Economics of Human-AI Interaction and AI-AI Interaction

Xingchen Xu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:
Yong Tan, Chair
Uttara M Ananthakrishnan
Stephanie Lee

Program Authorized to Offer Degree:
School of Business

University of Washington

Abstract

Economics of Human-AI Interaction and AI-AI Interaction

Xingchen Xu

Chair of the Supervisory Committee:
Yong Tan
Department of Information Systems and Operations Management

Artificial intelligence (AI) has evolved from a specialized tool into a central participant in economic life: it performs work, prices products, and decides what information people see. Economic outcomes are therefore no longer shaped by human decisions alone; they depend on how humans interact with AI. And as more market participants delegate decisions to algorithms, outcomes also depend on how AI systems interact with one another. This dissertation studies the economics of human-AI interaction and AI-AI interaction through three distinct yet interconnected essays. First, I study humans' strategic responses to the arrival of a powerful new AI that affects both sides of a market. I leverage comprehensive data from a leading online labor platform and a difference-in-differences design around the launch of ChatGPT. I find that LLM-based generative AI substantially displaces labor demand in exposed submarkets, while labor supply contracts more slowly, so competition intensifies. Surprisingly, rather than retreating from AI-exposed work, freelancers strategically transition toward programming-intensive jobs, the very domain where generative AI excels, because the technology also lowers the cost of acquiring new skills. This adaptation is driven primarily by high-skilled workers and carries significant consequences for earnings. Second, I examine what happens when many autonomous AI agents compete within a single market and a personalized AI intermediary stands between them and consumers. I combine a structural model of consumer search estimated from real-world data with large-scale simulations

of reinforcement-learning pricing algorithms. The objective of the platform's recommender system steers algorithmic competition: a revenue-maximizing design intensifies collusion, whereas a utility-maximizing design promotes competition. More surprisingly, enlarging the recommendation set does not consistently serve the platform's goals. This "more is less" effect arises because a larger set weakens the recommender system's ability to steer pricing algorithms through the rewards it controls. Third, I investigate what happens when a platform deliberately lowers the personalization level of its AI, a choice increasingly compelled by privacy regulations and computational costs. Through a large-scale randomized field experiment on a major social media platform, I find that depersonalizing search makes search measurably harder within the treated channel: users browse more content per query yet click less ordinary content, while their clicking on paid content barely moves. Surprisingly, the effects spill over to the untouched recommendation channel with the opposite sign. Users lower the standard they carry across channels, browse more, and click more ordinary content there. Paid content in the recommendation channel responds differently: its click-through rate is unchanged on average but falls among the platform's most active users, who already spend up to their limit, so their longer sessions spread a fixed amount of paid clicking more thinly. Taken together, the three essays follow AI across the life of a digital market: a powerful new AI enters and humans strategically adapt, autonomous AI agents fill the market and interact under the platform's mediation, and the platform in turn calibrates how much algorithmic mediation to supply. Ultimately, this dissertation serves as a foundation for future research on the evolving economics of AI, emphasizing the need for ongoing investigations that keep pace with markets in which humans and algorithms increasingly decide together.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I want to express my gratitude to my advisor, Professor Yong Tan. Since day one, he has encouraged me to think boldly and venture down less-traveled roads. This freedom has afforded me an atypical research path where I could explore unconventional topics, discern true uniqueness among seemingly novel directions, and pursue them with rigor. I will carry this spirit forward throughout my career.

I am also grateful to my reading committee members and coauthors, Professor Uttara M. Ananthakrishnan and Professor Stephanie Lee. From them, I have learned the art of both storytelling and analysis. They have also helped me "set boundaries, find peace," and embrace a more sustainable way of (professional) life.

Further, I want to thank Shangzi Zhou for her unwavering support and warmth over the years. With her by my side, I have had the luxury of celebrating every moment of happiness in daily life, exploring my interests freely, and continuing to pursue the aspirations on my life's checklist.

I also want to thank my friends and coauthors, Qili Wang and Yizhi Liu, for their steadfast friendship throughout these years. Together, we have overcome obstacles and shared moments of joy, both professionally and personally, and their presence has been a tremendous source of strength for me.

Lastly, I want to thank my parents, Yanfei Xu and Lihua Peng. Whatever I decide, they are always by my side with their unconditional love and support. The courage to explore is the greatest gift they have instilled in me since childhood, and I am proud to have come this far with their spirit always in my heart.

Chapter 1

INTRODUCTION

The rapid advancement and widespread adoption of artificial intelligence (AI) are fundamentally reshaping how digital markets operate. No longer a specialized tool at the margins of commerce, AI now sits at the center of economic activity, performing work that was once outsourced to people, setting prices on behalf of sellers, and deciding what information billions of users see. These capabilities offer immense potential for efficiency, personalization, and value creation. At the same time, they change the very structure of economic interactions. Market outcomes now emerge from the interplay between humans and AI, as people strategically adapt to algorithms that displace, assist, or inform them. Moreover, as more participants delegate their decisions to algorithms, a second layer emerges: AI systems increasingly interact with other AI systems, producing dynamics that none of their designers individually intended. Understanding these two layers of interaction is essential for businesses that deploy AI, for platforms that govern algorithmic marketplaces, and for policymakers who regulate them.

This dissertation examines the economics of human-AI interaction and AI-AI interaction in three interconnected essays. Each essay considers a digital market in which algorithms have become central actors, and each asks how the market and its participants respond when the role of AI changes. The first essay considers the arrival of a powerful new AI in a labor market and traces how human workers strategically adapt. The second essay considers a product market in which many autonomous AI agents compete and asks how the platform's personalized recommendation algorithm shapes their interaction. The third essay considers the reverse experiment, in which a platform deliberately reduces the personalization level of its AI, and follows the consequences for user behavior across the platform. Together,

the three essays move from humans responding to AI, to AI responding to AI, to platforms recalibrating AI itself. Chapter 2 reviews the literatures these essays build on, and Chapters 3 through 5 present the essays in turn.

The first essay focuses on humans' strategic responses when a powerful new AI arrives and shifts both sides of a market at once. As LLM-based generative AI can both substitute for human work and augment human capability, its arrival simultaneously moves the demand for labor and the productivity of those who supply it. In Chapter 3, I combine rich data from a leading online labor platform with a difference-in-differences design built around ChatGPT's release to estimate these effects. I find that generative AI substantially displaces labor demand in the submarkets most exposed to its capabilities, while labor supply contracts more slowly, so that competition among the remaining freelancers intensifies. Facing this pressure, freelancers do not exit passively: they strategically transition toward programming-intensive jobs, where generative AI lowers the cost of acquiring and exercising new skills, and it is the high-skilled workers who lead this adaptation, with sizable earnings consequences. This chapter speaks to the future of work in the AI era, showing that the labor market consequences of AI depend not only on what the technology can do, but also on how heterogeneous workers strategically reposition themselves around it. It underscores the importance of absorptive capacity and AI literacy for workers, and of upskilling programs for platforms and policymakers seeking to cushion the displacement that general-purpose AI can cause.

The second essay turns from human responses to the interactions among AI systems themselves. In modern online marketplaces, sellers increasingly delegate pricing to reinforcement-learning algorithms, while the platform's recommender system decides which products each consumer sees, so that the rewards from which pricing algorithms learn are themselves determined by another algorithm. In Chapter 4, I examine how this AI-AI interaction shapes market dynamics. Pairing a structural model of consumer search, estimated on real-world data, with large-scale simulations of reinforcement-learning sellers, I show that the platform's recommender system steers the outcome: its objective and the number of products it dis-

plays jointly determine whether algorithmic sellers drift toward collusive prices or are pushed toward competition, and how the resulting surplus is divided among sellers, consumers, and the platform. A recommender system designed to maximize consumer utility can discipline algorithmic sellers and mitigate collusion, whereas a revenue-maximizing design can sustain supra-competitive prices. Moreover, showing consumers more options does not always help the platform: expanding the recommendation set dilutes the recommender's influence over the rewards from which pricing algorithms learn, producing an unexpected "more is less" effect that horizontal differentiation further moderates. This chapter shows that the platform's information intermediation is not a neutral conduit but a market-design lever: the same population of pricing algorithms can produce starkly different competitive outcomes depending on the recommendation algorithm that mediates them. It also offers an alternative explanation for the correlated prices documented in empirical studies of algorithmic pricing and provides guidance for platforms and antitrust regulators who must evaluate markets in which the relevant decision-makers are no longer human.

The third essay investigates a platform's decision to deliberately weaken its own AI by reducing how much its rankings are personalized. Privacy regulations restrict the behavioral data on which personalization depends, and the computational cost of personalizing in real time grows with the user base, so platforms face mounting pressure to scale personalization back; search is the natural candidate because the user's query still anchors relevance without personal data. In Chapter 5, I study the consequences of such depersonalization through a large-scale randomized field experiment with roughly 71,000 users on a major social media platform, in which treated users received depersonalized search rankings while the recommendation channel remained untouched. Depersonalization makes search measurably harder: treated users browse more content per query yet click less of what they see. More surprisingly, the effects cross into the untouched recommendation channel and flip sign there: having lowered the bar that their search experience taught them to expect, users browse the recommendation channel more deeply and click more ordinary content, while paid-content monetization suffers precisely among the platform's most active users. A stylized model

in which a behavioral standard travels across channels rationalizes the full pattern. This chapter demonstrates that the value of personalization is heterogeneous, rising with the data each user has supplied, and that its removal propagates across the surfaces of a platform, so that single-channel evaluations of algorithmic changes can be systematically misleading for both platforms and regulators.

By combining insights from these three perspectives, this dissertation aims to provide a coherent account of how AI reshapes digital markets. The essays share a common lesson: what AI does to a market is determined not by the algorithm alone, but jointly by the algorithm, the humans who react to it, and the platforms that design and govern it. The same generative AI that displaces demand also lowers the cost of skill transition; the same pricing algorithms that can collude can be disciplined by a well-designed recommender; the same depersonalization that degrades one channel enlivens another. In each case, predicting the effect of AI requires modeling the strategic responses it sets in motion, whether those responses come from workers, from rival algorithms, or from users redistributing their attention.

The insights gleaned from this dissertation carry implications for businesses, platforms, policymakers, and researchers alike. For businesses and workers, the findings underscore that adapting to AI is a strategic problem: the returns to AI accrue disproportionately to those with the capacity to absorb it, and standing still in a market that AI has entered is itself a costly choice. For platforms, the findings elevate algorithm design from an engineering concern to a market-governance decision: the objective of a recommender system shapes competition among sellers, and the personalization level of a search ranking shapes behavior in channels far beyond search. For policymakers, the findings caution against evaluating algorithmic interventions in isolation: privacy rules aimed at one channel can push users toward more data-intensive channels, aggregate engagement can mask welfare losses, and collusion can arise or dissolve through design choices that no existing antitrust category cleanly captures. For researchers, the findings illustrate that the economics of AI cannot be reduced to the economics of any single algorithm.

This dissertation also highlights the value of methodological breadth in studying these questions. The essays combine causal inference on large-scale observational data, structural modeling and estimation, multi-agent reinforcement-learning simulation, randomized field experimentation, and analytical modeling, drawing on tools from information systems, economics, marketing, and computer science. No single method could answer the questions posed here: observational designs identify how humans respond to an AI shock that no experimenter could randomize, simulations reveal algorithmic dynamics that unfold over horizons no field study could observe, and field experiments isolate causal effects that no observational design could credibly recover. A multidisciplinary toolkit is, in this sense, not a stylistic choice but a requirement of the subject.

As AI continues to advance, the interactions studied in this dissertation will only deepen. Generative AI is spreading from text to every modality of work, autonomous agents are beginning to transact with one another directly, and platforms are continuously recalibrating how much algorithmic mediation to provide. It is therefore imperative for businesses, platforms, policymakers, and researchers to stay attuned to the evolving economics of these interactions. This dissertation underscores the importance of proactive engagement with both the opportunities and the risks: businesses must invest in the human capacity to work alongside AI, platforms must recognize and shoulder their role as governors of algorithmic markets, and policymakers must craft rules that protect individuals without foreclosing the efficiency gains that algorithmic mediation makes possible. By examining how humans respond to AI, how AI systems interact with one another, and how platforms calibrate the algorithms in between, this dissertation contributes a foundation for understanding markets in which humans and algorithms increasingly decide together.

Chapter 2

LITERATURE REVIEW

The theme of this dissertation is the economics of human-AI interaction and AI-AI interaction. It studies how humans respond when algorithms mediate their economic activity, and what happens when different algorithms interact with one another in a marketplace. The dissertation thus speaks to the general literature on the economics of AI, as well as to research on several specific aspects of AI: pricing algorithms and algorithmic collusion, recommender systems, and personalized ranking algorithms. The work is also relevant to the specific areas in which the essays are set. We therefore additionally review the literature on online labor markets and on technology and labor. This chapter discusses each of these literatures in turn.

2.1 Economics of AI

Broadly speaking, this dissertation is situated within the domain of the economics of AI, wherein economic principles are applied to evaluate, design, and implement AI techniques and architectures (Agrawal et al. 2019). A growing body of literature examines the impact of various AI systems across diverse contexts, such as AI assistants for customer response (Brynjolfsson et al. 2023), matching algorithms in the gig economy (Liu et al. 2024), and AI applications in creative works (Li et al. 2024). These insights inform the design of future AI systems to be more responsible and ethical, thereby enhancing overall welfare and promoting its equitable distribution (Maslej et al. 2023).

Within this broad domain, AI-based pricing algorithms, recommender systems, and ranking and personalization algorithms represent significant subfields. These areas have typically been studied separately, resulting in largely distinct streams of research (Calvano et al. 2020b,

Yoganarasimhan 2020, Fleder and Hosanagar 2009). In deployed marketplaces, however, the algorithms increasingly operate side by side and interact: as pricing algorithms adjust their strategies, the inputs to the recommender system shift accordingly, and the resulting recommendation decisions in turn redefine the rewards from which pricing algorithms learn and evolve. The economic phenomena that emerge when multiple types of AI operate jointly, and when humans interact with several algorithmic systems at once, remain comparatively underexplored. The following two sections review the streams most directly connected to these themes.

2.2 Algorithmic Collusion

A first specific stream is the burgeoning research on algorithmic collusion. With the prevalence of learning algorithms, scholars have investigated the impact of AI-based pricing algorithms on market competition and price equilibrium, particularly since the seminal work by Calvano et al. (2020b).

To describe and understand algorithmic collusion, some early works have employed theoretical models to illustrate how simple pricing strategies can decipher competitors' behavior and lead to collusive outcomes (Miklós-Thal and Tucker 2019, Salcedo 2015). However, as algorithms become more sophisticated, these models become analytically intractable, necessitating numerical simulations within a repeated game framework (Calvano et al. 2020b). Additionally, empirical evidence has documented instances of correlated and elevated prices among multiple vendors, although the underlying mechanisms remain challenging to verify empirically (Calder-Wang and Kim 2023, Wieting and Sapi 2021, Assad et al. 2020).

Most of this literature studies pricing algorithms that face consumer demand directly, abstracting from the platform that mediates the interaction. In practice, online marketplaces intermediate between sellers and consumers through recommender systems, which govern consumers' attention allocation and hence the rewards that pricing algorithms experience in each period. How such information intermediaries shape the learning dynamics of pricing algorithms, and whether they can help account for the correlated prices documented in the

empirical studies above, remains an open question. Related questions continue to emerge as new AI intermediaries enter the marketplace, such as AI shopping agents that provide consumers an alternative channel for discovering products (Lin et al. 2025).

