Firm Productivity and Learning with Digital Technologies: Evidence from Cloud Computing *

James Brand[†] Mert Demirer[‡] Connor Finucane[§] Avner A. Kreps[¶]

October 20, 2025

Abstract

Digital technologies have transformed firm production across all sectors. In this paper, we document new facts about firms' productivity with emerging technologies by studying how efficiently they use cloud computing. Leveraging high-frequency CPU utilization data from nearly 100,000 firms, we find large and persistent dispersion in firms' cloud productivity. Productivity, however, is highly dynamic in this setting: firms improve their cloud productivity by 33% in the first year after adoption, and reach a stable level only after four years, indicating a long adjustment period to learn the technology. While faster learning among initially less efficient firms reduces productivity dispersion by 60% over time, substantial heterogeneity remains even after 10 years. Finally, productivity improvements occur primarily within individual divisions of a firm, with minimal knowledge transfer across divisions.

JEL: D22, D24, L86

Keywords: Productivity, cloud computing, learning-by-doing, digital technologies, IT

*This paper was previously circulated under the title "Firm Productivity and Learning in the Digital Economy: Evidence from Cloud Computing." We thank Dan Ackerberg, Vivek Bhattacharya, Noman Bashir, Nick Bloom, Jeffrey Campbell, Eli Cortez, Ulrich Doraszelski, Luca Fontanelli, Ambar La Forgia, Sonia Jaffe, Bob Gibbons, Matthew Grennan, Igal Hendel, Patrick Hummel, Gaston Illanes, Kristina McElheran, Kyle Myers, Donald Ngwe, Scott Orr, Rob Porter, Devesh Raval, Michael Schwarz, Carolyn Stein, and Neil Thomson, as well as seminar and conference participants at CEPR Industrial Organization Conference; NBER Organizational Economics Meeting; NBER Productivity, Innovation, and Entrepreneurship Meeting; Northwestern University; Rice University; Stanford University; Tilburg University; UBC Sauder School of Business; Utah Business Economics Conference; University of California, Berkeley; University of California, Los Angeles; University of Maryland; University of Toronto; and ZEW Conference on the Economics of Information and Communication Technologies for their helpful conversations and comments. Aaron Banks provided excellent research assistance.

[†]Microsoft, jamesbrand@microsoft.com

[‡]MIT Sloan, mdemirer@mit.edu

[§]Microsoft, cfinucane@microsoft.com

[¶]Northwestern University, avner@u.northwestern.edu

1 Introduction

Digital technologies have transformed firm production across all sectors of the economy in recent years. Today, nearly 80 percent of firms rely on cloud computing as part of their production processes (Bloom and Pierri, 2018; Zolas et al., 2021), and firms are increasingly adopting artificial intelligence as the next frontier of the digital transformation (McElheran et al., 2023, 2025). These technologies have far-reaching implications: they have been shown to affect labor productivity, revenues, and organizational structure at the firm level (Brynjolfsson and Hitt, 2003; Bartel et al., 2007; Bloom et al., 2012, 2014), as well as concentration and the labor share at the industry level (Lashkari et al., 2024).

While an extensive literature in economics studies firm productivity (Syverson, 2004a,b; Foster et al., 2008; Bloom et al., 2013, 2019), there is limited large-scale evidence on how efficiently firms use and adopt these modern digital technologies. A key reason for this gap is the lack of granular data tracking firms' use of digital technologies. Most production datasets allow researchers to estimate total factor productivity (TFP), which, as a residual measure, captures various unobserved factors rather than isolating a specific aspect of production (Syverson, 2011). This underscores the importance of developing complementary measures that directly capture how efficiently firms use new technologies.

In this paper, we document new facts about firm productivity by studying firms' efficiency and learning in their use of cloud computing. Our analysis draws on high-frequency utilization data from over one billion virtual machines (VMs) used by nearly 100,000 firms across various industries and countries. Using this dataset, we develop a measure that quantifies the extent to which firms could produce the same computing output with fewer computing inputs. We call this measure *compute productivity*.[1] With this measure, we analyze dispersion in compute productivity across and within firms, document how firms learn to use cloud computing more efficiently, and examine the evolution of productivity dispersion as firms learn to use this technology.

We begin our paper by providing background information on cloud computing. Cloud computing gives firms on-demand access to IT resources over the Internet. Instead of making periodic capital investments, firms rent IT resources from cloud providers, which shifts IT from a fixed cost to a variable cost. Cloud computing is one of the most widely adopted technologies in recent times (Zolas et al., 2021; Kalyani et al., 2025), representing a substantial and growing share of firms' costs across sectors (Demirer et al., 2024).

In cloud computing, VMs serve as the primary unit of production, corresponding to

[1] Although we refer to this metric as "productivity," it differs from TFP because it measures how efficiently a firm uses one input, rather than the residual of a production function. Thus, it can be viewed as a factor-augmenting productivity or factor efficiency. We provide more exposition on this point below.

partitions of a physical server that firms rent on-demand to run their IT operations. A key feature of VMs is that firms can dynamically adjust compute resources within seconds in response to changing demand and only pay for the amount of time each VM is allocated to them. This flexibility reduces dynamic frictions in IT provisioning (Asker et al., 2014), eliminating the need to provision resources in advance and maintain excess capacity.

As a transformative technology reshaping how firms use IT, cloud computing requires adaptation for efficient use (Bresnahan and Trajtenberg, 1995; Bresnahan et al., 2002). Firms need to develop internal processes and monitoring mechanisms to accommodate the new cost structure since, unlike on-prem computing, marginal costs of compute are positive in the cloud. They also need new skills specific to cloud computing, such as using tools to select appropriately sized VMs for each task. Existing evidence highlights the difficulty of these adjustments and their value to firms: industry studies find substantial underutilization in the cloud (Cortez et al., 2017; Flexera, 2023a), which has fueled a multi-billion dollar consulting industry helping firms adopt efficient practices.

With these features, cloud computing provides an ideal setting to study firm productivity and learning with new technologies. First, despite its widespread use, its variable input nature and uniformity limit the role of idiosyncratic factors that typically complicate large-scale firm analyses (Syverson, 2011). Second, productivity can be estimated at high frequency at the machine level, offering insights into its short- and long-run dynamics. Third, we can estimate compute productivity for individual divisions within firms, improving our understanding of within-firm heterogeneity. Finally, we can quantify wasted economic resources due to inefficiencies, such as computing hardware and electricity, as our measure is quantity-based and attributable to a specific aspect of production.

To implement our empirical analysis, we construct our firm-level compute productivity measure using VM utilization data collected at 5-minute intervals. Following industry practice, we use CPU utilization patterns, which measure the share of a VM's compute capacity actually used by the firm. Our measure incorporates two sources of inefficient compute usage: idle and overprovisioned VMs. We define a VM as *idle* if the firm pays for but never uses it and *overprovisioned* if its peak compute load would fit within the capacity of a smaller but readily substitutable VM. The underlying idea is that firms could eliminate idle VMs or downsize overprovisioned ones without impacting performance, thus saving resources and money. Importantly, these concepts are defined using peak utilization over a seven-day period, ensuring that low average utilization due to volatile demand is not interpreted as inefficiency. By aggregating both sources of VM-level inefficiency, we obtain monthly firm- and division-level compute productivity estimates.

Although compute productivity is estimated from a novel dataset, it is conceptually

related to established measures in the productivity literature. In the context of production theory, compute productivity quantifies a firm's compute resource usage relative to a cost-minimizing benchmark that produces the same compute output using the same choice set of VMs. In this sense, it captures firms' efficiency in compute usage, in an analogous manner to factor-augmenting productivity measures in the literature (Doraszelski and Jaumandreu, 2018; Raval, 2019; Demirer, 2025).

Using this measure, we first characterize the cross-sectional distribution of compute productivity. We find large dispersion in compute productivity across firms that is persistent over short-term (1-month) and long-term (5-year) horizons. Controlling for industry, firms at the 90th percentile of the distribution are 3.5 times more productive at computing than those at the 10th percentile. Substantial dispersion also exists within firms: 55.7% of the variation in compute productivity is explained by within-firm heterogeneity. The levels of dispersion and persistence in compute productivity are comparable to prior findings in the productivity literature (Syverson, 2011; Bloom et al., 2019) and are not simply explained by observables such as industry, firm size, and VM characteristics.

We next ask what makes firms more productive in computing to analyze potential mechanisms that generate this dispersion. Although we do not observe specific firm practices beyond cloud computing, we analyze differences in VM deployment patterns that reflect heterogeneity in firm behavior. Three patterns differentiate high- from low-compute productivity firms. First, high-productivity (above-median) firms adjust better to demand fluctuations: on average, they reduce their provisioned compute resources by 75.2% more than low-productivity firms on weekends, during which compute demand declines substantially. Second, they tend to have better monitoring capabilities: when resources remain idle, they are more than twice as likely to shut down idle VMs as low-productivity firms. Finally, they take better advantage of the available menu of VMs: they use a greater variety of VMs and are less likely to put all their jobs on a single VM type.

We then shift our focus to learning, investigating whether firms improve their efficiency in computing as they gain experience with the cloud. We analyze the compute productivity trajectories of firms over time, considering both short- and long-term learning.

We find that new cloud adopters improve their compute productivity substantially over time, but take a long time to reach steady state. Firms with one year of experience are 33.1% more productive than they were initially on average. The rate of improvement slows after the first year, with firms exhibiting 40.5% higher compute productivity by the end of fourth year and making no improvements thereafter. The extended time to reach a stable level is longer than prior estimates from learning-by-doing studies (Benkard, 2000; Levitt et al., 2013), which could reflect the need for complementary investments and practices, as

is widely documented in the literature (Bresnahan et al., 2002; Bloom et al., 2012).

Our analysis also reveals that learning significantly influences dispersion in compute productivity. As firms gain experience, the 90th–10th percentile productivity ratio within a given cohort declines from 7.2 in their first month to 2.9 after four years. This trend is driven by differences in learning rates: firms that are initially less efficient learn much faster than their more efficient counterparts (but still do not catch up). At the same time, heterogeneity in the timing of adoption, through the presence of both more and less experienced firms at a given point in time, raises cross-sectional dispersion. Together, these patterns highlight that the maturity of a production technology is an important determinant of the level and dynamics of its productivity dispersion.

Next, we explore the mechanisms behind *how* firms learn. We decompose firms' compute productivity growth into (i) within-division learning, (ii) across-division reallocation, and (iii) division entry and exit (Foster et al., 2001). We find that firm-level learning is driven primarily by division-level learning, rather than by reallocation across divisions. At any level of firm experience, new divisions start with lower compute productivity than the firm as a whole and gradually improve their productivity over time. Notably, these new divisions follow a nearly identical trajectory to firms that are new to the cloud, even in experienced high-productivity firms. This pattern highlights the limited role of within-firm knowledge transfer. However, when we further decompose productivity within divisions across VM types, we find that experienced divisions achieve higher initial efficiency with new VM types than new divisions, indicating some knowledge transfer within divisions.

Our final analysis quantifies the aggregate impact of inefficient compute use on economic resources, capturing both private costs to firms (compute) as well as social costs (electricity). We find that a substantial share of economic resources is wasted due to inefficient computing use, indicating that compute inefficiencies do not come from only small firms or firms with negligible compute use. Raising all firms' compute productivity to the 80th percentile reduces the usage of compute hardware by 21.0% and its electricity use by 16.5%, which reach to 35.4% and 28.3% if all firms attain the highest possible productivity.

We take several steps to ensure our measure accurately captures compute efficiency while minimizing potential confounding factors. First, we adopt a conservative approach by focusing on peak CPU utilization measured over a seven-day period. Using peak utilization accounts for many factors that may lower overall utilization without indicating true inefficiency, such as fluctuating demand. Second, we show that our results are robust to controlling for a rich set of VM characteristics, suggesting that compute productivity is not mechanically driven by observable VM characteristics. Third, we review industry sources to demonstrate that our measure closely aligns with how firms measure compute

inefficiencies in practice. Fourth, we replicate our estimates using publicly available data from multiple cloud providers and find similar levels of dispersion. Fifth, we analyze other utilization metrics in computing (memory and networking) and find similar results.

We nevertheless acknowledge some limitations of our study. Our approach does not capture all forms of compute inefficiency, such as inefficiently written code. While important, such inefficiencies are fundamentally different from those we study, as they pertain to changing the production process rather than resource deployment. Moreover, our dataset is limited to compute inputs and does not include other firm inputs or outputs. As a result, we are unable to analyze the link between compute productivity and key firm attributes such as TFP, costs, and revenues. We believe this is a worthwhile trade-off given the extensive research on IT's impact on various firm-level measures as discussed below, while micro-analysis of IT usage remains scarce.

Contribution to the Literature. First and foremost, this paper contributes to the large literature on firm productivity (e.g., Bartelsman and Doms, 2000; Syverson, 2004a,b; Foster et al., 2008; Bloom and Van Reenen, 2007; Hsieh and Klenow, 2009; Syverson, 2011; Bloom et al., 2012, 2013; Backus, 2020; De Loecker and Syverson, 2021).[2] This literature has documented large dispersion in firm productivity and analyzed factors driving this dispersion. Our paper extends this literature by studying firm productivity in computing, an input of increasing importance in firm production. We document several new empirical facts about firm productivity in the context of a new technology, analyze productivity dynamics at a relatively granular level, and quantify the link between productivity and resource usage.

Our paper also contributes to the literature studying the effect of IT on firm outcomes (Brynjolfsson and Hitt, 1995, 1996, 2000, 2003; Black and Lynch, 2001; Bartel et al., 2007; Bloom et al., 2012, 2014; Tambe and Hitt, 2012; Brynjolfsson and McElheran, 2016; Brynjolfsson et al., 2023; Babina et al., 2024).[3] While this literature emphasizes the heterogeneous impact of IT across firms and the need for complementary investments to use IT effectively (Bresnahan et al., 2002), our paper directly measures how efficiently firms use a transformative digital technology at a large scale. As such, we provide detailed evidence on firms' use of an IT technology and complement the canonical findings in this literature.

[2]While many papers in this literature use plant-level TFP, several study efficiency measures based on utilization of inputs, as we do (Hubbard, 2003; Braguinsky et al., 2015; Butters, 2020). Our paper also relates to two other strands of literature: studies of dispersion in specific factor productivity measures—including labor (Fox and Smeets, 2011) and electricity (Davis et al., 2008)—and studies of how new technologies affect overall productivity (Van Biesebroeck, 2003; Collard-Wexler and De Loecker, 2015).

[3]Other contributions to this literature include case studies that investigate how IT affects firm outcomes (Baker and Hubbard, 2004; Miller and Tucker, 2011; Dranove et al., 2014), studies focusing on the decline in IT prices and its impact on production technology (Lashkari et al., 2024), macro-level studies (Jorgenson, 2001; Stiroh, 2002), and those studying IT's slow impact on aggregate statistics (Brynjolfsson et al., 2021).

By studying firm learning in cloud computing, we contribute to the literature on learning-by-doing (Darr et al., 1995; Benkard, 2000; Thornton and Thompson, 2001; Kellogg, 2011; Levitt et al., 2013; Hendel and Spiegel, 2014; Doraszelski et al., 2018; Tadelis et al., 2023; Backus et al., 2025). Much of this literature analyzes a single firm or a small number of firms within narrow industries and shows that productivity improves with firm experience. Our study instead focuses on learning in the context of a widely adopted and rapidly evolving technology, thus linking the learning-by-doing literature to the literature on general-purpose technologies (Bresnahan and Trajtenberg, 1995; Bresnahan, 2010).[4]

Finally, we contribute to the literature on the economics of cloud computing by analyzing firms' efficiency and learning in the cloud (Bloom and Pierri, 2018; Byrne et al., 2018; Greenstein and Fang, 2020; Jin and Bai, 2022; DeStefano et al., 2023; Jin et al., 2023; Demirer et al., 2024; Lu et al., 2024; Jin and McElheran, 2024; Caldarola and Fontanelli, 2024).

2 Background on Cloud Computing

Computing is an essential input to firm production across all industries. At a fundamental level, computing enables firms to process data and perform tasks that require calculations using a combination of hardware and software.[5] This section provides an overview of how firms use computing in the cloud, with additional details presented in Appendix A. Readers familiar with cloud computing can skip this section and proceed to Section 3.

2.1 What is Cloud Computing?

Traditionally, computing was performed on servers purchased and maintained by individual firms, known as "on-prem" computing. In this paradigm, computing resources are capital expenditures subject to a classical peak-load problem: firms make periodic investments to have enough capacity to handle peak demand, leading to underutilization during off-peak periods (Pindyck, 1986; Bloom et al., 2007). More recently, advancements in server and networking technology have given rise to cloud computing, which allows firms to access IT resources remotely over the Internet. In this model, the physical resources are owned by cloud providers, and firms have on-demand access to these resources on a pay-as-you-go basis, with usage typically billed by the minute. This flexibility allows firms to handle increased workloads during peak periods and scale down resources

[4]While there are multiple definitions of general-purpose technology in the literature, one basic characterization is that it is (i) widely used, (ii) capable of ongoing technical improvement, and (iii) enables innovation in application sectors (Bresnahan, 2010). Cloud computing satisfies these conditions.

[5]Firms' computing use falls into three main categories: production (e.g., streaming, mobile apps, payments), development (e.g., data analytics, computer-aided design), and administrative operations (e.g., HR, finance, sales) (Davenport, 1998; Greenstein, 2020). Production accounts for the majority of usage (Cortez et al., 2017).

Figure 1: Sample of VM Choices on Amazon Web Services

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
t4g.nano	$0.0042	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t4g.micro	$0.0084	2	1 GiB	EBS Only	Up to 5 Gigabit
t4g.small	$0.0168	2	2 GiB	EBS Only	Up to 5 Gigabit
t4g.medium	$0.0336	2	4 GiB	EBS Only	Up to 5 Gigabit
t4g.large	$0.0672	2	8 GiB	EBS Only	Up to 5 Gigabit
t4g.xlarge	$0.1344	4	16 GiB	EBS Only	Up to 5 Gigabit

Notes: This figure displays a selection of VM options from a specific Amazon Web Services machine series and details the configuration of each VM.

when no longer needed. Cloud computing thus flips the traditional on-prem paradigm, transforming computing from a capital investment into a variable operating cost.

Cloud computing has expanded rapidly, evolving from limited adoption in the 2000s to a central pillar of IT infrastructure today. Industry estimates place total US cloud spending at roughly $432 billion in 2024, which is projected to double by 2028 with the surge in AI adoption (IDC, 2025). As a result, cloud computing now represents the main element of firms' IT budgets, which itself accounts for a large share of firm revenue—19% in the software industry and 9-13% across various services sectors as of 2023 (Flexera, 2023b). To put this magnitude in perspective, aggregate cloud spending by US firms is now on par with the total energy expenditures of the industrial and commercial sectors combined.[6]

2.2 Virtual Machines in Cloud Computing

While a wide range of products are available in cloud computing, VMs are the primary cloud computing resource used by firms.[7] Cloud providers separate their physical servers into distinct partitions, each running its own isolated operating system. A VM is one such partition, allocated to a firm upon request and remaining active until the firm terminates it. Firms typically use tens or even hundreds of VMs simultaneously for their IT operations. Once requested, VMs are typically deployed within seconds (Nguyen and Lebre, 2017).[8]

Cloud providers offer a vast array of VM options. As illustrated in Figure 1, each VM comes with a configuration specifying memory, storage, network capability, and cores

[6]See Appendix D.1 for calculations.

[7]VMs fall under the category of Infrastructure as a Service (IaaS). Cloud providers also offer less customizable products that fall under Platform as a Service (PaaS), such as "serverless computing" and "containers," which are suitable only for some applications. Appendix A.1 provides an overview of these products.

[8]We provide more details on VMs and VM deployment in Appendix A.2. For technically-oriented readers, Appendix A.3 describes VM use for two real-world applications in web development and machine learning.

(CPU)—the number of independent processing units that determine the computing capacity of a VM. While the price of a VM can vary based on these attributes, firms are usually charged a fixed rate per unit of time a given VM is allocated to them, regardless of the use case or actual duration. Multiplying the number of cores by the VM duration gives *core-hours*, the unit of compute resource we use throughout the paper.

A key advantage of cloud computing is that firms can automatically deploy VMs without direct human intervention. This feature, known as autoscaling, allows firms to specify rules to adjust compute resources based on demand fluctuations. For example, an online retailer can schedule additional VM deployment every Monday morning when its traffic peaks, whereas if demand is unpredictable, firms can set rules to add more VMs when the average utilization of existing VMs exceeds a certain threshold. This feature makes it unnecessary to maintain excess capacity to meet peak demand.[9]

In summary, linear pricing, near-instant scalability, and minimal transaction costs make cloud computing a variable input with negligible adjustment costs and dynamic frictions (Asker et al., 2014). Moreover, all firms have access to the same technology and pricing model regardless of their location or industry. These features make the cloud an ideal setting for studying how firms integrate and use new technologies, as we can abstract away from many of the frictions associated with dynamic inputs and rule out external factors that directly affect firms (Syverson, 2011; Restuccia and Rogerson, 2017).

2.3 Determinants of Productivity in Cloud Computing

While cloud computing offers numerous advantages, it also introduces challenges that firms must address for efficient use. Academic literature and industry sources highlight two main challenges: the need for complementary organizational investments and the development of skills specific to cloud technology.

First, relative to on-prem computing, cloud computing can exacerbate latent monitoring problems within organizations due to its positive marginal cost and lack of firm-level capacity constraints. Engineers who manage VMs may not naturally have an incentive to be cost-conscious, creating principal-agent problems for the firm. Organizational inertia can make it hard to address these problems, as transitioning to the cloud requires new practices, referred to as "digital capital" by Tambe et al. (2020), that may be difficult or take time to implement (Bresnahan et al., 2002; Garicano, 2010; Bloom et al., 2012).[10] For example, 81% of respondents in an industry survey report that "their development teams

[9] All major cloud providers offer autoscaling with no additional fees on their platforms (Fazli et al., 2018). In addition, cloud providers offer predictive autoscaling, which attempts to spin up additional capacity in advance of, rather than in response to, demand spikes. See documentation from AWS; Azure; Google Cloud.

[10] These challenges were also observed in the previous major shift in computing, the transition from mainframe to client/server architectures, as documented by Bresnahan and Greenstein (1996).

are embracing the cloud . . . faster than the rest of the organization can adopt and manage them" (Couchbase, 2022). Another example is "shadow IT," the widespread practice where different units of the firm use IT resources without the oversight or knowledge of the IT department (Haag and Eckhardt, 2017).

Second, like many other technologies, cloud computing requires new skills (Bresnahan et al., 2002; Bartel et al., 2007). Chief among these are choosing the right VM and effectively using various tools available in the cloud, both of which require employee adaptation and learning. Indeed, one industry report states that finding the best match for a workload in a cloud provider's inventory is "easier said than done," with many teams "simply choos[ing] instances they know and have used before," which tends to "underutiliz[e] other resources that they have paid for" (CAST AI, 2024).

The challenge of optimizing cloud usage has led to the development of a plethora of first- and third-party tools to help firms become more efficient. First, cloud platforms themselves offer firms various ways to view and manage their usage and costs, often proposing steps to eliminate underutilized VMs. We provide an overview of these tools in Appendix A.2. Second, a large and growing cloud optimization consulting industry offers services to improve firms' efficiency. These consultants engage more deeply with the firm and provide more tailored recommendations than first-party tools do. This market was estimated to be worth $17.6 billion in 2022 and is projected to surpass $80 billion by 2030.[11]

To sum up, existing evidence suggests that firms' primary obstacles to using the cloud optimally are economic rather than technological in nature. Such economic factors have long been recognized as key drivers of productivity dispersion among firms (Leibenstein, 1966; Syverson, 2011). Moreover, the existence of the growing cloud consulting industry, combined with survey evidence showing that firms themselves view compute optimization as a top priority, indicates both strong demand for efficiency improvements and that such improvements are technologically feasible (Flexera, 2023a).

3 Data and Summary Statistics

This section introduces the datasets used in our analysis and presents summary statistics.[12] We provide a more detailed description of the data in Appendix B.

[11] Yahoo Finance— Global Cloud Computing Report. For an example of the tools provided by one of the startups in this industry, see Figure OA-7. Several case studies document significant cost savings from implementing best practices; for example How 6 Companies Saved up to 80% on Cloud Costs – Case Studies.

[12] Studies of IT typically use data on firm-level investments or adoption. Common sources include market research data, such as those from Computer Intelligence InfoCorp and Harte-Hanks (Bresnahan et al., 2002; Bloom et al., 2012), or administrative data, such as IT expenditures in France (Lashkari et al., 2024) and technology use in the US (Zolas et al., 2021). While our dataset is narrower in scope, covering a single technology, it offers more granular information, making it a useful complement to existing datasets.

3.1 CPU Utilization Data

Our primary dataset includes CPU utilization information from over one billion VMs provided by a global cloud computing provider. CPU utilization measures the percentage of a computer's processing capacity currently in use relative to its maximum capacity and is the primary metric for evaluating computing efficiency (Mason et al., 2018). CPU utilization is typically recorded at 5- or 10-minute intervals by computer systems. To make this raw dataset more manageable, we aggregate it to the VM-day level by recording the CDF of CPU utilization for each day the VM is active. We then impose sample restrictions to remove short-lived VMs as described in Appendix B.5.

The CPU utilization data cover nearly 100,000 firms intermittently between 2017 and 2023, with varying durations each year. In 2017, the dataset covers approximately 60 non-consecutive days, while in 2018 and 2019, it includes a 30-day period each year. Although no data are available for 2020 and 2021, we have 12 months of data from July 2022 to June 2023. While the intermittent nature of the data restricts some analyses, it still allows us to analyze both short-term and long-term dynamics over a six-year period.[13]

For each VM in our dataset, we observe its duration, an anonymized ID of the firm using the VM, and an anonymized unit ID for multi-unit firms. A "unit" in our data refers to a group of users who share a billing account with the cloud provider and an administrative structure overseeing the VMs. These units may correspond to product teams or functional divisions within the company, though no further information is available. As such, we will maintain the "unit" terminology in the rest of the paper.

We also observe various VM attributes, including the operating system, memory, cores, and region of the data center hosting the VM (EU, US, or other), along with anonymous identifiers indicating the hardware model and data center. Using this information, we categorize VMs into *family* (categories of workloads the VM is optimized for, such as compute- or memory-intensive), *series* (groupings based on performance or hardware), and *configuration* (the combination of memory, cores, operating system, VM series, and data center). Appendix B.2 contains more details on each of these categories.

3.2 Firm and Unit Level Data

In addition to our CPU utilization data, we have a balanced panel with the monthly cloud usage of all of the firms and units in our sample. These data cover the period from 2017 to mid-2023 and include each firm's and unit's normalized total computation in each month.

[13]The company stored these data in these periods for independent reasons and made them available to us for research. The dataset is sampled to reduce its size and exclude firms with very low cloud usage. See Appendix B.4 for details on the sampling procedure. Due to the sensitivity of our data, we do not report exact numbers of observations or levels of certain variables throughout the paper.

With this information, we can track firms' entry into or exit from the cloud provider and changes in compute usage over time.[14] In addition, we observe the firm and unit region (EU, US, or other), whether the firm is multinational, industry (2-digit SIC), and quartiles of a firm size measure.[15]

3.3 Publicly Available Cloud Data

We supplement our main dataset with publicly available CPU utilization data from Google Cloud and Microsoft Azure. These datasets provide additional information not present in our main data and allow us to validate our findings in other cloud computing environments. One such dataset is the 2019 Power Traces from Google Cloud, which records electricity consumption and CPU utilization of VMs at 5-minute intervals within a data center. Using this dataset, we estimate the relationship between CPU utilization and electricity consumption to quantify potential electricity savings from efficiency improvements.

3.4 Summary Statistics

Table 1 presents summary statistics. Column (1) of Panel A shows the firm shares across 1-digit SIC industry categories, highlighting the predominant sectors—services and software—which account for 36.3% and 23.1% of firms, respectively. Although smaller in share, our sample also includes more traditional industries such as manufacturing and transportation. Column (2) reports that the share of firms with multiple units that use the cloud ranges from 27.2% to 48.4% across industries, indicating that we can observe compute productivity at the unit level in many firms. Column (3) shows that firms' average cloud experience ranges from 3.5 to 4 years across industries.

Panel B reports the firm distribution by region, showing that 33.2% are located in the US, 22.4% in the EU, and 7.2% are classified as multinational. As expected, multinational firms have 0.7 years more experience on average than domestic firms. Lastly, Panel C presents summary statistics at the VM level. The average VM lifespan is 2.5 days, with a standard deviation of 13.7, indicating substantial heterogeneity in VM durations. Similarly, we observe considerable variation in the number of cores (mean = 7.9, s.d. = 12.0). Finally, 72% of the VMs are downsizable, as defined in Section 4.1, meaning that in most cases, firms can choose smaller-capacity substitute VMs if they overprovision.

[14] We observe data from one cloud provider only, and therefore cannot tell if a firm used a different cloud provider beforehand. As such, our measure of cloud experience is a lower bound on firms' actual experience.

[15] The firm size measure is constructed using internal data to approximate the number of employees. The 2-digit SIC industry classifications we observe are broader than the 3- or 4-digit classifications commonly used in the literature. Although this is a limitation of our dataset, it is likely less restrictive than in other settings because we study a productivity measure that is obtained from a uniform input without estimating industry-level production functions.

Table 1: Summary Statistics

	Share (%)	Multi-unit (%)	Average Cloud Experience (Years)
	(1)	(2)	(3)
	Panel A. Industry Category (1-digit SIC)		
Services	36.35	27.19	3.58
IT/Software	23.07	42.45	4.25
Retail Trade	12.58	31.20	3.58
Manufacturing	9.19	43.32	3.82
Public Administration	7.48	46.78	4.06
Transportation and Communications	6.09	44.26	3.94
Finance, Insurance, and Real Estate	4.53	48.35	3.83
Other	1.15	33.19	3.54
	Panel B. Firm Region		
Other	37.46	36.48	3.80
US	33.22	29.78	3.74
EU	22.37	35.29	3.76
Multinational	7.20	-	4.54
	Panel C. VM Statistics		
	Mean	SD	Mode
Duration (days)	2.52	13.75	1
Number of cores	7.88	12.04	4
Share downsizable	0.72	0.45	1

Notes: Panels A and B report summary statistics for firm, industry, and region characteristics in our sample, averaged across months over the sample period. Industries are classified by 1-digit SIC codes, with the exception of software firms, which are carved out of the services industry. Column (1) reports the unweighted share of firms in each category. Column (2) reports the share of firms with multiple units in each category, and Column (3) reports the average number of years since the firm first used cloud computing. Because multinationals, by definition, operate multiple units across various geographies, their multi-unit share is omitted. Panel C provides unweighted summary statistics for VMs created during a week in 2022, including their duration, number of cores, and the share of downsizable VMs. The definition of downsizability is provided in Section 4.1. Appendix B provides detailed information on the implementation of these estimation procedures.

4 Measurement of Productivity in Computing

This section describes our approach to measuring compute productivity. Our goal is to develop a measure that quantifies how efficiently firms use computing to study productivity with a new technology. As such, instead of focusing on the entire production function, we analyze a single input in minute detail using high-frequency data. Nonetheless, in Appendix C.2, we demonstrate that our measure can be interpreted as compute-augmenting productivity in a full production function (Doraszelski and Jaumandreu, 2018; Raval, 2019).

In developing our measure, we ask the following question: how would a cost-minimizing firm that has the same compute needs and faces the same menu of VM choices as a given

Table 2: Example of Cloud Usage for a Hypothetical Firm

VM	Capacity	Duration	Actual Input Use	Peak Util.	Peak Job Load	Efficient Usage	Efficient Input Use	Efficiency
	[a]	[b]	[c] = [a] × [b]	[d]	[e] = [a] × [d]		[f]	[g] = [f] ÷ [c]
A	2-core	10h	20 ch	0%	0 cores	**Eliminate**	0 ch	0%
B	4-core	5h	20 ch	25%	1 core	**Downsize**	10 ch	50%
C	8-core	10h	80 ch	75%	6 cores	**Maintain**	80 ch	100%
Total			**120 ch**				**90 ch**	**75%**

Notes: This table presents CPU utilization data for three different VMs used by a hypothetical firm. VM A, with a 2-core capacity and no utilization, suggests idleness (0% peak utilization). VM B, with a 4-core capacity used at 25% peak utilization, indicates overprovisioning. VM C shows the use of an 8-core capacity at 75% peak utilization, labeled as properly provisioned. The total CPU input across all VMs accumulates to 120 core-hours, whereas the efficient use is 90 core-hours, reflecting an overall efficiency of 75%.

firm deploy compute resources in the cloud? The share of resources this hypothetical firm uses relative to the firm's actual usage defines compute productivity.[16] Importantly, this approach goes beyond simply using the utilization rate as an efficiency measure and instead considers the menu of VMs the firm faces when deploying compute resources.

In what follows, we first describe how we measure compute productivity. We then argue that our method aligns with industry practices and has first-order relevance to measuring firm efficiency in computing. Finally, we compare and contrast our measure to TFP. Appendix C.1 provides a formal exposition of our measure, and Appendix D.2 presents additional implementation details omitted from the text.

4.1 Constructing Compute Productivity

To illustrate our measure, consider the usage pattern of a hypothetical firm documented in Table 2. Suppose there are three sizes of VMs available to the firm: 2-core, 4-core, and 8-core. The firm runs jobs on three VMs: VM A, a 2-core machine; VM B, a 4-core machine; and VM C, an 8-core machine. VMs A and C are active for 10 hours, and VM B lasts 5 hours. Therefore, the total computing resource the firm pays for—the firm's total input use—is $2 \times 10 + 4 \times 5 + 8 \times 10 = 120$ core-hours.

Now suppose that we observe the following utilization patterns. The firm did not actually utilize VM A at all; the peak load for the job was 0 cores. While the firm did use VM B, it utilized at most 25% of the computing capacity at any given moment. Therefore, the peak load of the job was $25\% \times 4 = 1$ core. Finally, on VM C, the peak utilization was 75%, meaning the peak load of the job was $75\% \times 8 = 6$ cores. These peak loads define the

[16] Our cost minimization framework bears similarity to the traditional cost-minimization assumption in neoclassical production theory (Shephard, 1953). Instead of firms taking input prices as given and finding the inputs that minimize costs of producing a given output level, they take a menu of VMs as given and find the VM that minimizes the cost of producing a given compute load.

number of cores that were needed to perform the observed workloads.[17]

What would a perfectly cost-minimizing firm have done if it had the same computing needs and faced the same set of available VMs? First, it would not provision VM A, which the firm did not use to compute anything. By doing so, the firm can avoid paying for 20 core-hours of compute. Second, given that the job run on VM B only requires a capacity of 1 core at peak, the firm would downsize it to a 2-core machine, reducing the input usage from 20 core-hours to 10 core-hours. Finally, since the job run on VM C requires a peak capacity of 6 cores, it cannot be downsized (the next smallest available machine is 4 cores); the firm would provision the same 8-core machine and use the same 80 core-hours of input. Overall, a cost-minimizing firm would have only used 90 core-hours, while this firm actually used 120 core-hours. As such, we conclude that this firm could have used 75% of the input it actually used to get the same output.

