, Erich Denk, Joowon Kim, and Cesare Righi (2020) "Declining industrial disruption," *Boston Univ. School of Law, Law and Economics Research Paper*, 20–28.

Bessen, James E, Erich Denk, and Chen Meng (2022) "The Remainder Effect: How Automation Complements Labor Quality," *James E. and Denk, Erich and Meng, Chen, The Remainder Effect: How Automation Complements Labor Quality (February 24, 2022)*.

Bessen, James, Martin Goos, Anna Salomons, and Wiljan Van den Berge (2019) "Automatic reaction-what happens to workers at firms that automate?".

Braxton, J Carter and Bledi Taska (2023) "Technological change and the consequences of job loss," *American Economic Review*, 113, 279–316.

Callaway, Brantly and Pedro HC Sant'Anna (2021) "Difference-in-differences with multiple time periods," *Journal of Econometrics*, 225, 200–230.

Card, David and Thomas Lemieux (2001) "Can falling supply explain the rising return to college for younger men? A cohort-based analysis," *The quarterly journal of economics*, 116, 705–746.

Choo, Eugene and Aloysius Siow (2006) "Who marries whom and why," *Journal of political Economy*, 114, 175–201.

David, H and David Dorn (2013) "The growth of low-skill service jobs and the polarization of the US labor market," *American economic review*, 103, 1553–97.

De Ridder, Maarten (2019) "Market power and innovation in the intangible economy."

Deming, David J (2017) "The growing importance of social skills in the labor market," *The Quarterly Journal of Economics*, 132, 1593–1640.

Deming, David and Lisa B Kahn (2018) "Skill requirements across firms and labor markets: Evidence from job postings for professionals," *Journal of Labor Economics*, 36, S337–S369.

Dillender, Marcus and Eliza Forsythe (2022) "Computerization of white collar jobs,"Technical report, National Bureau of Economic Research.

Dvorkin, Maximiliano A and Alexander Monge-Naranjo (2019) "Occupation mobility, human capital and the aggregate consequences of task-biased innovations," *FRB St. Louis Working Paper*.

Eckert, Fabian, Sharat Ganapati, and Conor Walsh (2022) "Urban-Biased Growth: A Macroeconomic Analysis,"Technical report, National Bureau of Economic Research.

Freyaldenhoven, Simon, Christian Hansen, and Jesse M Shapiro (2019) "Pre-event trends in the panel event-study design," *American Economic Review*, 109, 3307–38.

Hershbein, Brad and Lisa B Kahn (2018) "Do recessions accelerate routine-biased technological change? Evidence from vacancy postings," *American Economic Review*, 108, 1737–72.

Katz, Lawrence F and Kevin M Murphy (1992) "Changes in relative wages, 1963–1987: supply and demand factors," *The quarterly journal of economics*, 107, 35–78.

Krusell, Per, Lee E Ohanian, José-Víctor Ríos-Rull, and Giovanni L Violante (2000) "Capital-skill complementarity and inequality: A macroeconomic analysis," *Econometrica*, 68, 1029–1053.

Lashkari, Danial, Arthur Bauer, and Jocelyn Boussard (2018) "Information technology and returns to scale," *Available at SSRN 3458604*.

Lindenlaub, Ilse (2017) "Sorting multidimensional types: Theory and application," *The Review of Economic Studies*, 84, 718–789.

Lise, Jeremy and Fabien Postel-Vinay (2020) "Multidimensional skills, sorting, and human capital accumulation," *American Economic Review*, 110, 2328–76.

Mariscal, Asier (2018) "Firm organization and information technology: Micro and macro implications."

Webb, Michael (2019) "The impact of artificial intelligence on the labor market," *Available at SSRN 3482150*.

A List of Software types

- Accounting Software
- Advanced Microsoft Excel
- Android Development
- Animation and Game Design
- Application Programming
- Application Security
- Architectural Design
- Artificial Intelligence
- Audio Production
- AviationStandards
- Backup Software
- Big Data
- Business Communications
- Business Intelligence Software
- Business Management
- Cache computing
- C and C++
- Clinical Data Management
- Cloud Computing
- Cloud Solutions
- Computer Hardware
- Creative Design
- Customer Relationship Manag
- Cybersecurity
- Data
- Database Management System
- Dental Care
- Distributed Computing
- Document Management Systems
- Electrical and Computer Engineering
- Engineering Software
- Enterprise Management
- Extensible Languages
- Extraction Transformationa
- Financial Software
- Firmware
- Fundraising
- General Networking
- Geographic Information Systems
- Government Clearanceand Sec