2.3 Economics of Recommender Systems, Ranking, and Personalization

A second specific stream is the literature on the economics of recommender systems, which investigates the downstream impacts of these systems on various economically significant outcomes, driven by the complexity of consumer behavior in real-world settings.

For instance, Fleder and Hosanagar (2009) examine how recommender systems influence both individual-level and market-level sales diversity through analytical models and simulations. Building on this, Lee and Hosanagar (2019) empirically test the impact of recommender systems on diversity using field experiments. More recently, Lee and Hosanagar (2021) documents the heterogeneity in the economic impact of recommender systems, while Li et al. (2022) explore the causal pathways underlying these effects. In addition to studies that focus on the economic benefits that recommendations provide to sellers and platforms, there is also research that considers consumer welfare (Wan et al. 2024, Zhang et al. 2021) and examines the dynamic interactions between consumer choices and recommender systems (Zhou et al. 2023).

Most studies on recommender systems have typically treated product attributes, including price, as exogenous variables. However, a growing stream of literature considers participants' strategic behavior within a game-theoretic framework (Li et al. 2020, Ben-Porat et al. 2019, Che and Hörner 2018, Li et al. 2018). In these studies, strategic participants are governed by game-theoretic rules with clearly defined information sets, and their responses are characterized by closed-form equilibrium strategies (Li et al. 2020, 2018). Settings in which strategic responses instead emerge from algorithms that learn through trial and error across repeated interactions have received less attention.

The economics of recommender systems can also be regarded as part of the broader attention economics literature (Aridor 2025). From an economic perspective, when we abstract

away the technical details, recommender systems alter the attention allocation of consumers in the presence of information friction (Aridor et al. 2022). Adjacent literatures examine other information tools with the same character, such as assortment optimization, in which platforms choose product display to fulfill their objectives (Feldman et al. 2022), and advertising auctions that guide consumer attention.

Closely related is research on ranking and personalization, the algorithmic layer that determines what each individual user actually sees. Position effects are well documented: consumers disproportionately attend to highly ranked items, so the ranking itself shapes search behavior and transactions (Ursu 2018, Ghose et al. 2014). Personalizing the ranking can further improve the match between users and content: tailored search results raise click-through rates and reduce search frictions (Yoganarasimhan 2020), personalized recommendations affect consumer choice and welfare (Donnelly et al. 2024, Mao et al. 2023), and personalization more broadly underpins user engagement on content platforms (Kleinberg et al. 2024, Tam and Ho 2006). At the same time, personalization carries costs. It depends on behavioral data whose collection raises privacy concerns and is increasingly constrained by regulations such as the GDPR and the CCPA (Goldberg et al. 2024, Ke and Sudhir 2023, Jia et al. 2021), it can narrow the range of content users encounter (Pariser 2011), and the infrastructure behind it is computationally expensive at scale (Wu et al. 2022, Gupta et al. 2020). A small but growing body of work accordingly studies what happens when platforms scale personalization back, with existing evidence concentrated in e-commerce settings (Yuan et al. 2025, Chen et al. 2023); evidence from social media, where users discover content through multiple coexisting channels, remains scarce.

2.4 Online Labor Markets

Turning to the application areas, the first relevant stream concerns online labor markets, which have demonstrated a remarkable ability to effectively reconcile labor supply and demand across time and space with unparalleled flexibility (Stanton and Thomas 2025, Chen et al. 2019). A wealth of scholarly inquiries has been conducted to elucidate the complex

dynamics of these digital labor marketplaces. These investigations encompass the interplay between online labor markets and external socioeconomic fluctuations, the relationship between user behavior patterns and market design, and the advancements in matchmaking algorithms.

The first line of inquiry examines how offline conditions, such as unemployment rates and stages of national development, drive individual participation in online labor markets (Laitenberger et al. 2022, Huang et al. 2020, Kanat et al. 2018). Concurrently, related research investigates the reciprocal effects of online labor markets on the external environment, such as local employment and entrepreneurship outcomes (Guo et al. 2025, Burtch et al. 2018). Second, some prior work has focused on modeling user behavior and developing improved market designs within online labor platforms. For example, several studies analyze how employers dynamically learn to optimize hiring strategies for suitable workers (Bai et al. 2025, Daviss and Leung 2025, Kokkodis and Ransbotham 2023), while others explore the influence of factors such as betrayal aversion and real-effort incentives on workers' labor supply decisions (Benndorf et al. 2024, Hashim and Bockstedt 2024). Additional research examines the roles of reputation (Benson et al. 2020), monitoring systems (Liang et al. 2024), capacity constraints (Horton 2019), and wage regulations (Chen and Horton 2016) in shaping participant dynamics, market equilibrium, and social welfare, thereby informing the design and implementation of more effective market mechanisms (Garg and Johari 2021). Third, optimization algorithms have been refined to enhance job recommendations and streamline matchmaking processes within online labor markets (Aouad and Saban 2023, Kokkodis and Ipeirotis 2023).

Nevertheless, the advent of LLM-based generative AI leaves open the question of how such disruptive general-purpose technology reshapes competitive dynamics in online labor markets and how freelancers adapt to it.

2.5 Technology, Labor, and Human Capital

The second relevant application stream is the expanding literature on the technology–labor nexus and the role of human capital. A foundational concept is skill-biased technological change (SBTC), which posits that technological innovations disproportionately enhance the productivity and demand for skilled workers relative to their less-skilled counterparts (Acemoglu and Autor 2011, Autor et al. 2003). Task-based frameworks further suggest that technology complements workers engaged in abstract, non-routine cognitive tasks, while substituting for routine activities (Autor and Dorn 2013). Such dynamics can sometimes engender labor-market polarization: not only do high-skill roles benefit, but certain low-skill manual occupations (e.g., janitorial work) also exhibit lower susceptibility to automation, thereby expanding employment shares at both extremes of the skill distribution at the expense of middle-skill positions (Autor and Dorn 2013).

Beyond technology's differential impact on the demand for jobs with varying skill requirements, the process of skill formation itself warrants attention, as indicated by the human capital theory (Becker 1962). According to this theory, general human capital is transferable across firms, giving workers an incentive to finance their own training, whereas firm-specific skills are valuable only to the current employer, prompting firms to share training costs (Becker 1962). Some general human capital, however, is occupation-specific; this specificity can constrain workers' mobility (Neal 1995). Empirical evidence confirms that workers tend to switch to jobs requiring similar task profiles in order to leverage existing skills (Gathmann and Schönberg 2010). Although technology can enable workers to perform tasks beyond their original skill sets, successful adoption still (at both the organizational and individual levels) depends on absorptive capacity, defined as the ability to recognize the value of new knowledge, assimilate it, and apply it effectively (Hatch and Dyer 2004, Zahra and George 2002, Cohen and Levinthal 1990).

Recent advances in AI have pushed the frontier in the technology-labor interplay, enabling systems to accomplish tasks once thought beyond machine capability (Gallego and Kurer

2022, Paolillo et al. 2022, Agrawal et al. 2019). Previous AI applications, however, have remained purpose-built and narrow by design (Morris et al. 2023). In contrast, LLM-based generative AI constitutes a new class of powerful general-purpose technology capable of assisting with (or automating) a broad spectrum of activities (Eloundou et al. 2024, Morris et al. 2023). Initial evidence documents sizable productivity gains in specific settings, such as field studies in consulting (Dell'Acqua et al. 2023) and laboratory experiments in writing (Noy and Zhang 2023). These studies, however, focus on workers with well-defined roles; it remains unclear how freelancers, who confront all kinds of tasks and freely choose which projects to pursue, will adapt in the post-GPT era.

Chapter 3

"GENERATE" THE FUTURE OF WORK THROUGH AI: EMPIRICAL EVIDENCE FROM ONLINE LABOR MARKETS

3.1 Introduction

Online labor markets, particularly gig work platforms, have emerged as a pivotal component of the global labor landscape. According to the World Bank, as of 2023, the gig economy constitutes up to 12% of the global labor market and holds particular promise for vulnerable groups in developing nations[1]. The work-from-home and work-from-anywhere trends have continued to proliferate during and after the pandemic. As per Upwork's report, freelancers contributed $1.35 trillion to the U.S. economy in annual earnings in 2022, representing a $50 billion increase from the previous year, with 60 million American freelancers participating in this burgeoning sector[2]. Given their substantial impact on the economy, freelancer platforms have garnered significant attention from economic and information systems (IS) researchers (Stanton and Thomas 2025, Liang et al. 2024, 2022).

Nevertheless, the dynamics of the online labor market may have experienced a significant shift following the public release of ChatGPT on November 30, 2022. ChatGPT, along with other LLM-based generative AI[3], exhibits distinct technical features that set it apart from previous technologies. In particular, these models' unprecedented scale confers zero-shot proficiency across a broad spectrum of tasks (Kojima et al. 2022), prompting some scholars to regard them as the first generation of artificial general intelligence (AGI) (Morris et al.

[1]Please refer to https://openknowledge.worldbank.org/handle/10986/40066 for additional details.

[2]Please refer to https://www.businesswire.com/news/home/20221213005068/en/ for additional details.

[3]We hereafter collectively refer to LLM-based generative AI as ChatGPT in most instances to maintain brevity, except in formal research questions and hypotheses.

2023). Consequently, the advent of such tools holds the potential to directly automate certain tasks or to assist individuals in completing tasks more efficiently (Noy and Zhang 2023), thereby reducing the necessity to outsource work to others. This development is particularly consequential for online labor markets, where jobs are predominantly on-demand and short-term, rendering them especially vulnerable to fluctuations in external demand (Guo et al. 2025, Sundararajan 2017).

Given this potential demand shock, the resulting competitive landscape and dynamics on the platform are difficult to predict. On one hand, prior research suggests that freelancers are highly responsive to changes in earning opportunities on online labor markets and may exit the platform when conditions deteriorate (Chen and Horton 2016), thereby stabilizing the level of competition among remaining participants. On the other hand, because ChatGPT can reduce the time and effort required to complete tasks (Eloundou et al. 2024), it raises the net return per contract and may encourage freelancers to continue bidding even in a contracting market, thereby intensifying competition. These divergent perspectives motivate our first research question: (1) *How does LLM-based generative AI affect the competitive landscape of online labor markets?*

To address this question, we utilize a dataset comprising all job postings, bids, final transactions, and freelancer information from one of the leading freelancer platforms, covering the period from September 2021 to August 2023. We regard the launch of ChatGPT as an exogenous shock to the online labor market, given its sudden introduction and largely unexpected success. By applying natural language processing (NLP) techniques, we cluster jobs into submarkets and determine their treatment status based on the Language Modeling AI Occupational Exposure Index (LM-AIOE) (Eloundou et al. 2024, Felten et al. 2023). Using a standard difference-in-differences (DiD) framework, we identify a substantial demand displacement effect within the treated submarkets. Although labor supply in these submarkets also declines, the reduction is comparatively smaller, leading to increased competition as measured by the average number of bids per job.

Furthermore, the impact of ChatGPT may not be uniformly distributed across the treated

submarkets. At the time of our study, ChatGPT primarily generated responses in natural language or programming language (Achiam et al. 2023). Consistent with these output formats, writing and programming continue to be the core use cases when people use generative AI for work (Handa et al. 2025). However, the jobs requiring these two types of outputs differ substantially in their skill prerequisites, suggesting heterogeneous labor market effects. Specifically, natural language literacy can be acquired through everyday social interaction, such as conversations within families, without formal instruction. Programming, by contrast, is a skill that requires formal training and specialization (Vee 2013). This difference in skill acquisition is reflected in the population-level statistics: as of 2024, approximately 88% of adults worldwide possess basic natural language literacy[4], whereas the proportion of individuals who can write in a programming language is considerably lower, even in developed countries[5]. However, as a general-purpose technology, ChatGPT enables individuals across occupations to generate code through natural language prompts (Eloundou et al. 2024), potentially lowering the entry barrier to programming-intensive submarkets and facilitating transition. In addition, specialization barriers can give rise to wage premiums (Rosen 1983), which in turn create incentives for transition: on the platform we study, prior to the introduction of ChatGPT, programming-intensive jobs in the treated submarkets carry a higher average budget than jobs in other treated submarkets. To empirically examine this potential, we pose our second research question: (2a) *Does LLM-based generative AI induce skill transition toward programming in online labor markets?*

Building on the previously constructed dataset, we conduct difference-in-differences-in-differences (DDD) analyses at the submarket level to examine this potential transition. Our results show that programming-intensive submarkets experience a significantly smaller decrease in labor supply compared to other treated submarkets, a pattern that is not explained by heterogeneity in labor demand. We further investigate heterogeneity in new freelancer participation and rule out this alternative explanation. The remaining plausible explanation is

[4]See https://data.worldbank.org/indicator/SE.ADT.LITR.ZS for more details.

[5]See https://w3.unece.org/SDG/en/Indicator?id=115 for more details.

therefore a skill transition toward programming among existing freelancers. To substantiate this interpretation, we perform freelancer-level analyses and document an overall tendency for freelancers to shift toward programming-related tasks. Moreover, freelancers with greater exposure to ChatGPT (measured by the proportion of treated jobs they apply for) exhibit a stronger transition, suggesting that the demand shock also incentivizes adaptive behavior. Finally, we find that freelancers who transition realize significant economic benefits, reinforcing the strategic value of such skill transitions.

Despite the observed skill transition effect on average, a natural follow-up question is who participates in this transition and remains more resilient during the market downturn. The answer is difficult to predict a priori. On one hand, ChatGPT may primarily facilitate high-skilled freelancers in making this shift, as high-skilled workers tend to be more complementary to new technologies (Acemoglu and Autor 2011) and technology adoption requires absorptive capacity (Cohen and Levinthal 1990). On the other hand, ChatGPT's capabilities may lower the threshold for effective use, thereby compressing individual skill disparities and enabling lower-skilled freelancers to benefit disproportionately (Brynjolfsson et al. 2025, Noy and Zhang 2023). Given these competing perspectives, we propose to empirically examine heterogeneity in skill transition to complement the previous research question: (2b) *How does the skill-transition effect vary across freelancers with different skill levels?*

To examine such heterogeneity, we construct moderators indicative of freelancers' skill levels, including pre-treatment earning efficiency and review ratings. Using DDD analyses at the freelancer level, we find that high-skilled freelancers predominantly drive the skill-transition process. Specifically, freelancers who achieve higher revenue per bid and possess superior market ratings are the primary participants in this transition.

Figure 3.1 summarizes the key empirical evidence underpinning our analysis, and these results yield several significant implications. First, we contribute to the literature on online labor markets (Bai et al. 2025, Guo et al. 2025, Liang et al. 2024) by demonstrating that the emergence of powerful LLM-based generative AI generates a substantial external demand shock, and we provide comprehensive empirical evidence on how this technical breakthrough

alters the competitive landscape and elicits responses from freelancers. Second, our study informs theories concerning technology's impact on the labor force. Human capital theory posits that some skills are occupation-specific and that mobility across occupations entails frictions (Kleiner and Xu 2025, Becker 1962); our findings provide evidence that ChatGPT can facilitate such occupational mobility. Furthermore, we extend the skill-biased technical change (SBTC) theory by moving beyond within-occupation productivity effects to show that skill transition across occupations can also be skill-biased (Guo et al. 2025, Acemoglu and Autor 2011), disproportionately benefiting high-skilled workers. Finally, our research offers practical insights: policymakers and platform operators should closely monitor evolving demand dynamics and the upskilling opportunities introduced by disruptive technologies like ChatGPT, implementing training programs and market designs to support workforce transitions. For freelancers, proactively adapting to changes in the competitive landscape and undertaking skill transitions can confer a competitive advantage.