Our measure of compute productivity generalizes the logic of this example. We assign each VM j run by firm i on day t a productivity $\omega_{ijt} \in [0, 1]$, where, at a high level,

$$\omega_{ijt} = \frac{\text{Minimum number of cores needed for VM } j\text{'s job}}{\text{Actual cores used for VM } j}.$$

To determine the minimum number of cores needed for VM j's job, we use its peak utilization over a seven-day period.[18] Focusing on peak utilization accounts for many factors that may lower overall utilization without reflecting any actual inefficiency. For instance, our approach does not consider low average utilization due to fluctuating demand as inefficient, nor does it credit potential efficiency gains from turning a briefly idle VM off and on. We take peak utilization to be the 95th percentile of the CPU utilization distribution, following the recommendations made by cloud providers, as well as those in the computing literature (Reiss et al., 2012; Cortez et al., 2017).[19]

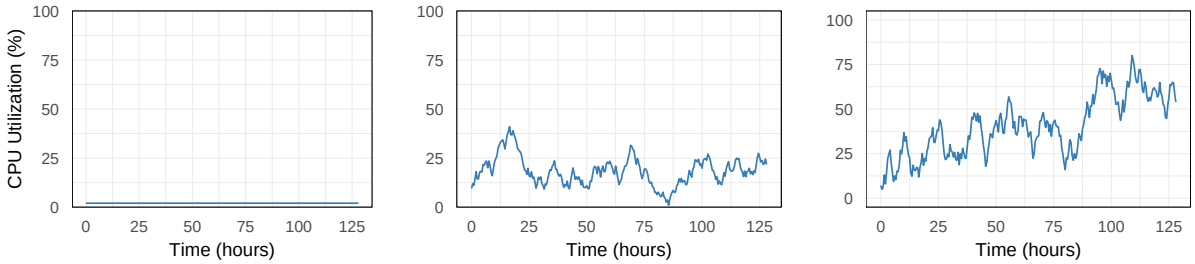
Our method identifies two distinct sources of inefficient VMs: idleness and overprovisioning. VM j is *idle* if its peak utilization is under 10%. This low level of utilization is explained by background CPU processes rather than any actual CPU usage by the user (Breitgand et al., 2014). Because an idle VM does not have any compute output, the minimum number of cores needed for its job is zero, hence $\omega_{ijt} = 0$.

[17] If the job's peak load exceeds the CPU capacity of the VM, the job will take longer to complete as tasks are added to the processor's execution queue faster than they can be executed (Hennessy and Patterson, 2012). This is different from exceeding the memory capacity, which can cause the job to crash.

[18] A job is a computation profile executed over the lifespan of a VM. For VMs shorter than seven days, we use the peak utilization observed during their lifetime. Appendix D.2 provides additional details and two illustrative examples from the data.

[19] The 95th percentile is less sensitive to measurement errors or spikes stemming from random events like software updates than the maximum CPU utilization.

Figure 2: Illustration of Different CPU Utilization Patterns



(a) Idle VM **(b)** Overprovisioned VM **(c)** Properly Provisioned VM

Notes: This figure displays three different CPU utilization patterns. Panel (a) shows the CPU utilization of an idle VM, maintaining a utilization near 0% throughout its duration. Panel (b) shows an overprovisioned VM with a peak utilization of about 40%. Panel (c) shows a properly provisioned VM with peak utilization above 75%.

VM j is *overprovisioned* if its job would have reached a peak utilization of 90% or less on a VM that has fewer cores but is otherwise similar (and is not idle). In this case, the minimum number of cores needed to run the job is that of the smallest such VM that fits the job's peak load. If such a smaller substitute exists—that is, if there exists a configuration with fewer cores but the same machine family, memory, data center, and operating system—we call the VM *downsizable*.[20] Typically, cores scale in powers of two, so an overprovisioned VM will often be resized to a VM with half the number of cores, in which case $\omega_{ijt} = 0.5$.[21] Finally, if a VM is neither idle nor overprovisioned, it is *properly provisioned*. In this case, $\omega_{ijt} = 1$.

Figure 2 is another illustration of our measure, with each panel displaying the CPU utilization of a VM over time. Panel (a) is an idle VM. While it was not utilized by the firm that provisioned it, this VM still consumes around 50% of the electricity of a fully utilized VM and is not typically allocated to another firm by the provider (Kansal et al., 2010).[22] Panel (b) is a potentially overprovisioned VM. Although the VM was continuously used, it could have also fit on a substitute VM with half the number of cores, and would be marked as overprovisioned if such a VM exists. Finally, panel (c) is a properly provisioned VM

[20] See Figure OA-3 for an illustration of downsizability, and Appendix E.5 for further discussion. We focus on these characteristics because VMs that share them are readily substitutable. Although networking is an attribute of a VM, we do not include it in our downsizability definition because it often does not change within a VM series (see Figure 1).

[21] In addition, in some cases, it is possible to downsize to a quarter of the number of cores, i.e., if the peak utilization is 20% and a VM with one-fourth of the number of cores is available. In this case $\omega_{ijt} = 0.25$. Although we omit this category from the description for brevity, as it is rarely observed in the data, we include it in our calculations.

[22] See, for example, Azure, which states that it does not oversubscribe resources, and Google Cloud and AWS, which explicitly list a small number of products that are technically feasible to oversubscribe. Cloud providers have service-level agreements (SLAs) with their clients, committing to maintain availability with extremely high probability (e.g., 99.99%, commonly referred to as "four 9s"). Therefore, reallocating idle VMs could risk violating these SLAs and pose monetary and reputational risks (Perez-Salazar et al., 2022).

with a peak utilization of 75%, which means the job would not fit on a smaller VM.[23]

After calculating VM-day-level estimates, we aggregate them to construct a firm-level compute productivity measure as in the example in Table 2. The productivity of firm i in month m with VMs J_{im} is the core-hour weighted average of VM-day-level productivities:

$$\omega_{im} = \frac{\sum_{j \in J_{im}} \sum_t \omega_{ijt} ch_{ijt}}{\sum_{j \in J_{im}} \sum_t ch_{ijt}} \tag{1}$$

where ch_{ijt} is the core-hours of VM j on day t. This measure essentially quantifies (one minus) the share of resources wasted by the firm in month m. Using the same procedure, we also estimate unit-month and firm-VM type-month level compute productivity to analyze the productivity of units and firms' productivity across different VM types. In addition, we estimate separate measures tracking idleness and overprovisioning, where each measure is defined as one minus the share of idle or overprovisioned core hours, so that higher values indicate greater efficiency.[24]

Finally, as described in Appendix D.2, we reestimate all our measures using fixed-effect regressions that control for job- and time-specific factors—such as the VM type and day of the week—to isolate the impact of observables on compute productivity.[25] Although we use the estimates without controls from Equation (1) for our baseline results, as the choice of VM may itself reflect productivity, we report our main results with various controls in Appendix H.

4.2 Discussion and Limitations of Compute Productivity

While our measure of compute productivity is novel to the economics literature, it aligns closely with industry practices and bears similarity to previously used utilization-based efficiency measures in economics. First, cloud providers' definitions of compute efficiency are similar to our own; for example, Microsoft Azure generates a "resize" recommendation if the 95th percentile CPU utilization of a job would be under 80% on a less expensive VM, nearly identical to our overprovisioning definition.[26] Second, the cloud optimization

[23] In practice, our data are at the VM-day level. Therefore, we classify each VM-day based on the peak utilization over the seven-day periods it is a part of. For example, we mark a VM-day as idle if it is part of a seven-day period in which the peak utilization is under 10%.

[24] These measures are useful because they likely reflect the distinct sources of inefficiencies discussed in Section 2. While idleness is mainly driven by incentive and monitoring frictions within the firm, overprovisioning may additionally reflect limited skill in VM selection.

[25] The firm fixed effects estimates from these regressions without any controls correspond to our baseline measures in Equation (1). In these fixed effect regressions, firm fixed effects can only be compared within a connected set of firms (Abowd et al., 1999; Metcalfe et al., 2023). Across all regressions, we find either a single connected set or that the largest connected set includes more than 99% of firms.

[26] Azure— How-to Guides. AWS and Google Cloud have their own similar definitions of idleness and overprovisioning. AWS—API Reference; Google Cloud—Overprovisioned VMs; Google Cloud—Idle VMs.

consultants described in Section 2 also focus on idleness and overprovisioning; one such startup states that the "best practices for optimizing cloud costs" are to "identify under-utilized resources, detect idle resources, [and] rightsize cloud resources," while another lists as the most common mistakes "picking oversize instances," "leaving instances running idle," and "forgetting to clean up stale resources."[27] Finally, the idea behind our measure—that the extent to which firms utilize their inputs is a dimension of firms' productivity—has been used in several other contexts in economics (Hubbard, 2003; Braguinsky et al., 2015; Butters, 2020, for example).[28]

Nevertheless, there are three caveats to note about our measure. First, it does not capture other kinds of computing inefficiencies like poorly written code that consumes excess compute. Our measure should therefore be viewed as efficiency due to provisioning decisions, conditional on all other factors affecting CPU utilization. Second, our measure excludes other VM utilization metrics and focuses solely on CPU utilization, which is the standard metric in the industry and the main driver of energy consumption (Rivoire et al., 2008). However, we perform robustness checks with our limited data on memory and network utilization in Appendix E.1. Finally, we do not account for more elaborate potential efficiency improvements, such as consolidating multiple VMs with 70% peak utilization into fewer VMs. While theoretically possible, such adjustments may not be practical and would require additional implementation assumptions.

4.3 Comparison of Compute Productivity with TFP

Since the productivity literature has overwhelmingly focused on TFP, it is important to discuss the relationship between TFP and our compute productivity measure. TFP is defined as the residual in the production function after accounting for the contributions of observed inputs (Syverson, 2011; Ackerberg et al., 2015). It can therefore correspond to various unobserved factors such as technology and management, offering limited insight into specific mechanisms by itself without further analysis.

Compute productivity, in contrast, captures a physical productivity measure specific to a single input that would otherwise be embedded within TFP if not modeled explicitly. To see this, suppose a firm transforms computing C and another input Z into output Y according to the Cobb-Douglas production function $Y = AC^{\alpha}Z^{\beta}$. However, the amount

[27] CASTAI— Cloud Cost Optimization and TechCrunch—The 10 Biggest Mistakes Made With AWS.

[28] A related measure called capacity utilization, which tracks the "ratio of the actual level of output to a sustainable maximum level of output" from manufacturing firms, is used in macroeconomics to study topics such as inflation prospects and business cycle volatility (Corrado and Mattey, 1997). The key distinction between this measure and our own is that in cloud computing, firms can adjust their VM provisioning in real time, whereas a manufacturing firm's production capacity is typically fixed in the short to medium run. As such, unutilized capital is not necessarily interpreted as inefficiency in that literature (Williamson, 1966).

of computing the firm actually receives is $C = \omega I$, where I is compute input and ω is compute productivity. Substituting, we get the production relationship in terms of inputs, $Y = A\omega^{\alpha} I^{\alpha} Z^{\beta}$, where the "true" TFP is $A\omega^{\alpha}$. In other words, ω is a component of TFP, and the component becomes more salient the larger α is, the firm's output elasticity of compute.[29] By focusing on this input-specific measure, we can analyze the mechanisms underlying compute inefficiencies, link our measure to actual resource usage (compute and electricity), and study firms' efficiency with new technologies in a way that is hard to analyze using TFP, thereby complementing the existing literature.

Of course, these advantages come with important limitations. First, as we do not consider the full set of inputs or output, we cannot link compute productivity to firm outcomes such as sales and profit to measure its overall impact. Second, since our compute productivity is utilization-based and bounded between 0 and 1, it differs conceptually from TFP, limiting direct comparisons between the two. We believe these trade-offs are worth making to study a transformative and widely used technology with the granularity we do. We refer readers to Section 1 for the extensive literature on the impact of digital technologies on firm outcomes and Appendix C for further discussion of compute productivity.

5 Empirical Facts on Compute Productivity

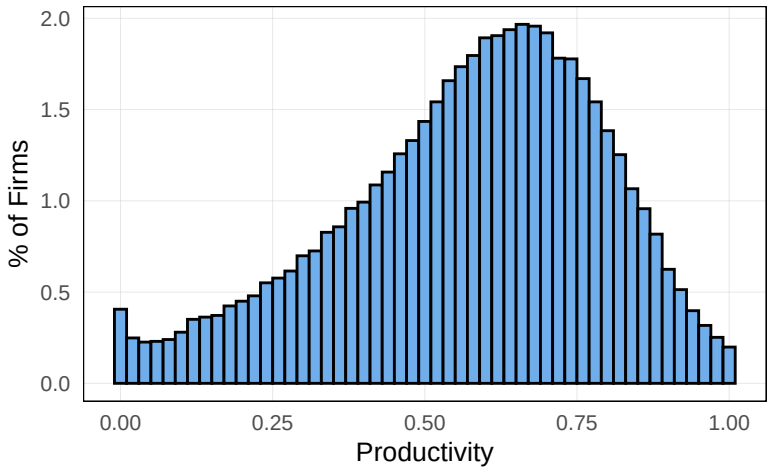
We begin our analysis by presenting several facts about dispersion, persistence, and other aspects of compute productivity. In the subsequent sections, we explore the mechanisms that make firms more productive in computing in more detail and analyze the evolution of compute productivity after adoption. Appendix D.3 provides details on the estimation procedures conducted in this section.

5.1 Dispersion and Persistence of Compute Productivity

Figure 3 shows the unconditional distribution of firm-level compute productivity, while Table 3 reports dispersion statistics estimated under various controls. The estimates, which range from 0 (fully idle VMs) to 1 (no idle or overprovisioned VMs), reveal substantial heterogeneity across firms. The distribution is roughly normal, with a median of 0.62 and a mean of 0.60, implying that firms use only 60% of their provisioned compute resources efficiently on average. However, there is a large dispersion around this mean: controlling for industry and time, the 90-10 percentile ratio is 3.53 and the interquartile ratio is 1.73, indicating that some firms use compute resources much more efficiently than others.

[29] For simplicity, our example uses a single input, but it extends to multiple inputs. This example also exploits a key property of the Cobb-Douglas functional form that all factor-augmenting productivity terms are Hicks-neutral. However, in a more general production function with nonseparable inputs, $Y = AF(C, Z)$, the omission of compute-augmenting productivity would still likely to be absorbed into the residual term A.

Figure 3: Dispersion of Firm-Level Compute Productivity



Notes: This figure shows the distribution of firm-level compute productivity as described in Section 4.1. The x-axis represents productivity levels ranging from 0 to 1, while the y-axis shows the percentage of firms. Productivity dispersion by industry is reported in Figure OA-4.

Next, we turn to the within-firm dispersion reported in Table 3. Different units within the same firm exhibit significantly different compute productivity: 56.4% of dispersion is explained by within-firm variation. We further decompose this within-firm variation into within- and between-region components using data from multinational firms and find that cross-region differences explain 18.2% of the within-firm variation. Although this share is modest, it indicates that geographic location contributes meaningfully to productivity differences even within the same firm. These regional differences could reflect heterogeneity in human capital, timing of cloud adoption, or managerial practices, as documented in Bloom et al. (2012). Overall, the large within-firm dispersion emphasizes the importance of within-firm dynamics, which we will revisit in Section 7.[30]

We next investigate another canonical finding on productivity: its persistence over time. The AR(1) estimates in Panel B of Table 3 indicate that compute productivity is highly persistent, with coefficients of 0.93 and 0.64 at 1-month and 1-year horizons, respectively. While persistence declines over longer horizons, it remains economically meaningful even at the 5-year horizon, with an AR(1) coefficient of 0.32. Moreover, as shown in Table OA-7, the different components of compute productivity—idleness and overprovisioning—are also persistent, with idleness being more persistent than overprovisioning in the long run.

Finally, we examine the extent to which observables explain the dispersion and per-

[30] Another important aspect relating to within-firm heterogeneity previously studied in the IT context is the degree of organizational centralization (McElheran, 2014). While more centralized decision-making could alleviate monitoring frictions, it could also reduce local information advantages. In Table OA-5, we observe a positive correlation between the level of centralization and productivity among multi-unit firms, suggesting that monitoring frictions may dominate local information advantages in the context of cloud computing.

Table 3: Dispersion and Persistence of Compute Productivity

	No Control (1)	Industry (2)	Time (3)	Industry/Time (4)
	Panel A. Productivity Dispersion Statistics			
Dispersion:				
Mean	0.60	-	-	-
Median	0.62	-	-	-
90-10th perc ratio	3.51	3.49	3.53	3.53
Interquartile Ratio	1.72	1.72	1.73	1.73
R^2	-	0.009	0.002	0.012
Within-Firm Decomp. (%):				
Between-firm	33.08	32.33	44.28	43.57
Within-firm	66.92	67.67	55.72	56.43
Within-Firm-Between-Region Decomp. (%):				
Between-region	5.88	-	18.22	-
Within-region	94.12	-	81.78	-
	Panel B. Persistence (AR(1) Coefficients)			
1-month persistence	0.93	0.93	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)
1-year persistence	0.64	0.63	0.64	0.63
	(0.00)	(0.00)	(0.00)	(0.00)
5-year persistence	0.32	0.30	0.32	0.30
	(0.01)	(0.01)	(0.01)	(0.01)

Notes: This table reports the dispersion and persistence of compute productivity conditional on different sets of controls. Panel A presents statistics on the productivity distribution, the decomposition of productivity dispersion into within- and between-firm components, and the decomposition of within-firm dispersion into between- and within-region components. Panel B shows the persistence of compute productivity through 1-month, 1-year, and 5-year AR(1) coefficients, with standard errors clustered at the firm level in parentheses. The industry refers to a 2-digit SIC code, and time refers to a month. Details on the estimation are provided in Appendix D.3, along with a visualization of persistence in Figure OA-5. "–" indicates that the corresponding control is not meaningful to include in that specification.

sistence in compute productivity. The R^2 estimates in Panel A suggest a limited role of observable factors: they explain at most 1.2% of variation in compute productivity, and the magnitudes of dispersion and persistence are similar across various control specifications. These results point to the role of unobserved heterogeneity across firms, mirroring common findings in the literature (Fox and Smeets, 2011; Metcalfe et al., 2023). Moreover, it is consistent with Murciano-Goroff et al. (2024), who find similar evidence in the context of digital technologies by showing that observed firm characteristics explain little variation in firms' propensity to use software with known vulnerabilities.

It is useful to interpret these findings in the context of the broader productivity literature. We should expect compute productivity dispersion to be directionally ambiguous

Table 4: Compute Productivity Differences by Firm Characteristics

	Compute Prod. (1)	Idleness Prod. (2)	Overprovisioning Prod. (3)
	Panel A: Firm Size (% difference relative to 1st quartile)		
2nd quartile	0.010	-0.009	0.061
	(0.003)	(0.002)	(0.005)
3rd quartile	0.028	-0.007	0.127
	(0.003)	(0.002)	(0.004)
4th quartile	-0.038	-0.060	0.075
	(0.003)	(0.002)	(0.004)
	Panel B: Region (% difference relative to EU)		
US firms	0.052	0.048	0.033
	(0.004)	(0.003)	(0.002)
Industry FE	X	X	X
Time FE	X	X	X

Notes: This table reports productivity differences by firm size and region. Panel A shows the productivity estimates for firms in different size quartiles relative to the 1st quartile, obtained from a regression of the logarithm of outcome variables on size quartiles. Panel B shows the differences between US and EU firms using the same method. Columns (1-3) report the results for compute productivity, idleness productivity, and overprovisioning productivity, respectively. The estimates are obtained from firm-month level regressions, where the outcome variable is the logarithm of productivity specified in columns, with the control variables given in the bottom of the table. The construction of firm-month level productivity estimates is described in Section 4.1 and Appendix D.2. Standard errors are clustered at the firm level and are reported in parentheses.

relative to canonical results: while focusing on a single input may reduce dispersion, the evolving nature of computing technology could lead to larger dispersion. Our findings indicate that dispersion in compute productivity is comparable to other estimates in the literature. For instance, Syverson (2004b) reports an average 90–10 percentile ratio of 2.45 in US manufacturing, while other estimates include an interquartile ratio of 1.76 in US retail (Foster et al., 2006) and a 90–10 ratio of 5 in Chinese and Indian manufacturing (Hsieh and Klenow, 2009). Our estimates of within-firm heterogeneity are also consistent with limited empirical evidence in the literature: Bloom et al. (2019) and Orr (2022) both find that within-firm dispersion explains 40% of the variation in managerial practices and product-specific productivity, respectively.

5.2 Compute Productivity Differences by Firm Characteristics

This section examines how compute productivity varies by key firm characteristics, such as size and location. The drivers of compute productivity described in Section 2 suggest no clear *a priori* relationship between firm size and compute productivity. While larger firms may benefit from greater resources for organizational adjustments and skill development, they might also face more organizational frictions. Our findings in Table 4

support this intuition as there is no consistent relationship between firm size and compute productivity. While firms in the 2nd and 3rd quartiles of the size distribution are 1.0% and 2.8% more productive than those in the 1st quartile, respectively, the largest firms are 3.8% less productive than the smallest firms. However, when we examine the components of productivity, the largest firms show 6.0% lower idleness productivity but 7.5% higher overprovisioning productivity compared to the smallest firms (i.e., larger firms are more likely to leave VMs idle but less likely to overprovision them). This pattern may reflect greater monitoring and incentive problems faced by larger firms, potentially resulting in more idle VMs, while their superior IT capabilities reduce overprovisioning.

Panel B focuses on firm location and reports the compute productivity differences between US and EU firms. We find that US firms outperform their European counterparts, achieving 3.3% to 5.2% higher productivity across different measures. This finding is consistent with Bloom et al. (2012), who find that "Americans do IT better," mainly due to differences in management practices.

Taken together, the empirical evidence on compute productivity aligns broadly with the extensive literature on firm productivity, despite differences in measurement approaches and our focus on computing. In this way, our paper complements the prior literature by showing that persistent productivity differences continue to exist with digital inputs.

5.3 Robustness Checks and Discussion

Many confounding factors could generate the dispersion in compute productivity we observe. In Appendix E, we go over several of them and present robustness checks that indicate they are unlikely to explain our results. We now describe these robustness checks.

One important potential explanation is demand volatility: seemingly inefficient firms may purposefully maintain idle and overprovisioned VMs as a precaution against rare demand spikes that ultimately do not materialize. As discussed in Section 2.2, we view this as a form of inefficiency in cloud computing, given the available tools that can automatically scale compute resources. Nevertheless, we conduct two exercises to show that this concern is unlikely to explain our results. First, we estimate multiple measures of firm-level demand volatility using our data and find that they explain less than 1% of compute productivity dispersion. Second, we calculate the probability that firms reach the capacity of their chosen VM and find that high-productivity (above-median) firms do not reach capacity more frequently than low-productivity firms.

Another explanation is that firms use computing for different purposes, which could inherently have different productivity patterns. Cloud providers offer a vast menu of VMs with different observable characteristics, often tailored for different use cases. Hence,

controlling for VM characteristics—such as type and memory—can help us condition on firms' use cases. We find that while VM characteristics have some explanatory power, more than 75% of compute productivity variation is among VMs with identical configurations.[31]

Our additional exercises also confirm the robustness of the results. First, we repeat our analysis using alternative definitions for downsizability and peak utilization and find that the results are robust to these choices. Second, we extend our analysis to other dimensions of VM utilization, memory and network, and find that they are positively correlated with CPU utilization, indicating that CPU-inefficient VMs also tend to be inefficient in these dimensions. Third, we show that idleness and overprovisioning productivity, which likely have different sensitivities to alternative explanations, are positively correlated. Fourth, we estimate compute productivity using publicly available datasets from Google Cloud and Microsoft Azure and find comparable magnitudes of dispersion. Finally, as an external validity check, we find that firms with below-median productivity are 60.0% more likely to exit the cloud than those above the median, similar to existing results in the productivity literature (Syverson, 2004a; Collard-Wexler and De Loecker, 2015; Foster et al., 2016).[32]

Beyond these robustness checks, many of our results suggest that compute productivity differences are not mechanical and are economically meaningful. For example, evidence of significant short- and long-term firm learning (Section 7) and within-firm productivity dispersion (Section 5) suggest that productivity is not solely determined by time-invariant firm characteristics or use cases. Moreover, the dispersion is not solely driven by economically insignificant small firms; we show in Section 8 that compute inefficiencies materially affect total compute resource and electricity usage. Finally, well-documented productivity dispersion in various industries makes heterogeneity in compute productivity plausible, and the close parallels between our results and prior findings reinforce this conclusion.

6 What Do More Efficient Firms Do?

We have established that firms differ markedly in how efficiently they use computing resources, and that these differences persist over time. We now turn to examining which aspects of firm behavior help explain these differences to better interpret what the compute productivity measure captures. Since we lack data on firms' internal practices, we cannot

[31] As shown in Figure OA-2, machine family and memory account for 4.1% and 6.8% of the variation in compute productivity, respectively. Even when controlling for the full configuration of a VM, only 23.9% of the variation is explained, some of which is mechanical, since configuration nests downsizability, a necessary condition for overprovisioning. Moreover, some variation in productivity is naturally attributable to VM characteristics if the ability to select appropriate configurations is itself part of firm productivity.

[32] Such a result would be inconsistent with the demand volatility explanation for productivity dispersion. Cloud is especially beneficial for firms facing large demand volatility. If volatility were the main driver of low productivity, low-productivity firms should be less likely to exit the cloud, not more.

directly test their role in explaining productivity dispersion. However, we observe firms' VM provisioning patterns under different circumstances, which can be used to evaluate the behaviors that make firms more productive in computing.

In this section, we examine the association between three such patterns and productivity: responsiveness to demand changes, attentiveness to idle resources, and greater knowledge about VM types. First, firms that leverage cloud capabilities more effectively should be better at responding to changes in demand. To test this, we study provisioning decisions on weekends, when firms face a sizable drop in computing demand. Second, when there are mistakes and resources are left idle, firms with better monitoring capabilities should shut down these idle resources faster. Finally, firms with more knowledgeable employees likely deploy a broader array of VM types rather than placing all workloads on a single type.

To study these factors, we perform the following exercise. We first classify firms as "high-productivity" or "low-productivity" based on their compute productivity in 2022 relative to their industry's median. We then analyze the differences in VM provisioning behavior between these two groups in 2023. In a manner similar to difference-in-differences, our analyses compare certain aspects of VM provisioning behavior across high- and low-productivity firms relative to their within-group baselines, thereby avoiding results being mechanically driven by the level differences that are used to classify firms.

6.1 Responsiveness to Demand Changes

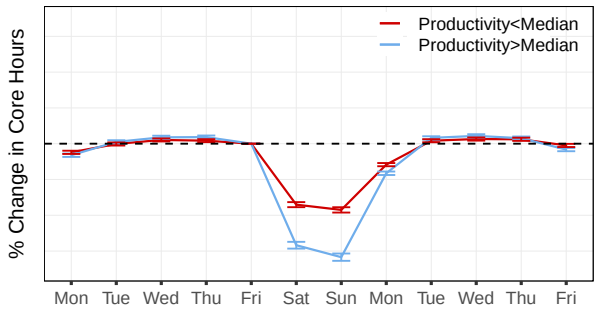
To analyze differences in responses to demand changes, we employ an event study approach and estimate the following specification:

$$y_{it} = \sum_{k=-6,k\neq 0}^{7} \beta_k^H D_{i,t-k}^H + \sum_{k=-6,k\neq 0}^{7} \beta_k^L D_{i,t-k}^L + \alpha_i + \gamma_{w(t)} + \varepsilon_{it}, \tag{2}$$

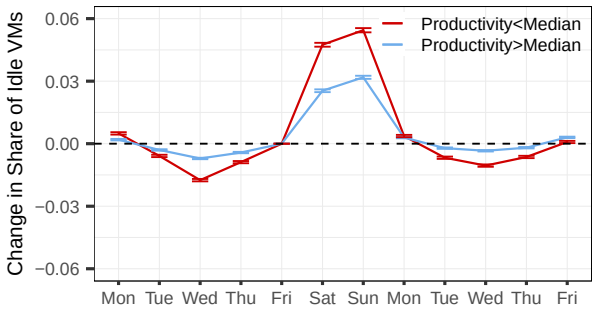
Here, y_{it} denotes the outcome variable for firm i on day t. $D_{i,t-k}^H$ and $D_{i,t-k}^L$ are indicator variables for high- and low-productivity firms, respectively, where k denotes the position of each day relative to the first Friday within a two-week period. We include firm fixed effects (α_i) to account for time-invariant firm characteristics and time fixed effects ($\gamma_{w(t)}$) for every two weeks to absorb common time trends. The coefficients of interest, β_k^H and β_k^L, represent the average change in the outcome variable relative to the first Friday for high- and low-productivity firms, respectively. We estimate this specification for two outcomes: the logarithm of total provisioned core-hours and the share of idle VMs.

Panels (a-b) of Figure 4 plot the coefficient estimates separately for high- and low-

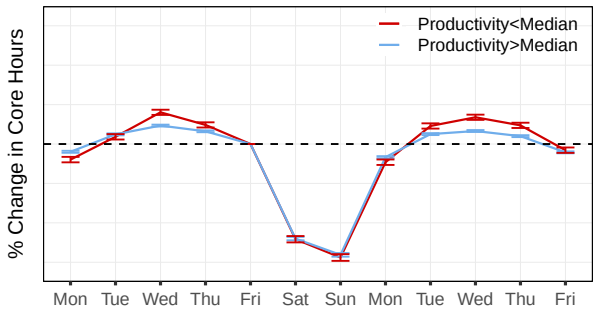
Figure 4: Firm Responses to Weekend Demand Changes by Productivity Level



(a) Change in Provisioned Resources



(b) Change in Share of Idle VMs



(c) Change in Compute Load

Notes: These figures show the estimates of β_k^H and β_k^L from Equation (2), for three outcome variables: log provisioned resources, the share of idle VMs, and log compute load. Firms are classified as above- and below-median using productivity estimates from the 2022 sample, while the regressions are estimated using the 2023 sample. The crossbars represent 95% confidence intervals clustered at the firm level. The y-axes of Panels (a) and (c) are obfuscated for confidentiality reasons.

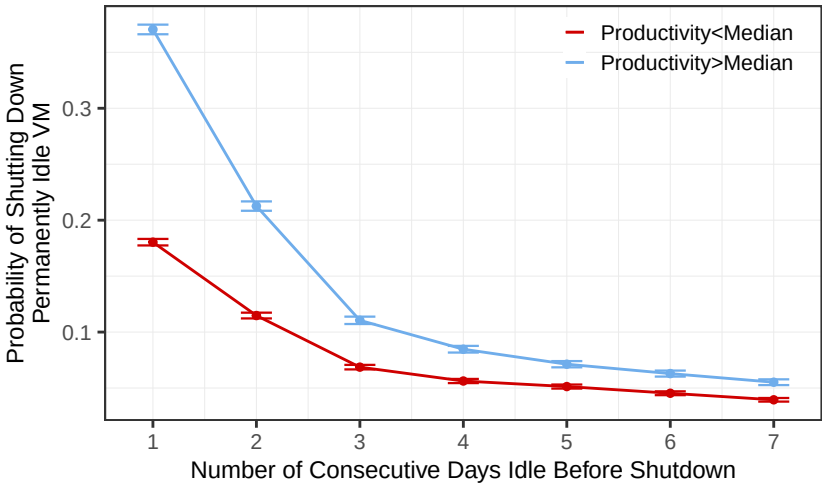
productivity firms. Both groups use fewer resources on weekends compared to weekdays. However, high-productivity firms reduce provisioned compute resources by 75.2% more than low-productivity firms, resulting in lower resource use. High-productivity firms also utilize their provisioned resources more effectively: their probability of leaving VMs idle rises by only 2.6 pp on weekends, compared to a 4.7 pp increase for low-productivity firms.

One potential concern is that these results are driven by differences in compute needs during weekends between low- and high-productivity firms. To address this, we estimate the same specification using the logarithm of compute load as the dependent variable and plot the results in Figure 4(c).[33] As expected, both groups experience declines in compute load during weekends. Notably, the magnitude of this decline is nearly identical across groups, indicating that differences in weekend provisioning behavior are not attributed to underlying weekend compute needs.[34]

[33] For each VM-day, load is given by multiplying the average CPU utilization by the VM's core-hours. The firm's total daily compute load is then determined by summing these VM-day loads across all active VMs.

[34] We interpret load as primarily driven by demand. One potential concern with this interpretation is that not all compute load is demand-driven—firms can have control over the timing of some compute needs, such as

Figure 5: Shutdown Probabilities of Permanently Idle VMs by Productivity Level



Notes: This figure displays the probability that a VM that is idle for multiple days at the end of its life is shut down after each given number of days, conditional on being idle for at least that many days. The red line represents low-productivity firms, while the blue line represents high-productivity firms. Productivity levels are estimated from 2022 data, while probabilities are estimated from 2023 data. The crossbars are 95% confidence intervals, with standard errors clustered by firm. Only VMs that last longer than one day and that end before the end of our sample period are included.

6.2 Attentiveness to Idle Resources

As discussed earlier, idle resources are a common phenomenon in the cloud that can occur when firms deploy VMs but fail to shut them off after use. In this section, we examine how quickly low- and high-productivity firms detect and terminate idle resources.

To test this, we first identify all VMs that are idle for at least one day at the end of their lifespan. We then calculate, separately for low- and high-productivity firms, the probability that a VM that has remained idle for a given number of days is shut down the following day. Put another way, we compute the hazard rate of shutting down an idle VM.[35] If more efficient firms are better at identifying idle resources, then they should shut down idle VMs faster, resulting in higher hazard rates. Since this analysis conditions on VMs being idle, it abstracts away from factors that lead to idle VMs in the first place; therefore, it speaks more directly to differences in firms' monitoring capabilities.

Figure 5 shows that high-productivity firms are significantly more likely to shut down idle VMs promptly. They exhibit a 37.0% probability of shutting down a VM on the day it becomes idle, compared to only 18.0% for low-productivity firms. Even if a VM is not shut down on the day it becomes idle, high-productivity firms are more likely to shut it

those for product development. To address this concern, we repeat the analysis focusing on software firms whose compute loads are more likely to be driven by customer demand and find similar results.

[35] While the base rate of idle resources is higher for less productive firms by construction, conditioning on a resource being idle eliminates this mechanical correlation from our analysis.

down on any subsequent day within the next week, if it gets to that day. These results suggest that more productive firms are better at detecting idle VMs, regardless of potential differences in the baseline likelihood of idle VMs.

6.3 Usage of a Variety of VM Types

Firms have access to various types of VMs, many of which are optimized for specific workloads. While the quantity of provisioned resources can be automated with tools like autoscaling, selecting the appropriate VM type typically requires active, manual decisions by cloud users. As such, all else equal, firms with employees who are more knowledgeable about VM types would be expected to deploy a broader range of VM types.