- Graphic and Visual Design
- Hardware Description Langua
- Human Resources Software
- IT Automation
- IT Management
- Information Security
- Integrated Development Envi
- Internet Protocols
- Java
- JavaScript and jQuery
- Law Enforcement and Criminal
- Learning Management Systems
- Legal Research
- Machine Learning
- Mainframe Technologies
- Manufacturing Standards
- Mathematical Software
- Medical Billing and Coding
- Microsoft Development Tools
- Microsoft Office and Product
- Middleware
- Natural Language Processing
- Network Configuration
- Network File System NFS
- Network Protocols
- Network Security
- Networking Hardware
- NoSQL Databases
- Occupational Health and Safety
- Online Marketing
- Operating Systems
- Oracle
- PHP Web
- Policy Analysis
- Productivity Software
- Project Management
- Project Management Software
- Property Management
- SAP
- SQL Databases and Programmin
- Scripting Languages
- Social Media
- Software development
- Specialized Accounting
- Statistical Software
- System Design and Implementa
- Tax Software
- Telecommunications
- Test Automation
- User Interface and User Exper
- Version Control
- Virtual Machines VM
- Web Analytics
- Web Content
- Web Design
- Web Development
- Web Servers

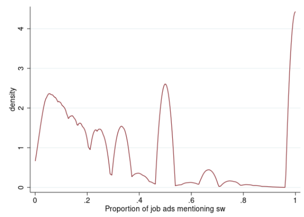
B Persistence of software adoption

We show some characteristics of our observed software adoptions. Firstly, software use is not all-or-nothing even within a firm-MSA's 6-digit occupation and month. Figure 8a shows the proportion of firm-occupations that require a software type in any month post adoption, conditional on at least one ad requiring it that month. We see that many firm-occupations require only a small proportion of potential hires to know the software. This could be because the workers are in somewhat different roles within the occupation.

While there is some persistence in software mentions post software adoption, it is not the case in reality that the firm goes from never requiring the software to always requiring the software. Figures 8b and 8c show the proportion of firm-occupations with *any* vacancy requiring the software in each month and year post adoption—conditional on posting any vacancy). Only 30% of firm-occupations require the software again after two years. The persistence of software use in the firm is likely higher than these estimates from vacancies suggest: workers hired with vacancies requiring knowledge of the software may continue to work at the firm, using the software; the firm may have started teaching the software internally and no longer need previous expertise with it. It is also possible that some of these firm-occupations do stop using the software, in which case our estimates of adopting a software will be lower in magnitude than the true effect of *using* the software. Reasons why a firm would stop using a software type are complicated, likely involving learning or changes in business directions, and outside the scope of this paper.

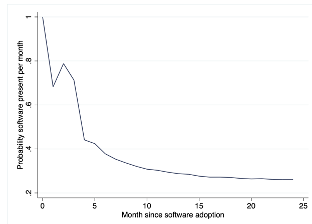
Figure 8: Persistence Graphs

(a) Software requirements within a job-year



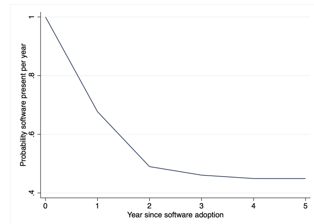
Notes: Proportion of vacancies in job requiring a software type, *conditional* on any vacancy requiring that software type

(b) Persistence over 24 months



Notes: Proportion of firm-occupations requiring the software conditional on posting any vacancies, by month after adoption event.

(c) Persistence over 5 years



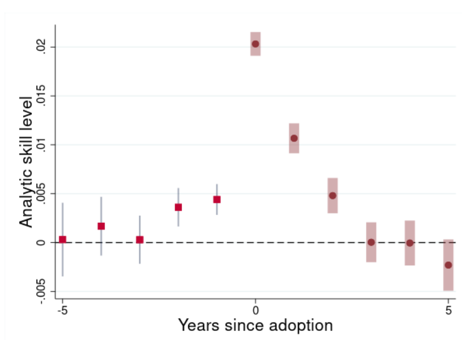
Notes: Proportion of firm-occupations requiring the software conditional on posting any vacancies, by year after adoption event

C Empirics Appendix

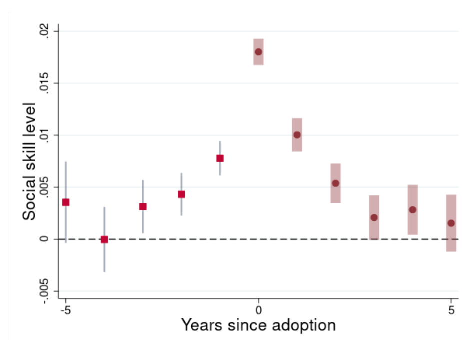
C.1 Event study estimates: Callaway Sant'anna estimator

Figure 9: Caption

(a) Analytic Skills



(b) Social Skills



C.2 Effects on other outcomes

Table 4: Effects of software adoption on auxiliary outcomes

	(1) Management	(2) IT	(3) BA degree	(4) Years experience	(5) Graduate education
Proxy	0.251	3.518*	2.190*	12.85*	-0.0519
	(0.498)	(2.029)	(1.215)	(7.083)	(0.171)
Post Adoption	0.0198***	0.0194***	0.00414	0.0333	-0.00146***
	(0.00130)	(0.00558)	(0.00353)	(0.0211)	(0.000502)
N	1211627	1211627	1215954	1215954	1215954