The subsequent sections of this chapter are organized as follows. Section 2 develops the theoretical hypotheses. Section 3 describes the empirical context, data collection, and variable construction methods. Section 4 outlines the identification challenges and details the selection of econometric specifications. Section 5 presents the empirical results in relation to the hypotheses. Section 6 reports robustness checks to validate our findings. Finally, Section 7 concludes by discussing detailed implications, as well as limitations and avenues for future research.

3.2 Hypothesis Development

In this section, we theorize the influence of LLM-based generative AI on market dynamics and freelancer behavior. Specifically, we develop hypotheses addressing (i) the overall impact of generative AI on the competitive landscape of online labor markets, (ii) the potential for freelancers to engage in skill-transition behaviors, and (iii) the heterogeneity of these skill-transition effects across different freelancer groups.

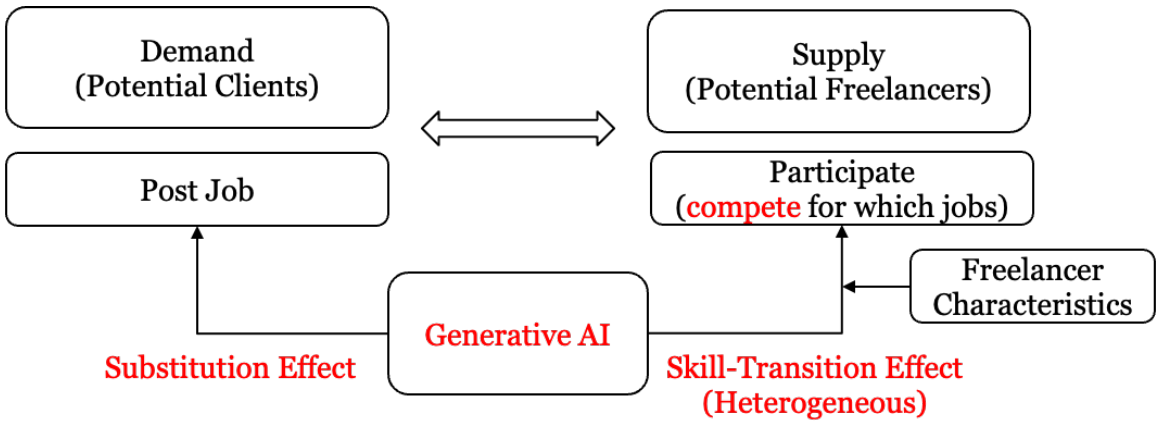


Figure 3.1: The Impact of Generative AI on Online Labor Market.

3.2.1 Demand Displacement Effect and Competition Change

We begin our theoretical framework by formulating hypotheses regarding the overall impact of LLM-based generative AI on the competitive landscape of online labor markets. As a powerful general-purpose technology capable of fully automating or assisting with a wide range of tasks (Eloundou et al. 2024), generative AI has the potential to help satisfy employer demand, thereby generating a displacement effect on the online labor force (Acemoglu and Restrepo 2019). As a result, the advent of generative AI may lead to a reduction in demand and an overall contraction of online labor markets. Nonetheless, despite this potential contraction, the resulting competition dynamics remain ambiguous; accordingly, we present two opposing theoretical predictions.

On the one hand, competition among freelancers may remain stable. A fundamental advantage of online labor markets lies in their flexibility for both sides. Consequently, a substantial body of literature documents freelancers' rapid responses to changing economic conditions (Huang et al. 2020) and other strategic behavior in online labor markets (Filippas et al. 2024). For instance, monitoring systems that mitigate the cold-start problem have been shown to significantly increase participation among less-experienced freelancers when

expected benefits rise (Liang et al. 2024). Additionally, freelancers exhibit prompt reactions to wage reductions by exiting the market (Chen and Horton 2016). Given freelancers' sensitivity to earning opportunities, it is plausible that they may quickly respond to decreases in demand by exiting, thereby maintaining a stable level of competition for these short-term tasks.

On the other hand, despite a potential decline in demand due to the displacement effect, freelancers may continue to participate in job bidding, thereby intensifying competition for several reasons. First, existing literature indicates that some freelancers are long-term participants in online labor markets, dedicating work hours comparable to full-time employment (Abraham et al. 2017). Such freelancers tend to exhibit market stickiness, allowing them to endure heightened competition and reduced earning opportunities (Gussek and Wiesche 2024). Second, generative AI can serve as a productivity-enhancing tool by assisting freelancers in task completion (Brynjolfsson et al. 2025, Noy and Zhang 2023), thereby increasing the marginal benefit of securing jobs and making higher competition levels more sustainable. Together, these factors may result in a smaller contraction in labor supply and elevate the equilibrium competition level among remaining jobs.

Based on the above reasoning, we propose a set of competing hypotheses for empirical testing.

Hypothesis 1a. LLM-based Generative AI decreases demand for exposed jobs without increasing competition.

Hypothesis 1b. LLM-based Generative AI decreases demand for exposed jobs and heightens competition.

3.2.2 Skill-Transition Effect

Next, we establish the theoretical foundations underlying the potential for freelancers to engage in skill-transition behavior within online labor markets following the introduction of LLM-based generative AI. As outlined in the development of Hypothesis 1, freelancers are strategic actors who respond adaptively to changes in the competitive landscape (Chen and

Horton 2016). For those who remain active in the market, it is plausible that they would change their participation across different job types as a strategic response. Specifically, human capital theory posits that jobs vary considerably in their human capital intensity, which influences wages and productivity (Becker 1962). In the digital era, programming-intensive occupations, characterized by higher cognitive complexity, typically introduce a wage premium (Brynjolfsson and McAfee 2014, Autor and Dorn 2013, Autor et al. 2003). However, because some skills tend to be occupation-specific, acquiring the competencies necessary for programming entails significant human capital investment and creates entry barriers (Acemoglu and Autor 2011, Gathmann and Schönberg 2010). The introduction of ChatGPT, which demonstrates substantial capability in assisting with programming tasks (Eloundou et al. 2024, Poldrack et al. 2023), may reduce these investment costs, thereby facilitating freelancers' skill transition.

While the above reasoning suggests that generative AI may promote skill transitions, additional frictions could weaken this effect. First, beyond the acquisition of technical skills, other forms of human capital, such as domain expertise and contextual knowledge, also entail substantial costs associated with learning through practice and accumulating work experience (Gathmann and Schönberg 2010, Polanyi 2009, Neal 1995). Consequently, ChatGPT may not sufficiently reduce the overall transition costs to induce widespread skill shifts. Second, the displacement effect (Hypothesis 1) may also impact programming jobs, thereby diminishing the relative benefits of transitioning, particularly for freelancers who are primarily engaged in tasks less susceptible to demand decreases.

Given these competing forces, whether the reduction in programming entry barriers enabled by ChatGPT is sufficient to overcome the aforementioned frictions remains an open empirical question. To guide our investigation, we propose the following hypothesis for empirical testing.

Hypothesis 2. LLM-based Generative AI introduces skill transitions towards programming.

3.2.3 Heterogeneous Skill-Transition Effect

We further investigate the heterogeneity of skill-transition behavior, contingent upon the positive impact of ChatGPT on skill transition being sufficiently strong (Hypothesis 2). Theoretically, this heterogeneity may unfold along two competing and plausible dimensions.

On one hand, existing literature generally suggests that high-skilled freelancers are more capable of adopting new technologies and enhancing their productivity (Autor and Dorn 2013, Acemoglu and Autor 2011). Although our context involves cross-occupational skill transitions that require acquiring and leveraging new competencies, similar patterns may emerge due to the superior abilities of high-skilled freelancers in utilizing external tools (Hatch and Dyer 2004, Zahra and George 2002, Cohen and Levinthal 1990). Consequently, ChatGPT may provide greater incentives for skill transition among these high-skilled freelancers.

On the other hand, given the considerable capabilities of ChatGPT, the performance gap between high- and low-skilled freelancers may narrow, particularly within the same occupation (Brynjolfsson et al. 2025, Noy and Zhang 2023). This dynamic may also apply to the skill-transition context, reducing the influence of ability differences on the transition cost (Hatch and Dyer 2004, Cohen and Levinthal 1990). Moreover, since lower-skilled freelancers start from a comparatively disadvantaged position in their original occupations, the relative returns from making a skill transition could be greater for them (Becker 1962). Accordingly, this cost-benefit tradeoff may encourage less-skilled freelancers to engage in skill transitions more actively.

Which of these competing forces dominates is thus an empirical question, and we propose the following pair of competing hypotheses.

Hypothesis 3a. The proposed skill-transition effect is stronger among skilled freelancers.

Hypothesis 3b. The proposed skill-transition effect is stronger among less-skilled freelancers.

3.3 Research Context, Data, and Variables

To empirically address our research questions, we collect data from a leading freelancer platform and construct variables for both market-level and freelancer-level analyses. Section 3.3.1 provides an overview of the platform, details the selection of the observation period, describes the raw dataset, outlines the job classification process based on clustering techniques, and explains the formation of submarkets. In Section 3.3.2, we define the treatment and control groups at both levels of analysis. Section 3.3.3 presents the development of the key variables used in our study.

3.3.1 Research Context and Submarket Construction

The focal online labor platform enables interactions between freelancers and clients across a diverse range of job categories. Clients post job listings (i.e., projects) that outline the specific requirements and essential skills necessary for successful completion. Freelancers evaluate these postings and submit bids detailing their qualifications and proposed compensation. Clients then select from among the bidders, thereby finalizing transactions with their chosen freelancers. A detailed description of this process is provided in Appendix A.1.

To investigate the impact of LLM-based generative AI, with a particular emphasis on ChatGPT, our observation window must encompass the launch date of ChatGPT, which occurred on November 30, 2022. To ensure sufficiently long observation periods before and after the launch time, while mitigating the confounding end-of-year effects, we establish a two-year observation window spanning from September 2021 to August 2023. Within this time period, we collect all job postings, associated job bids, and the data of freelancers who placed at least one bid on these projects.

When a job listing is published on the platform, the client specifies relevant skill tags that denote the technical or domain-specific competencies required for the task. Across 1.6 million jobs, 2,719 unique skill tags are observed, forming 485,370 distinct combinations of skill tags (hereafter referred to as skill sets). These tags facilitate efficient search and filtering,

as similar tags correspond to comparable skill requirements and attract overlapping pools of freelancers with relevant expertise. Conversely, divergent skill configurations reflect heterogeneous client demands, enabling differentiation among job types and appealing to diverse segments of the freelancer workforce. It is important to note that during this period, publicly accessible versions of text-to-image generative AI systems such as Dall-E and Midjourney were also released. We exclude the effects of these text-to-image AIs from the main analyses and address their potential influence separately in Appendix A.12. After excluding skill sets containing any skill tags related to image output, 409,119 skill sets remain. For these skill sets, we define submarkets as clusters of job opportunities requiring similar skills, allowing us to assess the differential effects of ChatGPT on demand and competitive dynamics across various job categories to test our hypotheses.

Specifically, based on the skill sets available for each job, we employ a multi-step clustering procedure to uncover underlying job categories. Initially, each job's set of skill tags is transformed into a fixed-length vector representation using embedding techniques, specifically BERT embeddings, which capture the job's latent position within a multi-dimensional space (Wang et al. 2023a, Devlin et al. 2019). Leveraging these embeddings, we apply the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm to group similar skill combinations into distinct clusters (McInnes et al. 2017). HDBSCAN is a density-based clustering method well-suited for high-dimensional, sparse, and unevenly distributed data and has been widely adopted in management research (Mousavi and Gu 2024, Nguyen et al. 2024). Unlike centroid-based algorithms such as k-means, HDBSCAN accommodates clusters of varying densities and does not require pre-specification of the number of clusters, rendering it particularly appropriate for the data-driven identification of submarkets on the platform. Consequently, HDBSCAN facilitates the detection of cohesive job clusters characterized by similar skill requirements. Based on the clustering results, we identify 1,082 distinct submarkets. Each submarket corresponds to a cluster of similar skill sets. Since each job is associated with a specific skill set, which maps uniquely to a single cluster, we are able to assign every job to its corresponding submarket. A detailed

description of this clustering procedure is provided in Appendix A.2.

3.3.2 Treatment Group, Control Group, and Treatment Timing

In the preceding subsection, we cluster the skill sets into 1,082 distinct submarkets and subsequently assign each job to one of these submarkets based on its associated skill set. Building on this submarket classification, we then define the treatment and control groups at both the market and freelancer levels.

We operationalize treatment assignment by quantifying the degree to which job skills are related to the capabilities of language models. Specifically, we utilize LM-AIOE, a variant of the AIOE, which measures the extent to which various occupations are related to language modeling AI tools (Felten et al. 2023). Our approach begins by computing the semantic similarity between each skill tag and the 774 occupations for which LM-AIOE scores are available. This enables us to assign an LM-AIOE value to each skill tag based on its most semantically aligned occupation. Subsequently, we compute the weighted average LM-AIOE score for each submarket, where weights correspond to the relative prevalence of skill tags within the submarket. Submarkets exhibiting higher average LM-AIOE scores are thus considered to be more closely related to the broad functionalities of LLMs.

However, directly using the continuous LM-AIOE score as the treatment variable entails assuming a parametric linear relationship between the LM-AIOE value and outcomes in the online labor market. This assumption is strong, as the LM-AIOE score reflects the relatedness between a skill and AI capabilities (Felten et al. 2023), yet the precise functional form linking this relatedness to labor market outcomes is not known *a priori*. Moreover, even in so-called "sharp" designs where treatment unambiguously corresponds to a continuous measure (e.g., dosage of a medication), estimating treatment effects at varying levels remains challenging and carries a high risk of bias (Callaway et al. 2024). Consequently, we adopt a more cautious approach, employing estimation methods with minimal parametric assumptions and conducting sensitivity analyses to evaluate robustness.

Specifically, we partition submarkets into treatment and control groups using a median

split of associated occupations' LM-AIOE values: submarkets with LM-AIOE above the median are assigned to the treatment group ($Treat = 1$), while those below the median form the control group ($Treat = 0$). Consequently, our submarket-level DiD analyses compare labor market outcomes between submarkets whose required skills are more closely related to LLM core functionalities and those less related. Detailed procedures for LM-AIOE computation and treatment assignment are provided in Appendix A.2. To verify robustness, we also experiment with alternative thresholds by varying the cutoff to the median LM-AIOE $\pm$ 0.05. The results, reported in Appendix A.13, further confirm the stability of our findings.

Regarding treatment timing, it is important to note that several models and their applications were released during the observation window. We designate November 30, 2022 as the treatment date, because this date corresponds to the public release of ChatGPT, the first LLM-based text generative AI and a milestone towards artificial general intelligence (Morris et al. 2023). Because this date falls at the end of November, we define December 2022 as the first post-treatment month in our month-level analyses. Accordingly, we set $Post = 1$ for all months from December 2022 onward and $Post = 0$ for earlier months. For week-level analyses, we designate the week containing November 30, 2022, as the initial post-treatment period.