With this idea in mind, we investigate VM selection behavior by testing whether high-productivity firms tend to use a wider array of VM series. To do so, we estimate the following regression at the firm level:

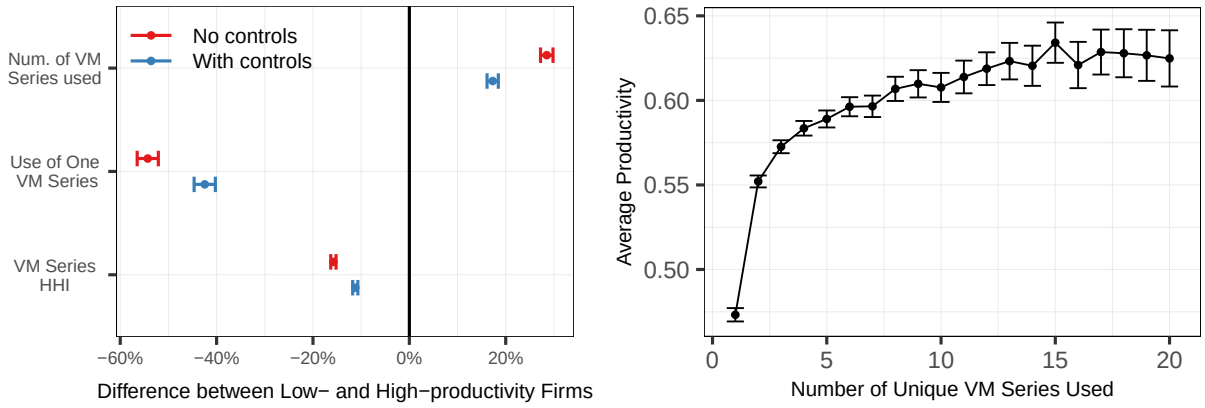
$$y_i = \beta D_i^H + Z_i'\gamma + \varepsilon_i. \tag{3}$$

Here, y_i represents a measure of dispersion in firms' usage of different VM series, including the Herfindahl–Hirschman Index (HHI; the sum of firms' squared usage shares), an indicator for whether the firm only uses one VM series, and the number of VM series the firm uses. The coefficient of interest, β, multiplies a binary variable D_i^H that indicates whether the firm is high-productivity. Finally, Z_i includes firm-level controls: firm size quartiles, industry, region, and cohort quarter fixed effects.

Figure 6(a) displays the results. We consistently find that high-productivity firms tend to use a wider array of VM series, both unconditionally and after controlling for firm-level covariates. For example, they exhibit a VM series HHI that is 0.067 lower than low-productivity firms—more than 11% of the mean HHI across firms. They also use a greater number of VM series and are significantly less likely to put all their usage on one VM series. These results are also reflected in the raw data: Figure 6(b) demonstrates a strongly monotonic and increasing relationship between the number of VM series used and compute productivity.

In summary, this section documents significant differences in behavior among firms. Some firms use resources more efficiently in both steady-state and under changing demand, more effectively monitor and shut down idle resources, and deploy a broader array of VMs. These differences point to inherent factors that are not tied to a specific external mechanism: some firms appear better at navigating internal frictions to reduce their costs (Leibenstein, 1966; Perelman, 2011). It is natural to expect that these factors evolve

Figure 6: Propensity to Use Different VM Series by Compute Productivity Level



(a) Regression Results **(b)** Productivity by Number of VM Series

Notes: Panel (a) displays the estimate of coefficient β from the regression shown in Equation (3), normalized by the mean of the dependent variables. The "No controls" red point includes the raw difference between the groups, while the "With controls" blue point controls for cloud adoption quarter fixed effects, industry (2-digit SIC code) fixed effects, region fixed effects, and firm size quartile fixed effects. The crossbars are 95% confidence intervals using standard errors clustered at the firm level. Table OA-6 reports the exact coefficient estimates corresponding to the visual summaries in Panel (a). Panel (b) reports the mean and 95th percentile of productivity across firms, grouped by the number of unique VM series they use. For both panels, productivity is taken from the 2022 sample, and the number and variety of VM series are estimated using the 2023 sample.

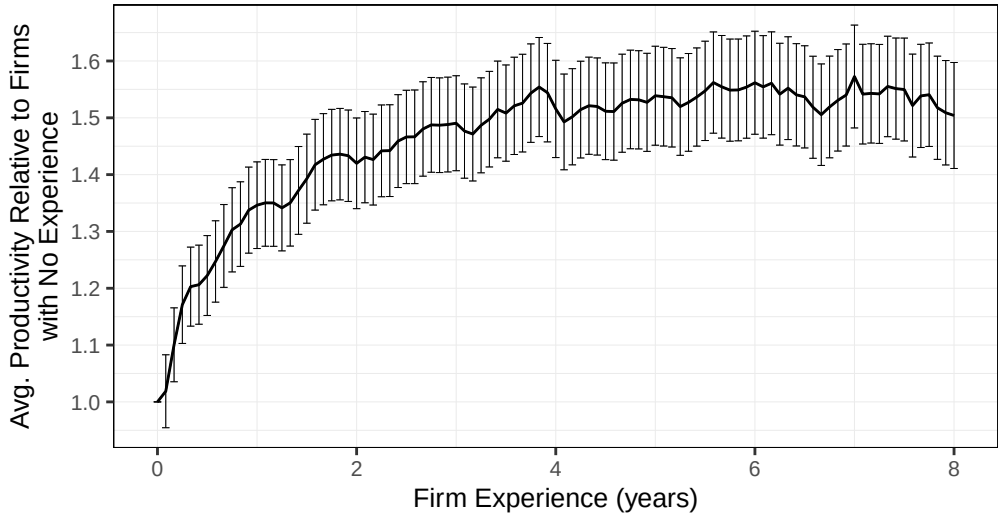
over time in the context of new technologies, as firms can monitor and improve their performance. We next analyze the extent to which this is the case.

7 Learning: How Compute Productivity Changes With Experience

In this section, we study learning: whether firms increase their compute productivity with experience and, if so, at what rate. It is natural to study learning in our setting because cloud computing is a relatively new technology, and firms typically require time to fully utilize and benefit from new technologies (Arrow, 1962; Helpman and Trajtenberg, 1998; Forman and Goldfarb, 2006). We aim to provide evidence on how firms learn to use a general-purpose technology that may require adjustments to use efficiently.

We present three sets of results. First, we document that firms improve their compute productivity over time—there is a strong relationship between a firm's productivity and its experience with the cloud. However, the learning process is slow, especially compared to existing evidence in the learning-by-doing literature, as it takes approximately four years for firms to reach steady state. Second, productivity dispersion is very high in the early stages of cloud adoption but narrows substantially over time as firms learn at different rates. Finally, nearly all productivity improvements result from the learning of individual units in the firm, suggesting limited within-firm knowledge transfer.

Figure 7: Compute Productivity vs. Firm Experience in July-September 2022



Notes: This figure shows the average productivity level as a function of firm experience, measured in years since the firm first began using the cloud. The productivity level is normalized relative to average productivity in the initial month (month 0). The crossbars indicate the 95% confidence intervals. The analysis is based on data from July-September 2022. The details of the estimation are provided in Appendix D.4.

7.1 Learning at the Firm Level

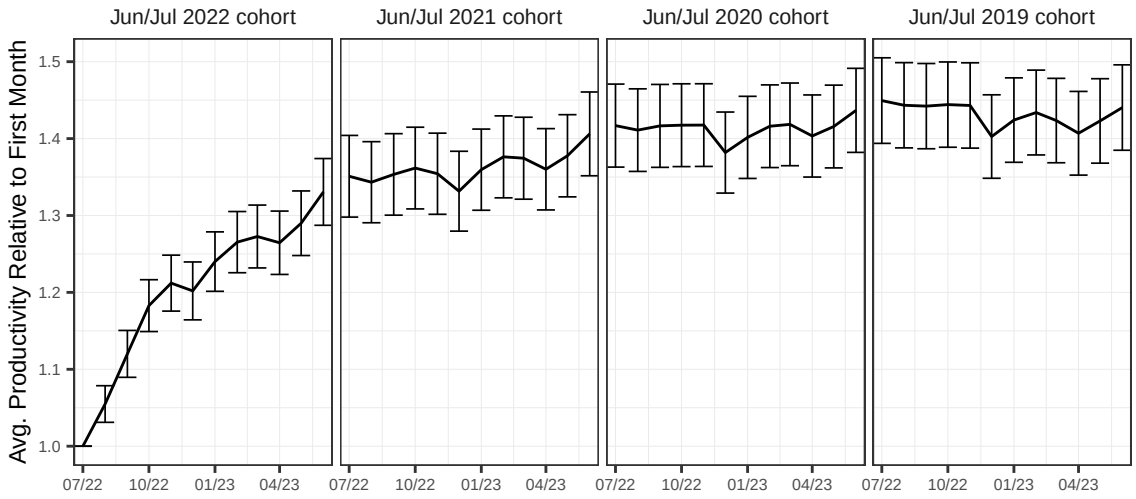
We study learning through two complementary approaches: (i) a cross-sectional analysis, which examines how compute productivity varies with firms' experience at a given point in time, and (ii) a cohort analysis, which tracks compute productivity over time within a group of firms.

At a given point in time, more experienced firms are substantially more efficient than less experienced firms. Figure 7 plots the normalized average compute productivity across firms from July to September 2022 against the duration each firm has been using the cloud as of that quarter.[36] The productivity gap is stark: firms with one year of experience are 34.6% more productive than new adopters. Experience is still positively correlated with productivity after the first year, but the relationship is weaker. Four-year-old firms are 51.6% more efficient than brand-new firms, with minimal efficiency gains thereafter.

While suggestive, the cross-sectional relationship between experience and compute productivity alone does not establish the existence of learning. Two potential forms of selection bias warrant consideration. First, early adopters of cloud technology may have done so because they anticipated higher productivity. Second, there might be survivorship bias if less productive firms tend to exit, which increases the average productivity of surviving firms over time independent of any actual learning effects.

[36]We focus on a three-month period to balance the stability of our estimates against minimizing the amount of time-series variation incorporated into our cross-sectional analysis.

Figure 8: Changes in Firm Compute Productivity Over Time by Cohort



Notes: This figure displays monthly average productivity estimates for four different cohorts of firms with different levels of experience (0-1, 1-2, 2-3, and 3-4 years) during the period between July 2022 and June 2023. The average productivity for the June-July 2022 cohort in July 2022 is normalized to 1. To be included in the analysis, a firm must have had nonzero usage every month from July 2022 to June 2023. Error bars indicate the 95% confidence intervals, with standard errors clustered by firm. The details of the estimation can be found in Appendix D.4.

To address both forms of selection, we focus on the productivity growth of individual firm cohorts over time, conditional on survival to the end of our sample. Figure 8 plots the compute productivity over the second half of 2022 and the first half of 2023 for four cohorts of firms: those that adopted the cloud in June-July 2022, 2021, 2020, and 2019. We find that firms' productivity dynamics over time exhibit similar patterns to the cross-sectional pattern discussed above.[37] Firms that started in the middle of 2022 experienced a productivity increase of 33.1% during their first year. Their productivity at the end of their first year was similar to the productivity of the 2021 cohort at the start of their second year, suggesting that productivity dynamics are similar across cohorts.

Over the subsequent years, the pace of learning slows substantially. Firms increase their compute productivity by 4.1% in their second year and by just 1.4% in their third year of cloud usage. Learning eventually plateaus, as evidenced by the 2019 cohort, which experiences no productivity gains in its fourth year. The relatively flat productivity levels of older cohorts also suggest that learning is not driven by aggregate cloud-specific trends; if this were the case, older cohorts would be becoming more productive, too.

Having presented visual evidence of learning, we next estimate a standard power-law specification to allow direct comparison between our findings and prior estimates from

[37]Figure 8 still includes some cross-sectional variation across different cohorts. We follow this approach because we do not have continuous productivity measures over four years due to the sparsity of our data. However, we can characterize long-term learning for the cohort that adopted the cloud in 2019 after four years, using only within-firm variation over time. This analysis still reveals clear evidence of learning among this cohort of firms in the long run. See Appendix E.9 for this result.

Table 5: Estimates of Learning Rate from Power Law Specification

	(1)	(2)	(3)	(4)	(5)
	Panel A: Cross-sectional analysis				
Learning rate (β)	0.162	0.162	0.134	0.140	0.100
	(0.004)	(0.004)	(0.003)	(0.003)	(0.003)
	Panel B: Cohort analysis				
Learning rate (β)	0.169	0.169	0.146	0.151	0.137
	(0.011)	(0.011)	(0.009)	(0.010)	(0.008)
Day of week + holiday FEs		X	X	X	X
VM family FEs			X		
VM family x data center region FEs				X	
VM series x data center x OS FEs					X

Notes: This table displays the slope coefficient from a regression of log productivity on log experience (in months) given in Equation (4). In panel A, we include all firms in July-September 2022, while in panel B, we include only firms that started using the cloud in June-July 2022, 2021, 2020, or 2019 and include data from July 2022-June 2023. To be included in Panel B, a firm must have had nonzero usage every month from July 2022 to June 2023. Each column includes a different set of controls when estimating firm-month level productivity as detailed in Appendix D.2. Clustered standard errors by firm in parentheses.

the learning-by-doing literature (Thompson, 2012; Levitt et al., 2013). In this analysis, productivity is modeled as a function of experience according to the following form:

$$\log(\omega_{im}) = \alpha + \beta \log(E_{im}) + \varepsilon_{im}, \tag{4}$$

where ω_{im} is firm i's productivity in month m, and E_{im} is its experience, defined as the length of time since the firm began using the cloud. The parameter β captures the rate at which productivity increases with experience, with larger values indicating faster learning.

Table 5 displays the results. Column (1) reports estimates using our baseline productivity measure as the dependent variable, while the remaining columns use the productivity estimates obtained under various controls as described in Appendix D.4. Each panel reflects a distinct source of variation; Panel A uses the cross-sectional variation reported in Figure 7, whereas Panel B uses the cohort-based variation shown in Figure 8.

Our baseline estimates in Panel B from the cohort analysis imply that when a firm's experience doubles, its compute productivity increases on average by $2^{0.169} - 1 = 12.4\%$. While controlling for time variables does not affect these results, controlling for VM characteristics slightly reduces the learning rate, which is consistent with the interpretation that part of the efficiency increase reflects firms becoming better at selecting VMs. These results are similar to the cross-sectional analysis reported in Panel A. Finally, as a robustness

check, we redefine experience as cumulative cloud usage instead of time—an alternative measure used in the learning-by-doing literature—and find similar results.[38]

Overall, our findings indicate that firms learn to use cloud computing more slowly than in most other contexts in the learning-by-doing literature. For example, Levitt et al. (2013) show that a car manufacturer's productivity in producing a new model plateaued after eight weeks; Kellogg (2011) finds that the cost of drilling oil wells flattened out after 20 weeks; and Thompson (2012) reviews evidence showing that the productivity of new shipyards in World War II converged within two years. Another commonly reported metric in the literature is the progress ratio, defined as the decrease in cost (or, in our case, inverse productivity) when experience doubles, corresponding to $2^{-\beta}$ in the power law model. Our estimates imply a progress ratio of 88.9%, which indicates slower learning relative to the literature.[39] One concern with these direct comparisons to prior studies is that they typically study cost reduction in firms' primary production input. In Figure OA-6, we find similar learning rates when focusing on the software industry, where computing is a major cost component.

We attribute the slower pace of learning in our setting relative to the literature to the nature of the technology learned by the firm. Cloud computing can be considered a general-purpose technology that simultaneously affects multiple parts of the firm and often requires complementary investments to realize its full potential (Bresnahan and Trajtenberg, 1995; Brynjolfsson et al., 2024). This contrasts with most other learning settings where firms or employees need to learn a new manufacturing technique.[40] With this interpretation, our results suggest that firms require more time to implement organization-wide changes compared to production-process-related improvements.

7.2 How Learning Affects Compute Productivity Dispersion

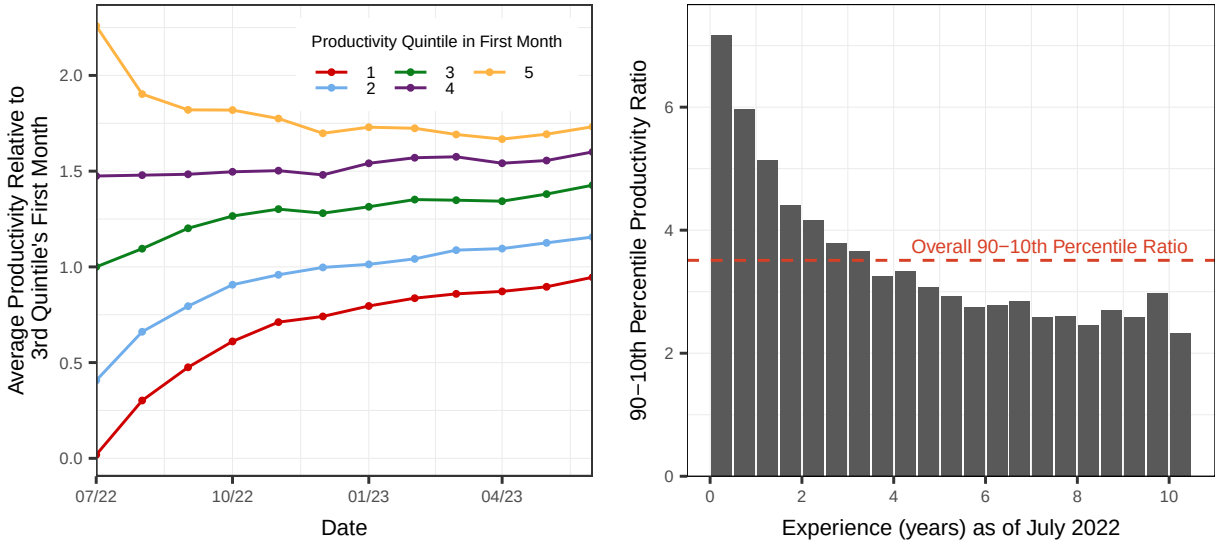
While experience raises average compute productivity, it is also important to consider its impact on the entire distribution. Figure 9 analyzes this question in two ways. Panel (a) reports the evolution of average productivity across initial productivity quintiles over the

[38] For the analogous regression to Column (1), we estimate $\beta = 0.149$ (SE = 0.009). We focus on calendar time for our main analyses as there is substantial heterogeneity in firms' compute demand, making a cross-sectional analysis of productivity improvements as a function of cumulative usage difficult. In the cohort analysis, we can control for this heterogeneity by adding firm fixed effects.

[39] For example, Levitt et al. (2013) finds a progress ratio of 81.2%, Benkard (2000) estimates a range up to 81.8%, and the literature reviewed by Thompson (2012) is concentrated between 70-80%.

[40] A notable exception is Doraszelski et al. (2018), who study the evolution of firm strategies after a change in the market structure. Similar to us, they find a learning duration of three-and-a-half to four years. Another useful comparison is the literature studying learning at the individual worker level (Jovanovic and Nyarko, 1995; Haggag et al., 2017), which finds an even shorter learning period, suggesting that as the size of the learning unit increases, the duration of learning also increases.

Figure 9: How Learning Affects Compute Productivity Dispersion



(a) Learning by Initial Productivity Level **(b)** Dispersion by Cloud Experience

Notes: Panel (a) displays monthly average productivity estimates for five equally sized groups of firms in the June-July 2022 cohort during the period between July 2022 and June 2023. The groups are divided based on their productivity in July 2022. The average productivity for the third initial productivity quintile of the June-July 2022 cohort in July 2022 is normalized to 1. To be included in the analysis, a firm must have had nonzero usage every month from July 2022 to June 2023. Panel (b) displays the ratio between the 90th percentile productivity and the 10th percentile productivity over July 2022-June 2023 within cloud experience bins as of July 2022. The dashed line is the value in the sample as a whole, 3.51. The details of the estimation are provided in Appendix D.4.

first year of usage, whereas Panel (b) reports the 90th-10th percentile productivity ratio by firms' experience levels.

Panel (a) shows that learning is driven mainly by the initially low-productivity firms, which exhibit faster learning relative to the top. On average, firms in the bottom three quintiles increase their productivity by 116.3% in the first year. In contrast, firms in the top two quintiles exhibit relatively stable productivity trajectories (aside from an initial drop among the top quintile, which could be attributed to mean reversion after a favorable initial draw). Despite the significant improvement among initially lower-productivity firms, the productivity ranking is preserved by the end of the period, consistent with persistent productivity differences.

Panel (b) directly analyzes cross-sectional dispersion by calculating the 90-10 percentile compute productivity ratio across firms' experience levels. Productivity dispersion is significantly higher in the early stages of technology adoption and narrows sharply over time: it stands at 7.21 for firms with no experience and declines steadily as they gain experience. Indeed, among firms with four or more years of experience, the 90-10 productivity ratio falls to 2.89, much closer to the US manufacturing average of 2.45 (Syverson, 2004b).

These results suggest that learning is an important determinant of dispersion patterns

in the context of new technologies. In particular, the dispersion in compute productivity can be partly attributed to cloud computing being a new technology with two contributing factors: there is heterogeneity in both the timing of adoption (which creates cross-sectional dispersion across experience levels) and the rate at which firms improve their productivity (which reduces the initially high within-cohort dispersion over time). As the technology becomes mature, we would expect both forms of dispersion to decrease over time.

7.3 Decomposing Firm-Level Learning Across Units

While our results so far establish that firms improve their compute productivity over time, they do not reveal the mechanisms behind these improvements. In the remainder of this section, we investigate how firms learn by decomposing their productivity growth.

We begin by studying whether compute productivity improvements happen primarily across or within units. On one hand, learning might be driven by firms devoting more resources to more productive units (across-unit); on the other, individual units themselves may become more productive (within-unit). To quantify these mechanisms, we decompose firm-level monthly productivity growth using the method from Foster et al. (2001). Firm i has a set of units K_{im} in month m; each unit $k \in K_{im}$ has monthly productivity ω_{ikm}. We can express firm i's productivity in month m as the core-hour weighted average productivity of its units; that is, letting s_{ikm} be the core-hour share of unit k in month m,

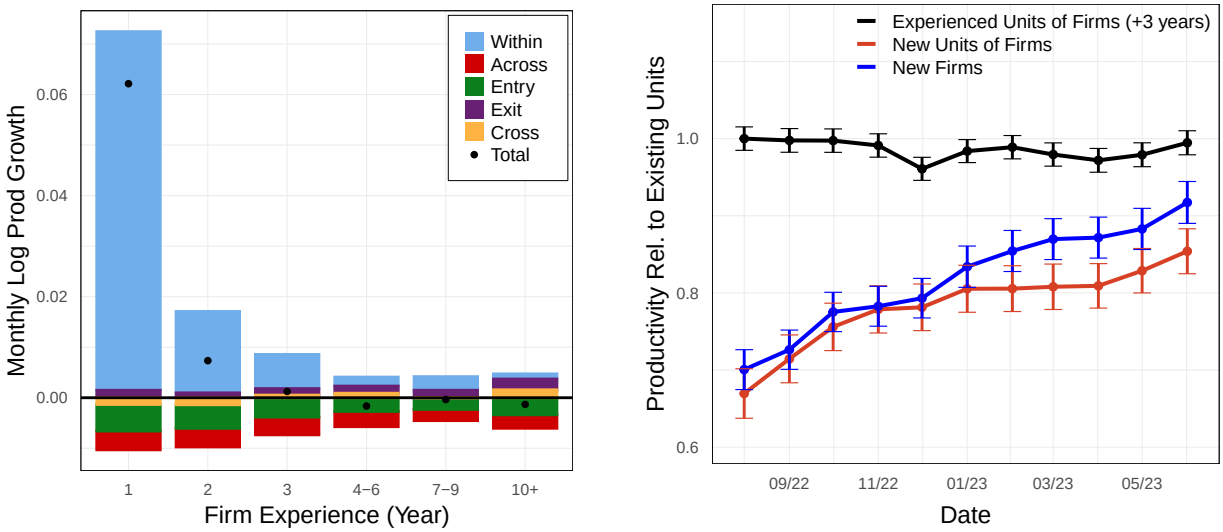
$$\omega_{im} = \sum_{k \in K_{im}} s_{ikm} \omega_{ikm}. \tag{5}$$

We are interested in decomposing $\Delta\omega_{im} = \omega_{im} - \omega_{i(m-1)}$, the change in firm i's compute productivity from month $m-1$ to month m. For ease of notation, we set $m = 1$. Let $S_{i1} = K_{i0} \cap K_{i1}$ be the set of units in firm i that used the cloud in both month 0 and month 1; $X_{i1} = K_{i0} \setminus K_{i1}$ the set of units that stopped using the cloud in month 1; and $E_{i1} = K_{i1} \setminus K_{i0}$ the set of units that started using the cloud in month 1. We can decompose $\Delta\omega_{i1}$ as follows:

$$\begin{aligned} \Delta\omega_{i1} = {} & \overbrace{\sum_{k \in S_{i1}} s_{ik0}(\omega_{ik1} - \omega_{ik0})}^{=Within} + \overbrace{\sum_{k \in S_{i1}} (s_{ik1} - s_{ik0})(\omega_{ik0} - \omega_{i0})}^{=Across} + \overbrace{\sum_{k \in S_{i1}} (s_{ik1} - s_{ik0})(\omega_{ik1} - \omega_{ik0})}^{=Cross} \\ & + \underbrace{\sum_{k \in E_{i1}} s_{ik1}(\omega_{ik1} - \omega_{i0})}_{=Entry} + \underbrace{\sum_{k \in X_{i1}} s_{ik0}(\omega_{i0} - \omega_{ik0})}_{=Exit}. \end{aligned} \tag{6}$$

The first term, *Within*, reflects the productivity growth coming from within-unit learning. The second term, *Across*, captures the productivity growth from the reallocation of com-

Figure 10: Decomposing Firm-Level Learning Within and Across Units



(a) Within-firm Learning Decomposition **(b)** New vs. Existing Unit Productivity

Notes: Panel (a) presents the decomposition of monthly log productivity growth by firms' years of experience with cloud computing, displaying the five components of Equation (6): within-unit, across-unit, entry, exit, and cross. The x-axis represents the firm's cloud experience in years, while the y-axis shows the monthly log productivity growth. The black dots indicate the average month-to-month productivity growth at the firm-level, whereas each bar represents a component of the decomposition. Panel (b) plots the average productivity of the units that joined in June-July 2022 against the average productivity of the older units in the same firm. The productivity of existing units in July 2022 is normalized to 1. The details of the estimation procedure are provided in Appendix D.5.

pute resources across units and is positive when units that grow are more productive than the firm's average productivity. The third term, *Cross*, represents the correlation between unit-level productivity growth and the growth of the unit's share. The fourth term, *Entry*, represents the contribution of units that are new to the cloud and would be positive if their initial productivity is higher than the firm. Finally, *Exit* reflects the contribution of units that stop using the cloud and would be positive if they are less productive than the firm.

We apply this decomposition to each firm separately and then average the components across firms. Figure 10(a) plots these averages by firms' years of experience with cloud computing (colored bars) together with the overall average monthly productivity growth (black dots). The results suggest that the observed flattening out of compute productivity over time masks substantial within-firm productivity dynamics. The *Within* components are large and positive during the first three years after adoption and remain positive yet small thereafter, indicating that (i) nearly all learning comes from units learning to be more productive and (ii) even if overall firm productivity growth plateaus after three years, individual units continue learning.

Other patterns in the data further confirm the role of within-unit learning. In Figure 10(b), we plot the compute productivity of units that newly adopt the cloud in the firm

(red) relative to the firm's existing units with at least three years of experience (black). New units start out with productivity 34.3% lower than experienced units in the same firm, yet they close about two-thirds of this gap within their first year. This pattern is similar to the overall learning pace of new firms, as shown by the blue line, suggesting that new units in experienced firms do not learn differently from new firms. Thus, our analysis provides no evidence for within-firm knowledge transfer and suggests that the changes required for efficient computing use occur at the unit level. These findings contribute to the growing literature emphasizing the role of within-firm factors in explaining firm performance (Hortaçsu and Syverson, 2007; Atalay et al., 2014; Hortaçsu et al., 2024).

7.4 Decomposing Unit-Level Learning Across VM Types

We further decompose learning within units to analyze how they become more productive. This time, we apply the decomposition in Equation (6) to within-unit compute productivity growth, breaking it down into VM-type components using the estimates of units' productivity in using different VM types from Section 4.[41] This analysis reveals how changes in a unit's productivity across different VM types contribute to the unit's overall productivity growth.

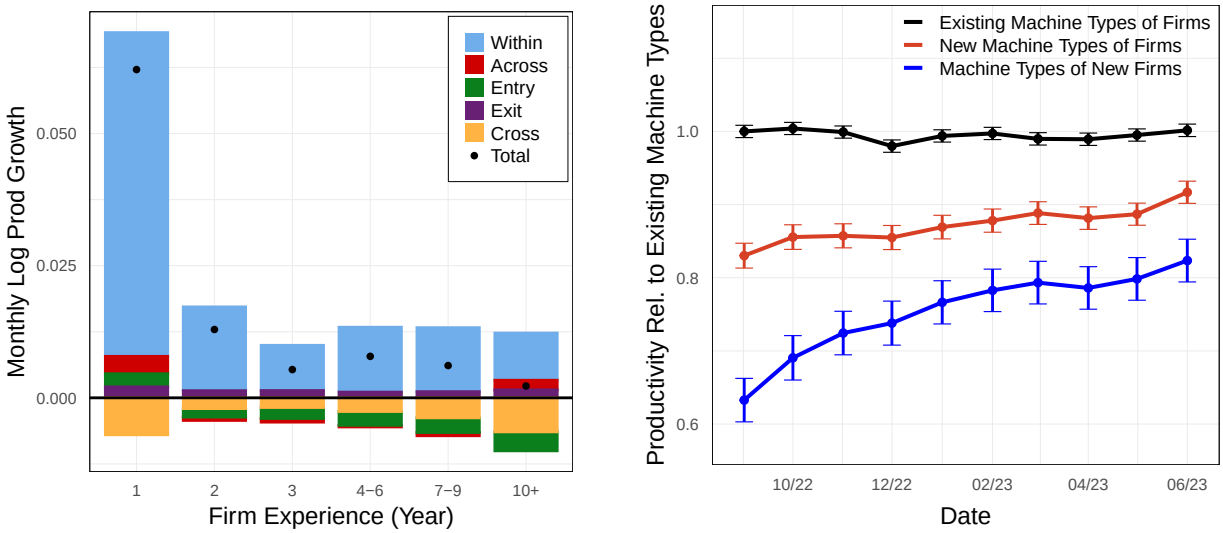
Results are reported in Figure 11(a). Units are broken out by the firm's cloud experience as before, and the black dots show the total within-unit component, which corresponds to the *Within* term of Figure 10(a). The results reveal that units' machine choices are not static; even more experienced firms try new machines and improve their productivity with the machine types they adopt. The *Entry* term is negative, reflecting the fact that units are less productive when they try machines for the first time. Units discard the less productive machines (the *Exit* term is positive) and get more productive with the machines they retain (the *Within* term is positive).

Figure 11(b) provides deeper insight into the *Within* component by comparing the average productivity level of machine types with which units have at least three years of experience (black) to that of machine types they newly adopt (red). Units are 16.5% less productive with newly adopted machines compared to their existing ones. This productivity gap is notably smaller than the 34.3% difference observed between the productivity of new and existing units in Figure 10(b) above, as well as the relative productivity of new firms (blue), indicating that experienced units achieve higher productivity with newly adopted machine types compared to new units and firms.

Taken together, our analysis suggests that although firms exhibit little knowledge trans-

[41] In this analysis, a VM type corresponds to a combination of VM series, data center, and operating system, which represent the primary VM dimensions along which learning can occur.

Figure 11: Decomposing Unit-Level Learning Within and Across VM Types



(a) Within-unit Learning Decomposition **(b)** New vs. Existing VM Productivity

Notes: Panel (a) presents the decomposition of monthly log unit-level productivity growth, displaying the five components of equation (6): within-type, across-type, entry, exit, and cross. The x-axis represents the firm's cloud experience in years, while the y-axis shows the monthly log productivity growth of the unit. The black dots indicate the average month-to-month productivity growth at the unit level, whereas each bar represents a component of the decomposition. Panel (b) plots the average productivity of the machine types that firms start using in August-September 2022 against the average productivity of the older machines in the same firm. The productivity of existing VMs in September 2022 is normalized to 1. The details of the estimation are provided in Appendix D.5.

fer across units, units do transfer some knowledge across machine types, while still demonstrating some learning at the machine level. Therefore, compute productivity growth at the unit level is a compositional effect of productivity changes from adopting new machines, learning to use the new machines efficiently, and discontinuing the less productive ones. These results suggest a potential role of experimentation as a driver of long-run changes in compute productivity.

8 Aggregate Impacts on Electricity and Compute Resources

This section analyzes the aggregate impacts of computing inefficiencies through a simple quantification exercise. Specifically, we estimate potential savings in the use of compute hardware and electricity if all firms were to reach a benchmark productivity level. Since electricity is the main input in computing, our analysis captures not only the private costs incurred by firms but also the social costs resulting from increased electricity consumption. We note that this analysis should be viewed as a quantitative exercise to illustrate the importance of inefficiencies in computing rather than a full modeling of the cloud market, as we do not consider several important factors, such as providers' capacity constraints,

potential price adjustments, or firms' behavioral responses to productivity improvements.

In our method, as detailed in Appendix D.7, we quantify the potential savings in core-hours and electricity consumption if all firms currently below the benchmark productivity level $\bar{\omega}_m^c$ were to reach this benchmark. Formally, the counterfactual productivity of firm i in month m is given by:

$$\omega_{im}^c = \bar{\omega}_m^c \cdot \mathbf{1}(\omega_{im} < \bar{\omega}_m^c) + \omega_{im} \cdot \mathbf{1}(\omega_{im} \geq \bar{\omega}_m^c). \tag{7}$$

To estimate the total core-hour savings, we first estimate each firm's core-hour consumption under this benchmark, compute the difference from actual usage to obtain the saved core-hours at the firm level, and aggregate these savings across firms.

To estimate the counterfactual electricity usage, we first establish the relationship between electricity consumption and VM utilization. As described in Appendix D.6, we estimate this relationship using data from Google Cloud, which includes 5-minute power consumption of 57 power domains and CPU utilization for all VMs within these domains. Our estimation shows that idle VMs consume approximately 50% of their maximum power, and each percentage-point increase in VM utilization results in an additional 0.5 pp increase in power consumption. Based on this finding, we propose the following functional form for the relationship between power usage and VM utilization:

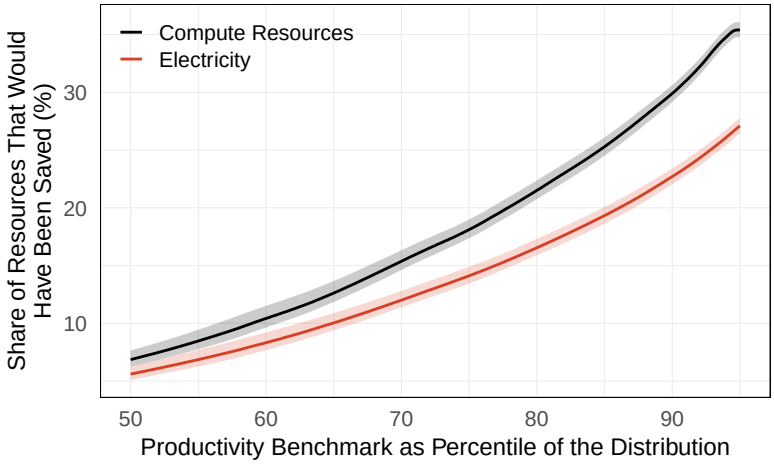
$$\mathbb{E}[p_{jt}] = (0.5 + 0.5u_{jt})k_j^{max}, \tag{8}$$

where k_j^{max} denotes the power consumption when VM j is utilized at 100%, $u_{jt} \in [0, 1]$ denotes the CPU utilization of VM j at time t and $p_{jt} \in [0, k_j^{max}]$ represents the power usage at u_{jt} utilization. Under this assumption and using the calculations detailed in Appendix D.7, we estimate the total electricity savings under counterfactual productivity.