Standard errors in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

C.3 Effects by previous software quantity

	≤ 2 sw analytic	> 2 sw analytic	≤ 2 sw social	> 2 sw social
Proxy	0.864**	-1.814	0.963***	-1.091
	(0.365)	(1.402)	(0.312)	(1.052)
Post adoption	0.00373	0.00917***	0.0126***	0.0109***
	(0.00523)	(0.00341)	(0.00446)	(0.00251)
N	432669	642612	432669	642612

Standard errors in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

C.4 Effects using alternate proxy

	(1) analytic	(2) analytic	(3) social	(4) social
agg_proxy_mean	2.556	3.430	2.847*	2.062
	(1.569)	(3.904)	(1.626)	(2.310)
after	-0.000589	-0.00371	0.00779	0.00313
	(0.0102)	(0.0145)	(0.0104)	(0.00861)
N	432669	642612	432669	642612

Standard errors in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

D Effect heterogeneity by type of software adopted

Software comprises a broad range of technologies, which may affect skill requirements differently. Since we observe adoption events separately by software type, we can re-run Equation 2 separately by software type. Due to power limitations from the smaller sample sizes, we use OLS. While the OLS coefficients cannot be interpreted as the causal effect for each software type, they suffice to show evidence of heterogeneity across types[43].

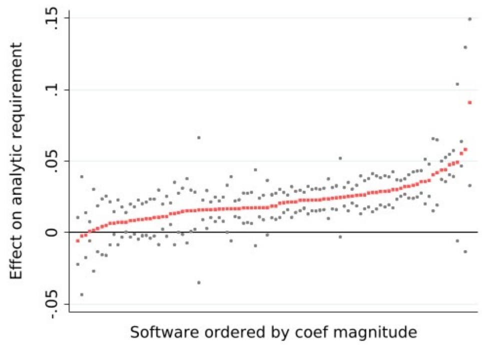
Figure 10a plots the effect of each software type on the analytic skill measure. The software types are ranked by the magnitude of their effect. We see that at the lower end, software has no discernible impact on analytic skills. The software types with the largest impacts—up to 5pp (compared to an average OLS estimate of 1.3) include software types like engineering and dentistry software that tend to be associated with highly analytically skilled tasks. We run a Pearson χ^2 test to test whether the observed heterogeneity in effects over software types is statistically significant (following Kline, Rose and Walters (2022)), and reject the null hypothesis that there is no heterogeneity over software types.

Figure 10b shows the software-specific effects for social skills. We see similar patterns, with the effect ranging from 0 to just under 5pp. Here software types are ranked by their effect on social skills. Again, a Pearson χ^2 test finds there is statistically significant heterogeneity between software types. We find that the

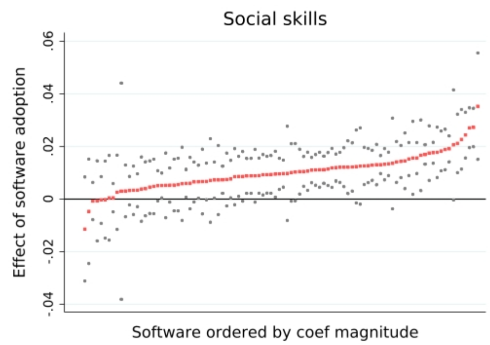
[43]Technically, the heterogeneity may also be driven by susceptibility of each software to the latent variable.

Figure 10: Heterogeneity in the impacts by software type

(a) Analytic Skills



(b) Social Skills



Notes: For each skill requirement, software types are ordered by magnitude of their impact on the skill requirement. We plot the OLS coefficient of a software adoption event on the skill requirement. Grey dots indicate confidence bands. Note that requirements are on average higher due to using OLS. A full list of software types used can be found in Appendix A.

correlation between effects on analytic and on social skills is 0.3. This means that software types that have a higher effect on analytic skills also, on average, have a higher impact on social skills.

E Parameter Estimates

Parameter	Definition	Estimated Value	(Prelim.) SE
γ_1	EOS between labor and sw occ 1	0.74	0.18
γ_2	EOS between labor and sw occ 2	0.92	0.003
σ	EOS between occupations	2.96	0.05
α_2	Relative software weight occ 2	22.53	0.00
A	TFP $\times$ price of final good	20.84	0.01
ω_2	Occupation 2 relative weight	10	0.03
p^{var}	Software variable cost	1.81	0.01
p^{fixed}	Software fixed cost	0.19	0.003
$\varkappa$	Variance of idiosyncratic software productivity	1.07	0.002

Table 5: Parameter Estimates from GMM