Finally, we construct treatment indicators at the freelancer level to examine heterogeneous responses and transitional behaviors, thereby empirically testing Hypotheses 2 and 3. We limit our sample to freelancers who placed at least one bid both before and after the treatment month, as the skill-transition process can only be measured for individuals active in both pre- and post-treatment periods (Wooldridge 2010, Angrist and Pischke 2009). This restriction yields 132,260 freelancers for our analysis, approximately 6.7% of the two million freelancers who bid during the observation window. Although this appears to be a small subset, it remains representative: most other freelancers bid only briefly after registration and then exit the market (Liang et al. 2024). Importantly, this cohort accounts for roughly 12 million bids, 60% of total bidding activity, making them the core contributors whose welfare and behavior underpin the platform's viability.

In the freelancer-level analysis, although freelancers may bid in multiple submarkets, most concentrate on a limited set of skills. As a result, the proportion of each freelancer's pre-treatment bids submitted in treated submarkets, which captures their initial skill relatedness to ChatGPT's capabilities, is typically close to either 0 or 1. Because this proportion resembles a binary indicator, using it as a continuous measure yields similar results to a binary treatment variable. We therefore adopt this proportion directly as the freelancer-level treatment variable ($Treat$).

3.3.3 Variable Descriptions

In this section, we define the primary variables used in our analysis and describe their construction. All market-level variables are aggregated at the submarket–month level. To assess labor demand and other facets of market dynamics, we compile several measures. Demand is proxied by the number of job postings in each submarket during a given month, denoted $Jobs$. Supply is captured by the total number of bids submitted to those postings, denoted $Bids$. To characterize competitive intensity and transaction value, we employ two additional variables: $Transaction$, the total monetary value of completed contracts, and Avg_bids, the average number of bids per job. We also construct several alternative measures for robustness; the corresponding results are reported in Appendix A.23.

Further, to evaluate potential transitions toward programming, we classify each submarket by its programming intensity. We first label every skill tag using the occupation mappings in Felten et al. (2023); tags linked to occupations whose descriptions primarily involve programming are marked as programming-intensive. A submarket, which comprises multiple skill tags, is considered programming-intensive ($Programming = 1$) if more than 50% of its tags are programming-intensive; otherwise, it is classified as non-programming-intensive ($Programming = 0$).

At the freelancer level, our primary interest is their transition toward programming. For each job a freelancer bids on, we compute its programming intensity, defined as the proportion of aforementioned programming-intensive tags among all tags. When constructing the panel

data, we average this intensity across all jobs a freelancer bids on within a given month; this measure is denoted $Programming\%$. To complement the panel, we calculate additional monthly variables for each freelancer that capture participation and earnings: the total number of bids submitted ($Bids$), the total value of completed transactions ($Transaction$), and the average revenue generated per bid (Avg_Earn_Bid). Several supplemental freelancer-level moderators are introduced in Section 3.5.3 and detailed in Appendix A.3.

Due to the left-skewed nature of the data distribution, we apply a log transformation to all the outcome variables in the subsequent analysis, with the exception of $Programming\%$, which ranges between 0 and 1. The descriptive statistics of all these variables are presented in Appendix A.3 and Appendix A.23.

3.4 Identification Challenges and Econometric Specifications

To rigorously investigate the impact of LLM-based generative AI on online labor markets, we discuss several identification challenges prior to conducting formal analyses.

The primary challenge arises from the potential interactions among market participants. As is common in research involving markets, units can influence each other through various mechanisms, such as competition and selection (Jensen and Miller 2018). In online labor markets, freelancers have the autonomy to select jobs and compete with one another (Liang et al. 2022), rendering the comprehensive capture of market dynamics a methodological conundrum, even with randomized experiments (Barach et al. 2020). Developing innovative statistical methods for analyzing matching markets is a promising avenue for future methodological research. However, given the currently available tools and the aspiration to expand the frontiers of our knowledge, we employ the following strategies: i) Rather than solely focusing on freelancer-level final transactional outcomes, we construct a series of submarkets to elucidate temporal dynamics in online labor markets, as detailed in Sections 3.5. By utilizing market-level data, we can directly examine heterogeneous changes on the demand side, which is mainly on-demand and suffers less from cross-unit interactions (Hong et al. 2016). ii) However, the presence of freelancers who are active for relatively long periods

and can transition across different submarkets raises concerns regarding the labor supply side analyses. Consequently, we exercise meticulous care in interpreting the effects. It is crucial to emphasize that our objective is not to capture the hypothetical labor change in the absence of freelancers' transition. Instead, at the market level, we capture the relative changes between submarkets whose required skills closely align with ChatGPT's capabilities and those less related, thereby allowing for freelancer transitions as a potential mechanism. (iii) To further elucidate the detailed transition processes on the supply side, we conduct freelancer-level analyses characterizing skill transition dynamics in Sections 3.5.2 and 3.5.3.

Another concern stems from the fact that all time-variant observables are potentially influenced by LLM-based generative AI, regardless of whether they exist within or outside the online labor markets. The inclusion of such post-treatment variables as controls can obstruct certain mechanisms and bias the overall effect estimation (Tafti and Shmueli 2020), a phenomenon known as the bad control problem (Cinelli et al. 2022). Therefore, we treat all observables as outcome variables and examine their changes, instead of presuming some to be exogenous control variables. In alignment with established practices (Braghieri et al. 2022), our primary specification incorporate two-way fixed effects without the introduction of control variables and presuming some to be exogenous:

$$Outcome_{it} = \beta_0 + \beta_1 * Treat_i * Post_t + u_i + T_t + \epsilon_{it} \tag{3.1}$$

Here, $Outcome_{it}$ denote the outcome of interest for unit i (either a submarket or a freelancer) during month t. The coefficient β_1 captures the treatment effect, with the treatment variable at both levels defined in Section 3.3.2. The term u_i represents the unit-level fixed effect, while T_t accounts for the year-month fixed effect. The error term ϵ_{it} encompasses unobserved factors influencing the outcome.

While the inclusion of certain post-treatment control variables can underestimate the treatment effects substantially, we perform robustness and sensitivity analyses to mitigate concerns regarding omitted variable bias and to provide more conservative interpretations of the coefficients. Further details are provided in Appendix A.19.

3.5 Main Analyses

In this section, we address our research questions and test the proposed hypotheses using a series of DiD analyses at both the market and freelancer levels. We first evaluate the overall impact of LLM-based generative AI on the competitive landscape of online labor markets (Hypothesis 1). We then examine freelancers' responses to this evolving landscape, assessing whether the introduction of ChatGPT has encouraged strategic skill transitions (Hypothesis 2). Finally, we analyze heterogeneity in these transitions, comparing the behavior of skilled and less-skilled freelancers (Hypothesis 3).

3.5.1 Empirical Results of Demand Displacement Effect and Competition Change (Hypothesis 1)

To assess the displacement effect of LLM-based generative AI and the resulting shifts in the online labor market, we employ the DiD framework outlined in Section 4 and estimate ChatGPT's influence on labor demand, labor supply, transaction values, and competition levels. Column (1) of Table 3.1 shows a pronounced, statistically significant 22.1% decline in labor demand (measured by job postings) in treated submarkets relative to control submarkets following ChatGPT's introduction ($\beta_1 = -0.2210$, $p < 0.01$). This contraction is consistent with a displacement mechanism: generative AI, as a powerful general-purpose technology, can automate or assist a wide array of tasks (Eloundou et al. 2024), thereby reducing people's reliance on online freelancers.

Column (2) of Table 3.1 show that the sharp decrease in demand is accompanied by a significant reduction in labor supply, reflected in a 16.8% decline in the total number of bids ($\beta_1 = -0.1680$, $p < 0.01$). Contraction on both sides of the market, in turn, translates into a decline in overall market value (Column (3)): total transaction volume falls by 31.5% ($\beta_1 = -0.3150$, $p < 0.01$). Notably, the decline in total transaction volume is not only statistically significant but also economically meaningful. In the post-treatment period, the average monthly transaction value in treated submarkets is \$2,030. Since our estimates indicate a

31.5% decrease attributable to generative AI, the counterfactual monthly transaction value in the absence of generative AI would be approximately $2,030/(1-0.315) \approx \$2,964$. This implies a monthly reduction of roughly \$934 per treated submarket, or equivalently, an annual loss of approximately \$11,202. Aggregating across the 917 treated submarkets in our sample, this translates into a platform-wide reduction in transaction value of roughly \$10.27 million per year.

Beyond the aggregate displacement effect, the reductions in jobs and bids are not proportional: job postings decline by 22.1%, whereas bids fall by 16.8%. This asymmetry manifests in the average number of bids per job (*Avg_Bids_job*). Column (4) of Table 3.1 reports a significant increase in this metric for treated submarkets ($\beta_1 = 0.138$, $p < 0.01$). Accordingly, competition within these submarkets intensifies after ChatGPT's introduction, lending support to Hypothesis 1b.

Table 3.1: The Impact of Generative AI at the Market Level

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
Treat $*$ *Post*	-0.2210***	-0.1680***	-0.3150***	0.1380***
	(0.0318)	(0.0476)	(0.0896)	(0.0357)
Constant	3.0190***	5.4060***	5.6250***	2.0880***
	(0.0101)	(0.0152)	(0.0286)	(0.0114)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5410

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

As posited in Hypothesis 1b, the observed intensification of competition may arise from the strategic, long-term engagement of core freelancers and from the productivity gains

that generative AI confers. A freelancer's bidding decision can be modeled as a utility-maximization problem that balances the expected probability of winning a contract against the expected benefit of completing it. Although increased competition can reduce the likelihood of winning, generative AI lowers the time and effort required to execute the work, thereby mitigating the decline in expected utility. As a result, some freelancers remain willing to participate even in a more competitive market. In other words, while the displacement effect reduces the number of available jobs, freelancers' participation diminishes less sharply because AI-driven productivity gains raise their expected utility and increase their tolerance for intensified competition.

3.5.2 Empirical Results of Skill-Transition Effect (Hypothesis 2)

As documented in Section 3.5.1, labor demand declines sharply, whereas labor supply contracts more modestly, thereby intensifying competition among remaining freelancers. To offset the lower probability of winning a contract under heightened competition, remaining freelancers are incentivized to increase the expected payoff conditional on winning. LLM-based generative AI can achieve this goal in two complementary ways: first, by reducing job completion costs through task assistance, and second, by enabling freelancers to upskill and bid for higher-value positions.

In the digital era, programming jobs entail greater cognitive complexity and command higher wages (Brynjolfsson and McAfee 2014, Autor and Dorn 2013). This premium is also salient in online labor markets, where programming posts constitute a substantial share of total listings and pay more than other jobs. Given ChatGPT's demonstrated effectiveness in coding assistance (Eloundou et al. 2024, Poldrack et al. 2023), we now empirically examine whether freelancers undertake skill transitions toward programming.

We first analyze evidence at the market level. As described in Section 3.3.3, we construct a binary indicator, *Programming*, that classifies submarkets as programming-intensive or non-programming-intensive based on their skill requirements. We then estimate a triple-difference (DDD) model (Equation 2), detailed in Appendix A.4, to assess how market outcomes vary

across these two submarket types. Using control submarkets as the baseline, the coefficient β_1 captures the post-ChatGPT change in non-programming-intensive submarkets, whereas β_2 measures the additional change observed in programming-intensive submarkets relative to β_1. The results are reported in Table 3.2.

$$Outcome_{it} = \beta_0 + \beta_1 * Treat_i * Post_t + \beta_2 * Treat_i * Post_t * Programming_i + u_i + T_t + \epsilon_{it} \quad (3.2)$$

Interestingly, while we find no significant differential effect on labor demand between programming-intensive and non-programming-intensive submarkets ($\beta_2 = 0.0313$, $p > 0.1$), programming-intensive submarkets experience a markedly smaller decline in labor supply ($\beta_2 = 0.153$, $p < 0.01$). Consistent with this pattern, the average number of bids per job (Avg_Bids_job) increases more sharply in programming-related submarkets ($\beta_2 = 0.116$, $p < 0.01$). Taken together, these results provide preliminary support for Hypothesis 2, suggesting that incumbent freelancers gravitate toward programming jobs and are willing to tolerate the heightened competition there.

However, it is important to note that the observed heterogeneity in programming-intensive submarkets may not be entirely attributable to skill transition behavior within the platform. An alternative explanation is that generative AI reduces entry barriers and enhances efficiency for individuals outside the online labor markets to engage in programming work, potentially attracting new market entrants. To assess whether the observed supply-side heterogeneity in programming-intensive submarkets is driven by the entry of new freelancers, we introduce a new outcome variable, $New\_Freelancer$, defined as the monthly count of freelancers who newly register and submit at least one bid within a given submarket. Using the same econometric specifications as in Equations (1) and (2), we conduct both DiD and DDD analyses. The results, presented in Appendix A.9, reveal no significant difference in the number of new freelancers between programming-intensive and non-programming-intensive submarkets ($\beta_2 = -0.0430$, $p > 0.1$), effectively ruling out new entrants as the explanation for the observed heterogeneity. Consequently, the aggregate shift of incumbent freelancers toward programming stands out as the sole plausible explanation for the market-level het-

Table 3.2: The Heterogeneous Impact of Generative AI at the Market Level

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2350***	-0.2390***	-0.3460***	0.0847**
	(0.0344)	(0.0518)	(0.0964)	(0.0385)
$Treat * Post *$ *Programming*	0.0313	0.1530***	0.0662	0.1160***
	(0.0226)	(0.0348)	(0.0674)	(0.0270)
Constant	3.0190***	5.4060***	5.6250***	2.0880***
	(0.0101)	(0.0152)	(0.0286)	(0.0114)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5410

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

erogeneity documented in Table 3.2.

Next, we extend our inquiry to a more granular freelancer-level analysis of skill transition. As explained in Section 3.3.3, we construct the outcome variable *Programming*%, defined as the monthly average programming intensity of the jobs each freelancer bids on. Because this measure is observed only when a freelancer submits at least one bid in a given month, we restrict the sample to freelancers who are active at least once in both the pre- and post-treatment periods; otherwise their time series cannot capture ChatGPT's influence. Using this balanced panel, we estimate an Interrupted Time Series Analysis (ITSA) to trace changes in programming emphasis following ChatGPT's release. The results, shown in Appendix A.5, reveal a significant post-treatment increase in *Programming*%, indicating that incumbent freelancers have, on average, redirected their bidding toward programming jobs. This evidence further supports Hypothesis 2.

Hypothesis 2 posits that a freelancer's decision to transition depends on the relative expected benefit of switching versus the cost incurred. Because ChatGPT induces a displacement effect, freelancers whose pre-treatment bids are less connected to ChatGPT's capabilities may face weaker incentives to transition. Consider an extreme case: if a freelancer bids exclusively on job types whose demand remains stable, when demand for programming jobs declines sharply, the relative benefit of shifting to programming diminishes, making continued bidding on original job types more attractive. By contrast, freelancers more heavily affected by displacement have stronger incentives to pivot toward programming to recoup lost earnings.

To test this mechanism, we estimate freelancer-level DiD models using Equation 3.1 with $Programming\%$ as the dependent variable. The treatment variable, $Treat$, equals the share of pre-treatment bids a freelancer placed in treated submarkets and thus proxies exposure to ChatGPT's displacement effect. Column (1) of Table 3.3 shows that freelancers with greater exposure are significantly more likely to shift their bidding activity toward programming work ($\beta_1 = 0.0229$, $p < 0.01$).