Figure 12 summarizes our results, showing core-hour and electricity savings (in percent) on the y-axis and benchmark productivity levels (ranging from the 50th to the 100th percentile of firm productivity) on the x-axis. The findings reveal large potential savings from increased compute productivity: simply raising all firms to median productivity levels results in a 6.9% decline in compute resources and a 5.6% decrease in electricity use. These savings grow to 21.0% and 16.5%, respectively, at the 80th percentile of productivity, and ultimately reach 35.4% and 28.3% if all firms were to exhibit full efficiency.

The results also highlight a nonlinear relationship between productivity improvements and resource savings, with core-hour savings consistently exceeding electricity savings. This gap arises because idle VMs fully use core hours but consume only a portion of

Figure 12: Resource Impacts Under Counterfactual Compute Productivity Levels



Notes: This figure illustrates the changes in the shares of core-hours and electricity under a counterfactual increase in productivity. The x-axis represents the productivity benchmark as a percentile of its distribution, while the y-axis shows the percentage of resources that would be saved under different scenarios. The blue line indicates the percentage change in core-hours, and the red line indicates the percentage change in electricity consumption. Shaded regions surrounding each line represent the 95% confidence intervals obtained using a bootstrap procedure. The details of the estimation are provided in Appendix D.7.

the full-load electricity. Further, the gap between compute and electricity savings grows as benchmark productivity rises, reflecting different elasticities of these resources with respect to productivity. These findings emphasize the importance of linking productivity to resource consumption to quantify the costs of inefficiencies in firm production.[42]

9 Concluding Remarks

A robust finding from the productivity literature is the high degree of dispersion in various firm outcomes, including productivity, markups, and the share of labor (Syverson, 2011; De Loecker et al., 2020; Autor et al., 2020). In this paper, we show that similar differences exist in how productively firms use new technologies by analyzing evidence from cloud computing. We also study the dynamics of firm productivity, which is central to understanding how firms adopt and efficiently use new technologies.

Our study uses CPU utilization data from over 1 billion VMs employed by nearly 100,000 firms. We develop a novel compute productivity measure and show substantial dispersion in the efficiency of compute usage across and within firms. We identify certain differences in firm behavior that contribute to this dispersion, such as effective responses to demand fluctuations and careful management of idle resources. Finally, we document

[42]These estimates are also informative about the potential cost savings at the industry-level. Multiplying the core-hour savings by industry-level cost shares in Section 2.1 indicates total potential cost reductions of up to 6.7% in the software industry and 3.1-4.6% in the services industry. These cost savings are even higher for low-productivity firms and are likely to increase as computing becomes more widespread in the economy.

firm learning that, while significant in magnitude, takes longer than previously studied learning-by-doing settings and significantly influences dispersion in cloud productivity over time.

Our results have several implications for the broader productivity literature. First, although our productivity measure is specific to a single input and is derived from more granular data than is typical, our estimates of productivity dispersion and persistence corroborate the general magnitudes seen in other studies. Second, we find that productivity dispersion is dynamic in the context of a new technology—initial dispersion is extremely high, but shrinks over time due to heterogeneous learning rates across firms and saturation of the technology as a whole. Finally, our results demonstrate that internal knowledge diffusion plays a limited role among firms learning to use a new technology, reinforcing that efficient technology use often emerges only gradually, even after widespread adoption.

References

Abowd, J. M., F. Kramarz, and D. N. Margolis (1999). High Wage Workers and High Wage Firms. *Econometrica 67*(2), 251–333.

Ackerberg, D. A., K. Caves, and G. Frazer (2015). Identification Properties of Recent Production Function Estimators. *Econometrica 83*(6), 2411–2451.

Arrow, K. J. (1962). The Economic Implications of Learning by Doing. *The Review of Economic Studies 29*(3), 155–173.

Asker, J., A. Collard-Wexler, and J. De Loecker (2014). Dynamic Inputs and Resource (Mis) Allocation. *Journal of Political Economy 122*(5), 1013–1063.

Atalay, E., A. Hortaçsu, and C. Syverson (2014). Vertical Integration and Input Flows. *American Economic Review 104*(4), 1120–1148.

Autor, D., D. Dorn, L. F. Katz, C. Patterson, and J. Van Reenen (2020). The Fall of the Labor Share and the Rise of Superstar Firms. *The Quarterly Journal of Economics 135*(2), 645–709.

Babina, T., A. Fedyk, A. He, and J. Hodson (2024). Artificial Intelligence, Firm Growth, and Product Innovation. *Journal of Financial Economics 151*, 103745.

Backus, M. (2020). Why Is Productivity Correlated With Competition? *Econometrica 88*(6), 2415–2444.

Backus, M., T. Blake, J. Pettus, and S. Tadelis (2025). Communication, Learning, and Bargaining Breakdown: An Empirical Analysis. *Management Science*.

Baker, G. P. and T. N. Hubbard (2004). Contractibility and Asset Ownership: On-Board Computers and Governance in U.S. Trucking. *The Quarterly Journal of Economics 119*(4), 1443–1479.

Bartel, A., C. Ichniowski, and K. Shaw (2007). How Does Information Technology Affect Productivity? Plant-Level Comparisons of Product Innovation, Process Improvement, and Worker Skills. *The Quarterly Journal of Economics 122*(4), 1721–1758.

Bartelsman, E. J. and M. Doms (2000). Understanding Productivity: Lessons from Longitudinal Microdata. *Journal of Economic Literature 38*(3), 569–594.

Benkard, C. L. (2000). Learning and Forgetting: The Dynamics of Aircraft Production. *American Economic Review 90*(4), 1034–1054.

Black, S. E. and L. M. Lynch (2001). How to Compete: The Impact of Workplace Practices and Information Technology on Productivity. *Review of Economics and Statistics 83*(3), 434–445.

Bloom, N., S. Bond, and J. Van Reenen (2007). Uncertainty and Investment Dynamics. *Review of Economic Studies 74*(2), 391–415.

Bloom, N., E. Brynjolfsson, L. Foster, R. Jarmin, M. Patnaik, I. Saporta-Eksten, and J. Van Reenen (2019). What Drives Differences in Management Practices? *American Economic Review 109*(5), 1648–1683.

Bloom, N., B. Eifert, A. Mahajan, D. McKenzie, and J. Roberts (2013). Does Management

Matter? Evidence from India. *The Quarterly journal of Economics 128*(1), 1–51.

Bloom, N., L. Garicano, R. Sadun, and J. Van Reenen (2014). The Distinct Effects of Information Technology and Communication Technology on Firm Organization. *Management Science 60*(12), 2859–2885.

Bloom, N. and N. Pierri (2018). Cloud Computing Is Helping Smaller, Newer Firms Compete. *Harvard Business Review 94*(4).

Bloom, N., R. Sadun, and J. V. Reenen (2012). Americans Do IT Better: US Multinationals and the Productivity Miracle. *American Economic Review 102*(1), 167–201.

Bloom, N. and J. Van Reenen (2007). Measuring and Explaining Management Practices Across Firms and Countries. *The Quarterly Journal of Economics 122*(4), 1351–1408.

Braguinsky, S., A. Ohyama, T. Okazaki, and C. Syverson (2015). Acquisitions, Productivity, and Profitability: Evidence from the Japanese Cotton Spinning Industry. *American Economic Review 105*(7), 2086–2119.

Breitgand, D., Z. Dubitzky, A. Epstein, O. Feder, A. Glikson, I. Shapira, and G. Toffetti (2014). An Adaptive Utilization Accelerator for Virtualized Environments. In *2014 IEEE International Conference on Cloud Engineering*, pp. 165–174.

Bresnahan, T. and S. Greenstein (1996). Technical Progress and Co-Invention in Computing and in the Uses of Computers. *Brookings Papers on Economic Activity. Microeconomics 1996*, 1–83.

Bresnahan, T. F. (2010). General Purpose Technologies. In B. H. Hall and N. Rosenberg (Eds.), *Handbook of the Economics of Innovation*, Volume 2, pp. 761–791. Elsevier.

Bresnahan, T. F., E. Brynjolfsson, and L. M. Hitt (2002). Information Technology, Workplace Organization, and the Demand for Skilled Labor: Firm-Level Evidence. *The Quarterly Journal of Economics 117*(1), 339–376.

Bresnahan, T. F. and M. Trajtenberg (1995). General Purpose Technologies 'Engines of Growth'? *Journal of Econometrics 65*(1), 83–108.

Brynjolfsson, E. and L. Hitt (1995). Information Technology as a Factor of Production: The Role of Differences Among Firms. *Economics of Innovation and New Technology 3*(3-4), 183–200.

Brynjolfsson, E. and L. Hitt (1996). Paradox Lost? Firm-level Evidence on the Returns to Information Systems Spending. *Management Science 42*(4), 541–558.

Brynjolfsson, E. and L. M. Hitt (2000). Beyond Computation: Information Technology, Organizational Transformation and Business Performance. *Journal of Economic Perspectives 14*(4), 23–48.

Brynjolfsson, E. and L. M. Hitt (2003). Computing Productivity: Firm-Level Evidence. *Review of Economics and Statistics 85*(4), 793–808.

Brynjolfsson, E., W. Jin, and S. Steffen (2024). Do IT Capabilities Still Drive Productivity and Innovation in the Digital Age? *SSRN Working Paper, No. 4765508*.

Brynjolfsson, E., W. Jin, and X. Wang (2023). Information Technology, Firm Size, and

Industrial Concentration. *NBER Working Paper, No. 31065*.

Brynjolfsson, E. and K. McElheran (2016). The Rapid Adoption of Data-Driven Decision-Making. *American Economic Review 106*(5), 133–139.

Brynjolfsson, E., D. Rock, and C. Syverson (2021). The Productivity J-Curve: How Intangibles Complement General Purpose Technologies. *American Economic Journal: Macroeconomics 13*(1), 333–372.

Butters, R. A. (2020). Demand Volatility, Adjustment Costs, and Productivity: An Examination of Capacity Utilization in Hotels and Airlines. *American Economic Journal: Microeconomics 12*(4), 1–44.

Byrne, D., C. Corrado, and D. E. Sichel (2018). The Rise of Cloud Computing: Minding Your P's, Q's and K's. *NBER Working Paper, No. 25188*.

Caldarola, B. and L. Fontanelli (2024). Scaling Up to the Cloud: Cloud Technology Use and Growth Rates in Small and Large Firms. *Available at SSRN 5202110*.

CAST AI (2024). 2024 Kubernetes Cost Benchmark Report.

Collard-Wexler, A. and J. De Loecker (2015). Reallocation and technology: Evidence from the us steel industry. *American Economic Review 105*(1), 131–171.

Corrado, C. and J. Mattey (1997). Capacity Utilization. *Journal of Economic Perspectives 11*(1), 151–167.

Cortez, E., A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini (2017). Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 153–167.

Couchbase (2022). 2022 Couchbase Cloud Evolution Report.

Darr, E. D., L. Argote, and D. Epple (1995). The Acquisition, Transfer, and Depreciation of Knowledge in Service Organizations: Productivity in Franchises. *Management Science 41*(11), 1750–1762.

Davenport, T. H. (1998). Putting the Enterprise into the Enterprise System. *Harvard Business Review 76*(4), 121–131.

Davis, S. J., C. Grim, and J. Haltiwanger (2008). Productivity Dispersion and Input Prices: The Case of Electricity. *US Census Bureau Center for Economic Studies Paper, No. CES-WP-08-33*.

De Loecker, J., J. Eeckhout, and G. Unger (2020). The Rise of Market Power and the Macroeconomic Implications. *Quarterly Journal of Economics 135*(2), 561–644.

De Loecker, J. and C. Syverson (2021). An Industrial Organization Perspective on Productivity. In *Handbook of industrial organization*, Volume 4, pp. 141–223. Elsevier.

Demirer, M. (2025). Production Function Estimation with Factor-Augmenting Technology: An Application to Markups. *Working Paper*.

Demirer, M., D. J. J. Hernández, D. Li, and S. Peng (2024). Data, Privacy Laws and Firm

Production: Evidence from the GDPR. *NBER Working Paper, No. 32146*.

DeStefano, T., R. Kneller, and J. Timmis (2023). Cloud Computing and Firm Growth. *The Review of Economics and Statistics*, 1–47.

Doraszelski, U. and J. Jaumandreu (2018). Measuring the Bias of Technological Change. *Journal of Political Economy 126*(3), 1027–1084.

Doraszelski, U., G. Lewis, and A. Pakes (2018). Just Starting Out: Learning and Equilibrium in a New Market. *American Economic Review 108*(3), 565–615.

Dranove, D., C. Forman, A. Goldfarb, and S. Greenstein (2014). The Trillion Dollar Conundrum: Complementarities and Health Information Technology. *American Economic Journal: Economic Policy 6*(4), 239–270.

Fazli, A., A. Sayedi, and J. D. Shulman (2018). The Effects of Autoscaling in Cloud Computing. *Management Science 64*(11), 5149–5163.

Flexera (2023a). 2023 State of the Cloud Report.

Flexera (2023b). 2023 Tech Spend Pulse.

Forman, C. and A. Goldfarb (2006). ICT Diffusion to Businesses. In *Handbook of Economics and Information Systems*, Volume 1, pp. 1–52. Elsevier.

Foster, L., C. Grim, and J. Haltiwanger (2016). Reallocation in the Great Recession: Cleansing or Not? *Journal of Labor Economics 34*(S1), S293–S331.

Foster, L., J. Haltiwanger, and C. J. Krizan (2006). Market Selection, Reallocation, and Restructuring in the US Retail Trade Sector in the 1990s. *The Review of Economics and Statistics 88*(4), 748–758.

Foster, L., J. Haltiwanger, and C. Syverson (2008). Reallocation, Firm Turnover, and Efficiency: Selection on Productivity or Profitability? *American Economic Review 98*(1), 394–425.

Foster, L., J. C. Haltiwanger, and C. J. Krizan (2001). Aggregate Productivity Growth: Lessons from Microeconomic Evidence. In *New Developments in Productivity Analysis*, pp. 303–372. University of Chicago Press.

Fox, J. T. and V. Smeets (2011). Does Input Quality Drive Measured Differences in Firm Productivity? *International Economic Review 52*(4), 961–989.

Garicano, L. (2010). Policemen, Managers, Lawyers: New Results on Complementarities Between Organization and Information and Communication Technology. *International Journal of Industrial Organization 28*(4), 355–358.

Greenstein, S. (2020). Digital Infrastructure. In *Economic Analysis and Infrastructure Investment*, pp. 409–447. University of Chicago Press.

Greenstein, S. and T. P. Fang (2020). Where the Cloud Rests: The Location Strategies of Data Centers. *Harvard Business School Working Paper, No. 21-042*.

Haag, S. and A. Eckhardt (2017). Shadow IT. *Business & Information Systems Engineering 59*, 469–473.

Haggag, K., B. McManus, and G. Paci (2017). Learning by Driving: Productivity Improvements by New York City Taxi Drivers. *American Economic Journal: Applied Economics 9*(1), 70–95.

Helpman, E. and M. Trajtenberg (1998). *Diffusion of General Purpose Technologies*, Chapter 4. MIT Press, Cambridge, MA.

Hendel, I. and Y. Spiegel (2014). Small Steps for Workers, a Giant Leap for Productivity. *American Economic Journal: Applied Economics 6*(1), 73–90.

Hennessy, J. L. and D. A. Patterson (2012). *Computer Architecture: A Quantitative Approach, Fifth Edition*. Elsevier.

Hortaçsu, A., O. R. Natan, H. Parsley, T. Schwieg, and K. R. Williams (2024). Organizational Structure and Pricing: Evidence from a Large U.S. Airline. *The Quarterly Journal of Economics 139*(2), 1149–1199.

Hortaçsu, A. and C. Syverson (2007). Cementing Relationships: Vertical Integration, Foreclosure, Productivity, and Prices. *Journal of Political Economy 115*(2), 250–301.

Hsieh, C.-T. and P. J. Klenow (2009). Misallocation and Manufacturing TFP in China and India. *The Quarterly Journal of Economics 124*(4), 1403–1448.

Hubbard, T. N. (2003). Information, Decisions, and Productivity: On-Board Computers and Capacity Utilization in Trucking. *American Economic Review 93*(4), 1328–1353.

IDC (2025). International data corporation, worldwide cloud spending forecast: $432 billion and doubling by 2028. Technical Report PRUS52460024, IDC.

Jin, C., S. Peng, and P. Wang (2023). Sticky Consumers and Cloud Welfare. *SSRN Working Paper, No. 4573326*.

Jin, W. and J. J. Bai (2022). Cloud Adoption and Firm Performance: Evidence from Labor Demand. *SSRN Working Paper, No. 4082436*.

Jin, W. and K. McElheran (2024). Economies Before Scale: IT Strategy and Performance Dynamics of Young US Businesses. *Management Science*.

Jorgenson, D. W. (2001). Information Technology and the U.S. Economy. *American Economic Review 91*(1), 1–32.

Jovanovic, B. and Y. Nyarko (1995). A Bayesian Learning Model Fitted to a Variety of Empirical Learning Curves. *Brookings Papers on Economic Activity. Microeconomics 1995*, 247–305.

Kalyani, A., N. Bloom, M. Carvalho, T. Hassan, J. Lerner, and A. Tahoun (2025). The Diffusion of New Technologies. *The Quarterly Journal of Economics*.

Kansal, A., F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya (2010). Virtual Machine Power Metering and Provisioning. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, pp. 39–50.

Kellogg, R. (2011). Learning by Drilling: Interfirm Learning and Relationship Persistence in the Texas Oilpatch. *The Quarterly Journal of Economics 126*(4), 1961–2004.

Lashkari, D., A. Bauer, and J. Boussard (2024). Information Technology and Returns to Scale. *American Economic Review 114*(6), 1769–1815.

Leibenstein, H. (1966). Allocative Efficiency vs." X-efficiency". *The American Economic Review 56*(3), 392–415.

Levitt, S., J. List, and C. Syverson (2013). Toward an Understanding of Learning by Doing: Evidence from an Automobile Assembly Plant. *Journal of Political Economy 121*(4), 643–681.

Lu, Y., G. M. Phillips, and J. Yang (2024). The Impact of Cloud Computing and AI on Industry Dynamics and Firm Financing. *SSRN Working Paper, No. 4480570*.

Mason, K., M. Duggan, E. Barrett, J. Duggan, and E. Howley (2018). Predicting Host CPU Utilization in the Cloud Using Evolutionary Neural Networks. *Future Generation Computer Systems 86*, 162–173.

McElheran, K. (2014). Delegation in Multi-Establishment Firms: Evidence from IT Purchasing. *Journal of Economics and Management Strategy 23*(2), 225–258.

McElheran, K., J. F. Li, E. Brynjolfsson, Z. Kroff, E. Dinlersoz, L. S. Foster, and N. Zolas (2023). AI Adoption in America: Who, What, and Where. *NBER Working Paper, No. 31788*.

McElheran, K., M.-J. Yang, Z. Kroff, and E. Brynjolfsson (2025). The Rise of Industrial AI in America: Microfoundations of the Productivity J-Curve(s). *Center for Economic Studies, U.S. Census Bureau Working Paper, No. CES-25-27*.

Metcalfe, R. D., A. B. Sollaci, and C. Syverson (2023). Managers and Productivity in Retail. *NBER Working Paper, No. 31192*.

Miller, A. R. and C. E. Tucker (2011). Can Health Care Information Technology Save Babies? *Journal of Political Economy 119*(2), 289–324.

Murciano-Goroff, R., R. Zhuo, and S. Greenstein (2024). Navigating Software Vulnerabilities: Eighteen Years of Evidence from Medium and Large US Organizations. *NBER Working Paper, No. 32696*.

Nguyen, T. L. and A. Lebre (2017). Virtual Machine Boot Time Model. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, pp. 430–437. IEEE.

Orr, S. (2022). Within-Firm Productivity Dispersion: Estimates and Implications. *Journal of Political Economy 130*(11), 2771–2828.

Perelman, M. (2011). Retrospectives: X-efficiency. *Journal of Economic Perspectives 25*(4), 211–222.

Perez-Salazar, S., I. Menache, M. Singh, and A. Toriello (2022). Dynamic Resource Allocation in the Cloud with Near-Optimal Efficiency. *Operations Research 70*(4), 2517–2537.

Pindyck, R. S. (1986). Irreversible Investment, Capacity Choice, and the Value of the Firm. *NBER Working Paper, No. 1980*.

Raval, D. R. (2019). The Micro Elasticity of Substitution and Non-Neutral Technology. *The*

RAND Journal of Economics 50(1), 147–167.

Reiss, C., A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch (2012). Towards Understanding Heterogeneous Clouds at Scale: Google Trace Analysis. *Intel Science and Technology Center for Cloud Computing*.

Restuccia, D. and R. Rogerson (2017). The Causes and Costs of Misallocation. *Journal of Economic Perspectives 31*(3), 151–174.

Rivoire, S., P. Ranganathan, and C. Kozyrakis (2008). A Comparison of High-Level Full-System Power Models. *HotPower 8*(2), 32–39.

Shephard, R. W. (1953). *Cost and Production Functions*. Princeton University Press.

Stiroh, K. J. (2002). Information Technology and the US Productivity Revival: What Do the Industry Data Say? *American Economic Review 92*(5), 1559–1576.

Syverson, C. (2004a). Market Structure and Productivity: A Concrete Example. *Journal of Political Economy 112*(6), 1181–1222.

Syverson, C. (2004b). Product Substitutability and Productivity Dispersion. *Review of Economics and Statistics 86*(2), 534–550.

Syverson, C. (2011). What Determines Productivity? *Journal of Economic literature 49*(2), 326–365.

Tadelis, S., C. Hooton, U. Manjeer, D. Deisenroth, N. Wernerfelt, N. Dadson, and L. Greenbaum (2023). Learning, Sophistication, and the Returns to Advertising: Implications for Differences in Firm Performance. *NBER Working Paper, No. 31201*.

Tambe, P., L. Hitt, D. Rock, and E. Brynjolfsson (2020). Digital Capital and Superstar Firms. *NBER Working Paper, No. 28285*.

Tambe, P. and L. M. Hitt (2012). The Productivity of Information Technology Investments: New Evidence from IT Labor Data. *Information Systems Research 23*(3-part-1), 599–617.

Thompson, P. (2012). The Relationship Between Unit Cost and Cumulative Quantity and the Evidence for Organizational Learning-by-Doing. *Journal of Economic Perspectives 26*(3), 203–24.

Thornton, R. A. and P. Thompson (2001). Learning from Experience and Learning from Others: An Exploration of Learning and Spillovers in Wartime Shipbuilding. *American Economic Review 91*(5), 1350–1368.

Van Biesebroeck, J. (2003). Productivity Dynamics with Technology Choice: An Application to Automobile Assembly. *The Review of Economic Studies 70*(1), 167–198.

Williamson, O. E. (1966). Peak-load pricing and optimal capacity under indivisibility constraints. *American Economic Review 56*(4), 810–827.

Zolas, N., Z. Kroff, E. Brynjolfsson, K. McElheran, D. N. Beede, C. Buffington, N. Goldschlag, L. Foster, and E. Dinlersoz (2021). Advanced Technologies Adoption and Use by US Firms: Evidence from the Annual Business Survey. *NBER Working Paper No. 28290*.

Firm Productivity and Learning with Digital Technologies: Evidence from Cloud Computing

James Brand Mert Demirer
Connor Finucane Avner A. Kreps

Appendix - For Online Publication

Contents

A Additional Institutional Details . . . . . **OA - 3**
A.1 Details on Cloud Computing OA - 3
A.2 Details of VM Deployment OA - 5
A.3 Two Examples of VM Deployment OA - 7

B Data Appendix . . . . . **OA - 11**
B.1 CPU Utilization Data OA - 11
B.2 VM Data OA - 11
B.3 Firm and Unit Level Data OA - 13
B.4 Sampling Details OA - 13
B.5 Data Cleaning Procedure OA - 14
B.6 Publicly Available CPU Utilization Data OA - 14

C Additional Details on Productivity Measurement . . . . . **OA - 16**
C.1 Formal Exposition of Compute Productivity OA - 16
C.2 Microfoundation as Factor-Augmenting Productivity OA - 19

D Additional Details on Estimation . . . . . **OA - 19**
D.1 Details on Cloud and Electricity Spending OA - 19
D.2 Details of Productivity Estimation OA - 20
D.3 Details of Dispersion and Persistence Estimation OA - 24
D.4 Details of Learning Estimation OA - 26
D.5 Details of Learning Decomposition Analysis OA - 28
D.6 Details of CPU Utilization and Electricity Relationship Estimation OA - 29
D.7 Details of Counterfactual Resource Calculations OA - 32

E Robustness Checks . . . . . **OA - 36**
E.1 Robustness to Other Utilization Measures OA - 36
E.2 Robustness to Duration of Peak Utilization Measurement OA - 37
E.3 Robustness to Controlling for VM Characteristics OA - 37
E.4 Robustness to Compute Load Volatility Measures OA - 38
E.5 Robustness to Measurement of Downsizability OA - 39
E.6 Correlation Between Idleness and Overprovisioning Productivity OA - 40

E.7 External Validity: Relationship Between Productivity and Firm Exit OA - 40
E.8 External Validity: Dispersion in Publicly Available CPU Utilization Data OA - 41
E.9 Long-Term Learning OA - 44

F Additional Figures . . . . . OA - 46

G Additional Tables . . . . . OA - 53

H Robustness Results . . . . . OA - 56

A Additional Institutional Details

In this Appendix, we provide additional details about cloud computing that were omitted from the main text. This information was primarily compiled from the websites of major cloud computing providers and our conversations with industry professionals. This section aims to reflect the widely adopted practices in cloud computing that are relatively common across cloud providers. However, it may not apply universally to all providers.

A.1 Details on Cloud Computing

A.1.1 Types of Cloud Computing Products

Cloud computing products can be classified into three categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS differs fundamentally from PaaS and IaaS, as it encompasses a broader range of applications—including email services and productivity tools—and often targets end consumers rather than exclusively businesses. Therefore, we limit the discussion in this section to PaaS and IaaS.

PaaS products encompass a range of services that abstract away from the underlying infrastructure, allowing users to focus more on application development and deployment rather than managing physical or virtual hardware. Examples of these services include containers and serverless computing. Containers package applications with their dependencies, ensuring consistent and reliable performance across different environments, while container orchestration tools manage the deployment, scaling, and operation of these containers. Serverless computing enables developers to run code without provisioning or managing servers, as the cloud provider handles the infrastructure, scaling, and execution of code in response to events.

Despite the convenience of PaaS services, IaaS remains more common among firms, especially during our sample period, because it provides a high degree of flexibility and control, allowing users to tailor the underlying infrastructure to their specific needs. This level of control is important for complex, custom applications and enterprise environments that require precise configurations. Additionally, IaaS can accommodate a wide range of workloads, from legacy applications to modern microservices, making it more versatile than PaaS services.

A.1.2 Resource Availability in Cloud Computing

In cloud computing, firms typically request a quota that specifies the maximum amount of computing resources they may provision at any given time. Once this quota is established,

users can access the resources up to the quota whenever they need them. Except for a few specialized VMs, firms can adjust their quota at any time, at no additional costs, and quickly. Even though the quota cannot guarantee the immediate availability of resources, the high reliability of cloud services ensures that firms rarely face situations where their resource requests cannot be fulfilled.

Cloud computing providers ensure sufficient capacity is available by investing in data centers. While predicting the needs of individual firms can be challenging, cloud providers can more accurately forecast aggregate demand across all users by leveraging the law of large numbers. Therefore, they can predict compute demand more accurately than individual firms and invest accordingly to maintain sufficient infrastructure.

However, aggregate demand remains volatile, requiring cloud providers to balance supply and demand in real time. Cloud providers use a mechanism known as spot instances to manage this. Via spot instances, cloud providers rent out unused capacity at steep discounts, offering a cost-effective option for firms with workloads that are less sensitive to interruptions. However, these instances come with the caveat that they can be reclaimed at any time with short notice. Essentially, cloud providers invest in infrastructure to ensure they can meet peak demand and then monetize the temporary excess capacity due to demand volatility through the spot market.

A.1.3 Pricing in Cloud Computing

Cloud computing primarily employs an on-demand pricing model, allowing users to pay solely for the resources they actually provision. This model typically involves linear prices, meaning that costs scale directly with the amount of resource requested—whether it is computing power, storage, or bandwidth.

For VMs, providers implement a granular billing model based on per-minute—or in some cases, per-second—usage rather than billing in hourly increments. The billing system records the precise start and termination times of each VM and computes the total active duration in the time intervals. If the active period falls below a defined minimum unit, the usage is rounded up to that smallest increment. The applicable rate per time unit is determined by the VM's configuration details, including the number of virtual CPUs, the amount of allocated memory, storage type, and the geographic region in which the VM is deployed.

In addition to this on-demand structure, cloud providers offer more specialized pricing options tailored to different usage patterns. Reserved instances, for example, enable users to commit to a specified amount of resource usage over an extended period in exchange for discounted rates, benefiting workloads that are predictable and steady. Alternatively, as

described above, spot instances provide resources at significantly reduced rates but with the caveat that providers can reclaim those resources with minimal notice. Despite these varied pricing strategies, cloud costs can still be thought of as mainly variable costs from the firms' point of view, as the price paid (negotiated or not) is directly tied to the level of resource consumption.

A.2 Details of VM Deployment

This section provides details of deploying VMs separately for (i) manually through browser-based platforms and (ii) automatically through tools like autoscaling.

A.2.1 VM Creation and Selection

All cloud providers offer a browser-based platform with step-by-step instructions for generating a VM. These instructions typically apply for first-time VM creation and may not be as relevant for firms' day-to-day VM deployments. In this section, we detail the steps involved in selecting a VM. In the following subsection, we review the tools commonly used by firms for VM deployment and management.

Account Creation: Before deploying VMs, an account needs to be created and configured—this can be either an individual user account or a business account. This involves setting up the necessary permissions and roles to ensure that users or services have the appropriate access levels to manage and interact with the VMs.

Resource Allocation: During deployment, specific amounts of CPU, memory, and storage must be chosen for the VM.

Network Configuration: If the VM needs connectivity, a network connection step is needed. This involves assigning IP addresses, configuring network interfaces, and setting up necessary VLANs (Virtual Local Area Networks).

Storage Provisioning: VMs require storage for the operating system, applications, and data. This can involve allocating space on local disks or choosing a storage system linked to the VM.

Security Settings: Security configurations need to be applied during deployment. This includes setting up firewalls, defining access controls, and implementing any required encryption.

Region Choice: The user selects the geographic region of the data center for the VM. This choice can impact performance due to latency and varying redundancy policies.

This procedure applies when creating a VM for the first time. In practice, developers use VM images, which are templates that store information such as the OS, security settings, and pre-installed software, allowing users to create VMs without redefining these

parameters. Firms often maintain a library of standardized images for various purposes, facilitating the VM deployment process.

A.2.2 Automated Tools For VM Deployment

As we argue in Section 2.3, there are many available tools in cloud computing that help dynamically adjust compute capacity both horizontally and vertically. In this section, we list some of these tools and provide a brief description.[43]

Cloning a VM: Cloning technology allows users to create new VMs from the running state of an existing VM. The cloned VM is identical to the source VM and can be created quickly at any point in time. This method is suitable for large-scale applications and enables the creation of many VMs on a single host. All major cloud providers offer some form of cloning capability.

Auto Shutdown: In cloud computing, users have the capability to schedule or automatically shut down VMs to help manage costs and optimize resource usage. This functionality allows users to define specific times for VMs to stop and start, eliminating the need for constant monitoring.

Live Migration: Live migration allows users to move running VMs from one physical host to another without downtime. With this technology, applications can continue to operate during maintenance, load balancing, or hardware upgrades.

Automatic Redundancy and Fault Tolerance: Cloud providers offer built-in automatic redundancy and fault tolerance to ensure high availability and reliability of services. These features include distributing data and workloads across multiple servers, data centers, or geographic regions to prevent single points of failure. In case of hardware or software failures, automatic failover mechanisms can redirect traffic to healthy instances, minimizing downtime and maintaining service continuity.

Autoscaling: Autoscaling refers to the automatic adjustment of compute resources based on the current demand. This capability allows cloud environments to dynamically allocate or deallocate resources such as CPU, memory, and storage to applications or services as needed. When demand increases, autoscaling provisions additional instances to handle the load, ensuring that applications can maintain performance levels. Conversely, when demand decreases, it reduces the number of active instances, scaling down the resources in use. The process of autoscaling involves automatic monitoring of the performance metrics and resource utilization of applications. Firms can set autoscaling rules and policies based on their specific requirements, such as thresholds for CPU usage, memory consumption, or network traffic. For example, if CPU utilization exceeds a certain threshold, the autoscaling

[43] AWS—Cost Optimization; Azure—Deployment Optimizer; Google Cloud—Cost Management.

mechanism adds more instances to distribute the load. Similarly, if resource usage falls below a specified level, it scales back the instances. These rules can be configured through cloud provider dashboards or APIs.

Predictive Autoscaling: Rather than adjusting compute resources in response to current demand, predictive autoscaling attempts to adjust compute resources in advance of upcoming demand fluctuations. This is suitable for latency-sensitive workloads such as streaming platforms. All major cloud providers offer some form of predictive autoscaling, in which they attempt to predict users' workloads based on past workload patterns of both those users and other similar users. Users are also able to develop their own predictive models themselves.

Load Balancers: Load balancers distribute incoming network traffic across multiple VMs. By evenly distributing the load, load balancers ensure that no single server hits capacity. They achieve this by monitoring the health of instances and redirecting traffic away from failing or underperforming instances. Load balancers complement autoscaling by working together to optimize resource utilization and application performance. While autoscaling dynamically adjusts the number of active VMs based on demand, load balancers distribute the incoming traffic among these VMs.

Cost Monitoring Tools: Cloud providers offer a range of tools to help users monitor and manage their resource utilization and costs. For example, AWS CloudWatch provides monitoring for resources and applications across AWS environments. It includes visualization tools, automated alarms, and integration with other AWS services to help identify and manage idle resources. Similarly, Google Cloud Operations offers integrated monitoring, logging, and tracing for applications and systems on Google Cloud. These tools may also include the observability of metrics specific to detecting and managing idle resources.

A.3 Two Examples of VM Deployment

In this section, we provide two examples of deploying IT resources in cloud computing to help readers understand their use cases. While this section is quite technical, it may help inform readers about the day-to-day work of software developers in cloud environments.