Finally, we assess the economic consequences of skill transition to further corroborate Hypothesis 2. Using the same DiD specification as in Equation 3.1, we compare ChatGPT's impact on transaction volume, bidding activity, and earning efficiency across different freelancer segments. Appendix A.7 reports that freelancers who earned income in the pre-treatment period and relied exclusively on treated submarkets experienced an 11.6% decline in monthly transaction value relative to those who bid only in control submarkets. More strikingly, freelancers who continued to depend on treated submarkets after ChatGPT's introduction without transitioning saw a 20.8% reduction. Given that actively earning freelancers generate an average monthly transaction value of $493, the average annual earnings for these freelancers would be roughly $5,916, close to the average GDP per capita ($6,800) in the developing economies from which 92% of freelancers originate[6]. The observed decline trans-

[6]Please refer to https://www.imf.org/external/datamapper/NGDPDPC@WEO/OEMDC/ADVEC/WEOWORLD for more details.

lates into a loss of approximately $1,230 per year, or about 18% of GDP per capita, posing a substantial threat to their livelihoods. By contrast, treated freelancers who undertake a skill transition realize an 11.75% increase in earnings per bid and avoid the loss in total earnings (a modest 6.95%, which is positive but not significant).

Taken together, these findings support Hypothesis 2 and illuminate the underlying mechanism: LLM-based generative AI reduces the human-capital barriers to programming, enabling freelancers to transition into higher-paying roles while preserving their earning efficiency despite intensified competition.

Table 3.3: Freelancer-Level Skill-Transition Effects

	Programming%				
	(1)	(2)	(3)	(4)	(5)
$Treat \times Post$	0.0229***	0.0697***	0.0138***	0.1079***	-0.0072***
	(0.0015)	(0.0036)	(0.0017)	(0.0025)	(0.0019)
Constant	0.3121***	0.3596***	0.2998***	0.3779***	0.2820***
	(0.0005)	(0.0012)	(0.0006)	(0.0008)	(0.0007)
Rating	Both	High	Low	-	-
Earning_Bid_Pre	Both	-	-	High	Low
Freelancer FE	Yes	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes	Yes
Observations	711,738	140,079	571,659	218,791	492,947
R^2	0.8576	0.8636	0.8545	0.8924	0.8347

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

3.5.3 Empirical Results of Heterogeneous Skill-Transition Effect (Hypothesis 3)

Section 3.5.2 demonstrates that LLM-based generative AI encourages freelancers to shift their skills toward programming. Hypothesis 3, however, posits that this transition may not be uniform: the propensity to move may vary with freelancers' skill levels. High-skilled

freelancers could be more inclined to switch occupations because their stronger absorptive capacity allows them to adopt and exploit new technologies more effectively. Conversely, ChatGPT's powerful capabilities can lower the barriers to programming, thereby reducing the skill gap and offering larger relative gains to low-skilled freelancers, whose lower starting point leaves greater room for improvement. These two mechanisms operate in opposite directions, making it unclear *ex ante* which group will exhibit the stronger tendency to transition.

To adjudicate between these competing perspectives and test Hypothesis 3, we employ two proxies for freelancer skill level: (i) reputation, measured by the average rating received (*Rating*), and (ii) pre-treatment earning efficiency, calculated as total transaction value divided by total bids ($Avg\_Earn\_Bid\_Pre$). Both measures enter freelancer-level DDD regressions reported in Appendix A.6. For interpretability, we report four subsample analyses (two proxies $\times$ two skill strata) here, which yield consistent results.

Reputation is highly skewed. Because of the cold-start problem (Liang et al. 2024), 75% of freelancers receive no reviews, whereas those who do are almost always rated at the maximum (median = 5). This bimodal pattern (missing vs. 5) leads us to classify freelancers with $Rating = 5$ as high-skill and all others ($Rating < 5$ or unrated) as low-skill. A similar distribution characterizes earning efficiency: many freelancers earn nothing prior to treatment, leaving $Avg\_Earn\_Bid\_Pre = 0$ for a sizable share of the sample. A median split, therefore, assigns freelancers with positive pre-treatment earnings to the high-skill group and those with zero earnings to the low-skill group. For each subgroup, we replicate the DiD specification used in Section 3.5.2 (Equation 3.1) to estimate the treatment effect on freelancers' programming participation intensity, measured by $Programming\%$.

Columns (2) and (3) of Table 3.3 report the estimates that use *Rating* as a skill proxy. High-rated freelancers exhibit a significant post-treatment rise in the share of programming jobs ($\beta_1 = 0.0697$, $p < 0.01$), an increase of roughly 19% relative to their pre-treatment mean of 0.359. By contrast, low-rated freelancers show only a modest gain ($\beta_1 = 0.0138$, $p < 0.01$), or about 4.9% above their baseline mean of 0.284, indicating a far weaker propensity to

transition. Columns (4) and (5) use pre-treatment earning efficiency ($Avg\_Earn\_Bid\_Pre$) as an alternative skill measure. Freelancers with higher earning efficiency experience a sizable increase in programming intensity ($\beta_1 = 0.1079$, $p < 0.01$), representing a 28% uptick over their pre-treatment average of 0.386. In contrast, low earners even register a small decline ($\beta_1 = -0.0072$, $p < 0.01$), underscoring their difficulty in shifting toward programming work.

Taken together, these findings support Hypothesis 3a: freelancers with higher overall skill levels account for most of the observed transitions. Their superior prior skills signal greater individual absorptive capacity, which remains important for cross-occupational moves even though ChatGPT is easy to use and aids skill acquisition.

3.6 Robustness Checks

To further strengthen causal inference, we implement a series of robustness analyses addressing potential threats to identification.

First, to ensure our results are not driven by the specific threshold used to define treatment and control groups, we test alternative cutoff values (Appendix A.13), replace the binary indicator with the continuous LM-AIOE score (Appendix A.25), adopt a tercile-based classification (Appendix A.26), and employ an alternative exposure measure (Appendix A.27), all yielding consistent results. Second, to mitigate potential seasonal biases, we conduct a placebo test by assigning November 30, 2021, the same calendar day in the prior year, as a pseudo treatment date and reestimate the DiD specification. The resulting coefficients are statistically insignificant, thereby ruling out year end or seasonal confounders (Appendix A.14). Third, to assess whether our results could simply reflect random fluctuations, we randomly reassign treatment status across observations and repeat the DiD estimation for each permutation. The distribution of placebo estimates is centered on zero, and the observed treatment effects lie in the tails of this distribution, indicating they are unlikely to be driven by random variation (Appendix A.15). To address concerns about group comparability, we estimate a relative time model confirming parallel pre-treatment trends between treated and control submarkets (Appendix A.16). We further enhance com-

parability using propensity score matching, which yields consistent results (Appendix A.17). In addition, we apply ITSA to the treated submarkets, with results that align closely with our main estimates (Appendix A.18). To mitigate omitted variable bias from unobserved external or internal factors, we add relevant controls and conduct a sensitivity analysis following Altonji et al. (2005). The similarity of coefficients with and without these controls indicates robustness to unobserved confounders (Appendix A.19). We also test robustness through alternative modeling and data structures. Specifically, we use a Negative Binomial specification to account for count outcomes (Appendix A.20), apply weekly data aggregation (Appendix A.21), and include separate year and month fixed effects (Appendix A.22). Moreover, we examine three additional outcome variables to assess market dynamics from different angles (Appendix A.23). Lastly, excluding extreme submarkets at both ends of the distribution yields consistent results (Appendix A.24). Taken together, all these comprehensive checks reinforce the robustness of our findings. To further strengthen our conclusions, we conduct a series of extended analyses. First, given the rise of text-to-image generative AI tools during the observation period, we examine their impact on image-related submarkets. To capture both direct effects and potential publicity spillovers from ChatGPT's launch, we run two complementary analyses (Appendix A.12). Neither yields significant results, suggesting that current text-to-image models lack the general-purpose capabilities needed to influence online labor markets. Further, while our main analysis shows a smaller decline in labor supply within programming-intensive occupations, heterogeneity may exist within programming jobs. To explore this, we categorize programming languages into scripting and compiled types. Results indicate that scripting-language-dominated submarkets do not experience significant decline relative to the control group, further illustrating the direction of skill transition (Appendix A.8). Additionally, we examine other demand-side dynamics in treated submarkets over the observation period, focusing on budget, required skills, and the share of jobs related to programming. These results are presented in Appendices A.10 and A.11.

3.7 Implications and Conclusion

3.7.1 Summary of Key Findings

In this study, we conduct a comprehensive analysis of the impact of LLM-based generative AI on the online labor market. Leveraging rich data from a leading freelancer platform and a DiD framework, we uncover key shifts in market dynamics. Specifically, we find a significant demand-side displacement effect in submarkets closely aligned with ChatGPT's core capabilities. While labor supply in these submarkets also declines, the reduction is smaller, intensifying competition among freelancers. More interestingly, we identify a notable skill-transition effect, with freelancers increasingly shifting their focus toward programming. Moreover, this transition exhibits substantial heterogeneity, revealing that it is primarily driven by high-skilled freelancers.

3.7.2 Theoretical Contributions

Our study contributes to the literature on online labor markets and the interplay among technology, labor markets, and human capital.

First, it expands the body of research examining how offline socioeconomic conditions influence online labor markets, such as the impact of local unemployment on freelancers' participation (Laitenberger et al. 2022, Huang et al. 2020, Kanat et al. 2018). While existing literature typically focuses on large-scale external socioeconomic shifts, our research highlights that technological innovations can also exert significant effects. Specifically, we empirically investigate how the advent of LLM-based generative AI substantially disrupts online labor demand and, notably, motivates supply-side dynamics and intensifies market competition.

Second, our research also contributes to the literature on the strategic behaviors of participants in online markets. For instance, clients may deliberately avoid high-quality yet overly popular freelancers due to capacity constraint concerns (Horton 2019). Further, the implementation of monitoring systems influences freelancers to adjust their bidding strate-

gies, particularly affecting inexperienced workers (Liang et al. 2024). Our work extends this line of inquiry by identifying an additional type of strategic behavior, specifically freelancers' skill transition toward different job categories triggered by the introduction of ChatGPT.

Third, our study extends the SBTC theory by examining the effects of a general-purpose technology (GPT) on cross-occupation transitions. SBTC theory traditionally emphasizes how technological advances disproportionately enhance productivity and demand for high-skilled labor, potentially raising wages and attracting additional workers (Acemoglu and Autor 2011, Autor et al. 2003). In contrast, we document a novel mechanism whereby even a uniformly negative demand impact from GPT across certain related job categories can induce skill-transition behavior among workers toward occupations requiring higher skills. This transition aligns with human capital theory, which models workers' skill acquisition decisions as a strategic trade-off between benefits and associated costs (Becker 1962). As GPT reduces the skill acquisition costs for higher-skilled and higher-paying jobs, workers become more likely to transition into these roles. Consequently, our research identifies a GPT-enabled cross-occupation transition towards more skill-intensive jobs on the supply side, a phenomenon distinct from the demand-driven effects commonly highlighted in SBTC literature (Autor and Dorn 2013).

Finally, building on the observed skill-transition effects, we further demonstrate that this transition is itself skill-biased. Specifically, our findings indicate that freelancers with higher general abilities on the platform contribute disproportionately to the skill transition. This observation aligns with absorptive capacity theory, which underscores the critical role of an individual's ability to efficiently assimilate and utilize new technologies (Zahra and George 2002, Cohen and Levinthal 1990). While SBTC traditionally highlights heterogeneous productivity gains within occupations as the primary consequence of technological advancements, we extend this perspective by showing that cross-occupation transitions are also skill-biased, favoring higher-skilled workers. These theoretical insights hold significant implications for future advancements in general-purpose technologies, particularly in the trajectory towards AGI, where powerful AI could potentially both displace and augment human

labor across diverse occupational contexts.

3.7.3 Managerial Implications

Our findings yield several practical implications for policymakers, platform operators, and freelancers. First, the centrality of individual absorptive capacity in enabling skill transitions implies that platforms should extend their training offerings beyond tool-specific and task-oriented competencies to include meta-learning strategies that help freelancers assimilate new knowledge amid rapid technological change. This emphasis parallels organizational absorptive capacity development: the long-term competitiveness of online labor platforms depends on the collective adaptability of their workforce to continuously attract demand. Second, in the AI era, fostering foundational AI literacy—that is, the ability to understand, evaluate, and leverage diverse AI applications—can convert emerging technologies from potential risks into strategic opportunities. Realizing these outcomes requires a coordinated effort: platforms must design targeted programs to strengthen freelancers' absorptive capacity and AI literacy, while freelancers themselves must engage in ongoing, proactive skill development. Third, policymakers may also incorporate these insights into public upskilling and reskilling initiatives. Although the specific occupations affected may vary across broader labor markets, the underlying technology-enabled skill-transition dynamics remain relevant, making absorptive capacity and AI literacy critical considerations for effective workforce policy.

Further, AI's impact on freelancers is bidirectional: while powerful AI can enable workers to perform tasks previously beyond their capabilities, it also carries the risk of substantial demand displacement, as clients may deploy the same tools to meet their needs. Consequently, a lower strategic transition barrier does not necessarily constitute a favorable opportunity; freelancers must closely monitor shifts in demand to select appropriate skill-transition pathways, rather than blindly acquiring easily attainable skills. Furthermore, because peers are simultaneously adjusting their skill portfolios, the competitive landscape is in constant flux; freelancers should evaluate their relative positioning within the marketplace and plan tran-

sitions with strategic foresight. Platforms can also capitalize on this dynamism through targeted market and information designs. For example, by analyzing demand data, a platform might identify specific freelancer cohorts and nudge them toward alternative skills when reallocating labor would be welfare-enhancing and transitions are feasible with AI assistance.

3.7.4 Limitations and Future Directions

Several limitations of this study point to promising directions for future inquiry. First, the two-year observation window yields a relatively brief post-treatment period of only nine months, which may not fully capture longer-term market adjustments or the gradual adaptation of freelancers. Although we observe temporal dynamics within this interval, effects may evolve differently over an extended horizon. To address this limitation, future research should broaden the observation period and even apply network-analytic techniques alongside dynamic, game-theoretic models to more precisely trace how freelancer–client interactions unfold and adapt over time.

Second, this study focuses exclusively on a single online labor platform. Although our theoretical framework enhances the potential for generalization, alternative platform architectures and traditional labor markets exhibit unique institutional and operational characteristics that may produce context-specific outcomes. Accordingly, future research should conduct more analysis across a variety of settings, such as gig platforms with different governance models or offline labor exchanges, to develop a more holistic understanding of generative AI's impact on labor markets.

Third, our dataset does not include detailed measures of ChatGPT usage by individual freelancers or clients. Consequently, we analyze the overall changes following ChatGPT's introduction, rather than isolating the direct effects of adoption. Future research could obtain granular usage data to revisit these dynamics with deeper mechanism explorations. For example, it would be interesting to implement a field experiment that provisionally grants a random sample of freelancers access to specific tools and tracks subsequent shifts in their bidding strategies over time.

Furthermore, over the two-year observation window, macroeconomic fluctuations and other exogenous shocks may have influenced overall market dynamics. To mitigate these concerns, we exclude text-to-image generative AI from our primary analyses (see Appendix A.12) and incorporate a set of internal and external control variables in robustness checks (see Appendix A.19). We also conduct sensitivity analyses to assess the potential impact of unobserved confounders (Appendix A.19). Nonetheless, fully isolating every external influence remains a limitation of this study.

In addition, client participation on the platform is highly sporadic, rendering the construction of a reliable client-side panel infeasible. Moreover, we lack data on clients' trade-offs between alternative sourcing options and platform engagement, which precludes a comprehensive understanding of demand-side dynamics. Future research could address this gap by examining demand-side behavior at a more granular level to complement our supply-focused analysis. For example, scholars might investigate how employers' hiring decisions evolve after they begin integrating generative AI tools into their working processes.

Finally, our outcome measures have room for refinement. We infer work quality solely from client ratings, which are both intermittently recorded and prone to inflation—a bias noted in prior research (Filippas et al. 2018). Moreover, our analysis of skill-transition effects focuses on movement into programming roles, constrained by the platform's predefined skill tags. Future studies should develop more nuanced measures of transition pathways and examine the corresponding quality outcomes to capture the full spectrum of skill-upgrading dynamics.

In summary, ongoing advances in technology, particularly in AI, are continually reshaping labor markets and open a wealth of avenues for future research and practical innovation for both scholars and practitioners.

Chapter 4

ALGORITHMIC COLLUSION OR COMPETITION: THE ROLE OF PLATFORMS' RECOMMENDER SYSTEMS

4.1 Introduction

4.1.1 Motivation and Research Focus

With the advancement of technology and the proliferation of sophisticated machine learning algorithms, the use of automated pricing systems has increased significantly across various sectors. For instance, a substantial share (34%) of residential units in the United States currently utilize RealPage software, which provides instantaneous recommendations for optimal rental rates (Calder-Wang and Kim 2023). Artificial Intelligence (AI)-driven pricing programs, particularly those employing reinforcement learning techniques, have thus emerged as compelling alternatives to conventional pricing models (Spann et al. 2024). These AI-powered systems typically require minimal human intervention and exhibit remarkable adaptability to rapidly changing economic conditions.

However, both researchers and policymakers have voiced concerns about the potential detrimental effects of these algorithms on consumers, particularly regarding algorithmic collusion. Algorithmic collusion refers to the phenomenon whereby algorithms learn to establish supra-competitive prices through repeated interactions (Calvano et al. 2020a). In this setting, sellers merely set a revenue-optimizing objective, and the pricing algorithms can independently learn to charge supra-competitive prices without direct communication or explicit instructions on how to collude. The pricing algorithms can consistently maintain these higher prices at the expense of consumers (Calvano et al. 2020a).

Given the growing attention to this issue, related legislative actions and legal cases have begun to emerge. For example, the "Preventing Algorithmic Collusion Act" was proposed in

2024[1], and RealPage has been sued for deploying pricing algorithms that allegedly harmed millions of renters[2]. In parallel, a growing body of academic literature has sought to address this phenomenon. On the empirical front, studies have provided suggestive evidence that the adoption of pricing algorithms is associated with increased price synchronization among firms and elevated price levels (Calder-Wang and Kim 2023, Wieting and Sapi 2021, Assad et al. 2020). To better understand the underlying mechanisms, numerical experimental approaches have been widely employed since the seminal work of Calvano et al. (2020b) (Dou et al. 2024, Rocher et al. 2023).

One of the most prevalent settings for pricing algorithms is online marketplaces, such as Amazon (Chen et al. 2016). On these platforms, recommendation algorithms typically select and rank a subset of products to display to consumers. In the presence of information frictions, recommendations directly shape consumers' attention allocation and thereby determine product exposure (Li and Tuzhilin 2024b, Wang et al. 2024). Consequently, although platforms optimize recommendations according to their own objectives, doing so also influences sellers' revenue by altering product exposure, which in turn changes the learned reward structure and dynamics of the pricing algorithms. In fact, the coexistence of recommender systems and pricing algorithms on online platforms gives rise to a unique form of AI-AI interaction: pricing algorithms alter the price inputs to the recommender system, while the recommender system reshapes the rewards of the pricing algorithms in each period. This feedback loop cannot be adequately captured by either stream of previous literature concerning (i) the economics of recommender systems or (ii) algorithmic collusion on its own. Recognizing this gap, we aim to address it by examining *how various recommendation algorithms affect pricing dynamics and equilibrium outcomes in the algorithmic pricing game.*

---

[1]See https://www.congress.gov/bill/118th-congress/senate-bill/3686 for more details.

[2]See https://www.cbsnews.com/news/realpage-lawsuit-price-fixing/ for more details.

4.1.2 Research Framework

To examine the algorithmic pricing game under the influence of recommender systems, it is essential to develop a modified repeated game framework. Calvano et al. (2020b) provides a seminal foundation for the study of algorithmic collusion, offering a framework that has been widely adopted and adapted in subsequent research (Dou et al. 2024, Rocher et al. 2023, Banchio and Skrzypacz 2022). Calvano et al. (2020b) specifies three key elements: (1) the consumer demand model, where consumer decisions are governed by a canonical logit model; (2) the pricing algorithms, in which sellers employ basic Q-learning algorithms to adaptively set prices; and (3) the game sequence, where at each discrete period, sellers simultaneously make pricing decisions and observe the resulting demand responses.

Given our research focus, we need to specify and integrate the recommender system as a fourth element within the framework. Furthermore, it is also necessary to adjust the other three components accordingly to ensure coherent integration, which presents certain challenges. In the presence of recommender systems, a key challenge is to model consumer decision-making when consumers are exposed to multiple recommended products simultaneously. Typically, consumers observe a ranked list of top-K products and need to expend time and effort to evaluate the options of interest (Moraga-González et al. 2023, Mortensen 2011). Additionally, position biases may influence their choices (Ursu 2018). To accurately capture these considerations and establish a solid economic foundation, we develop a customized structural search model that incorporates heterogeneous utility functions and search costs (Chung et al. 2024, Ursu et al. 2023). We then estimate the model parameters using real-world data, employing an innovative combination of Hermite approximation methods and simulated maximum likelihood estimation to reduce computational complexity. The estimated model subsequently generates demand responses within the repeated game framework.

With this demand model in place, we also specify a recommender system that optimizes the platform's objective, such as overall revenue or consumer utility (Compiani et al.

2024, Wang et al. 2024, Zhang et al. 2024, Li and Tuzhilin 2024a). However, when the demand model is grounded in a sophisticated framework such as sequential search, closed-form expressions for consumer choice probabilities and optimal recommendations are generally unavailable. As a result, we follow established optimization literature and employ numerical integration methods to identify recommendation actions that maximize the specified objective function (Compiani et al. 2024).

For the pricing algorithm and game sequence, we adopt the Q-learning approach and the simultaneous game structure outlined by Calvano et al. (2020b). Nonetheless, Q-learning inherently assumes forward-looking behavior, prompting us to examine an alternative in the form of a simpler bandit algorithm to ensure robustness. Similarly, to further test the flexibility of our framework, we also consider an asynchronous game sequence as an additional robustness check.

By integrating all of these elements, we can analyze the impact of recommender systems on algorithmic pricing. However, even the original framework proposed by Calvano et al. (2020b) poses significant analytical challenges: in a multi-agent environment, each agent's setting is non-stationary, rendering both the learning of strategies and the determination of equilibria analytically intractable. Introducing a recommender system further complicates the analysis, as the absence of closed-form expressions for consumer choice probabilities precludes straightforward derivation of sellers' demands and revenues. To address these complexities, we extend the approach of Calvano et al. (2020b) by relying on numerical simulations to trace price evolution and identify equilibrium outcomes. In each period, after sellers have set prices and the platform has determined recommendations, we use the empirical distribution of consumer attributes and the estimated demand model to compute each seller's expected reward through multiple layers of numerical integration. This procedure enables us to efficiently execute simulations and conduct counterfactual analyses by altering any component within the proposed framework.

4.1.3 Findings

Based on the aforementioned simulation setup, we can adjust the elements within the framework to examine how such changes influence price dynamics and equilibrium outcomes. From an economic perspective, recommender systems optimize the platform's objective by selecting and potentially ranking a subset of products from all available options to display to consumers (Feldman et al. 2022). From classical collaborative filtering to deep learning, advances in the technical literature have significantly improved models' ability to map product selections to objectives (Li and Tuzhilin 2024b, Wan et al. 2024, Zhang et al. 2019, Adomavicius and Tuzhilin 2005). We instead focus on two economic dimensions that need to be specified in the recommendation setting: what to achieve (the recommendation objective) and how many products to display (the recommendation size). While specific methods determine how the product set is identified, these two dimensions are generally applicable from an economic standpoint (Compiani et al. 2024). Therefore, as shown in Figure 4.1, our first two research questions seek to identify the objective effect and the size effect of recommender systems on the algorithmic pricing game:

> *RQ1: How does the objective of the recommender system affect the algorithmic pricing game?*

> *RQ2: How does recommendation size affect the algorithmic pricing game?*

For the recommendation objective effect (RQ1), previous research on product display algorithms in two-sided markets predominantly considers two key objectives. The first, *Revenue-Maximization*, seeks to maximize total platform-wide revenue, thereby aligning with the platform's economic incentives through commission fees. The second, *Utility-Maximization*, aims to enhance consumer utility, which may indirectly benefit the platform by improving consumer retention (Compiani et al. 2024, Feldman et al. 2022, Derakhshan et al. 2022, Ursu 2018). On some platforms, recommender systems may balance the interests of both sides and establish a weighted objective (Wang et al. 2024). Different objectives

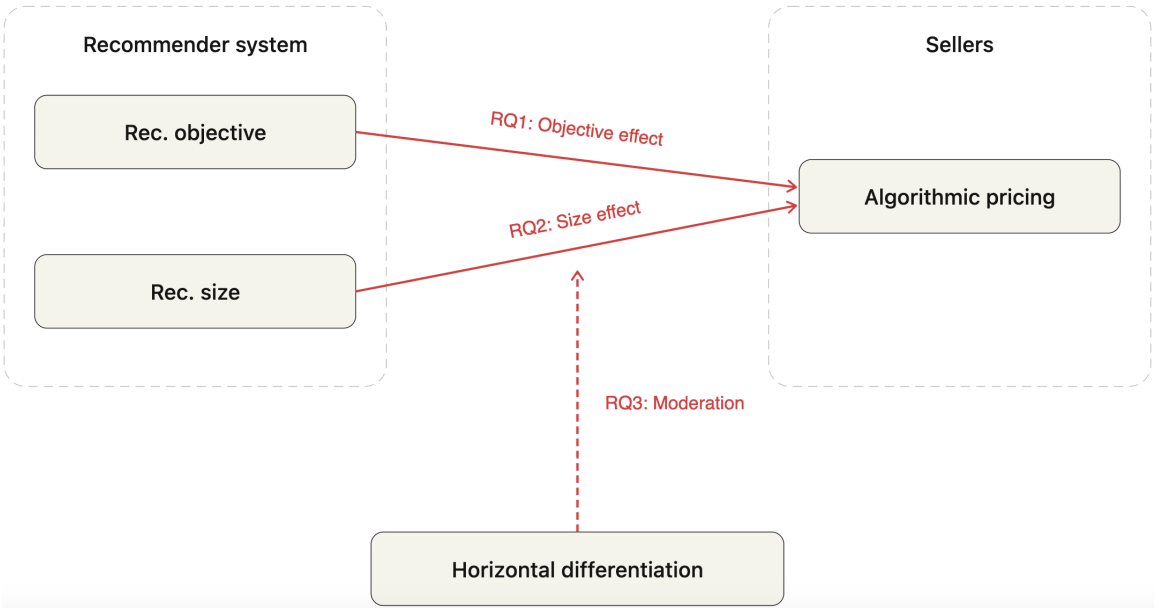


Figure 4.1: Summary of Research Questions for AI-AI Interaction.

can substantially alter which products are prioritized in recommendations and consequently influence sellers' learned strategies as well as price dynamics.

Leveraging the proposed repeated game framework, we conduct two sets of analyses to explore how different recommendation objectives affect price dynamics and the final equilibrium. First, we conduct numerical simulations under two distinct recommender system objectives, revenue maximization and utility maximization, and compare these results to a baseline scenario in which product display is randomized. We find that revenue-maximization recommender systems tend to exacerbate algorithmic collusion, leading to elevated market prices. In contrast, utility-maximization systems yield non-collusive outcomes characterized by lower equilibrium prices. Second, we examine a weighted objective that interpolates between revenue maximization and utility maximization, capturing recommender systems that balance the interests of multiple stakeholders. We find a positive association between the weight placed on revenue and both equilibrium price and seller revenue, but a negative asso-

ciation with consumer utility. Comparing each weighted outcome against the baseline using a grid search method, we identify two notable points at which the recommender system matches the baseline along different dimensions: one weight achieves a similar price level while delivering substantially greater consumer utility, and the other weight yields comparable consumer utility while generating considerably higher seller revenue. These two points suggest two distinct criteria for evaluating the consumer welfare implications of recommender systems and indicate potential room for Pareto improvement.

Beyond the objective of the recommender system, another significant factor it can predefine is the number of products displayed (RQ2). This dimension holds considerable economic importance, as it directly influences search friction and thus the intensity of competition among sellers (Moraga-González et al. 2023). Without the interaction between recommender systems and pricing algorithms, one may expect that displaying more products always improves the platform's objectives, based on the choice modeling literature (Train 2009). The underlying rationale is that, with a wider range of product options, consumers can bypass items they deem less desirable and choose those that offer higher utility. Therefore, when the platform optimizes consumer utility, the welfare derived from a product set increases as more products are made available (Train 2009). Meanwhile, when the platform optimizes revenue, a wider product set can expand the relative value of the recommendation set versus the outside option (i.e., not buying from the platform), thereby increasing revenue (Train 2009).

By running simulations with varying numbers of displayed products within the same repeated game framework, we uncover a counterintuitive "more is less" effect: under a given objective, increasing the number of displayed products can sometimes reduce the achievement of that objective. This outcome emerges from two competing mechanisms. Under the utility-maximization objective, on one hand, offering more products improves consumers' chances of discovering those that yield higher utility (Train 2009). On the other hand, expanding the set diminishes the platform's leverage in guiding prices towards competitive levels. Consequently, high-priced products retain exposure and continue to attract some con-

sumers, experiencing relatively smaller reductions in their rewards and thus sustaining higher equilibrium prices. When this latter effect dominates, providing more products ultimately harms consumer welfare. Under the revenue-maximization objective, although offering more products leads to larger sales chances on the platform (Train 2009), the platform's ability to push prices towards collusive levels is also weakened, thereby lowering the revenue gained.

From a theoretical perspective, the level of horizontal differentiation in the market tends to moderate the aforementioned mechanisms and market dynamics. This is because, on one hand, greater horizontal differentiation may allow consumers to find products that better match their preferences, thereby increasing the objective gain from a larger recommendation size. On the other hand, greater horizontal differentiation also softens price competition and lowers the difficulty of collusion, thereby changing the relative importance of the recommender systems' price manipulation ability. Therefore, we ask our third research question:

> *RQ3: How does horizontal differentiation moderate the recommendation size effect?*

To answer RQ3, we adjust the level of horizontal differentiation and rerun the simulations for RQ2. Interestingly, we find divergent patterns in the moderating role under different recommendation objectives: the "more is less" effect is more salient under high horizontal differentiation for the utility-maximization scenario, and more salient under low horizontal differentiation for the revenue-maximization scenario. This is because, under utility-maximization, as horizontal differentiation increases, the potential gains from searching among a larger recommendation set also increase. Meanwhile, higher horizontal differentiation enables sellers to collude more easily when more products are displayed. Consequently, two competing mechanisms intensify with higher horizontal differentiation, and the consumer utility loss from higher collusion levels can outweigh the potential utility gains derived from increased product variety. Under revenue-maximization, however, lower horizontal differentiation is where collusion is difficult to sustain and where the recommender system's power is most needed to achieve higher revenue. Moreover, with lower horizontal differentiation, the sales gain

from displaying more products is also limited. Therefore, both mechanisms help cultivate the "more is less" effect on revenue at lower levels of horizontal differentiation.