A.3.1 "Create, Read, Update, Delete" Application

The first toy example we consider is the deployment of a simple CRUD (Create, Read, Update, Delete) application. This refers to a general class of applications that provide a user interface for the typical operations involved in persistent storage. A common example of this application is a blog or message board. In this example, "Create" corresponds to a user making a new post or comment, "Read" corresponds to the functionality of listing all

posts and comments corresponding to some filter, "Update" corresponds to editing posts or profile information, and "Delete" corresponds to the deletion of posts or comments. The basic components of the architecture of this application include a web server, which is accessible by users on the public internet, and a database server, which is typically only accessible by the web server itself. The web server's responsibilities include authenticating users, producing HTML that provides the information and features available to the user, and issuing control commands and queries to the database based on the user's request.

We describe specific deployment scenarios of this application type on the Microsoft Azure cloud. For the first example, we consider hosting this application only in a single region, with a fixed resource footprint, and with manual processes to deploy resources. To complete this deployment, we:

1. Create a Resource Group (RG), a logical group that will contain all of the resources for this deployment.

2. Create a Virtual Network (VNet) and create a subnet within this VNet.

3. Select and provision an appropriate VM to run the web application based on our application's requirements, budget, and account quota. Additionally, we must:

   (a) Select the proper region and operating system for the application.

   (b) Add the VM's network interface to the appropriate subnet within the VNet.

   (c) Create a Network Security Group (NSG) and add rules that restrict the incoming traffic to SSH/RDP and HTTPS traffic from internet IP Addresses.

   (d) Give the VM a static IP address. It is also possible to give it a human-readable alias using Azure DNS.

4. Create an Azure Database for PostgreSQL to serve as the persistent storage back-end for the application.

   (a) Similar to the web server VM, we must choose an appropriate virtual core count, virtual memory amount, storage size, storage scale rule, and storage performance tier based on our requirements and budget.

   (b) We will place the database in the same region as our web server.

   (c) We will disallow public IP access to the database and integrate it into the existing VNet and the same or new subnet.

At this point, our resources are established and we can install the application onto the web server and complete the connection between the application and the database to be able to service user requests using the persistent storage. We can monitor the health and utilization of the instances using Azure Monitor.

The above architecture leans towards using IaaS solutions and is not capable of scaling horizontally. It is a simple deployment method that can be hard to maintain, and the architecture will likely be insufficient to handle dynamic loads. Since we can only control the size of a single instance for the web server and database, respectively, it will be challenging to avoid being under- or over-provisioned, and we will incur downtime in the application if we need to scale instances up or down. A common way to handle this is by using IaaS offerings for the web server, such as a Virtual Machine Scale Set (VMSS, Azure's autoscaling solution) and Load Balancer (LB). The VMSS is a collection of identical VMs in a single region. With this deployment method, we can define conditions on the pool of VMs that will trigger custom scale-up and scale-down actions. These rules or conditions are defined by statistics on the time series of instance-level counters such as CPU, Network, and Disk Utilization. The LB can be configured with a front-end IP address to accept user traffic and then given the VMSS as a back-end pool to distribute requests over. This architecture allows us to scale horizontally instead of vertically, increasing our ability to handle dynamic loads, making us less likely to be under or over-provisioned at any given time, and decreasing our application's expected downtime.

A.3.2 Data Analytics and Machine Learning Application

Another common use of modern public cloud infrastructure is building pipelines for data analytics and machine learning use cases. Since many of these use cases can be implemented purely as PaaS services, we will focus on a sample architecture that favors IaaS resources for batch training of a custom machine learning model or a numerical simulation. In these settings, it is common to have a highly parallelizable workload that consists of iterating over a set of parameters or hyperparameters for an underlying model or simulation, where for each parameter setting, we wish to construct the model or simulation according to the specified parameters, and then train the model or execute the simulation and store the relevant results from the process. We could implement a system like this using Azure Blob Storage or Azure Data Lake Storage (ADLS) to store the model or simulation parameter configurations and training/simulation results, Azure Batch to acquire the required capacity and allocate jobs to nodes, and then either a single dedicated VM or cluster of dedicated VMs that orchestrate the Azure Batch node pool and collate results from the storage. The exact steps to provision these resources and develop the

code to orchestrate and execute the jobs are involved, but we can describe an outline of the process and architecture:

1. Provision a storage account and create a container that will store model/simulation result blobs and another container to store blobs defining the model/simulation parameters.[44]

2. Provision an Azure Container Registry, construct a Docker image, which defines the functionality of the model or simulation given the parameter data, and write that Docker image into the container registry.[45]

3. Given an estimate of model training time and the desired parameters to iterate over, estimate the number of worker nodes needed to complete the entire batch job in a reasonable time period.

4. From the orchestration nodes, construct the parameter data structures, write them to storage, and then define jobs using the Azure Batch API, which points to the parameters in storage and the image stored in the container registry.

5. Start the job and monitor the job status for task-level failures from the orchestration nodes.

6. Detect when the job completes and collate the results to make model parameter selection or generate simulation reports on the orchestration nodes.

This architecture still leverages some PaaS services to manage persistent storage and container images. Azure Batch is capable of acquiring very large amounts of capacity (tens of thousands of instances) relatively quickly and efficiently, allocating work to the acquired nodes.

All of the examples listed above comprise only a small fraction of all of the use cases of modern public clouds like Azure. However, even among this small sample, we see significant variability in terms of the PaaS services coupled with our deployed VMs and the potential utilization patterns of the deployed VMs themselves. The latter utilization variability by use case is relevant to the VM counter data examined in this paper. For example, in the CRUD application case, we are subject to an uncontrollable and variable

[44] A blob, short for Binary Large Object, is a collection of binary data stored as a single entity in a database management system. In the context of cloud storage, blobs are used to store large amounts of unstructured data such as text, images, videos, or, in this case, model/simulation results and parameters.

[45] Docker is a platform that uses OS-level virtualization to deliver software in packages called containers. These containers are lightweight, standalone, and executable packages of software that include everything needed to run an application: code, runtime, system tools, system libraries, and settings.

amount of user requests in the future and must design our architecture to handle both the expected and unexpected variability in this load. In the model training pipeline, we can learn more about the workload ahead of time. Instead, we need to focus on ensuring that the worker pool is sized correctly and that we efficiently pack jobs onto the nodes.

B Data Appendix

B.1 CPU Utilization Data

CPU utilization is a fundamental metric in computing that quantifies the workload on a computer's central processing unit. It is typically expressed as a percentage, representing the proportion of time the CPU spends executing non-idle tasks relative to its total available processing time (Gregg, 2014).

The most common method for measuring CPU utilization relies on system counters provided by the operating system. These counters continuously track the CPU's state, recording the time spent in various modes such as user mode (executing application code), system mode (executing kernel-level operations), and idle mode. By sampling these counters at regular intervals, typically every few milliseconds, the operating system can calculate the percentage of time the CPU spends in non-idle states (Gregg, 2014). These data are then aggregated over longer periods (e.g., seconds or minutes) to provide a meaningful representation of CPU usage.

The raw data we have access to are aggregations of counter readings to the 5-minute level, taking the maximum utilization reading in each 5-minute interval. Therefore, for each VM, at a 5-minute interval, we have the maximum CPU utilization. Since we have data on more than 1 billion VMs, the data at this granularity are not manageable. We further aggregate these data to the VM-day level by calculating the inverse CDF of the distribution of maximum CPU utilization recordings in 5% intervals. In particular, for each 5% increment, we calculate the share of 5-minute max CPU utilization recordings under 5%, 10%, . . ., 95% utilization. We also record the maximum CPU utilization and the total number of hours the VM is running during that day. This sample forms the primary dataset for all analyses conducted in the paper.

B.2 VM Data

Together with the information on CPU utilization of VMs, we also collect information on the important characteristics of VMs to understand their usage patterns and performance. Our data include the data center of the VM, which is anonymized for privacy and security reasons. However, we observe the geographical region of the data center, categorized into

US, EU, and Other. This helps identify the geographical location of the firm and whether firms and units run jobs outside of their domestic country. We also observe the series to which each VM belongs. Cloud providers group VMs of similar sizes, hardware, and features into the same families, typically referred to as machine series or instance types, depending on the cloud provider.[46] The VM series information is anonymized, and we only observe a unique identifier for confidentiality reasons.

Another important piece of information is the VM family, which categorizes VMs based on their primary purpose, such as general-purpose, compute-optimized, memory-optimized, and storage-optimized. We observe the actual values of these variables, allowing us to analyze the families of VMs used by each firm. Additionally, we observe other key VM characteristics, including the operating system (Linux or Windows), memory, and the number of cores.

These VM characteristics include the following set of VM groupings, which we use throughout the text.

1. The *VM family*, as described above, categorizes VMs based on their primary purpose. It takes the values of general-purpose, compute-optimized, memory-optimized, storage-optimized, HPC (high-performance computing), or GPU (graphics processing unit).

2. The *VM series* is a more granular indicator of the architecture of the VM. Each VM series is defined by a combination of hardware, such as the processing chip, and certain proportions or features, such as the available memory and CPU combinations. Generally speaking, VMs within the same VM series will only differ in terms of the data center in which they are physically located, size attributes such as cores or memory, and their operating system; the VM series defines all other attributes of the VM.

3. The *VM product* is a unique combination of VM series, data center, and operating system. This variable encompasses all the characteristics of a VM, except for its memory and CPU capacity.

4. The *VM configuration* is a unique combination of a VM series, data center, operating system, number of cores, and amount of memory. This variable represents the exact hardware and operating system that the VM user chooses and is the most granular variable that classifies a VM.

[46] For examples of machine series from top providers, see these links: Google Cloud— Machine Families, AWS EC2— Instance Types

B.3 Firm and Unit Level Data

There are two ID variables associated with the creator of each VM in our CPU utilization data. The first one is the unit ID. The unit ID collects all users that share a system administrative structure for oversight of the VMs and a payment/billing account with the cloud provider. Each unit ID is then associated with a higher-level firm ID.

Although our CPU utilization data are intermittent between 2017-2023, we have unit-month and firm-month level panel datasets that cover the entire sample period from 2017 until mid-2023. These panel datasets contain monthly normalized statistics on the cloud usage of each unit and firm, including the number of VMs deployed, the number of active VM days, a measure of cloud spend, and the total number of hours and core-hours across all VMs on the cloud. The values are normalized relative to the first usage month of each firm.

In addition to these panels, we also have datasets containing detailed information on each unit and firm in our sample, including their industry classification. Notably, our classification method differs from the standard NAICS and SIC systems; it follows the "vertical" definition—a common term in industry jargon. This vertical classification provides a level of granularity that falls between 2-digit and 3-digit SIC codes, which we subsequently map to 2-digit SIC codes.

Regarding location information, each firm is associated with a billing address. Although we do not have direct access to these addresses, we observe indicators that indicate whether a firm has a billing address in the US, EU, or other regions, as well as an indicator for whether a firm is multinational (i.e., has billing addresses in multiple countries). Additionally, we derive a "usage region" for each firm and unit based on the data center locations where they have the most VM usage. Because network latency increases with the distance between the user and the data center, cloud users typically select data centers within their own region. Thus, the region with the highest compute usage serves as a useful proxy for the firm's or unit's operational location. We use this information for a small number of firms for which the billing location is not available.

Finally, our data also includes quartiles of an independent and internal measure of firm size for each firm, which proxies the number of employees at the firm. These quartiles are computed separately within each region-industry-year combination.

B.4 Sampling Details

In addition to aggregating our data to the VM-day level, we implement several additional measures to reduce data volume and fully anonymize firm identities. With our VM-day dataset, we perform three such sampling steps. First, we filter out the firms that fall

below fixed thresholds for total usage and consistency of usage. This step eliminates a small number of firms with negligible cloud usage. Second, we thin the right tail of the distribution by sampling the firms in the top few percentiles of the distribution (measured by total usage) at an undisclosed sample rate. This sampling is necessary to (i) reduce the data size further, as firms in the top percentiles may have a very large number of VMs, and (ii) eliminate the possibility of identifying large firms. This step still maintains a sample that is representative of a wide array of firms, only reducing the number of very large firms in the sample. Finally, we sample VM-day observations for every firm at a fixed undisclosed rate that is between 70% and 100%.

B.5 Data Cleaning Procedure

In addition to the initial cleaning and sampling applied to the raw data, we further filter the VM-day data to exclude firms with low usage and eliminate short-duration VMs.

First, we remove VMs with a duration of less than 20 minutes. Such VMs are typically used to initiate another job, such as testing configurations, starting batch processing, or briefly scaling to manage temporary demand spikes, and they account for a negligible share of total core hours. We also exclude a very small fraction (less than 0.01%) of VMs running operating systems other than Linux or Windows, or those missing key information such as memory or VM configuration.

At the firm level, we exclude firms that are inactive for more than 80% of the months between their cloud entry and exit month and those with less than 100 hours, 500 core hours, or 200 VM-days of usage. These criteria ensure that each firm included in our analysis has a sufficient sample size to accurately estimate its productivity. These cleaning steps affect fewer than 1% of the firms in the raw data.

B.6 Publicly Available CPU Utilization Data

There are several publicly available datasets that detail CPU utilization information for both cloud and cluster environments. These datasets are referred to as *traces*. These traces often provide data on users and compute hardware as well.

Traces are collected from real-world computing environments. Providers of these computing resources make their traces publicly available for research, analysis, and educational purposes. Below, we describe three usage traces. The first dataset we describe was released by Google Cloud (GC) and is a trace of a high-performance computing cluster. The second dataset is a public *power* trace provided by GC (Google, 2019). The third was released by Microsoft Azure (Azure) and is a trace of a cloud computing environment.

As in the main text, the key variables in these datasets pertain to the characteristics of

provisioned VMs and their utilization information. Each of the datasets described in this section is a trace of cluster information about users' resources, compute activity, and the networks within which their VMs run workloads.

B.6.1 Google Cloud Platform Cluster Trace

The contents of this trace allow studying of job scheduling and cluster management (Verma et al., 2015; Tirmazi et al., 2020). However, our main interest in these data is to use the information provided on CPU utilization to measure the compute productivity of users. The GC trace was sampled in May 2019 and pertains to eight clusters utilizing Google's *Borg* cluster manager. These clusters are located in data centers in New York and Chicago in North America, as well as in Helsinki and Brussels in Europe, and in Singapore in Asia.

The unit of observation in the usage component of the trace data is a *task*. Tasks are processes that originate from programs running as part of jobs submitted to the cluster manager by users. The tasks detailed in this trace are either the result of jobs run by Google engineers, or they are run within resources available to Google services used by internal or external users (Wilkes et al., 2020). Tasks are executed either within resource allocations (similar to a VM) or directly on machines. Observations are recorded every five minutes.

On the user end, these data contain the requested VM, its usage statistics, and user-configured constraints on the requested resources. The resources users can request are memory (RAM) and CPU cores. CPU requests are measured in internal "Google Compute Units" (GCUs), which are similar to CPU cores but enable the comparison of compute hardware across machines in the data (Wilkes et al., 2020). The GCU measurements are reported after being normalized by a constant factor. CPU usage (i.e., GCU usage) is measured in CPU-seconds and reported after being normalized. Therefore, dividing the normalized CPU usage by the normalized CPU requested provides the utilization information. The data also contains information about machine attributes, machine availability, obfuscated user and job identifiers, and variables that track events, missing data, and reasons for task failures.

B.6.2 Google Cloud Platform Power Trace

The content of this trace complements the trace above in that it provides information on the power consumption of the physical machines from the GC cluster trace during the same sampling period, May 2019. This enables us to measure the relationship between CPU utilization and power consumption.

The power consumption data is available at the power distribution unit (PDU) level. The power drawn into the data center is stepped down into a PDU, which serves as the

main distributor of power in a data center. PDUs manage the flow of power to equipment and monitor environmental factors, such as temperature and humidity. Modern cluster computing produces large amounts of heat. Therefore, measuring power consumption requires accounting for the composite power supply to both IT (e.g., servers) and non-IT (e.g., coolers) resources (Singh et al., 2015; Athavale et al., 2018). Since PDUs manage power supplied to clusters and their supporting IT equipment, as well as non-IT equipment such as cooling resources, the power trace provided by GC incorporates both IT and non-IT power demands.

The GC dataset includes power utilization levels of 55 PDUs, which manage the power supply to each of the clusters in the GC cluster trace. The power for each cluster is managed by multiple PDUs. The data include two key variables: total power utilization and production power utilization. Total power utilization, measured in 5-minute intervals, indicates the percentage of available power capacity consumed through a PDU for all IT and non-IT equipment, including coolers. Similarly, production power utilization is an estimated measure provided by GC that details the power consumption attributable to production workloads, including power consumed by non-IT equipment.

B.6.3 Microsoft Azure Trace

Azure publicly provides a number of cloud traces. We focus on a 2019 trace containing a representative subset of Azure VM workloads (Microsoft Azure, 2019).

The dataset spans 30 consecutive days and covers over two million VMs. It is an unbalanced panel with 5-minute CPU utilization readings totaling nearly 1.25 billion measurements from over 5,000 users. Key variables include sanitized user, VM, and deployment IDs; timestamps in seconds that are recorded every 5 minutes; indicators for VM creation and deletion; the number of VMs created; deployment size; CPU utilization statistics (maximum, minimum, average, and 95th percentile); VM virtual core count; and requested VM memory in GB.

C Additional Details on Productivity Measurement

In this Appendix, we first provide a formal definition of our compute productivity measure. Then, we provide some conditions under which compute productivity can be interpreted as factor-augmenting productivity in a full production function.

C.1 Formal Exposition of Compute Productivity

As in the main text, index firms by i, jobs by j, and days by t. Firm i assigns job j, which runs for h_{ijt} hours on day t, to VM v_{ij}. Each VM v is defined by a tuple $(c(v), x(v))$, where

$c(v) \in C \subset \mathbb{N}$ is the number of cores of VM v and $x(v) \in X$ denotes the VM's characteristics, which include the VM's machine series, memory, data center, and operating system.

On day t, we observe n_{ijt} snapshots of CPU utilization $\{u_{ijst}\}_{s=1}^{n_{ijt}}$. By multiplying the utilization with the capacity of the chosen machine, we get n_{ijt} snapshots of the load of each job: $\{\ell_{ijst}\}_{s=1}^{n_{ijt}}$, where $\ell_{ijst} := u_{ijst}c(v_{ij})$. We assume that the load for each job is exogenous — that is, we take it as given that each firm must use the exact same amount of computing power that we observe them using in the data.

On each day, there is a set of VMs available for the firm to choose from. Let V_t be the set of VMs available on day t, and $V_t(x) = \{v \in V_t : x(v) = x\}$ be the set of VMs available on day t that have characteristics x. We also define the outside option "VM" v_0 as the 0-core VM that represents not running a job. Similar to the load, we assume that the characteristics $x(v)$ of a VM are exogenous, and we take it as given that the firm chose these correctly. Therefore, we infer the choice set of firm i for job j on day t to be $V_{ijt} = V_t(x(v_{ij})) \cup \{v_0\}$.

We compare the firm's provisioning decision v_{ijt} with the decision of a hypothetical cost-minimizing firm v^*_{ijt}. To do so, we need to model the optimal provisioning process. For our baseline analysis, we assume the following:

Assumption 1. *The cost-minimizing firm provisions based solely on the peak load of the VM, which we take to be the 95th percentile load over the period in which the VM is being provisioned.*

Assumption 2. *If a VM has a peak utilization of under 10% over a given time period, then the firm does not receive any benefit from that job over that time period.*

Assumption 3. *The cost-minimizing firm will downsize a machine only if the peak utilization on that machine would be less than 90%.*

Assumption 4. *After the initial provisioning decision, it is only worthwhile for a firm to change its provisioning decision if a VM is improperly provisioned over a seven-day period or longer.*

As discussed in the main text, Assumption 1 is relatively standard, both in industry and literature definitions of improper provisioning. Assumption 2 is justified by a 10% peak CPU utilization being explainable by background processes of the CPU and not by any foreground processes run by the user. Assumption 3 comports with the rightsizing recommendations given by cloud providers to their clients. Finally, Assumption 4 is justified by firms facing sufficiently high switching costs from reconfiguring a job to a new type of VM.[47]

[47] In practice, some major cloud providers have processes to reprovision running jobs to VMs of different sizes without any interruption in service. For example, see AWS—Resizing clusters. Thus, we view this assumption as conservative.

Let $\mathcal{T}_t^k = \{t, t+1, \ldots, t+k-1\}$ be defined as the set of k consecutive days starting with day t. Let $\overline{\ell}_{ij}(\mathcal{T}_t^k)$ be the peak load over $\mathcal{T}_t^k$:

$$\overline{\ell}_{ij}(\mathcal{T}_t^k) = \max\left\{\ell : \frac{\sum_{r=t}^{t+k-1}\sum_{s=1}^{n_{ijr}} \mathbf{1}(\ell > \ell_{ijsr})}{\sum_{r=t}^{t+k-1} n_{ijr}} \leq 0.95\right\} \tag{9}$$

Define the peak utilization $\overline{u}_{ij}(\mathcal{T}_t^k)$ analogously. Let T be the length of job j in days and, for ease of exposition, relabel the days so that job j lasts from day 1 to day T.

First, suppose $T \geq 7$. Given our assumption, the cost-minimizing firm's decision on each day $t = 1, \ldots, T$ solves:

$$\begin{aligned} v_{ijt}^* &= \arg\min_{v \in V_{ijt}} c(v) \\ \text{s.t.} \quad \min_{r \in \{\max\{1,t-6\},\ldots,\min\{t,T-6\}\}} & \overline{\ell}_{ij}(\mathcal{T}_{t-r}^7)\mathbf{1}(\overline{u}_{ij}(\mathcal{T}_{t-r}^7) \leq 0.1) \leq c(v) - 0.1 \cdot \mathbf{1}(v \neq v_{ij}) \end{aligned} \tag{10}$$

It is easiest to interpret the constraint of expression (10) in words. The left-hand side of the constraint searches over all of the sets of seven consecutive days that include day t. If day t is part of a seven-day stretch in which the peak utilization is under 0.1, then it is idle, the left-hand side of the constraint will evaluate to zero, and any VM will cover the load over those seven days. In this case, the cost-minimizing firm will choose to deprovision the job, i.e., select $v_{ijt}^* = v_0$ for those seven days. Otherwise, firm i will take the smallest VM that will cover the peak utilization of job j over a seven-day stretch that includes day t. This will always include as a possibility the actual VM that the firm chose, v_{ij}, but could include a smaller VM if the VM is downsizable (there exists a smaller VM with the same characteristics) and the peak load over the seven-day period is small enough to be covered by this smaller VM with at most a peak utilization of 90%. If this indeed is the case, then $v_{ijt}^* \neq v_{ij}$ and we say that job j is overprovisioned over those seven days. In practice, because the number of cores in a given VM nearly always scales by powers of two, a VM will be overprovisioned if a smaller VM exists and its 95th percentile CPU utilization over a seven-day period is under 45%.

For VMs that are shorter than seven days ($T < 7$), we evaluate only the initial provisioning decision and do so over the entire length of the VM. That is, the cost-minimizing firm's provisioning decision solves

$$v_{ijt}^* = \arg\min_{v \in V_{ijt}} c(v) \quad \text{s.t.} \quad \overline{\ell}_{ij}(\mathcal{T}_1^T)\mathbf{1}(\overline{u}_{ij}(\mathcal{T}_1^T) \leq 0.1) \leq c(v) - 0.1 \cdot \mathbf{1}(v \neq v_{ij}) \tag{11}$$

As discussed in the main text, the final productivity measure ω_{ijt} is the ratio between

resource usage of the cost-minimizing firm and firm i's actual resource usage on job j on the day t: $\omega_{ijt} = c(v^*_{ijt})/c(v_{ij})$.

C.2 Microfoundation as Factor-Augmenting Productivity

We note that while ω_{ijt} is fundamentally a measure of how effectively firm i solves a cost minimization problem, holding output fixed, it also has an interpretation as traditional factor-augmenting productivity.

Suppose that firm i produces a single product sold at an exogenous price normalized to 1. For ease of exposition, suppose the firm only uses compute for one job j. Let the firm's production function at a given moment in time s be $f_{is}(\ell_{ijst}, z)$, where ℓ is compute load and z are other inputs that are assumed fixed in the short run. Assume the price of compute is linear in the amount of compute used with price p. Further, let the time length of each moment s be h hours. Then, the firm solves:

$$\max_{\{\ell_{ijst}\}_s, v} \sum_s f_{is}(\ell_{ijst}, z) - phc(v) \quad \text{s.t.} \quad v \text{ satisfies the constraint of (10).} \tag{12}$$

For a given path of compute load $\{\ell_{ijst}\}_s$, denote the cost-minimizing firm's choice of VMs as v^*_{ijt}.

Now suppose that firm i choosing VM v_{ij} results in compute productivity ω_{ijt} on day t. The definition of ω_{ijt} implies that

$$\frac{p}{\omega_{ijt}} hc(v^*_{ijt}) = phc(v_{ij}). \tag{13}$$

As such, firm i's profits on day t are given by

$$\sum_s f_{is}(\ell_{ijst}, z) - phc(v_{ij}) = \sum_s f_{is}(\ell_{ijst}, z) - \frac{p}{\omega_{ijt}} hc(v^*_{ijt}). \tag{14}$$

As shown in Appendix A of Demirer (2025), this formulation of ω_{ijt} is equivalent to factor-augmenting productivity because ω_{ijt} scales the input price for compute.

D Additional Details on Estimation

This appendix provides the details of the estimation procedures used in the main text.

D.1 Details on Cloud and Electricity Spending

To calculate the total energy spending of US businesses, we combine the data from two EIA surveys. The 2018 Commercial Buildings Energy Consumption Survey provides the

total energy consumption of non-manufacturing businesses. This number was $114 billion in 2018. The second survey is the Manufacturing Energy Consumption Survey, according to which the total energy consumption of manufacturers was $142 billion.[48] This sums to $256 billion. If we incorporate a roughly 20% increase in electricity and natural gas prices since 2018, the number approximately reaches $300 billion.[49]

For cloud computing spending, there are no official numbers; however, there are multiple surveys and estimates by market research companies. These sources indicate that U.S. public-cloud spending is well into the $300–400 billion range today and will continue growing—approaching $700-800 billion by 2030—as enterprises shift ever more of their IT budgets to cloud platforms.[50]

One caveat for this calculation is that the EIA does not include commercial transportation in its estimates of industrial and commercial energy spending. To approximate total transportation energy use, we can turn to data from the Transportation Energy Data Book, which shows that 27% of all gasoline consumption is due to heavy trucks (Transportation and Energy Data Book). Multiplying that 27% by the EIA's $600 billion figure for total transportation spending (EIA Today in Energy) adds roughly $162 billion in commercial energy expenditures attributable to transportation. Even after including this adjustment, total cloud spending is only slightly lower than commercial energy spending today and is expected to exceed it in the coming years.

D.2 Details of Productivity Estimation

We use the following procedure to implement the compute productivity measures presented in Section 4. First, for each possible VM configuration (a combination of VM series, data center, operating system, memory, and cores) a firm could choose, we evaluate whether that configuration is downsizable on each day (another VM configuration of the same family, data center, operating system, and memory, but half the number of cores is available on that day). We also evaluate whether the configuration is *twice downsizable*, which is defined as there existing a VM that the configuration could be downsized to that is itself downsizable. As discussed in the main text, cores scale in powers of two; therefore, if a VM is twice downsizable, this means there exists a VM with the same machine family,

[48] U.S. Energy Information Administration, 2018 Commercial Buildings Energy Consumption Survey, non-manufacturing business energy expenditures (CBECS 2018); U.S. Energy Information Administration, 2018 Manufacturing Energy Consumption Survey, manufacturing energy expenditures (MECS 2018).

[49] See Fred Electricity Prices and EIA Commercial Natural Gas Prices.

[50] International Data Corporation estimates cloud spending at $432 billion in 2024 and $697 billion in 2027 (Worldwide Software and Public Cloud Services Spending Guide). Statista reports $471 billion in 2025 (Statista Public Cloud). Grand View Research estimates over $250 billion today and $813 billion in 2030 (Cloud Computing Market Outlook).

data center, operating system, and memory that has a quarter of the number of cores.

Second, for each VM on each day, using the daily inverse utilization CDF, we compute the peak (95th percentile) CPU utilization for all seven-day streaks that include that day. For VMs that last for fewer than seven days, we compute the peak CPU utilization over the life of the VM. We then assign the productivity measure ω_{ijt} to each VM-day using the following hierarchical definition:

1. If a VM-day is part of a seven-day streak with a peak CPU utilization lower than 10%, it is idle and assigned a value of $\omega_{ijt} = 0$.

2. Else if a VM-day is part of a seven-day streak with a peak CPU utilization lower than 20% AND the VM configuration is twice downsizable, it is overprovisioned, with the correct configuration being a VM that is a quarter the size and assigned a value of $\omega_{ijt} = 0.25$.

3. Else if a VM-day is part of a seven-day streak with a peak CPU utilization lower than 45% AND the VM configuration is downsizable, it is overprovisioned, with the correct configuration being a VM half the size, and assigned a value of $\omega_{ijt} = 0.5$.

4. Else a VM-day is properly provisioned and assigned a value of $\omega_{ijt} = 1$.

Once we have defined the productivity of a VM-day ω_{ijt}, we proceed by calculating the firm-month level productivity estimates using the aggregation given in Equation (1). This constitutes our baseline productivity measure that is used throughout the main text. In addition to these firm-month level estimates, we also calculate (i) firm, (ii) unit, (iii) unit-month, (iv) unit-month-VM product and (v) firm-month-VM product level.[51] These measures use the same formula as given in Equation (1):

$$\omega_{im} = \frac{\sum_{j \in J_{im}} \sum_t \omega_{ijt} ch_{ijt}}{\sum_{j \in J_{im}} \sum_t ch_{ijt}} \tag{15}$$

but instead of summing over the jobs used by the firm i in a given month m, it sums over the corresponding level of aggregation. For example, to calculate firm-month-VM product level productivity, we aggregate all ω_{ijt} of the VMs that are used by the firm for a given VM product.

We also define alternative variables that decompose productivity from idleness and from overprovisioning separately. To study idleness, we define a dummy variable $y_{ijt}^{\text{idle}} = \mathbf{1}(\omega_{ijt} = 0)$, while to study overprovisioning, we use an indicator for whether the VM

[51] A VM product is a unique combination of VM series, data center, and operating system.

is overprovisioned $y_{ijt}^{\text{overprov}} = \mathbf{1}(\omega_{ijt} \in (0,1))$. We follow the analogous procedure to compute core-hour weighted averages of each firm's idleness inefficiency y_{im}^{idle} and overprovisioning inefficiency y_{im}^{overprov}. To remove the negative mechanical correlation between these two variables, we estimate a firm's overprovisioning inefficiency, excluding all idle observations; that is, overprovisioning inefficiency will be the share of VMs that are overprovisioned, conditional on not being idle. Before analysis, we compute "idleness productivity" as $\omega_{im}^{\text{idle}} = 1 - y_{im}^{\text{idle}}$ so that higher values continue to correspond to more efficient usage, and similarly for $\omega_{im}^{\text{overprov}} = 1 - y_{im}^{\text{overprov}}$.

We also calculate all of our productivity measures controlling for time- and machine-varying factors using fixed effects by regressing ω_{ijt} on a fixed effect that is either at the firm, firm-month, firm-month-VM product, unit, unit-month, or unit-month-VM product level. These regressions use the following estimating equation (here, for firm-month level productivity):

$$\omega_{ijt} = \omega_{im} + Z'_{jt}\beta + \varepsilon_{ijt}. \tag{16}$$

Here, Z_{jt} are the controls, and ω_{im} are the firm-month-level productivity estimates as fixed effects. In these regressions, we weigh each observation by core-hours in order to properly account for the resources used by each VM on each day. Our baseline estimates in Equation (16) correspond to a regression with no controls since the resulting fixed effects simply represent the weighted average of the dependent variable. We run these regressions controlling for (i) day-of-week fixed effects and an indicator for whether there is a holiday in the region on the given date; (ii) day-of-week plus holiday plus machine family fixed effects; (iii) day-of-week plus holiday plus machine family interacted with data center region fixed effects; and (iv) day-of-week plus holiday plus VM product fixed effects.[52] Results with these alternative levels of controls are located in Appendix H. We also use y_{ijt}^{idle} and $y_{ijt}^{\text{overprov}}$ as dependent variables and convert the resulting fixed effects to idleness productivity and overprovisioning productivity by subtracting the fixed effects from 1 as above.

In these regressions, a location normalization is to be made — one can add and subtract a constant from two different fixed effect levels and arrive at the exact estimates for all units. Our normalization is to make the average productivity according to each of these fixed effect regressions equal to the average productivity without controls. Finally, for all the alternative controls, we verify that the controls form a connected set, and therefore, the fixed effects resulting from the estimation procedure are directly interpretable and comparable with one another.

[52] For the firm-month-VM product and unit-month-VM product regressions, the final three sets of controls are extraneous because they are nested by VM product.

D.2.1 Two Examples of Productivity Calculations

Table OA-1 presents usage data from two VMs in our sample to help describe how we compute our productivity estimates at the VM-day level. For each VM on each day, we display the amount of time in hours the VM was active, the core-hours used by that VM, and relevant selections from the inverse CDF of the VM's CPU utilization: the fraction of recordings the VM was less than 10% utilized, 20% utilized, and 45% utilized respectively.[53] We also display ω_{ijt}—the productivity we assign to each VM-day—which is a function of the shown columns.

Panel A displays a 4-core general-purpose Linux VM that is active over a three-day period. This VM's usage is volatile: it is highly utilized on the day it is created, as it is above 45% utilization approximately 5/6 of the time, but is essentially unused thereafter, spending nearly all its time under 10% utilization. However, since this VM is shorter than seven days, what matters for our productivity assignment is the total distribution of CPU utilization over the entire life of the VM, displayed in the bottom row. Because the peak utilization of this VM is above 45%—i.e., the value of the inverse CDF at 45% is under 0.95—this VM is marked as properly provisioned for its entire life, with $\omega_{ijt} = 1$ on every day.

Panel B displays an 8-core memory-optimized Linux VM that is active over a 10-day period. 2- and 4-core memory-optimized VMs with the same amount of memory and the same operating system were available in the same data center throughout this time period; therefore, this VM is twice downsizable. Because this VM is active for longer than seven days, the relevant distributions for assigning VM-day productivity are the running seven-day totals displayed in the bottom panel. Over days 1-7, the peak CPU utilization of the VM was above 20% (as the inverse CDF value at 20% was under 0.95) but below 45% (as the inverse CDF value at 45% was above 0.95). While this implies that the VM was not idle over this seven-day period, and the firm could not have downsized the VM to a 2-core machine, they could have downsized it to a 4-core machine and been under 90% utilization. Therefore, this 7-day streak is marked as (once) overprovisioned. By similar logic, the 7-day streaks starting on days 2 and 3 are also marked as overprovisioned. However, in the last seven days, the VM is active—days 4-10—the peak utilization of the VM is under 10%, meaning the VM is idle for these seven days. Thus, all days between 4-10 are assigned a productivity of $\omega_{ijt} = 0$, as they are part of a 7-day period where the VM is marked as idle. Days 1-3 are not part of such a period, but they are each part of

[53] Though we do not show the remainder of the columns, our data include the inverse CDF in 5 percentage point increments. The only points on the inverse CDF that are relevant for our procedure are the ones displayed in the table.