4.1.4 Summary of Contributions

Our work provides methodological, theoretical, and practical contributions. From a *methodological* perspective, we incorporate recommender systems into a repeated game framework and accordingly adjust the other model components. In doing so, we address the estimation of a complex structural search model by introducing the Hermite method into the simulated maximum likelihood procedure. We also derive the sequence of the game and equilibrium using multi-layer numerical integration, while relaxing several key assumptions from Calvano et al. (2020b). The resulting framework offers a valuable foundation for future analyses of analogous research questions.

From a *theoretical* perspective, we contribute to the broader economics of AI literature by examining the economic effects of cross-category AI-AI interactions between recommender systems and pricing algorithms. Our findings cannot be easily deduced from either the economics of recommender systems or the algorithmic collusion literature alone because of the feedback loop: the price input to the recommender system is generated by pricing algorithms that adapt based on a learned reward structure from past interactions rather than following canonical game-theoretic functions (Li et al. 2020); as for the pricing algorithms (Calvano et al. 2019), the reward generation is now affected by the recommender system's ability to manipulate consumers' attention allocation based on the platform's objective. This unique framework gives rise to our recommendation objective effect, recommendation size effect, as well as the moderating role of horizontal differentiation. Moreover, our weighted objective analyses yield two potential definitions to characterize and evaluate the fairness of recommender systems in the presence of pricing algorithms. Further, the "more is less" effect highlights a unique mechanism through which recommendation size translates to downstream welfare allocation via pricing manipulation ability, beyond the traditional choice modeling mechanism (Train 2009).

From a *practical* perspective, our results yield important insights for regulators, platforms, and sellers, which will be discussed more in Section 4.5. For instance, the regulators now need to regulate recommender systems for supracompetitive pricing instead of pricing algorithm providers alone, and the definitions for fairness need to be updated when both recommender systems and pricing algorithms are at play. Platforms can refine their recommendation policies by considering the pricing algorithms' dynamic learning process. Similarly, sellers might benefit from strategically adjusting the product differentiation from other sellers under different recommendation conditions.

In the subsequent sections, we present our repeated game framework, detailing its components and the challenges addressed in Section 4.2. Section 4.3 outlines our parameterization strategy, describes the empirical data, and explains the procedures for estimating the structural search model. Building on this foundation, we present our findings through numerical simulations in Section 4.4. Finally, Section 4.5 concludes by discussing the *practical implications* of our study and suggesting directions for *future research*.

4.2 Dynamic Pricing Competition Moderated by the Platform's Recommendation System

4.2.1 General Repeated Game Framework

Despite its critical importance, the role of platform recommendations in shaping algorithmic pricing strategies remains underexplored in the existing literature. Previous studies on algorithmic collusion, largely following the framework established by Calvano et al. (2020b), typically include a logit demand model, employ Q-learning as the pricing algorithm, and consider a simultaneous game sequence. To more effectively address our research questions, we develop a novel repeated game framework that incorporates the platform's recommendation system and modifies these three elements accordingly.

This section presents the overall structure of our repeated game model, specifies the sequence of play, and provides a concise summary of the principal challenges encountered. In the subsections that follow, we discuss each key element in detail: Section 4.2.2 introduces

the demand model, Section 4.2.3 details the pricing algorithm, and Section 4.2.4 outlines the platform's recommendation algorithm. For ease of reference, Appendix B.1 compiles the central notations employed in this study. Although additional symbols may emerge as we formally characterize and estimate the customer demand model, we refrain from listing them here to preserve clarity.

Specifically, consider a marketplace comprising J heterogeneous products ($j \in \{1, 2, \ldots, J\}$)[3], each associated with an attribute vector $\boldsymbol{X}_j$, as well as an outside option ($j = 0$). At each discrete time period t, sellers independently and simultaneously set their prices p_j^t. Concurrently, a cohort of consumers, indexed by i and each characterized by an attribute vector $\boldsymbol{A}_i$, enters the platform. The platform then constructs a recommendation set $\boldsymbol{B}_i$ for each consumer and establishes product rankings $pos_{i,j}$ to optimize its own objective. Upon viewing these recommendations, each consumer i undertakes a search process and ultimately purchases product k, yielding utility $u_{i,k}$. As a result, each seller realizes a reward r_j^t, enabling it to iteratively refine its pricing algorithm over time. Meanwhile, the platform also accrues a reward m^t. This pseudocode for the cyclical process is also summarized in Appendix B.2.

The repeated game process described above builds on the foundational work of Calvano et al. (2020b), a benchmark study in the field of algorithmic collusion that has influenced numerous subsequent investigations (Dou et al. 2024, Rocher et al. 2023). However, incorporating recommendation systems into this framework introduces a set of additional methodological complexities.

First, the introduction of recommender systems necessitates modeling consumer decision-making amid multiple simultaneously presented product recommendations. Consumers typically encounter a ranked list of products and need to devote time and effort to assess those of interest (Moraga-González et al. 2023). Additionally, position biases may influence their choices (Ursu 2018). In response, we develop a model that incorporates both consumer search behaviors and position biases, as discussed in Section 4.2.2.

[3]We assume a one-to-one mapping between sellers and products. If a seller offers multiple products with distinct dynamic pricing strategies, each product can be treated as a separate seller.

Second, consumers exhibit inherent heterogeneity, requiring sufficient flexibility in both the demand model and the associated personalized recommendation algorithms (Chung et al. 2024, Morozov et al. 2021, Zhang et al. 2019). Accounting for such realities complicates the estimation of demand models and the optimization of recommendation strategies. We address these challenges in Sections 4.2.2, 4.2.4, and 4.3.

Finally, while the framework introduced by Calvano et al. (2020b) is notably adaptable and involves relatively fewer stringent assumptions, it still presupposes forward-looking pricing algorithms and simultaneous decision-making among sellers. In Section 4.4.4, we critically examine these assumptions and relax them, thereby extending the model's scope and robustness.

4.2.2 Consumer Demand Model

In this subsection, we examine a structural search model to elucidate the decision-making processes of consumers when presented with platform-based recommendations. We adopt the sequential search framework for several reasons. First, consumers typically lack complete information about all available products in the market and face significant costs when seeking information (Moraga-González et al. 2023, Mortensen 2011). This aligns with the objectives of recommender systems, which aim to alleviate the challenges of information overload and search frictions (Zhang et al. 2019). The sequential search model addresses these issues by explicitly incorporating the search costs incurred by consumers in accessing product information (Ursu et al. 2023, Weitzman 1979). Second, when multiple products are recommended, position bias may emerge, a phenomenon that the sequential search framework can effectively capture (Chung et al. 2024, Ursu 2018). Third, the sequential search model allows for the incorporation of heterogeneous preferences and search costs among consumers, thereby facilitating personalized recommendations within this framework (Ursu et al. 2023, Morozov et al. 2021).

Specifically, each consumer i is presented with a recommendation set $\boldsymbol{B_i}$, resulting from

the platform's top-K recommendation algorithm ($K \leq J$)[4]. Here, $\boldsymbol{B_i}$ denotes a consumer-specific subset of all products ($j = 1, 2, \ldots, J$) in the marketplace. The position of product j for consumer i is represented by $pos_{i,j}$ ($pos_{i,j} \in \{1, 2, \ldots, K\}$). From the consumer's perspective, the value of each product is quantified by the purchase utility function. This utility, $u_{i,j}$, derived by consumer i from product j, is decomposed into two components:

$$u_{i,j} = v_{i,j} + \xi_{i,j} \tag{4.1}$$

The first component, $v_{i,j}$, represents the utility observable on the search results page *prior* to clicking. In contrast, the second component, $\xi_{i,j}$, which follows a normal distribution $N(0, \delta_\xi^2)$, denotes the utility revealed only *after* navigating to the product's webpage.

Moreover, each product j is characterized by its price p_j and a vector of attributes $\boldsymbol{X_j}$, which are disclosed to consumers prior to their search. Similarly, consumer i has a series of attributes $\boldsymbol{A}_i$. Consequently, the decision-making context for each consumer i includes the following elements: the consumer i with attributes $\boldsymbol{A_i}$, product j with its price p_j and attributes $\boldsymbol{X_j}$ ($j \in \boldsymbol{B_i}$), and the ranking of these product $\psi_{i,j}$. Consistent with prior research on sequential search models, we postulate a linear parametric relationship between these observed attributes and the consumer's pre-search expected utility (Chung et al. 2024, Ursu 2018):

$$v_{i,j} = \beta_i * p_j + \boldsymbol{\Gamma'} * \boldsymbol{X_j} + \boldsymbol{\Theta'} * \boldsymbol{A_i} \tag{4.2}$$

In addition to the utility derived from purchasing products, consumers incur a cost associated with the search required to ascertain $\xi_{i,j}$. This cost is encapsulated by the search cost function $c_{i,j}$, which, in accordance with extant literature, adheres to an exponential function contingent upon the product's ranking:

$$c_{i,j} = \exp\left(\tau_i + \varphi \cdot pos_{i,j}\right) \tag{4.3}$$

Given consumers' uncertainty regarding the precise magnitude of $\xi_{i,j}$—with only its probabilistic distribution known—it becomes necessary for consumers to balance the potential

[4]In this section, the time index t is omitted for simplicity, although p_j^t and A_i^t may vary over time.

utility gains from discovering products that offer higher purchase utility, $u_{i,j}$, against the associated search costs, $c_{i,j}$. This trade-off in consumer decision-making is encapsulated by the concept of reservation utility (Weitzman 1979). The reservation utility for a product, denoted by $z_{i,j}$, is defined as the utility level at which the consumer is indifferent between continuing the search for product j or obtaining a utility of $z_{i,j}$ with certainty. Mathematically, this represents the equivalence between the marginal utility derived from searching for product j and the marginal cost incurred, expressed as:

$$\begin{aligned} c_{i,j} &= \int_{z_{i,j}}^{\infty} (u_{i,j} - z_{i,j})\, f(u_{i,j}) du_{i,j} \\ &= \int_{z_{i,j}-v_{i,j}}^{\infty} (v_{i,j} + \xi_{i,j} - z_{i,j})\, f(\xi_{i,j}) d\xi_{i,j} \end{aligned} \tag{4.4}$$

Incorporating purchase utility and search costs, the framework that governs consumer search and selection behavior is delineated by three fundamental rules: the selection, stopping, and choice rules. These rules comprehensively characterize consumer behavior, as detailed below:

1. **Selection Rule**: This rule stipulates that consumers rank products in descending order based on their reservation utilities, denoted by $z_{i,j}$. This ranking determines the sequence in which products are searched. Specifically, the product with the highest reservation utility, $z_{i,j}$, is selected as the next candidate for evaluation.

2. **Stopping Rule**: After evaluating the set of products $\boldsymbol{H_i}$ and observing their post-search utilities[5], consumer i needs to decide whether to continue the search or terminate it. The search proceeds if the highest utility realized from the searched products $\boldsymbol{B_i}$ is less than the maximum reservation utility among the yet-to-be-searched options, $\boldsymbol{B_i} \setminus \boldsymbol{H_i}$:

$$\max_{j \in \boldsymbol{H_i}} u_{i,j} < \max_{j \in \boldsymbol{B_i} \setminus \boldsymbol{H_i}} z_{i,j} \tag{4.5}$$

[5]Consistent with prior literature, we assume that consumers are aware of their outside option before searching recommended alternatives. Consequently, we further assume $H_i(0) = 0$ for notational simplicity (Ursu et al. 2023).

Conversely, the search is terminated if the above condition is not met. In other words, the maximal utility among all the searched products exceeds the reservation utilities of all the unsearched products. Additionally, the search process also terminates if all options have been searched.

3. **Choice Rule**: Upon cessation of the search, the consumer selects the product k that exhibits the highest utility from all the searched products:

$$u_{i,k} \geq \max_{j \in \boldsymbol{H_i}} u_{i,j} \tag{4.6}$$

By integrating these rules, consumers make search and purchase decisions based on available information. The entire process is visualized in Appendix B.4. Section 4.3 provides further details on how to estimate this model and derive parameters from the data.

4.2.3 Pricing Algorithms Adopted by Sellers

We employ canonical Q-learning to model the pricing algorithm, consistent with prior research on algorithmic collusion (Calvano et al. 2020b, Dou et al. 2024, Wang et al. 2023b). Q-learning is well-suited for this setting for three reasons: it underpins many state-of-the-art reinforcement learning algorithms, its non-parametric reward structure and minimal parameterization facilitate clear interpretation, and it requires no prior knowledge of the economic environment, mirroring real-world sellers with limited market information.

Within this framework, each pricing agent observes a state defined by the previous period's prices across all products and selects a price from a discrete set of feasible actions. The agent then receives a reward, corresponding to the revenue generated by the demand model, and observes the transition to the next state. The agent's objective is to maximize total discounted cumulative revenue over an infinite horizon.

The core of Q-learning is the iterative estimation of a Q-function, which maps each state-action pair to the expected discounted reward of taking that action and behaving optimally thereafter. Starting from an arbitrary initialization, the agent updates its Q-values each

period by blending the previous estimate with newly observed information, at a rate governed by a learning rate parameter α. To balance the tension between exploring unfamiliar actions and exploiting currently known good ones, the agent follows an ϵ-greedy policy: it selects a random action with probability ϵ and the best-known action otherwise. The exploration rate decays exponentially over time at rate ω, reflecting growing confidence in the learned Q-values. A complete formal treatment, including the Bellman equation, Q-function update rule, and notation, is provided in Appendix B.6.

4.2.4 Recommender Systems Adopted by the Platform

In this subsection, we delineate the specific recommender systems employed by the platform. Although our study primarily investigates the economic impact of recommender systems rather than seeking technical enhancements (Li et al. 2020, 2018, Fleder and Hosanagar 2009), we aim to ensure that the algorithm aligns rigorously with the economic setting, as established in prior works on product display algorithms (Compiani et al. 2024, Derakhshan et al. 2022), while maintaining the ease of tracking economic properties and conducting counterfactual evaluations.

Achieving these objectives collectively presents inherent complexity. Even minor extensions to the choice model can render closed-form expressions for choice probabilities intractable, let alone the subsequent implications such as expected revenue or consumer utility (Train 2009). Consequently, even when the platform possesses precise knowledge of the demand model parameters and no heterogeneity exists, determining the optimal assortment and rankings typically necessitates a brute-force approach (Compiani et al. 2024). In our case, the demand model incorporates both observed and unobserved heterogeneity, further complicating the derivation of a recommendation strategy. To overcome these challenges, we perform multi-layer numerical integration over the distributions of these heterogeneities and the taste shocks, followed by a brute-force search to identify the optimal recommendations based on specific objective functions (Compiani et al. 2024).

As discussed in Section 4.1, previous economic and optimization studies predominantly

consider revenue and consumer utility as the primary objective functions for platforms (Compiani et al. 2024, Derakhshan et al. 2022). Consequently, we design the recommendation algorithms to optimize these two objectives, aligning with our first research question: *How does the objective of the recommender system influence the algorithmic pricing game?* For comparative analysis, we also consider a baseline scenario in which the platform presents each consumer with a randomly assorted and ranked selection of products.