Table OA-1: Example of CPU Utilization Data from Two VMs

Day	Hours active	Core-hours	Share of CPU util recordings below X% 10%	20%	45%	ω_{ijt}
			Panel A: VM 1 (3 days)			
1	20.97	83.87	0.158	0.158	0.167	1
2	24	96	0.985	0.996	1.000	1
3	23.67	94.67	0.984	0.996	1.000	1
Total			0.737	0.745	0.750	
			Panel B: VM 2 (10 days)			
1	16.33	130.67	0.778	0.894	0.995	0.5
2	24	192	0.304	0.485	0.996	0.5
3	24	192	0.477	0.602	1.000	0.5
4	24	192	0.996	1.000	1.000	0
5	24	192	0.996	1.000	1.000	0
6	24	192	0.992	1.000	1.000	0
7	24	192	0.951	0.996	1.000	0
8	24	192	0.964	0.996	1.000	0
9	24	192	0.872	0.996	1.000	0
10	21.03	168.27	0.991	1.000	1.000	0
				7-day totals		
1-7			0.786	0.853	0.999	
2-8			0.811	0.868	0.999	
3-9			0.893	0.942	1.000	
4-10			0.966	0.998	1.000	

Notes: This table displays selected points from the inverse CDF of CPU utilization for two VMs in our data. VM 1 is a downsizable 4-core general-purpose Linux VM. Although it is rarely used on its second and third days of activity, its usage on the first day is enough such that it is marked as properly provisioned. VM 2 is a twice downsizable 8-core memory-optimized Linux VM. Its usage patterns result in being marked as overprovisioned on its first three days of usage and idle on its fourth through tenth days of usage.

7-day periods where the VM is marked as (once) overprovisioned, so they are assigned $\omega_{ijt} = 0.5$.

D.3 Details of Dispersion and Persistence Estimation

This section provides the details of the estimations presented in Table 3.

D.3.1 Dispersion

In our dispersion analysis reported in Panel A, we employ firm-month level compute productivity estimates as detailed in Section D.2. We restrict our sample to firms with an average of at least 50 VM-day observations per month, ensuring that only firms with

precisely estimated productivity levels are included. Column (1) reports statistics (mean, median, 90-10th percentile ratio, interquartile ratio, and R^2) computed from the firm-month data without any controls. For Columns (2)–(4), we calculate the same statistics within each reported group—namely by industry, month, and industry-by-month. We then calculate a weighted average of each statistic across these groups, using the number of firms in each group as weights. This approach prevents smaller industries with fewer firms from disproportionately influencing the overall statistics.

D.3.2 Within-Firm Decomposition

In our within-firm decomposition analysis reported in Panel A, we estimate the variance explained by within-firm and between-firm heterogeneity as follows:

$$\text{Var}(\omega_{im}^k - \bar{\omega}_m) = \text{Var}(\omega_{im}^k - \bar{\omega}_{im}) + \text{Var}(\bar{\omega}_{im} - \bar{\omega}_m)$$

where i denotes firm, k denotes unit, and m denotes month. $\bar{\omega}_m$ denotes the mean productivity in month m and $\bar{\omega}_{im}$ is the mean productivity of firm i in month m.

To implement these decompositions, we use a regression analysis and obtain the adjusted R^2 from regressions. Specifically, for the within-firm decomposition, we regress unit-level productivity on firm fixed effects and take the R^2 from that regression as the between-firm component. We repeat the same exercise while including the controls reported in Columns (2-4) of Table 3. For these specifications, the goal is to quantify the variance explained by firm fixed effects in unit-level variance after taking out the variance explained by the control variables. We implement this by first running the fully saturated regression with the control variables and recording the resulting R^2 as R_0^2. Here, $1 - R_0^2$ quantifies the remaining variance after including the control variables. Then, we include firm fixed effects by interacting them with the control variables and record the resulting R^2 as R_1^2. To find the variance explained by between-firm variation, we calculate $(R_1^2 - R_0^2)/(1 - R_0^2)$.

D.3.3 Within-Firm-Between-Region Decomposition

In our within-region across-region decomposition analysis reported in Panel A, we decompose the within-firm variance as follows:

$$\text{Var}(\omega_{im}^k - \bar{\omega}_{im}) = \text{Var}(\omega_{im}^{kr} - \bar{\omega}_{im}^r) + \text{Var}(\bar{\omega}_{im}^r - \bar{\omega}_{im})$$

where r denotes a region. For this analysis, we only use multinational firms, which are firms that have units in multiple geographic regions, classified as US, EU, and domestic.

The calculation of within- and between-region decomposition is similar to the within-firm decomposition. We restrict the sample to multinational firms and estimate the variance explained by region-fixed effects after accounting for firm-fixed effects as between components, with or without control variables specified in the table. Since this analysis quantifies the contribution of the region in explaining variance within a firm, industry controls are fully absorbed and, therefore, do not apply. Those values are labeled as "-" in the table.

D.3.4 Persistence

For persistence results in Panel B, we use the month-firm level data and regress the compute productivity measure ω_{im} on 1-month, 1-year, and 5-year lagged values separately for overall productivity, idleness productivity, and overprovisioning productivity. Results in Columns (2-4) run the same regressions by adding the corresponding control variable specified in the column title to the regression. In these regressions, standard errors are clustered at the firm level. We report the results for the overall productivity in Table 3 and idleness and overprovisioning productivity in Table OA-7.

D.4 Details of Learning Estimation

In our learning analysis, we remove all firms with an average of fewer than 50 VM-days per month over the 2022-2023 timeframe.

D.4.1 Cross-Sectional Learning Analysis

For the learning analyses that are based on cross-sectional variation (Figure 7, Figure 9(b) and Table 5, Panel A), we begin by computing firm-level productivity for a given time period by averaging each firm's productivity across months, weighted by core-hours. For Figure 7 and Table 5, Panel A, this includes July-September 2022; for Figure 9(b), this includes July 2022-June 2023. We compute each firm's experience as the number of months between their first cloud usage and the beginning of the sample period (July 2022). We then compute statistics (means and quantiles) and run regressions unweighted across firms. The overall 90-10 percentile ratio in Figure 9(b) is taken from Table 3.

D.4.2 Timeseries Learning Analysis

For the learning analyses that are based on within-cohort variation in productivity over July 2022-June 2023 (Figures 8 and 9(a) and Table 5, Panel B), we limit to a balanced panel of firms, i.e., firms that have usage in each month from July 2022 to June 2023, that started using the cloud in either June-July 2022, 2021, 2020, or 2019. For Figure 9(a), we only

include firms that started using the cloud in June-July 2022 and classify them into five equally sized groups based on their productivity in July 2022. We then compute statistics unweighted across firms.

D.4.3 Power Law Estimation

For the power law estimation in Table 5, we regress log productivity on an intercept and log experience in months, given each of the samples described above. Experience is as of the end of the month—that is, a firm that started using the cloud in July 2022 has one month of experience in July 2022. We floor productivity by 0.01 before taking the log to avoid missing values.

D.4.4 Normalization and Standard Error Calculation

In several figures in this section, we normalize the productivity estimates for firms in each month by dividing them by the productivity estimate of a specific group of firms. In Figure 7, we normalize by the average productivity of firms in their first month; in Figure 8, we normalize by the average productivity of the June-July 2022 cohort in their first month; and in Figure 9(a), we normalize by the average productivity of the third quintile of initial productivity in their first month.

For plots with standard errors, we then compute the standard errors of the ratio between the productivity estimate of firms in a given month and the productivity estimate of firms in their first month using the delta method. In particular, suppose that $\overline{\omega}_t$ is the expected productivity of firms that are t months old, and σ_t is the standard error. Using the delta method, a first-order approximation of σ_t^{norm}, the standard error of $\overline{\omega}_t/\overline{\omega}_0$, is:

$$\sigma_t^{\text{norm}} \approx \frac{1}{\overline{\omega}_0}\sqrt{\sigma_t^2 - \frac{2\overline{\omega}_t}{\overline{\omega}_0}\text{cov}(\overline{\omega}_0, \overline{\omega}_t) + \frac{\overline{\omega}_t^2}{\overline{\omega}_0^2}\sigma_0^2} \tag{17}$$

To compute an estimated $\widehat{\sigma}_t^{\text{norm}}$, we plug in the estimated average productivities, along with the estimated variances and covariances from the coefficient covariance matrix of the regression of productivity on firm experience indicators. In this regression, standard errors are clustered by the firm; this implies a nonzero covariance across months in the first year in Figure 8. In Figure 7, the unit of observation is the firm, and therefore, the standard errors are computed using the empirical standard deviations, and the covariance in the estimates is assumed to be zero. The exception is when $t = 0$, in which case we know that $\text{cov}(\overline{\omega}_0, \overline{\omega}_0) = \sigma_0^2$, and therefore this expression simplifies to $\sigma_0^{\text{norm}} = 0$.

D.5 Details of Learning Decomposition Analysis

D.5.1 Within-Firm Learning Decomposition

In our learning decomposition, we restrict our sample to the period from July 2022 to June 2023 to be able to calculate month-to-month productivity growth. Following Melitz and Polanec (2015), we decompose the log productivity change into five components specified in the main text: (i) within, (ii) across, (iii) cross, (iv) entry, and (v) exit. In some rare cases, firm or unit productivity in a given month is zero, reflecting that all VMs are idle. For those months, we set productivity to 0.01 so that we do not drop those observations when we take the logarithm. In other rare cases where a firm or unit does not have any usage in a given month but we observe usage afterward, we do not treat those months as entry and exit, but we impute the productivity of that unit from the previous month, and we set its core hours to zero. We also restrict the sample to firms with at least two units to be able to implement the decomposition.

We first implement the decomposition given in Equation (6) for each firm and calculate the five decomposition components. We then take an unweighted average of each component across firms or units within each firm experience group given in the x-axis of Figure 10(a): 0-1, 1-2, 2-3, 4-6, 7-9 and 10+ in years. The firm experience is measured as of the end of the sample, June 2023.

In the calculations of Figure 10(b), we first subset the data to firms that are more than three years old and have a unit that began using cloud computing in July 2022 and continued usage through June 2023. These units constitute our sample of new units within experienced firms, which are reported in red color. We then subset the units of these firms that started using cloud computing before July 2019 so that these units have at least three years of experience by July 2022, which are reported in black color. These units constitute our sample of experienced units. We then calculate the average productivity of these groups for each month from July 2022 to June 2023 and report the productivity levels by normalizing them relative to the productivity level of experienced units in July 2022.

D.5.2 Within-Unit Learning Decomposition

The implementation of within-unit decomposition Figure 11(a) proceeds similarly to the within-firm decomposition. Instead of decomposing a firm's month-to-month productivity changes into unit-level components, we decompose a unit's month-to-month productivity into machine type components using the unit productivity estimates for each machine type. Here, a machine type corresponds to a combination of "VM series, data center, and operating system". We make this choice because these are the VM dimensions for which

it is most likely that learning will occur. This analysis is restricted to firms with multiple units so that the sample is consistent with the sample of within-unit decomposition. We also restrict the sample to the units that use multiple machine types to implement the decomposition.

The implementation of Figure 11(b) follows that of Figure 10(b) with one key distinction. Unlike firms and units—for which we observe the exact dates of entry and exit from the cloud—we lack separate data on when a firm/unit began using a particular machine type. Instead, we infer this information from the VM-level dataset by recording the first time a firm/unit uses a VM type. Since our 2022 data begins in July, we identify new machine types as those first used by a firm in August 2022. This approach may introduce small errors if a firm had used a machine series prior to July 2022 but did not use it in July 2022. However, such cases should be rare, and when they occur, they render our results more conservative. After identifying the first use date of a machine type, we proceed with the analysis as described in the previous paragraph.

D.6 Details of CPU Utilization and Electricity Relationship Estimation

This section outlines our method of measuring the power consumption of VMs based on their utilization levels and provides background information on the power consumption characteristics of VMs.

One unique aspect of computing hardware is that it consumes a significant amount of electricity even when idle because it needs to maintain essential functions such as operating systems, network connections, and cooling systems (Meisner et al., 2009; Duan et al., 2015).[54] Therefore, understanding the relationship between utilization and electricity consumption is crucial for calculating the overall resource impact of productivity dispersion, as different types of computing inefficiencies (idleness and overprovisioning) result in different levels of inefficient electricity use.

We combine the public CPU utilization and power data provided by Google Cloud (GC) to estimate the relationship between CPU utilization and power consumption. Appendix B.6 offers a detailed description of the datasets. First, we explain how we aggregate VM-level utilization information to the level at which power consumption is measured. Next, we outline the specifications used to estimate power consumption as a function of utilization and then present the results.

[54] Additionally, peripheral components like hard drives and power supplies continue to draw power, and the servers must remain in a ready state for quick activation.

D.6.1 *Aggregating CPU Utilization from VMs by PDU*

The GC cluster trace denotes each of the eight clusters contained in the data by a through h. When compressed, the full size of the cluster trace alone is nearly 2.6 terabytes. Therefore, in order to attenuate the computational burden of aggregating and merging the entire data, we focus on cluster a.

In Appendix B.6, we described the "task" as the unit of observation in the usage data of the cluster trace. Each task in the cluster trace is identified by an index relative to a *collection ID*. A collection is either a set of resources where jobs executing tasks run or stand-alone jobs submitted directly to the scheduler to run on a machine. Hence, the unique usage observation is identified by the pairing (collection ID, task ID).

Each PDU is uniquely associated with a cluster. Moreover, each PDU supplies power to a specific subset of physical machines within a cluster. Every task is scheduled on a single machine. As noted in Appendix B.6, CPU utilization (in terms of GCUs) is reported after being normalized. However, the normalizing factor is the same across all observations. Hence, for each 5-minute interval t and PDU p, we compute CPU utilization at the PDU level as

$$U_{pt} = \frac{\frac{1}{c}}{\frac{1}{c}} \cdot \frac{\sum_{m_{jt} \in \mathcal{M}(p,t)} \sum_{i \in m_{jt}} u_{i,m_{jt}}}{\sum_{m_{jt} \in \mathcal{M}(p,t)} \sum_{i \in m_{jt}} r_{i,m_{jt}}}, \tag{18}$$

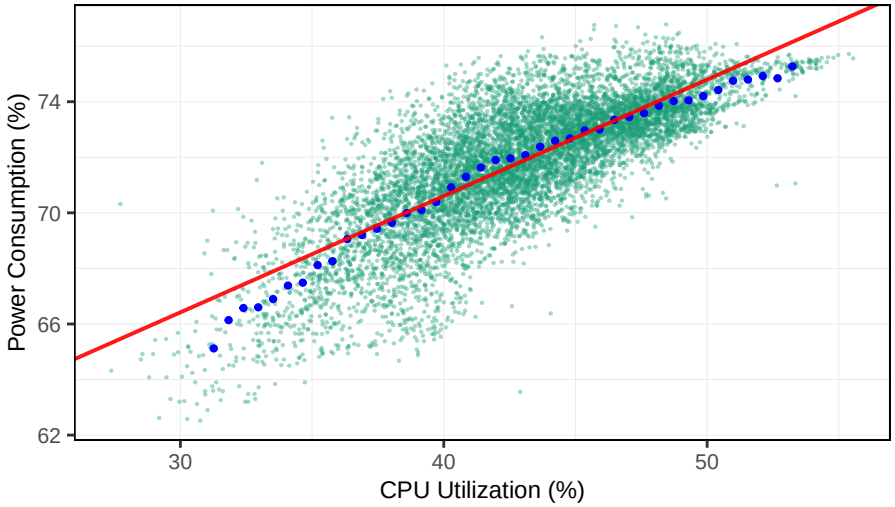
where c is the resource specific normalizing factor, $\mathcal{M}(p,t)$ denotes the set of active machines belonging to PDU p at time t, i indexes the pair (collection, task) at t, and $u_{i,m_{jt}}$ and $r_{i,m_{jt}}$ are the used and requested CPU resources of i, respectively. Since c enters the reported measures linearly, it gets canceled out in the computation of U_{pt} to yield a CPU utilization measure in percentage terms.

D.6.2 *Estimating the Relationship between CPU Utilization and Electricity Consumption*

In the computer science and electrical engineering literature, researchers have estimated the relationship between CPU utilization and power consumption through a combination of regression analysis and experimental methods (Husain Bohra and Chaudhary, 2010; Kansal et al., 2010; Waßmann et al., 2013; Jiang et al., 2013). Experimental studies utilize machines with fixed characteristics and controlled computing environments to generate data on CPU utilization and power consumption.

In our paper, we will use a regression-based method to analyze the power consumption of

Figure OA-1: Relationship Between Power and CPU Utilization



Notes: This figure shows the estimated relationship between CPU utilization and power consumption for VMs. The scatter plot shows individual data points representing power consumption (%) as a function of CPU utilization (%) in 5-minute intervals. Blue points show binned scatter points, and the red line shows the regression line.

VMs. In particular, we estimate the following regression model:

$$P_{pt} = \alpha + \beta \cdot U_{pt} + \varepsilon_{pt},$$

where P_{pt} represents the total power consumption of PDU p at time t. The intercept, α, indicates the baseline power consumption when VMs are idle (i.e., zero CPU utilization), while the coefficient β measures the effect of a one percentage point increase in CPU utilization on power consumption. The term ε_{pt} captures the error. We estimate this model using 5-minute interval data on utilization and power consumption via ordinary least squares.

D.6.3 Results

Figure OA-1 displays the results from the regression of electricity usage on utilization along with a binscatter plot. The results reveal several noteworthy findings. We see that the relationship between power consumption and CPU utilization is approximately linear with a regression coefficient of 0.5, indicating that every percentage point (pp) increase in utilization leads to a 0.5 pp increase in power consumption.[55]

[55] Although the relationship becomes nonlinear at the boundary of the utilization support, it is not precisely estimated due to a lack of data. Additionally, the line has to cross the point (1,1) by construction, supporting the linearity assumption.

In our sample, as can be seen from Figure OA-1, CPU utilization seldom falls outside the range of 30%-50%. This is because, in real-world cluster traces, we rarely observe VMs that consistently use 0% CPU. Therefore, the constant term in the linear model, α, is estimated through extrapolation of the observed relationship to the utilization at 0% CPU. We estimate this term to be 0.5, which suggests that idle VMs consume about 50% of their full capacity electricity. This value is consistent with the experimental literature discussed above.

D.7 Details of Counterfactual Resource Calculations

This section details the counterfactual resource calculations described in Section 8. We first present the methodology for calculating core-hours usage under counterfactual productivity improvements and then detail the corresponding electricity calculations.

D.7.1 Counterfactual Core-Hour Calculations

Our goal is to calculate the total core-hours that would be saved if all firms below the benchmark productivity level reached the productivity level $\bar{\omega}_{im}$. For this exercise, we denote the counterfactual productivity ω^c_{im} as:

$$\omega^c_{im} = \bar{\omega}^c_m \cdot 1(\omega_{im} < \bar{\omega}^c_m) + \omega_{im} \cdot 1(\omega_{im} \geq \bar{\omega}^c_m). \tag{19}$$

where ω_{im} is firm i's actual productivity level in month m. For brevity, we suppress the month subscript m as the time dimension is not important in this analysis. In this analysis, we need to calculate the total core-hours a firm would use if its productivity changes from ω_i to ω^c_i. This calculation is relatively straightforward because it does not depend on whether firms increase productivity through idleness or overprovisioning; the core-hours that would be saved will be the same regardless of the mechanism of improvements.

Let s_i denote the total core-hours used by firm i. We categorize these core-hours into three types of machine utilization patterns:

$$s_i = s^{id}_i + s^{op}_i + s^{pp}_i \tag{20}$$

where s^{id}_i, s^{op}_i, and s^{pp}_i denote idle, overprovisioned, and properly-provisioned core-hours of firm i, respectively. Following the analysis in Section 4.1, we denote the output from these different types of VMs as $y^{id}_i = 0$, $y^{op}_i = 0.5 s^{op}_i$, and $y^{pp}_i = s^{pp}_i$; thus, $y_i = y^{id}_i + y^{op}_i + y^{pp}_i$ represents the total compute output. Let s^c_i denote the total core-hours needed for firms to

produce the same output under the counterfactual productivity ω_i^c. Therefore, we have:

$$\omega_i = \frac{y_i}{s_i}, \qquad \omega_i^c = \frac{y_i}{s_i^c}. \tag{21}$$

By taking the ratio, we can find s_i^c as:

$$s_i^c = s_i \frac{\omega_i}{\omega_i^c}.$$

This calculation shows that the mechanism by which firms achieve productivity gains does not affect the counterfactual calculations; s_i^c depends only on the level of counterfactual and factual productivity. By aggregating s_i^c across firms, we can calculate the total counterfactual core hours, s^c as:

$$s^c = \sum_i s_i \frac{\omega_i}{\omega_i^c}.$$

The total percentage of resource savings is given by the ratio between counterfactual and actual core-hours:

$$\Delta s = \frac{s - s^c}{s}.$$

D.7.2 Counterfactual Electricity Calculations

This section describes the calculation of the percent electricity savings under counterfactual productivity. Let s_{ij} denote the core-hour of VM j, used by firm i, and let $u_{ijt} \in [0, 1]$ denote the utilization at time t, which indicates a five-minute interval. Based on the relationship between utilization and electricity estimated in Section D.6, we assume that the relative power consumption related to CPU utilization is according to the following form:

$$p_{ijt} = (0.5 + 0.5u_{ijt})k_{ij}^{max}, \tag{22}$$

where k_{ij}^{max} represents the power consumption when VM j is utilized at 100%. This functional form assumes that when the VM is idle, the power consumption is 50% of maximum power consumption and then increases linearly as utilization increases.

We further assume that $k_{ij}^{max} = k \cdot c_{ij}$, where c_{ij} is the number of cores of VM j. This assumption is reasonable because the computation power required for a VM typically increases linearly with the number of cores. We further normalize $k = 1$ as we are interested in percent savings relative to observed baseline consumption.

The functional form in Equation (22) is particularly convenient because additivity is preserved under integration, meaning that the core hours of a VM and average utilization are sufficient statistics to calculate total power consumption. To see this, the power consumption of VM j during its duration is given by:

$$\begin{aligned} p_{ij} = \int p_{ijt}\, dt &= \int (0.5 + 0.5u_{ijt})c_{ij}\, dt = 0.5c_{ij}T_{ij} + 0.5c_{ij}\int u_{ijt}\, dt \\ &= 0.5(1 + \bar{u}_{ij})s_{ij} \end{aligned}$$

where $\bar{u}_{ij}$ is the average utilization of VM j, T_{ij} is its duration, and $s_{ij} = c_{ij}T_{ij}$ is the total core-hours of VM j. Furthermore, we can aggregate this at the firm level as follows:

$$\begin{aligned} p_i &= \sum_{j(i)} 0.5(1 + \bar{u}_{ij})s_{ij} = 0.5\sum_{j(i)} s_{ij} + 0.5\sum_{j(i)} \bar{u}_{ij}s_{ij} \\ &= 0.5s_i + 0.5\bar{u}_i s_i \end{aligned}$$

where $\bar{u}_i$ is the firm i's average utilization and $j(i)$ is the set of VMs used by firm i. This form suggests that a firm's total power requirement depends on the number of core-hours they use and the average utilization. This makes counterfactual power calculations challenging because the change in the average utilization will depend on whether the firm reaches counterfactual productivity by improving idleness or overprovisioning.

To make progress, we introduce additional notation to separate efficiency gains from changes in idleness and over-provisioning. Let $s_i^{id,c}$, $s_i^{op,c}$, and $s_i^{pp,c}$ denote the counterfactual idle, overprovisioned, and productive core-hours respectively, and $\Delta s_i^{id} = s_i^{id} - s_i^{id,c}$, similarly for other utilization types. We have that:

$$\frac{s_i^{pp} + 0.5s_i^{op}}{s_i} = \omega_i, \qquad \frac{s_i^{pp,c} + 0.5s_i^{op,c}}{s_i^c} = \omega_i^c. \tag{23}$$

Moreover,

$$s_i^{pp} + 0.5s_i^{op} = s_i^{pp,c} + 0.5s_i^{op,c} \tag{24}$$

since we require the counterfactual core-hours to produce the same observed output. This implies that:

$$\Delta s_i^{pp} = -0.5\Delta s_i^{op}.$$

Therefore, an X core-hours reduction in overprovisioned VMs should add $X/2$ core-hours of properly-provisioned VMs. Using Equations (23) and (24), we also obtain:

$$\frac{s_i^{pp} + 0.5 s_i^{op}}{s_i} = \omega_i, \qquad \frac{s_i^{pp,c} + 0.5\, s_i^{op,c}}{s_i - (0.5\,\Delta s_i^{op} + \Delta s_i^{id})} = \omega_i^c.$$

where the denominator in the second equation specifies the total core-hours used under the counterfactual productivity. This gives:

$$0.5\Delta s_i^{op} + \Delta s_i^{id} = s_i \left(\frac{\omega_i^c - \omega_i}{\omega_i^c} \right).$$

This provides an equation with two unknowns Δs_i^{op} and Δs_i^{id}, so it is not sufficient to identify Δs_i^{op} and Δs_i^{id} separately. We need another assumption to identify Δs_i^{op} and Δs_i^{id}. We make the following assumption:

$$\frac{s_i^{id} - \Delta s_i^{id}}{s_i^{id}} = \frac{s_i^{op} - \Delta s_i^{op}}{s_i^{op}} \qquad \Longrightarrow \qquad \frac{\Delta s_i^{id}}{s_i^{id}} = \frac{\Delta s_i^{op}}{s_i^{op}}. \tag{25}$$

The underlying idea behind this assumption is that core-hour savings from each mechanism are proportional to the initial inefficiencies from each mechanism. Without additional information on how firms would reach counterfactual productivity, this assumption seems reasonable and assumes that firms split efforts equally between different mechanisms. Now, under the assumption given in Equation (25), one can compute Δs_i^{id} and Δs_i^{op} as follows:

$$\Delta s_i^{id} = s_i \left(\frac{\omega_i^c - \omega_i}{\omega_i^c} \right) \left(\frac{s_i^{id}}{s_i^{id} + 0.5 s_i^{op}} \right), \qquad \Delta s_i^{op} = s_i \left(\frac{\omega_i^c - \omega_i}{\omega_i^c} \right) \left(\frac{s_i^{op}}{s_i^{id} + 0.5\, s_i^{op}} \right).$$

These equations provide the counterfactual values of idle, overprovisioned, and properly provisioned core-hours. We can calculate the average counterfactual utilization of firm i as follows:

$$\bar{u}_i^c = \bar{u}_i \frac{s_i}{s_i^c}.$$

Thus, we can calculate both factual and counterfactual firm-level power consumption:

$$p_i^c = 0.5 s_i^c + 0.5 \bar{u}_i^c s_i^c, \qquad p^c = \sum_i p_i^c.$$

The total power saving in the economy is given by:

$$\Delta p = \frac{p - p^c}{p}.$$

E Robustness Checks

This appendix provides the details of robustness checks summarized in Section 5.3.

E.1 Robustness to Other Utilization Measures

In cloud computing, network and memory utilization are commonly monitored alongside CPU utilization to measure the performance and efficiency of VMs.

Network utilization refers to the amount of data being transferred in and out of a VM relative to its capacity. High network utilization reflects significant data traffic, while low utilization indicates minimal use of the available bandwidth. Memory utilization measures the amount of allocated memory that a VM is actively using relative to the total memory allocated to that VM.

In calculating our measure of computing productivity, we focus on CPU utilization because it is the most relevant metric in the industry, and the CPU is the most resource-intensive component of computing infrastructure. CPUs typically consume the majority of power in servers, making their efficient use crucial for minimizing energy consumption. By concentrating on CPU utilization, we align our analysis with industry practices and address the most significant aspect of resource management in cloud environments.

Still, one potential concern is that firms with high CPU utilization efficiency might underperform in memory or network resource usage, creating inefficiencies in other VM performance dimensions. To address this, we conduct a robustness check to verify that memory or network utilization patterns do not contradict our CPU utilization findings.

We have limited data on memory and network utilization, covering one-month periods in 2022 and 2023. Using these data, we estimate the correlation between CPU utilization and other utilization measures. The direction of this correlation is unclear in advance. Some jobs may be memory-intensive, using more memory and less CPU, while others may be compute-intensive, relying more on CPU than memory. This variation could result in a negative correlation between these utilization measures. Conversely, if a job is truly idle, it would likely use neither memory nor CPU, potentially resulting in a positive correlation between the two measures.

In Figure OA-16, we report the correlation between utilization measures and find that CPU utilization is positively correlated with both network and memory utilization. This

result suggests that firms identified as inefficient in terms of CPU utilization also tend to be inefficient in other dimensions of computing resource utilization.

E.2 Robustness to Duration of Peak Utilization Measurement

As mentioned in Section 4.1 and detailed further in Appendices C.1 and D.2, we define productivity and inefficiency using peak VM utilization over a seven-day period. This approach ensures our measure remains conservative regarding potential costs associated with short-term provisioning adjustments, especially given the predictable volatility in load related to days of the week. Additionally, this definition aligns with the internal measures used by cloud providers.

However, one might be concerned about the sensitivity of our results to this choice. To address this, we re-estimate our productivity measures using alternative periods for calculating peak productivity: one-day, three-day, and 15-day intervals. We then repeat our analyses using these alternative productivity measures to verify that our primary findings are robust to these different definitions. Figures OA-11 and OA-14, as well as Table OA-8, present the outcomes of these robustness checks, which are broadly consistent with the results from our main specification.

E.3 Robustness to Controlling for VM Characteristics

As explained in Section 4.1, we first estimate the productivity of individual VMs based on their idleness and overprovisioning, and then aggregate these measures to the firm level at different frequencies. In our main specification, we treat all VMs equally, simply summing their VM-level efficiencies to the firm level. One potential concern with this approach is that VMs may differ in characteristics that could influence utilization. We believe focusing on peak utilization mitigates this concern, as peak utilization is less sensitive to VM type. For example, memory-intensive or network-intensive workloads naturally exhibit lower average utilization because the CPU may spend idle time waiting for data transfers; however, such factors should not significantly impact peak utilization. Nevertheless, we conduct several robustness checks to confirm that our findings are not driven by differences in VM characteristics.

To account for these differences, as described in Appendix D.2, we aggregate VM-day-level productivity to the firm level using a weighted regression that controls for several job characteristics, as follows:

$$\omega_{ijt} = \omega_{im} + \beta' Z_{jt} + \varepsilon_{ijt}.$$

This regression essentially estimates firm-month level fixed effects ω_{im} while controlling

for systematic productivity differences across VM types within various control bins.

Our first control includes day-of-week and holiday fixed effects to account for temporal variations in productivity differences. For example, if less productive firms, for unrelated reasons, tend to run jobs on weekends—and if weekend jobs are systematically less productive—this specification would accommodate such patterns. Our second control adds product fixed effects by interacting day-of-week and holiday fixed effects with VM configurations. We then gradually introduce additional controls, including machine fixed effects and region-by-machine fixed effects. These VM characteristics help account for potential differences arising from hardware specifications.

In these fixed-effect regressions, we can only compare firm fixed effects within a connected set (Abowd et al., 1999; Metcalfe et al., 2023). This means that firms must be directly or indirectly linked through the graph of firm-to-machine characteristics. In our context, due to the large number and variety of VMs used by firms, we either have all firms in one connected set or one large connected set covering more than 99% of firms, along with a few smaller sets consisting of firms that utilize only specialized VMs. Consequently, we can compare nearly all firms, even when controlling for detailed VM characteristics.

The results from these robustness checks are reported in Figure OA-10, Figure OA-13, and Table OA-7. The findings remain similar to our main specification, with the notable exception that controlling for VM characteristics reduces the magnitude of long-term learning in the cohort-by-cohort analysis. This reduction likely occurs because part of the observed learning arises from firms improving their choice of VM types, as documented in Section 7. Consequently, controlling for VM types captures this mechanism, reducing the estimated learning effect, particularly in the long run when VM types undergo significant changes over time.

E.4 Robustness to Compute Load Volatility Measures

One important identification threat in our analysis is that inefficiency might be rational if firms maintain idle capacity to handle volatile workloads, thus reducing the risk of hitting capacity constraints. Although we argue that, given the nature of cloud computing and available tools such as autoscaling, firms have no reason to maintain idle VMs, we nevertheless examine whether productivity measures correlate with key outcomes related to load volatility and capacity constraints.

To address this concern, we compute the following volatility measures: (1) standard deviation; (2) coefficient of variation; (3) fourth moment; (4) tail event type 1, defined as the probability of experiencing a load greater than mean + 2×standard deviation; (5) tail event type 2, defined as the probability of experiencing a load greater than mean +

3×standard deviation; and (6) tail event type 3, defined as the probability of experiencing a load greater than mean + 4×standard deviation. In these calculations, the load refers to the firm's daily compute load, which is estimated by integrating the area under the CPU utilization curve.

We then regress firm-month level compute productivity on these measures of time-varying firm-level demand volatility. This regression suggests that demand volatility measures explain only 1.8% of the variation in firm productivity across firms.

E.5 Robustness to Measurement of Downsizability

An important aspect of measuring compute productivity is the concept of downsizability: identifying alternative VMs that a firm could select if a VM is overprovisioned. When discussing downsizability in VMs, it is important first to establish the criteria for an appropriate substitute with fewer cores. A good substitute VM should maintain equivalent performance across all specifications, except for having a reduced number of CPU cores.

Several key factors should be considered when defining a substitute VM. These are primarily memory, VM family, operating system, region, and data center. For example, the memory capacity should remain the same or be higher to ensure that the job can run in the alternative VM. The operating system should also remain the same to maintain software compatibility. Another less clear dimension is the VM family. VMs are different in many dimensions, including family, manufacturer, and series. In principle, the same job can be run on different hardware versions and even on hardware from different manufacturers. However, firms might prefer to maintain the same VM family for consistency, performance predictability, and ease of management.

Another critical factor when considering VM downsizing is the geographical region or data center. The location of the data center might be important as firms tend to choose data centers close to their customers or employees to reduce latency (Greenstein and Fang, 2020). Moreover, firms might prefer to use a particular data center because their data is stored there. Finally, regulatory compliance and data sovereignty requirements can dictate the need for specific regional or data center locations.

Given these factors, we define various levels of downsizability. In each measure, we maintain the constraint that the alternative VM should have the same memory and operating system while allowing for variations in VM family and location.

- "Data Center-Machine Series-OS-Memory" Downsizing: This is the most restrictive measure, requiring VMs to be in the same data center, VM type, and series, with identical OS and memory.

- "Region-Machine Series-OS-Memory" Downsizing: This variation relaxes the data center requirement to the regional level while maintaining other restrictions.
- "Region-Machine Family-OS-Memory" Downsizing: This measure allows for different machine series within the same region, family, OS, and memory specifications.
- "Region-OS-Memory" Downsizing: This variation permits downsizing across different machine families within the same region, maintaining OS and memory consistency.
- "OS-Memory Downsizing": The least restrictive measure, allowing downsizing across different regions, only requiring the same OS and memory specifications.

In our baseline specification, we choose "Region-Machine Family-OS-Memory Downsizing" to balance the need for consistent performance with the flexibility of using different machine families within the same region. However, we also conduct robustness checks using other downsizability measures to ensure the robustness of our findings. The results for these robustness checks are given in Table OA-9.

E.6 Correlation Between Idleness and Overprovisioning Productivity

In this robustness check, we analyze the relationship between idleness and overprovisioning productivity. A positive relationship between these two productivity measures would suggest that inefficiency is not driven by a particular mechanism that generates only one type of inefficiency. For example, one explanation for inefficiency could be that a firm's workload is volatile, so firms overprovision to ensure they can meet additional demand. However, this explanation is less likely to account for idleness because an efficient firm could easily manage volatility across VMs using available tools.

For this robustness check, we regress overprovisioning productivity on idleness productivity using firm-month-level data, controlling for industry and time fixed effects. This regression yields a coefficient of 0.064 with a standard error (clustered at the firm level) of 0.004, suggesting that firms with idle VMs also tend to have overprovisioned VMs. This result indicates that the underlying factors driving inefficiency in computing simultaneously lead to both idleness and overprovisioning, providing evidence that inefficiency does not originate from a single source.

E.7 External Validity: Relationship Between Productivity and Firm Exit

There is an extensive literature demonstrating that less productive firms are more likely to exit (Foster et al., 2016). Although we do not observe actual firm exits, we can conduct a

similar analysis by examining the relationship between a firm's compute productivity and its probability of leaving the cloud. It is important to note that our measure of exit captures only when a firm leaves our specific cloud provider and does not necessarily indicate that the firm stops using cloud services entirely, as it might switch to another provider. Nevertheless, this interpretation remains consistent with the intent of our analysis: firms with lower productivity at our cloud provider are more likely to exit that provider.

This result also provides evidence against the demand volatility hypothesis for productivity dispersion. Cloud computing is particularly advantageous for firms that experience significant fluctuations in demand. If demand volatility were the primary driver behind low productivity, we would expect that low-productivity firms, which benefit most from the cloud's flexibility, would be less likely to exit.

E.8 External Validity: Dispersion in Publicly Available CPU Utilization Data

We use publicly available CPU utilization datasets, described in Appendix B, to assess the external validity of our findings. These datasets consist of short panels and lack user-specific information, limiting our analysis to only productivity dispersion.

The datasets, sourced from Google Cloud and Microsoft Azure, differ from our primary data in both structure and duration. Therefore, we provide detailed explanations of how we construct the samples used for productivity analysis in each dataset. In processing both traces, we either directly apply the assumptions used in cleaning our primary data or select observations that closely resemble the operational focus of our analysis.

E.8.1 Azure Cloud Trace

The 2019 Azure dataset provides one month of VM utilization data and characteristics. Regarding utilization, we observe average and maximum CPU utilization recorded at five-minute intervals throughout the VM's lifetime. For VM characteristics, we have data on the number of requested cores and amount of memory, as well as timestamps indicating when each VM was created and deleted. Additionally, the dataset includes a "machine category" variable that describes whether a VM is "delay insensitive," "interactive," or "unknown."

As expected, we observe core and memory levels that primarily scale by factors of two. For cores, we observe request levels of 2, 4, 8, 16, and 30. For memory, we observe request levels of 2, 4, 8, 32, 64, and 70 gigabytes. As in the main text, we define downsizeability in terms of cores for a given set of VM characteristics. Specifically, given a VM's memory request level and machine category, the VM is considered downsizeable if there exists another VM with the same memory and category but with half the number of cores. This

definition differs slightly from our main downsizeability definition, as we lack certain VM features in the Azure data, such as VM series and data center regions. Additionally, consistent with our main analysis, we measure productivity based on the 95th percentile of maximum CPU utilization.

Before computing productivity, we clean the trace data using the same sampling procedures as those applied to our primary dataset. Specifically, we drop all users who are observed with fewer than 10 VMs throughout the trace period, resulting in a loss of 2.8% of users. Additionally, we remove VMs with lifetimes of less than 20 minutes, which eliminates approximately 39% of the available VMs in the data. The final sample contains resource usage information from 3,503 users operating a total of 1,632,952 VMs.

Next, for VM j of user i on day t, we define a productivity measure $\omega_{ijt} \in [0, 1]$, consistent with the definition provided in Section 4. Using this measure, we aggregate VM-level productivity to obtain user-day-level productivity. Specifically, the productivity for user i on day t across the set of VMs J_{it} is defined as follows:

$$\omega_{it} = \frac{\sum_{j \in J_{it}} \omega_{ijt} ch_{ijt}}{\sum_{j \in J_{it}} ch_{ijt}}, \tag{26}$$

where ch_{ijt} is the core-hours of job j on day t. Unfortunately, the Azure trace only contains timestamps in seconds, and it is unknown when the trace began. Hence, we have no mapping between timestamps and particular dates. Thus, we determine the days where observations fall to be the modulus of the timestamp divided by the number of seconds in a day.

Appendix Figure OA-17(a) shows the distribution of user-day-level productivity throughout the duration of the Azure trace. There is a notable concentration at the productivity level of one-half, primarily due to the relatively small sample size in the data. Nonetheless, both the dispersion and the overall distribution of productivity are broadly similar to our main results shown in Figure 3.

E.8.2 Google Cluster Trace

Similar to the Azure trace, the cluster data provided by GC traces one month of CPU usage and characteristics from May 2019. The Google cluster trace concerns cluster usage by jobs submitted by Google engineers and services (see Appendix Section B.6). We utilize the detailed information in the trace provided by GC to focus on jobs that resemble the structure of VMs as closely as possible.

As described in Appendix Section B.6, the task is the fundamental unit of observation

in the usage data in this trace. Tasks are executed as instances of jobs that either run independently and directly on a physical machine or as part of an *alloc set*, which represents sets of fixed resources. In the GC trace, jobs and alloc sets are referred to as *collections*. For each collection, we observe whether auto-scaling is enabled and, if so, whether it is constrained. For this data, we use collections without auto-scaling enabled as our VM-like objects. Henceforth, we will refer to these collections as VMs. We focus on these VMs because the provisioning decisions for CPU and memory are made by the user rather than the cluster manager. In this way, this subset of collections helps us focus on compute resources most similar to our sample, enabling us to concentrate on user-made provisioning decisions.

The GC trace contains observations of nearly 5.2 million collections. Only for about 360,000 (6.8%) of these, the requested CPU capacity is fixed. Since these VMs run on a cluster, they start using compute resources when scheduled on a machine. Thus, we consider the lifetime of the VM as beginning at its scheduled time. We remove all collections that do not have an explicit scheduling event. This preserves 99.5% of the collections. We also know exactly when the trace started and ended, so we do not encounter the timestamp-to-date conversion difficulties that are present with the Azure trace.

Some VMs have multiple scheduling events observed. To the best of our knowledge, these cases correspond either to VMs that failed and were restarted or to VMs that were booted from a VM due to a higher-priority VM needing to be scheduled. For these kinds of VMs, we take the minimum scheduling event observed as the start time of the VM. As in the main text, we remove VMs with a lifetime of less than 20 minutes and all users with less than 10 VMs. This results in a sample of 24,909 VMs. Although this is a small subset of all available collections, it was created to ensure that the VM-like objects we analyze closely resemble the VMs in our data.

In our data and the Azure trace, we observe core and memory requests that scale by a factor of two. This is not the case in this trace. Since our VMs are cluster jobs, they can request memory in terms of bytes rather than just gigabytes, and therefore, requests can differ at a much more granular level. Given the availability of provisioning resources at such a granular level, users can provision VMs efficiently at virtually any level of compute usage. Thus, we consider all VMs to be downsizeable, and hence, the productivity of a given VM will be solely based on its CPU utilization.

As in the main text, we define the productivity of the VM j of user i on the day t by the measure ω_{ijt}. In this case, ω_{ijt} is zero for VMs with CPU utilization under 10%, one-half for VMs with between 10% and 45% utilization, and one for VMs with greater than 45% utilization. As normal, we consider the 95th percentile of maximum CPU utilization. With

this, we aggregate VM-level productivity to the user-day level using a similar notation as above:

$$\omega_{it} = \frac{\sum_{j \in J_{it}} \omega_{ijt} ch_{ijt}}{\sum_{j \in J_{it}} ch_{ijt}}. \tag{27}$$

Figure OA-17(b) presents the distribution of productivity calculated in Google Cloud. A few points to note: First, similar to Azure traces, we observe a mass at 0.5, coming from users with only over-provisioned resources. Second, we see that the productivity distribution is more skewed to the right than our main result in Figure 3. This is likely due to the fact that Google Cloud data only includes internal users or jobs of external customers implemented by Google engineers. It is likely that Google engineers are more productive than those in typical firms. However, despite this, we still observe substantial productivity dispersion in this sample, where productivity ranges from 0 to 1, with a substantial mass concentrated between 0.5 and 1.

E.9 Long-Term Learning

Our primary dataset contains one consecutive year of data, from mid-2022 to mid-2023. Using these data, we characterize the productivity improvements of different cohorts over that year in Section 7. By comparing the productivity growth of different cohorts over that year, we can estimate long-term learning in a way that minimizes the amount of cross-cohort variation in our estimates. However, estimating learning over a period longer than a year using one year of data necessarily requires comparing the productivity growth of firms in different cohorts.

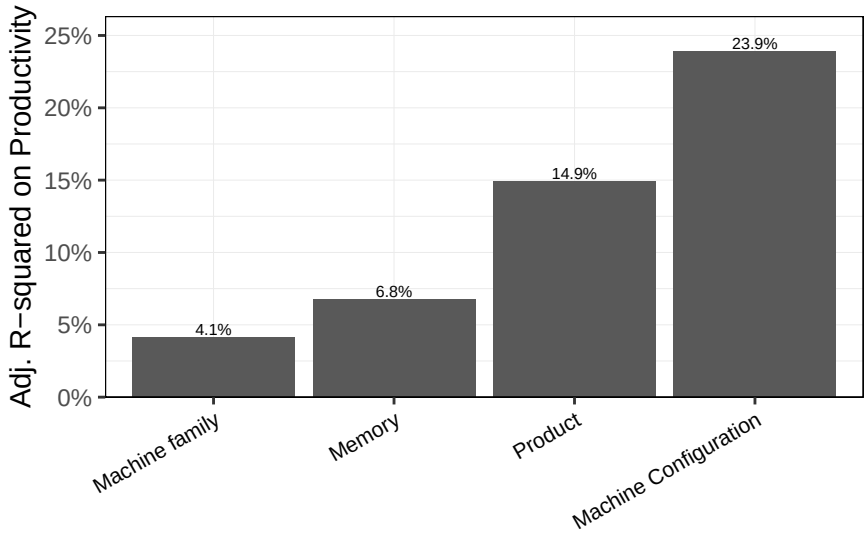
To address this problem, we can utilize our limited data from 2017 to 2019 to estimate the productivity growth of a single cohort of firms over a longer period. To do so, we require sufficient data for one period to reliably estimate the productivity of a group of firms in their initial month. Given the intermittency and timing of our data, the best opportunity to do so is in 2019, when we have a consecutive month of data that is at a similar time of year to the data at the end of our sample.

We find that firms that joined the cloud in mid-2019 have a 27.6% higher productivity at the end of our sample roughly four years later. While still providing clear evidence of learning, this estimate is lower than the 44.0% long-term productivity improvement that we find in our cohort analysis. This difference could be in part due to survivorship bias: we are able to condition on firms in the 2019 cohort we study surviving for four years, whereas we can only condition on the June-July 2022 cohort surviving for one year, which—since exit is negatively correlated with productivity—implies a higher base rate

of productivity for the 2019 cohort. Another reason could be differences across cohorts: firms that are initially more productive may join the cloud earlier, which reduces the scope of learning. Consistent with both interpretations, we find that the difference in estimates is driven in large part by differences in initial productivity: firms in the 2019 cohort that survive for four years have a 13.8% higher productivity in their first month than firms in the June-July 2022 cohort that survive for one year in their first month. Due to the sampling procedure applied to the raw data, we are unable to compare the unconditional distributions of firms in their first months across cohorts, which would help indicate which effect is more prevalent in this setting.

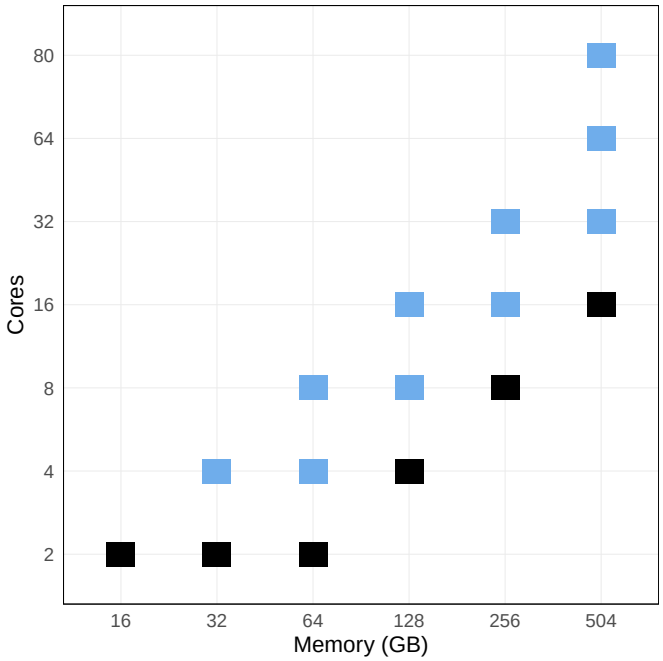
F Additional Figures

Figure OA-2: Explanatory Power of VM Characteristics in Productivity Dispersion



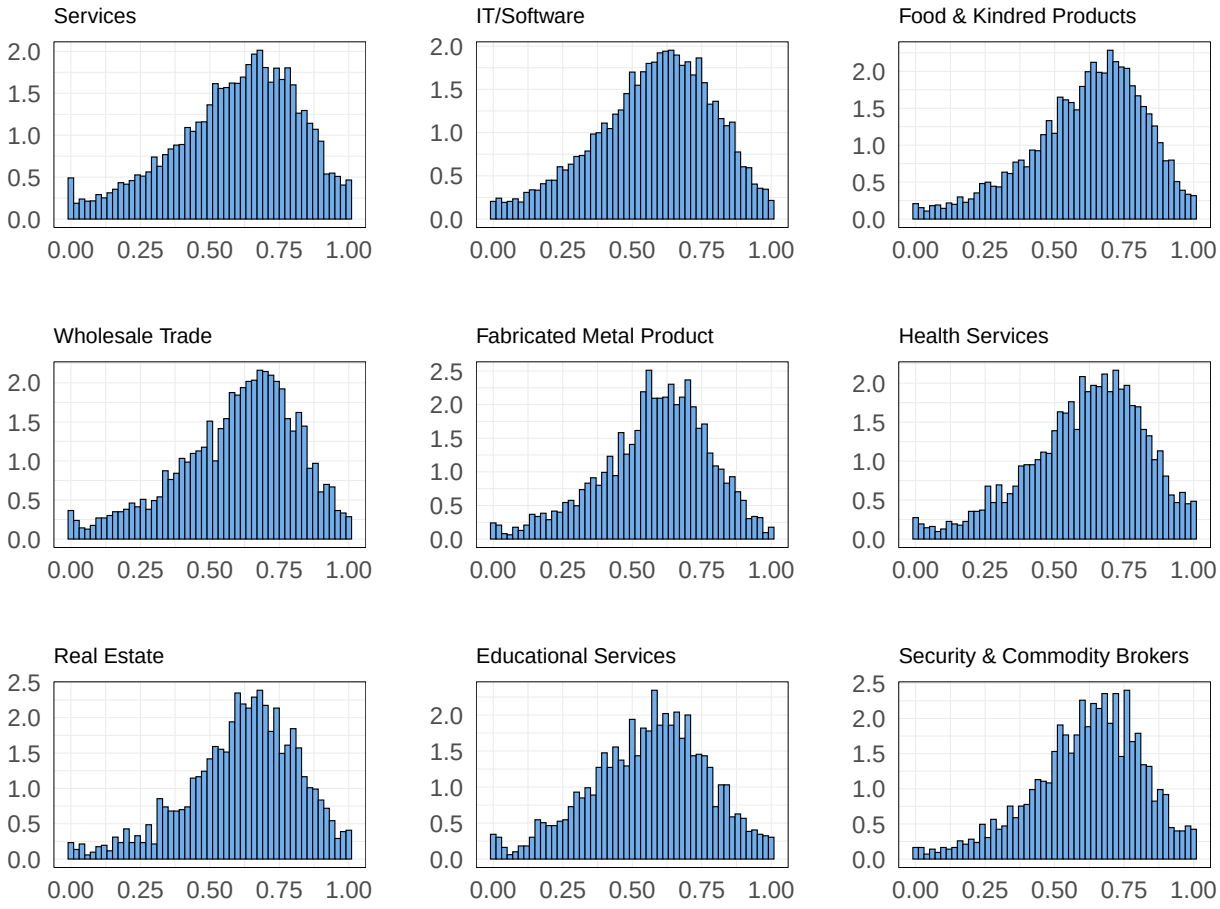
Notes: This figure displays adjusted R^2 from a regression of VM-day level productivity on increasingly detailed levels of fixed effects. Fixed effects included in bars to the right always nest the fixed effects used in preceding (left) bars.

Figure OA-3: Downsizability Example: Memory and Core Combinations



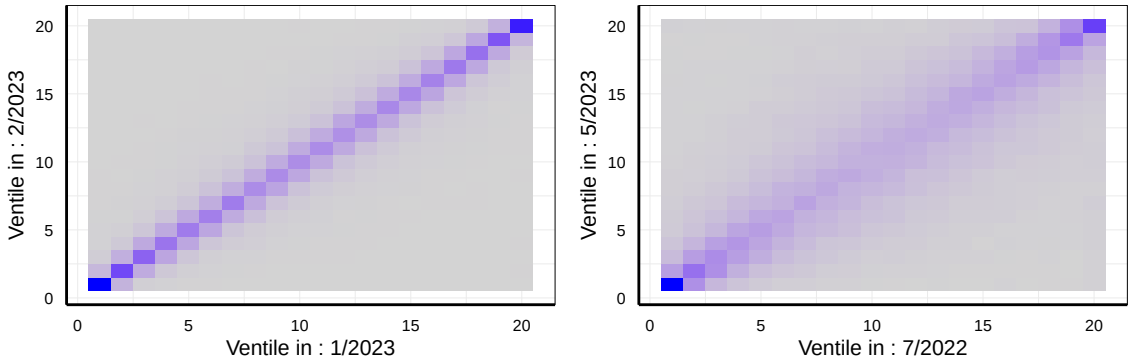
Notes: This figure displays all available combinations of memory (GB) and cores for one VM series from our data. Each point represents a specific VM configuration. The blue-colored points indicate downsizable machines, where an alternative VM exists with the same memory capacity but fewer cores.

Figure OA-4: Productivity Dispersion by Industry



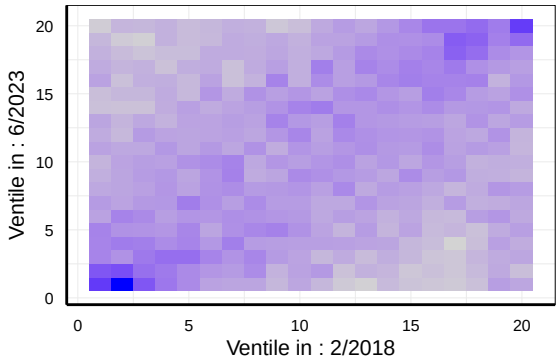
Notes: This figure displays the productivity distribution by industry, calculated in the same way as in Figure 3. Industry classification is based on 1-digit SIC codes.

Figure OA-5: Persistence of Productivity in the Short, Medium, and Long Run



(a) 1-Month Persistence

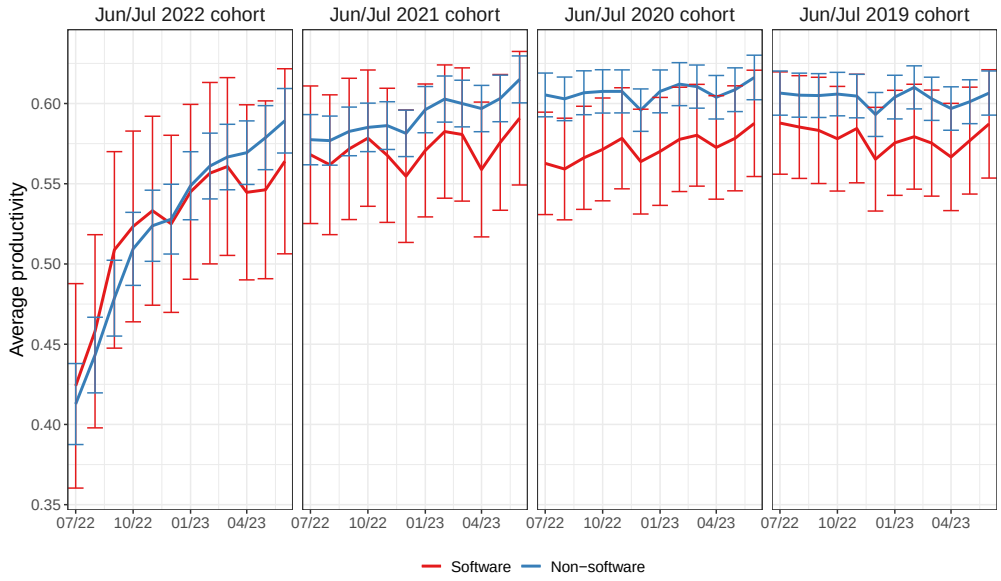
(b) 1-Year Persistence



(c) 5-Year Persistence

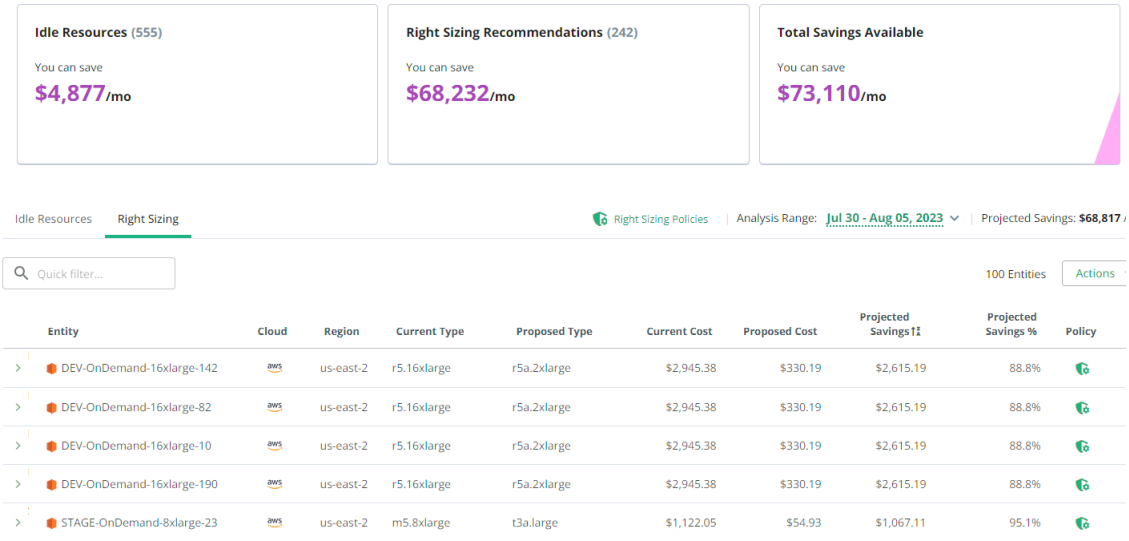
Notes: This figure presents heatmaps illustrating productivity persistence across three different time horizons: (a) 1-month, (b) 1-year, and (c) 5-year. Each heatmap's axes are divided into 20 equally sized bins, representing the ventiles of the productivity distribution. The x-axis shows the ventile at the start of the period, while the y-axis shows the ventile at the end of the period. Each cell's color intensity corresponds to the frequency of firms moving from one ventile to another over the specified time horizon. Panel (a) depicts 1-month persistence from January 2023 to February 2023, panel (b) shows 1-year persistence from July 2022 to June 2023, and panel (c) illustrates 5-year persistence from 2018 to June 2023.

Figure OA-6: Changes in Software and Non-Software Firm Compute Productivity Over Time by Cohort

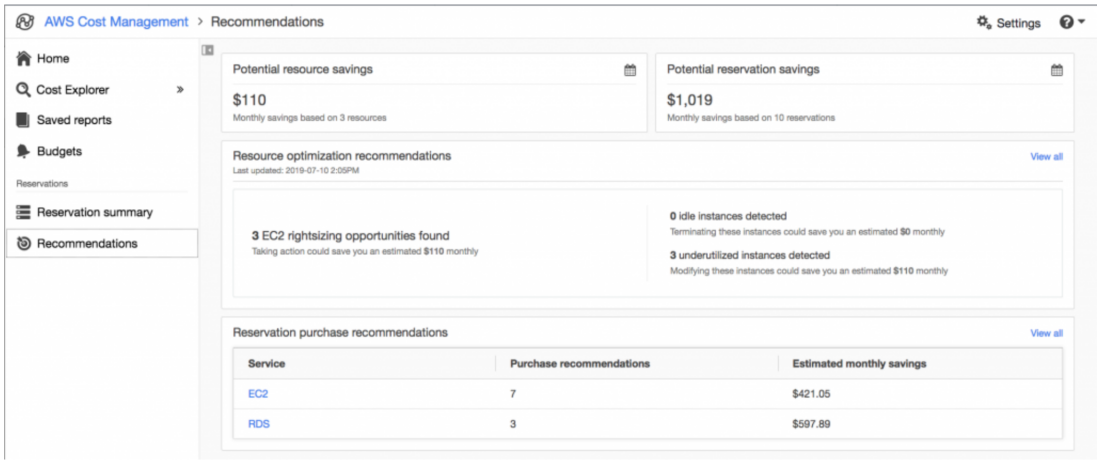


Notes: This figure displays monthly average productivity estimates for four different cohorts of firms with different levels of experience (0-1, 1-2, 2-3, and 3-4 years) during the period between July 2022 and June 2023. The average productivity for the June-July 2022 cohort in July 2022 is normalized to 1. To be included in the analysis, a firm must have had nonzero usage every month from July 2022 to June 2023. Firms are separated based on whether their 1-digit SIC code is in the Software/IT category. Error bars indicate the 95% confidence intervals, with standard errors clustered by firm. The details of the estimation can be found in Appendix D.4.

Figure OA-7: Examples of First- and Third-Party Cost Management Tools



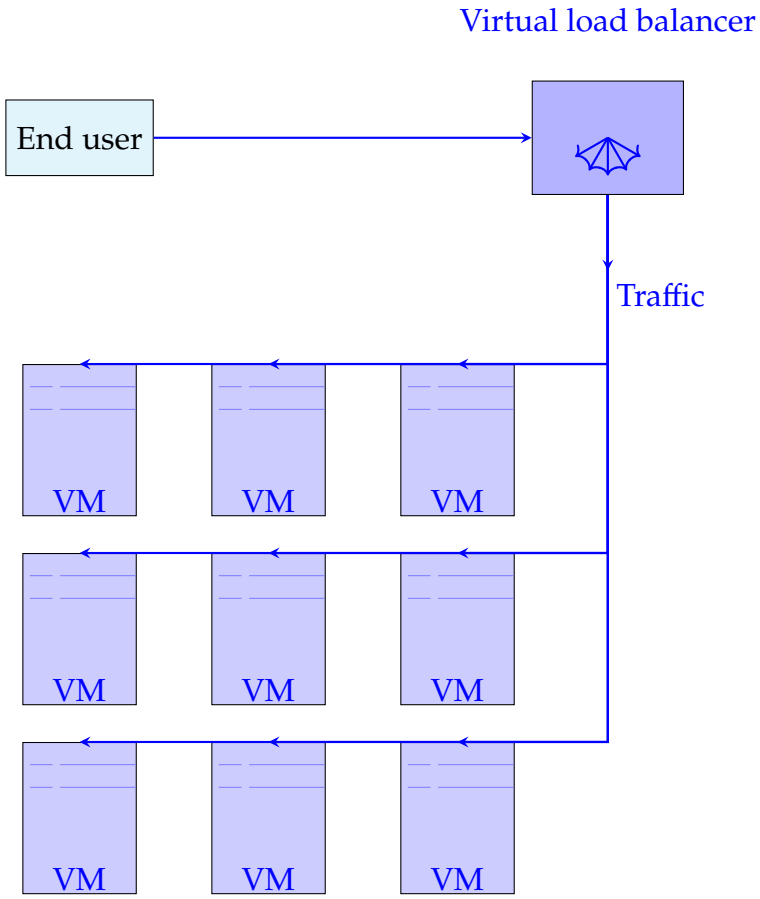
(a) Virtana Cost Management Tool



(b) AWS

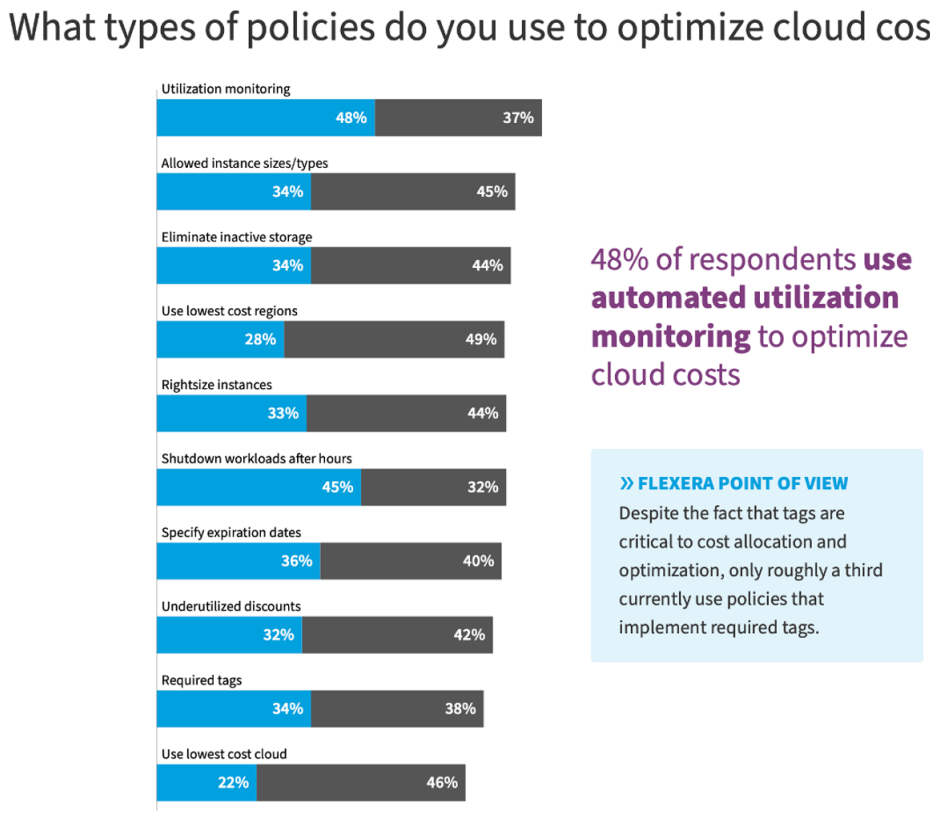
Notes: This figure presents two examples of tools for detecting idleness and overprovisioning in cloud computing. Panel (a) shows a dashboard from cloud optimization startup Virtana, taken from Virtana Cloud Cost Management while Panel (b) displays the cost management interface of AWS, obtained from AWS Resource Optimization Recommendations.

Figure OA-8: Representation of Load Balancer

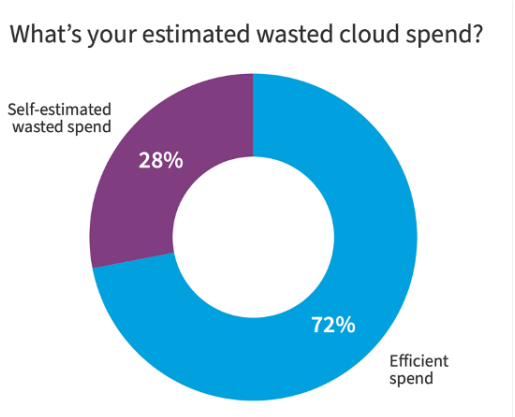


Notes: This figure illustrates a virtual load-balancing architecture in cloud computing. It depicts the flow of traffic from end users through a virtual load balancer, which then distributes the requests across multiple VMs. The load balancer directs traffic (indicated by a blue arrow) to three rows of VMs, each row containing three VMs. This architecture is designed to optimize resource utilization and improve system performance by efficiently distributing incoming requests across available compute resources.

Figure OA-9: Surveys About Cloud Optimization



(a) Survey Response to a Question about Cloud Optimization Tools



(b) Survey Response to a Question about Spending

Notes: This figure is a screenshot from a survey conducted by Flexera titled "State of the Cloud" (Flexera, 2023). It shows the responses to two questions asked in the survey. Panel (a) illustrates the types of policies companies use to optimize cloud costs, while Panel (b) displays the respondents' estimates of their wasted cloud spend.

G Additional Tables

Table OA-2: List of Machine Families Offered by Major Cloud Providers

VM Family	AWS	Azure	GCP
General Purpose	M5, T3	B-series, Dsv3-series	E2, N1, N2, N2D
Compute Optimized	C5	Fsv2-series	C2
Memory Optimized	R5, X1	Esv3-series, Mv2-series	M1, M2
Storage Optimized	I3, D2	Lsv2-series	-
GPU	P3, G4	NC-series, NV-series	A2
High Performance	-	H-series	-
Shared-core	-	-	f1-micro, g1-small

Notes: This table summarizes the main families of virtual machines offered by Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). The categories are general and may not be exhaustive. Each provider offers multiple sizes and series within each family. "-" indicates that the provider doesn't have a direct equivalent or the information wasn't specified in the given context.

Table OA-3: Advertised Use Cases of Different Machine Families

VM Family	**Key Considerations**	**Ideal For**
General Purpose	Cost-effective, balanced CPU, memory, temporary storage	Web servers, application servers, development environments, small to medium databases
Compute Optimized	High core counts, faster CPUs	Scientific computing, HPC, video editing, simulations
Memory Optimized	Large RAM capacities	Databases, caching layers, in-memory analytics
Storage Optimized	Local SSDs, high I/O performance	Large databases, data warehousing, Big Data analytics, real-time applications
GPU	Diverse GPU types and configurations	Machine learning, deep learning, video editing, scientific simulations
High-Performance	Exceptionally high compute power, massive memory, ultra-fast storage	Scientific modeling, simulations, weather forecasting

Source: www.cloudoptimo.com

Table OA-4: First-Party Cloud Cost Optimization Tools

Cloud Provider	Cost Optimization Tool	Description
AWS	AWS Cost Explorer	Interface to view costs, usage, and ROI for AWS services, with data for the past 13 months and forecasting capabilities.
	AWS Budgets	allows setting and enforcing budgets for AWS services, with notifications when budgets are exceeded or reached.
	AWS Trusted Advisor	provides automated recommendations for cost optimization, including EC2 reserved instance optimization and idle resource identification.
	Amazon CloudWatch	Monitoring service that can set alarms based on metrics, commonly used for cost optimization by identifying underutilized resources.
	AWS Instance Scheduler	Automates starting and stopping of EC2 and RDS instances based on defined schedules to save costs.
	AWS Pricing Calculator	Estimates the cost of use cases on AWS, helping to model solutions and explore pricing points before deployment.
Azure	Azure Cost Management and Billing	Provides cost analysis, budgeting, and recommendations for cost optimization, integrated with the Azure portal.
	Azure Advisor	Offers personalized best practices and recommendations to optimize Azure resources, including cost optimization.
	Azure Pricing Calculator	helps estimate costs for Azure services and solutions, allowing users to model and forecast expenses before deployment.
GCP	Google Cloud Cost Management	Includes tools for cost visibility, budgeting, and recommendations to optimize cloud spending.
	Google Cloud Pricing Calculator	Estimates costs for Google Cloud services, allowing users to model and forecast expenses before deployment.
	Google Cloud Recommender	Provides recommendations for cost optimization, including rightsizing VM instances and identifying idle resources.
	Google Cloud Budgets and Alerts	allows setting budgets and receiving alerts when costs exceed predefined thresholds, integrated with Google Cloud Console.

Notes: This table summarizes the main cost optimization tools offered by Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Table OA-5: Relationship Between Firm Centralization and Compute Productivity

Indep. var.	(1)	(2)
HHI (0 to 1)	0.102	
	(0.007)	
HHI quartile 2		0.016
		(0.004)
HHI quartile 3		0.031
		(0.004)
HHI quartile 4		0.069
		(0.005)

Notes: The coefficients in this table come from a regression where the dependent variable is productivity in 2022-2023, and each observation is a multi-unit firm. In regression (1), the independent variable is the firm's HHI of usage across all units, while in regression (2), there are independent dummy variables for four equally sized groups based on the firm's unit usage HHI (with the lowest HHI the omitted group). Both regressions control for industry (2-digit SIC) and cohort year fixed effects. Standard errors clustered by industry and cohort year in parentheses.

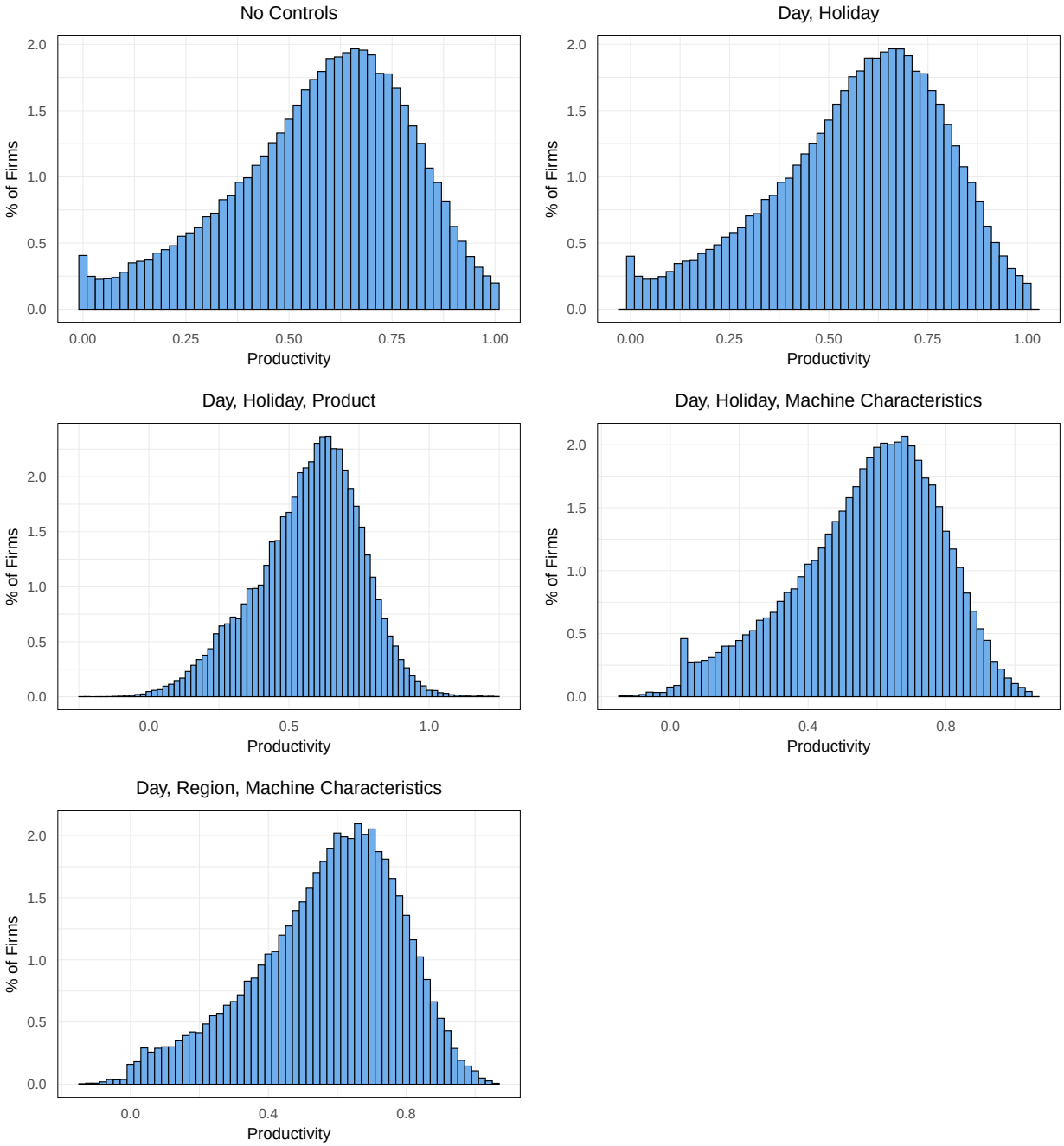
Table OA-6: Propensity to Use Different VM Series by Productivity Level

	No controls		With controls	
Dependent variable	prod >median	% of mean	prod >median	% of mean
HHI of usage across VM series	-0.094	-15.8	-0.067	-11.2
	(0.002)		(0.002)	
1(all usage on one VM series)	-0.118	-54.3	-0.092	-42.5
	(0.002)		(0.002)	
Number of VM series used	1.260	28.5	0.764	17.3
	(0.029)		(0.026)	

Notes: This table displays the estimate of coefficient β from the regression shown in Equation (3), along with the ratio between the coefficient and the mean of the dependent variable. The left set of columns includes the raw difference between the groups, while the right set controls for cloud adoption quarter fixed effects, industry (2-digit SIC code) fixed effects, region fixed effects, and firm size quartile fixed effects. Firms are classified as above- and below-median using productivity estimates from the 2022 sample, while the regressions are estimated using the 2023 sample. Standard errors clustered at the firm level are in parentheses.

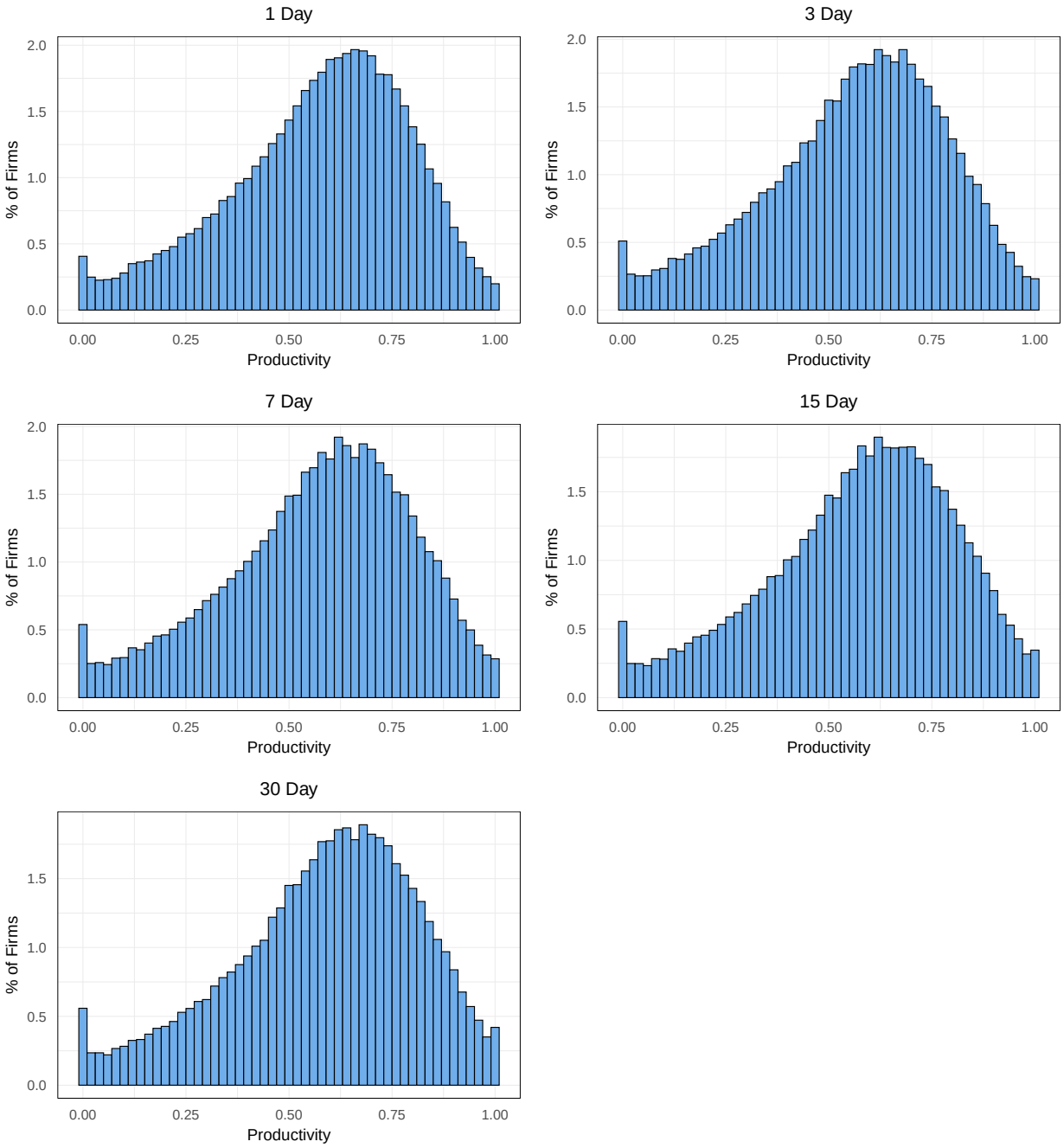
H Robustness Results

Figure OA-10: Productivity Distribution Under Different Controls



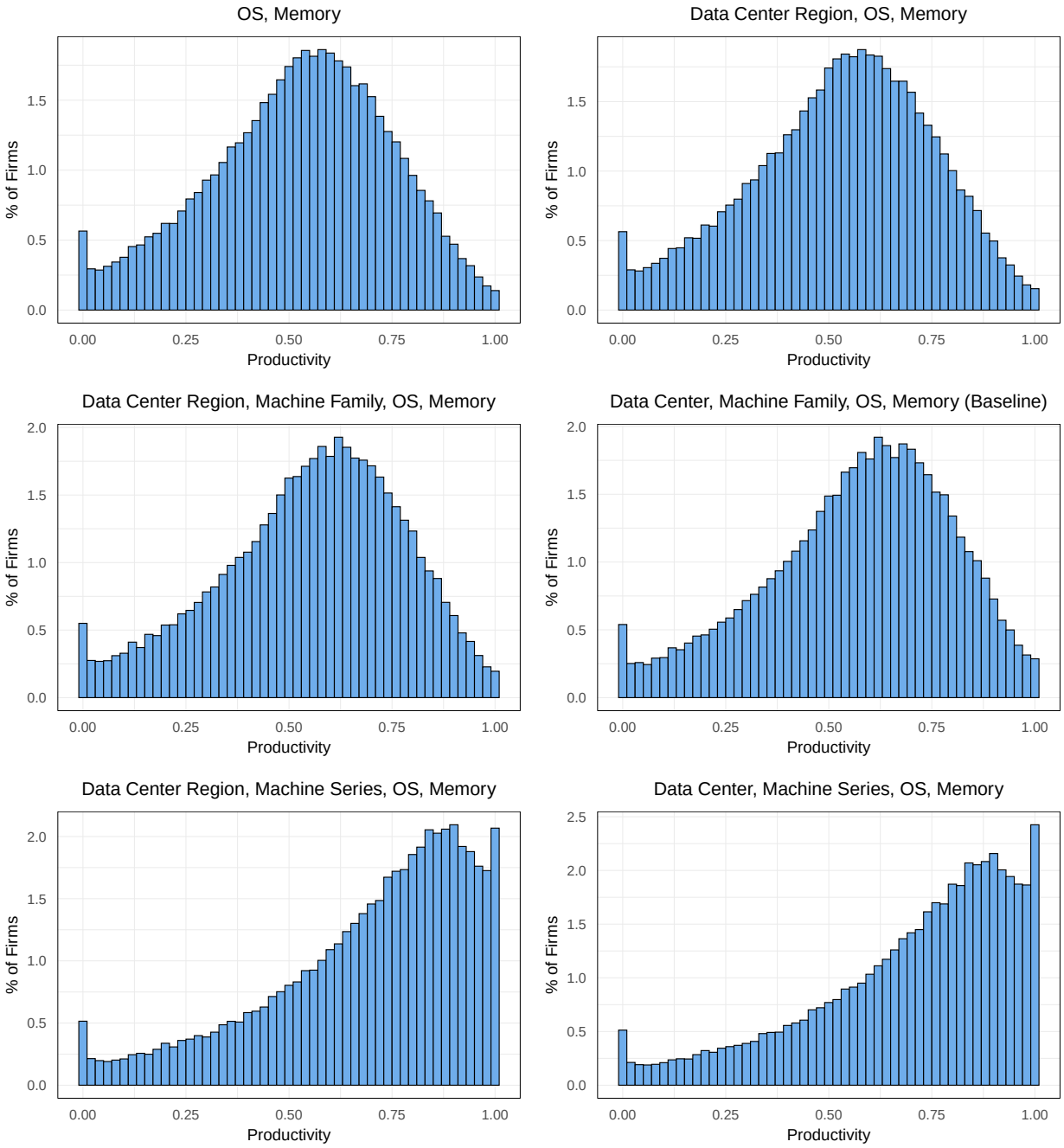
Notes: This figure presents the distribution of firm-level productivity estimates using productivity measures that control for different machine characteristics, as detailed in Section E.3. The histograms show the dispersion in productivity under various control specifications, including the day of the week, holiday, product ID, and machine family.

Figure OA-11: Productivity Distribution Under Different Peak Utilization Durations



Notes: This figure presents the distribution of firm-level productivity estimates using productivity measures that control for different days of measurement of peak utilization as detailed in Appendix E.2.

Figure OA-12: Productivity Distribution Under Different Downsizability Definitions



Notes: This figure presents the distribution of firm-level productivity estimates using productivity measures that use different downsizability definitions as detailed in Appendix E.5.

Table OA-7: Productivity Dispersion and Persistence Under Different Controls

	No Controls	Day Holiday	Day, Holiday, Product	Day, Holiday, Machine Char.	Day, Region, Machine Char.
	(1)	(2)	(3)	(4)	(5)
			Panel A. Dispersion		
Dispersion:					
Mean	0.60	0.60	0.58	0.59	0.59
Median	0.62	0.62	0.60	0.61	0.62
10-90th perc ratio	3.51	3.51	3.08	3.41	3.39
Interquartile Ratio	1.72	1.72	1.65	1.71	1.70
Within-Firm:					
Between-firm	33.08	33.10	30.00	33.12	33.10
Within-firm	66.92	66.90	70.00	66.88	66.90
Within-Firm-Between-Region:					
Between-region	5.88	5.89	5.70	5.82	5.89
Within-region	94.12	94.11	94.30	94.18	94.11
			Panel B. Persistence (AR(1) Coefficients)		
1-month persistence:					
Productivity	0.93	0.93	0.92	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.93	0.93	0.92	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.91	0.91	0.90	0.91	0.91
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
1-year persistence:					
Productivity	0.64	0.64	0.59	0.64	0.64
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.66	0.66	0.61	0.66	0.66
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.60	0.60	0.55	0.58	0.58
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
5-year persistence:					
Productivity	0.32	0.32	0.26	0.34	0.34
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.33	0.33	0.24	0.33	0.33
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.10	0.10	0.14	0.12	0.12
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)

Notes: This table reports the dispersion and persistence of productivity measures across different specifications that differ by the control variables included in Equation (16). Panel A presents the dispersion of compute productivity. Panel B shows the persistence of productivity, idleness, and overprovisioning measures with 1-month, 3-month, and 5-month autoregressive (AR(1)) coefficients, including their standard errors in parentheses. The control variables in each column are (2) day-of-week and holiday fixed effects, (3) day-of-week, holiday, and product ID fixed effects, (4) day-of-week, holiday, and machine family fixed effects, (5) day-of-week, holiday, data center region, and machine family fixed effects.

Table OA-8: Productivity Dispersion and Persistence Under Different Peak Utilization Durations

	1 Day (1)	3 Days (2)	7 Days (3)	15 Days (4)	30 Days (5)
			Panel A. Dispersion		
Dispersion:					
Mean	0.60	0.59	0.60	0.60	0.61
Median	0.62	0.61	0.62	0.62	0.63
10-90th perc ratio	3.23	3.47	3.51	3.48	3.42
Interquartile Ratio	1.68	1.72	1.72	1.71	1.69
Within-Firm:					
Between-firm	33.46	33.08	33.08	32.96	32.93
Within-firm	66.54	66.92	66.92	67.04	67.07
Within-Firm-Between-Region:					
Between-region	5.98	5.90	5.88	5.84	5.79
Within-region	94.02	94.10	94.12	94.16	94.21
			Panel B. Persistence (AR(1) Coefficients)		
1-month persistence:					
Productivity	0.94	0.94	0.93	0.93	0.94
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.94	0.94	0.93	0.93	0.94
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.91	0.92	0.91	0.91	0.92
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
1-year persistence:					
Productivity	0.66	0.65	0.64	0.64	0.64
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.68	0.67	0.66	0.65	0.65
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.61	0.62	0.60	0.59	0.58
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
5-year persistence:					
Productivity	0.32	0.32	0.32	0.31	0.31
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.34	0.34	0.33	0.33	0.32
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.10	0.11	0.10	0.10	0.10
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)

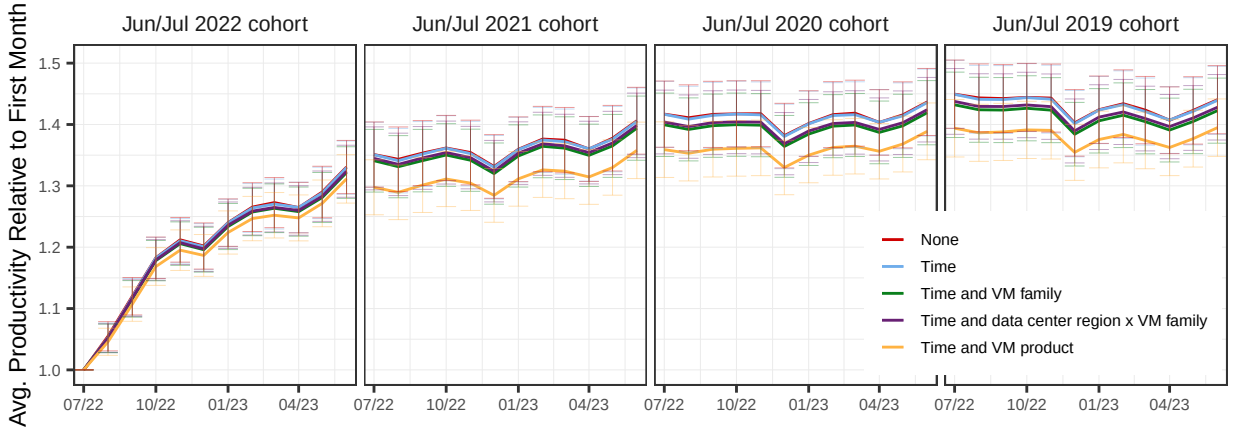
Notes: This table reports the dispersion and persistence of productivity using productivity measures that control for different days of measurement of peak utilization as detailed in Appendix E.2. Panel A presents the dispersion of compute productivity across different productivity measures. Panel B shows the persistence of productivity, idleness, and overprovisioning measures with 1-month, 3-month, and 5-month autoregressive (AR(1)) coefficients, including their standard errors in parentheses.

Table OA-9: Productivity Dispersion and Persistence Under Different Downsizability Definitions

	OS, mem	Region, OS, mem	Region, family, OS, mem	DC, family, OS, mem	Region, series, OS, mem	DC, series, OS, mem
	(1)	(2)	(3)	(4)	(5)	(6)
			Panel A. Dispersion			
Dispersion:						
Mean	0.55	0.56	0.58	0.60	0.71	0.72
Median	0.56	0.57	0.60	0.62	0.77	0.78
10-90th perc ratio	3.76	3.70	3.56	3.51	3.06	3.01
Inter Quartile Range	1.82	1.81	1.75	1.72	1.64	1.63
Within-Firm:						
Between-firm	33.87	33.98	33.31	33.08	32.34	32.23
Within-firm	66.13	66.02	66.69	66.92	67.66	67.77
Within-Firm-Between-Region:						
Between-region	5.76	5.76	5.71	5.88	5.42	5.45
Within-region	94.24	94.24	94.29	94.12	94.58	94.55
			Panel B. Persistence (AR(1) Coefficients)			
1-month persistence:						
Productivity	0.93	0.93	0.93	0.93	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.93	0.93	0.93	0.93	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.90	0.90	0.91	0.91	0.93	0.93
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
1-year persistence:						
Productivity	0.65	0.65	0.65	0.64	0.64	0.65
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.66	0.66	0.66	0.66	0.66	0.66
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.60	0.61	0.61	0.60	0.58	0.58
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
5-year persistence:						
Productivity	0.32	0.32	0.32	0.32	0.32	0.32
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Idleness	0.33	0.33	0.33	0.33	0.33	0.33
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Overprovisioning	0.11	0.10	0.09	0.10	0.16	0.19
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)

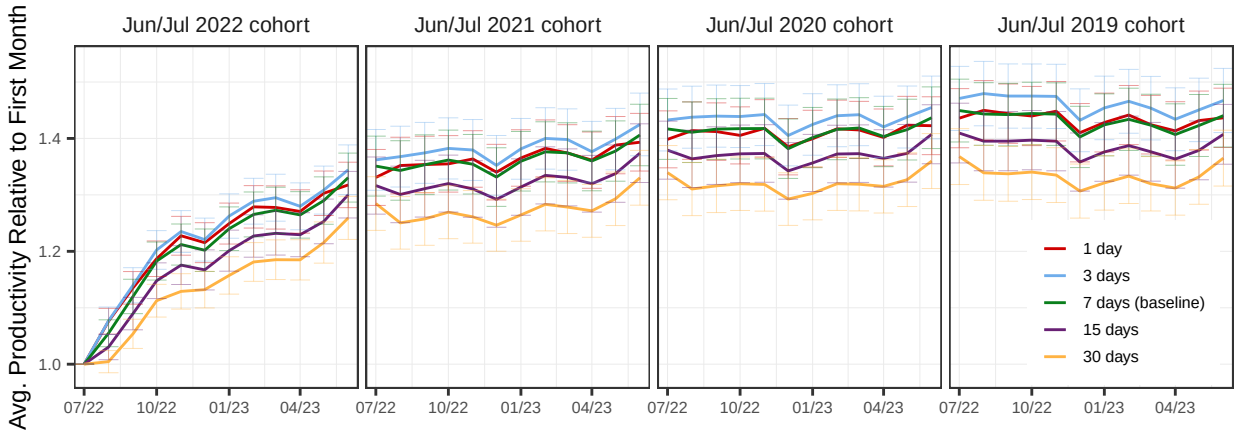
Notes: This table reports the dispersion and persistence of productivity measures under different downsizability definitions given in Appendix E.5. Panel A presents the dispersion of compute productivity across different productivity measures. Panel B shows the persistence of productivity, idleness, and overprovisioning measures with 1-month, 3-month, and 5-month autoregressive (AR(1)) coefficients, including their standard errors in parentheses. The columns represent different downsizability definitions: (1) "OS, mem" for OS and memory; (2) "Region, OS, mem" for data center region, OS, and memory; (3) "Region, family, OS, mem" for data center region, machine family, OS, and memory; (4) "DC, family, OS, mem" for data center, machine family, OS, and memory (baseline); (5) "Region, series, OS, mem" for data center region, machine series, OS, and memory; and (6) "DC, series, OS, mem" for data center, machine series, OS, and memory.

Figure OA-13: Cohort Learning Analysis Under Different Controls



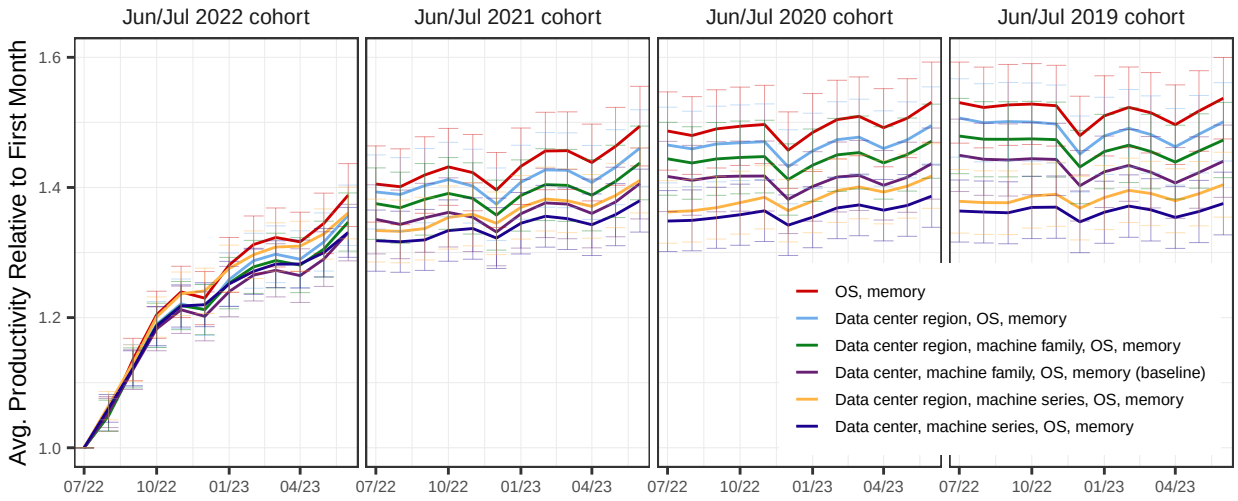
Notes: This figure shows the learning results on Figure 8 using different control variables in the productivity estimation.

Figure OA-14: Cohort Learning Analysis Under Different Peak Utilization Durations



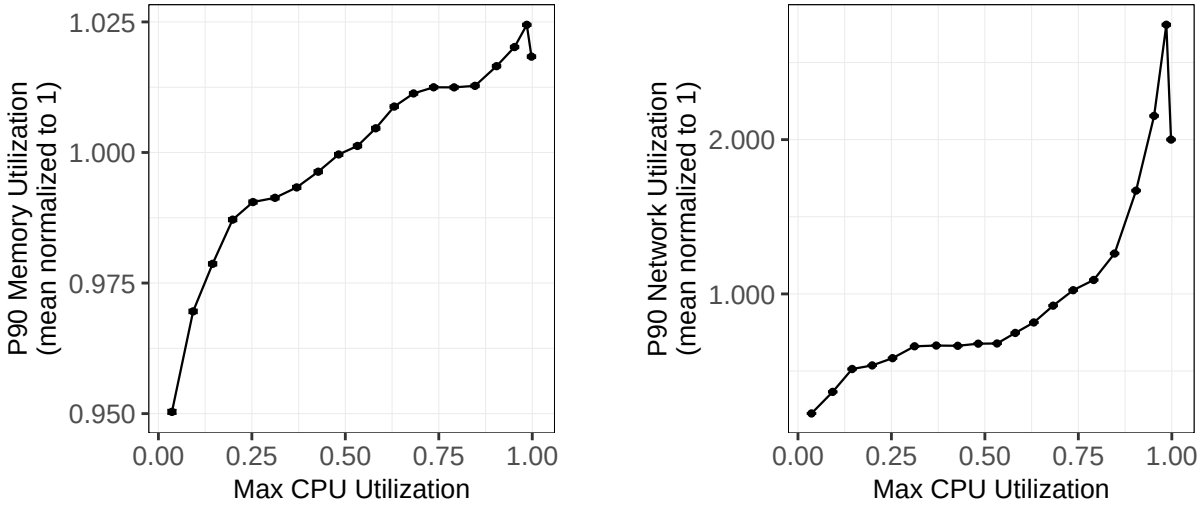
Notes: The learning results in Figure 8 using different numbers of days in the peak utilization definition.

Figure OA-15: Cohort Learning Analysis Under Different Downsizability Definitions



Notes: This figure shows the learning analysis of Figure 8 using different downsizability definitions.

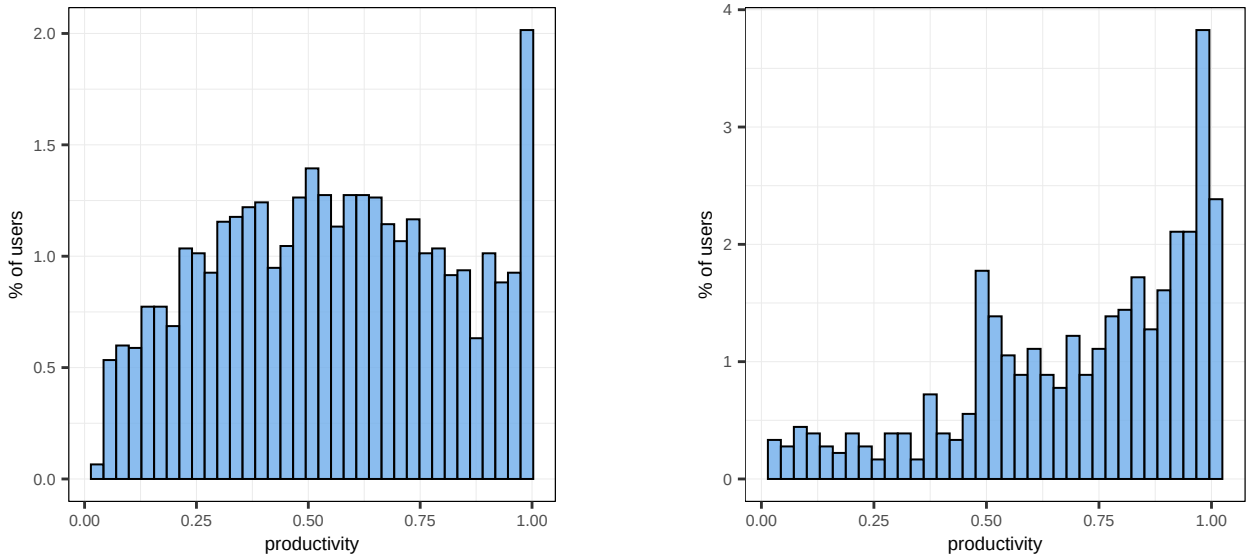
Figure OA-16: Correlation of CPU Utilization with Memory and Network Utilization



(a) CPU and Memory Utilization **(b)** CPU and Network Utilization

Notes: This figure illustrates the correlation between max CPU utilization and other resource measures. For each figure, we divide VM-days into twenty equally sized bins based on max CPU utilization, then plot the average max CPU utilization of VMs in that bin against the max memory utilization and 90th percentile network utilization of VMs in that bin. The average max memory and the average 90th percentile network across all bins are normalized to 1.

Figure OA-17: Productivity Distribution in Publicly Available CPU Utilization Data



(a) Azure **(b)** GCP

Notes: These figures illustrate the distribution of user-level compute productivity, estimated using the entire sample, weighting each VM by core-hours as shown in equations (26) and (27) for Azure and GCP, respectively. The x-axis represents productivity levels ranging from 0 to 1, while the y-axis shows the percentage of firms. Each observation corresponds to a user, and the histogram bars reflect the unweighted distribution of users across different productivity intervals.

References for Online Appendix

Abowd, J. M., F. Kramarz, and D. N. Margolis (1999). High Wage Workers and High Wage Firms. *Econometrica 67*(2), 251–333.

Athavale, J., M. Yoda, and Y. Joshi (2018). Thermal Modeling of Data Centers for Control and Energy Usage Optimization. In E. M. Sparrow, J. P. Abraham, and J. M. Gorman (Eds.), *Advances in Heat Transfer*, Volume 50, pp. 123–186.

Demirer, M. (2025). Production Function Estimation with Factor-Augmenting Technology: An Application to Markups. *Working Paper*.

Duan, L., D. Zhan, and J. Hohnerlein (2015). Optimizing Cloud Data Center Energy Efficiency via Dynamic Prediction of CPU Idle Intervals. In *2015 IEEE 8th International Conference on Cloud Computing*, pp. 985–988. IEEE.

Flexera (2023). 2023 State of the Cloud Report.

Foster, L., C. Grim, and J. Haltiwanger (2016). Reallocation in the Great Recession: Cleansing or Not? *Journal of Labor Economics 34*(S1), S293–S331.

Google (2019). Cluster Power Data 2019. Last accessed on 2024-06-24.

Greenstein, S. and T. P. Fang (2020). Where the Cloud Rests: The Location Strategies of Data Centers. *Harvard Business School Working Paper, No. 21-042*.

Gregg, B. (2014). *Systems Performance: Enterprise and the Cloud*. Pearson Education.

Husain Bohra, A. E. and V. Chaudhary (2010). VMeter: Power Modelling for Virtualized Clouds. In *2010 IEEE IPDPSW*, pp. 1–8.

Jiang, Z., C. Lu, Y. Cai, Z. Jiang, and C. Ma (2013). VPower: Metering Power Consumption of VM. In *2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp. 483–486.

Kansal, A., F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya (2010). Virtual Machine Power Metering and Provisioning. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, pp. 39–50.

Meisner, D., B. T. Gold, and T. F. Wenisch (2009). Powernap: Eliminating Server Idle Power. *ACM SIGARCH Computer Architecture News 37*(1), 205–216.

Melitz, M. J. and S. Polanec (2015). Dynamic Olley-Pakes Productivity Decomposition with Entry and Exit. *The RAND Journal of Economics 46*(2), 362–375.

Metcalfe, R. D., A. B. Sollaci, and C. Syverson (2023). Managers and Productivity in Retail. *NBER Working Paper, No. 31192*.

Microsoft Azure (2019). Azure Public Dataset V2. Last accessed on 2024-06-25.

Singh, R., M. A. Qureshi, and K. Annamalai (2015). A Brief Overview of Recent Developments in Thermal Management in Microelectronics. *Journal of Electronic Packaging 137*(4).

Tirmazi, M., A. Barker, N. Deng, M. E. Haque, Z. G. Qin, S. Hand, M. Harchol-Balter, and J. Wilkes (2020). Borg: the Next Generation. In *Proceedings of the Fifteenth European Conference on Computer Systems*, pp. 1–14.

Verma, A., L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes (2015). Large-Scale Cluster Management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys)*.

Waßmann, I., D. Versick, and D. Tavangarian (2013). Energy Consumption Estimation of Virtual Machines. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 1151–1156.

Wilkes, J., C. Reiss, N. Deng, M. E. Haque, and M. Tirmazi (2020). Google Cluster-Usage Traces V3. Last accessed on 2024-06-24.