The **revenue-maximization** recommender system aims to maximize the total revenue across all sellers on the market (Compiani et al. 2024). This objective closely aligns with platforms that generate income by collecting a portion of sellers' revenue as commission fees, as observed in prominent platforms such as Amazon and Tmall. For each consumer, the platform observes her demographic attributes but not her exact unobserved preference heterogeneities or post-search match values; only their distributions are accessible. The platform therefore enumerates all feasible recommendation sets (i.e., assortment and ranking combinations), estimates the expected revenue of each set through simulation-based numerical integration over both layers of unobserved heterogeneity, and selects the set that yields the highest expected revenue. The estimated structural search model from Section 4.2.2 is used within this simulation to predict consumer purchase decisions. The formal algorithm and notation are provided in Appendix B.7.

The **utility-maximization** recommender system is instead designed to enhance the utility experienced by consumers on the platform (Compiani et al. 2024). Utility serves as an economic measure of consumer satisfaction, which benefits the platform by improving consumer retention rates (Zhang et al. 2019, Ursu 2018). The computational procedure mirrors that of the revenue-maximization system, with the key distinction that the objective being integrated and compared across recommendation sets is consumer utility (purchase utility minus total search costs) rather than revenue. The detailed procedure is also provided in Appendix B.7.

4.3 Data, Structural Estimation, and Parameterization

Although we aim to design our framework to be robust against parameter variations and minimize reliance on assumptions, certain parameters still need to be specified prior to conducting numerical experiments. In this section, we describe the estimation of the structural search model using real-world data, the setup of the pricing algorithms, and the integration of these components for step-by-step simulations.

4.3.1 Parameterization of the Consumer Demand Model

In modeling consumer demand, we aim to derive consumers' price sensitivity, attribute preferences, search costs, and positional biases within a *real-world* context. To achieve this, we draw upon established literature and estimate our model using customer search and choice data made publicly available by Expedia (Compiani et al. 2024, Ursu 2018).

Data

The Expedia dataset is provided at the level of *search impressions*, defined as ordered lists of hotels and their characteristics presented to consumers in response to their trip queries. Notably, this dataset originates from a field experiment wherein consumers searching for hotels during the observational window were randomized into two groups: (i) those viewing Expedia's standard ranking, where hotels are ordered by relevance, and (ii) those viewing a random ranking, where hotels are displayed in a random order. Consequently, for the randomized group, the hotel display is exogenous to their attributes, eliminating the influence of the platform's existing algorithms on the customer modeling process. We utilize this random-ranking sample for our model estimation (Chung et al. 2024, Ursu 2018).

Following prior literature, we also partition the dataset by travel destination, recognizing that each destination may differ significantly in terms of hotels and travelers, and that hotels in different destinations do not directly compete with each other (Chung et al. 2024, Ursu 2018). We thus focus on the destination with the *largest* number of observations (destination

ID 8192). After applying additional filtering procedures to exclude null values and outliers, we obtain 35,769 observations, where each observation represents one available alternative. Further details on data cleaning, variable descriptions, and summary statistics are provided in Appendix B.11.

Estimation

As researchers, we observe consumer covariates, product attributes, consumers' search sequences, and their final purchase decisions. However, the utility parameters and search cost parameters are unobserved and must be estimated from the data. Following the sequential search literature, we assume that the post-search shocks are drawn from a standard normal distribution (Ursu 2018, Weitzman 1979).

The estimation exploits three key implications of the optimal sequential search rule: (i) a product searched later must have a reservation utility exceeding the realized utilities of previously searched products; (ii) all unsearched products must have reservation utilities below the highest realized utility among searched products; and (iii) the purchased product must yield the highest utility among all searched products. Together, these conditions define the probability of observing each consumer's search sequence and purchase decision as a function of the model parameters.

To estimate the model, we address two primary challenges. First, structural search models typically lack closed-form expressions relating parameters to choice probabilities. Consistent with prior literature, we therefore employ simulated maximum likelihood (SML) estimation using a logit-smoothed accept-reject simulator (Ursu et al. 2023). Second, to enhance the model's flexibility in analyzing the pricing game under the influence of recommendations, it is essential to account for unobservable heterogeneities in price sensitivity and search costs. We assume that both the price coefficient and search cost follow log-normal distributions, allowing them to vary across consumers while being partially linked to observable characteristics such as display position.

Although SML is commonly utilized for econometric models with random coefficients

(Train 2009), incorporating an additional layer of simulation to capture unobservable heterogeneities substantially increases the computational burden. To mitigate this, we innovatively integrate Gaussian Hermite quadrature into the estimation procedure (Akşin et al. 2013). To the best of our knowledge, this represents the first application of the Hermite approach to estimate a structural search model incorporating unobservable heterogeneities. The formal likelihood construction, distributional assumptions, and detailed estimation procedure are provided in Appendix B.5. After addressing these challenges, we obtain the model estimates presented in Table B.3.

4.3.2 Parameterization of the Pricing Algorithms

In our repeated game framework, the pricing algorithms necessitate the specification of several hyperparameters prior to simulation. Since these parameters do not influence the qualitative outcomes, we adopt the hyperparameter settings from Calvano et al. (2020b), specifically assigning a learning rate of $\alpha = 0.15$ and a discount factor of $\delta = 0.95$. To accommodate the presence of three agents, we enhance exploration by utilizing a smaller exploration rate, $\omega = 2 \times 10^{-7}$. We define the minimum and maximum permissible prices for sellers as the 10th and 90th percentiles of the dataset's prices (i.e., 0.5 and 2.59), respectively. This price range is discretized into 14 equal intervals, resulting in 15 available pricing actions.

4.3.3 Integration into the Simulation

To evaluate the outcomes and address our research questions within the repeated game framework, we implement several additional steps. First, for our primary simulation, we consider a scenario involving three sellers, where the platform recommends two out of the three available products to each consumer in each period. Consistent with prior studies, this configuration represents the most fundamental case in which assortment and position biases are relevant, while avoiding excessive computational complexity (Calvano et al. 2020b).

Second, to establish the static characteristics of the sellers, we utilize the median values from the dataset, thereby creating a balanced market that facilitates easy interpretations

of effect sizes. However, this approach results in homogeneity among alternatives aside from price. To introduce horizontal differentiation, we add a zero-mean term of the form $(\mu \cdot N(0,1))$, where μ can be adjusted to simulate various economic environments and address our third research question[6].

Third, we employ the empirical distribution of all consumers from the dataset. This distribution, combined with the estimated structural search model, enables us to generate demand and thereby determine the rewards for both sellers and the platform. To compute the expected value of these rewards, we follow the same numerical integration procedure used for the recommender system, accounting for both consumer heterogeneity and taste shocks. Having established these elements, we integrate the estimated structural search model and the pricing algorithms into the repeated game framework to carry out the simulations.

4.4 Results

After completing these aforementioned steps, we proceed to conduct numerical experiments to address our research questions. Specifically, as outlined in Algorithm 2 of Appendix B.2, each simulation run extends over multiple periods. At the beginning of each period, each seller utilizes its respective pricing algorithm, as detailed in Section 4.2.3, to determine pricing decisions. Subsequently, the platform generates recommendations for each consumer following the strategy described in Section 4.2.4. Consumers then observe the recommended set and make purchase decisions based on the estimated structural search model presented in Section 4.2.2. Each seller realizes revenue and updates its algorithm accordingly. With parameters fixed as described in Section 4.3, the simulation framework is fully specified. In the subsequent three subsections, we vary different elements of the framework to evaluate how these variations produce different counterfactual outcomes, thereby systematically addressing each of our research questions.

---

[6]Since our utility magnitudes are comparable to those in Calvano et al. (2020b), we also vary μ from 0 to 0.5 to sufficiently explore its impact and employ $\mu = 0.25$ to illustrate the main results.

4.4.1 Effect of the Recommender Systems' Objectives

As motivated in our *RQ1*, the underlying objective of a recommender system directly shapes which products are highlighted for consumers and, consequently, influences sellers' rewards and strategic behavior. To evaluate how different objectives influence the dynamics of the algorithmic game, we conduct a series of numerical experiments comparing equilibrium outcomes (prices, overall revenue, and consumer utility) under three distinct scenarios as specified in Section 4.2.4: (i) a revenue-maximization recommender system, which seeks to increase total platform-wide revenue and aligns with the platform's economic interests through commission fees, (ii) a utility-maximization recommender system, which aims to enhance consumer surplus and potentially benefits the platform by fostering consumer retention, and (iii) a baseline scenario without recommendation optimization as a point of reference.

For each scenario, we conduct 50 simulation runs, each spanning 20,000,000 periods[7]. At each period, we record the average price and revenue for the three sellers, as well as the average consumer utility across all consumers. These values are then averaged over the 50 runs, and their means from the final 1,000 periods are reported in Table 4.1. Standard deviations, presented in parentheses, provide insight into the degree of end-game fluctuation.

Table 4.1: Price, Revenue, and Utility Comparison Across Objectives

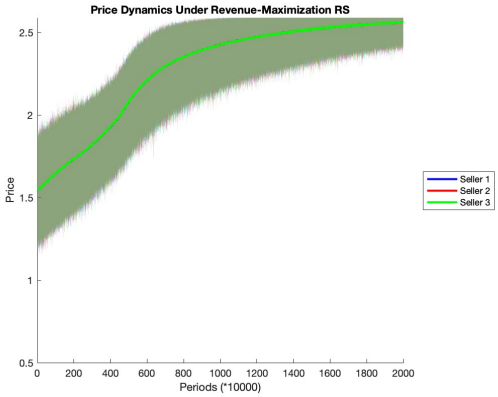
	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	2.5657 (0.0159)	1.0078 (0.0509)	1.7108 (0.0159)
Revenue	0.2681 (0.0006)	0.1278 (0.0043)	0.2228 (0.0013)
Utility	0.3273 (0.0016)	0.6336 (0.0070)	0.4668 (0.0024)

Our findings indicate that the objectives of recommender systems substantially shape

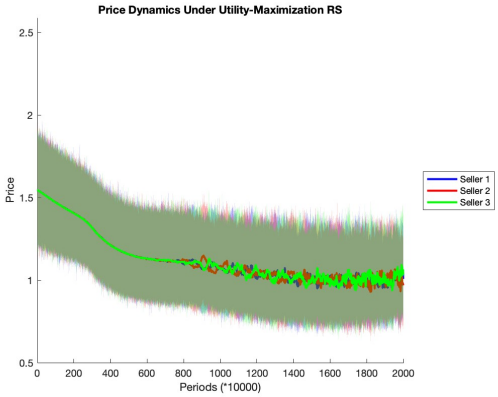
[7]Q-learning agents often converge slowly because only one element is updated in each period (Calvano et al. 2020b). However, as shown in Figure 4.2, the game's duration primarily affects the effect magnitude rather than the direction.

the trajectory of pricing strategies and their eventual equilibrium outcomes. Under revenue-maximization, sellers gravitate toward more collusive behavior, resulting in higher equilibrium prices (2.5657) and revenue (0.2681) at the expense of consumer utility (0.3273). Conversely, when the system aims to maximize consumer utility, the equilibrium is less collusive, featuring lower prices (1.0078) and higher consumer utility (0.6336), but reduced seller revenue (0.1278).

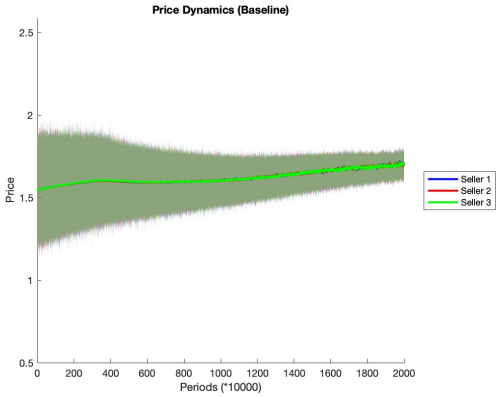
Figure 4.2 illustrates the temporal evolution of prices under the three scenarios, further corroborating our findings. In the revenue-maximization setting, sellers gradually and stably converge to higher prices. Under utility-maximization, they tend to settle at lower prices but exhibit greater volatility in their pricing behavior.



(a) Revenue-Maximization RS



(b) Utility-Maximization RS



(c) Baseline

Figure 4.2: Learning Dynamics

Note: The points on the continuous lines depict the moving average, with a window size of 10,000, of the average price observed across 50 experiments at each t. The shaded region's upper and lower limits indicate the highest and lowest values occurring within the window.

The fluctuation under utility-maximization recommender systems is noteworthy. Each seller seeks to optimize its own revenue, which diverges from the platform's objective of enhancing consumer utility. Higher-priced sellers face reduced exposure and fewer sales, yet each transaction generates greater revenue per unit. Lower-priced sellers benefit from

increased exposure and sales volume, but at lower per-sale margins. This creates opposing incentives: the highest-priced sellers are motivated to lower prices to compete for exposure, while the lowest-priced sellers are tempted to raise prices to capture higher margins while retaining their favorable positioning. When prices are close or tied, sellers face a dynamic trade-off between undercutting to secure the recommender system's exposure advantage and raising prices to extract more revenue per sale, depending on the competitive configuration. Consequently, although the utility-maximizing recommender system penalizes higher-priced sellers and pushes prices toward competitive levels, this tension between exposure and margin incentives sustains considerable price fluctuation.

Conversely, the revenue-maximization scenario converges rapidly and maintains relative stability. Both sellers and the platform are incentivized to sustain higher prices. When a seller raises its price, the resulting decline in purchase likelihood may be offset by increased product exposure from the revenue-maximizing recommender system, which favors higher-margin products. Prices can therefore escalate to collusive levels where all products in the recommendation set receive equal exposure. If a seller deviates from this equilibrium, the platform responds by deprioritizing the deviator, as the deviation reduces overall revenue from the recommendation set. The deviating seller consequently suffers significantly lower exposure and rewards, while non-deviating sellers receive even higher rewards, creating no incentive for others to follow suit. This self-enforcing mechanism ensures that the collusive price equilibrium is both achievable and stable.

Furthermore, certain recommender systems may balance the interests of various stakeholders. To capture this, we modify the objective of the recommender system to a weighted average of revenue and consumer utility, with the weight ranging between 0 (utility-maximizing) and 1 (revenue-maximizing). For each weight, the dynamics and resulting equilibrium differ. In general, there is a positive association between the weight and both price and revenue, but a negative association between the weight and consumer utility, as shown in Appendix B.18. Across the continuous weight space, we conduct a grid search to compare each outcome

against the baseline scenario[8]. Interestingly, we identify two distinct points that align with the baseline scenario along different dimensions: when the weight is 0.38, the equilibrium price (1.7131) is close to the baseline price (1.7108), with a similar level of revenue (0.2288 versus 0.2228)[9], but substantially greater consumer utility (0.4884 $>$ 0.4668). When the weight is 0.43, consumer utility is similar to the baseline (0.4673 versus 0.4668), yet seller revenue is considerably higher (0.2398 $>$ 0.2228), along with the price (1.9433 $>$ 1.7108). These two points suggest distinct criteria for evaluating the consumer protection role of recommender systems, one based on price and the other on utility, and the discrepancy between them reveals potential room for Pareto improvement. The values are also shown in Appendix B.8 and further discussion is provided in Section 4.5.

4.4.2 Effect of the Recommendation Size

Apart from the objectives of recommender systems, another economic feature of the systems are the recommendation size. As motivated in our second research question (*RQ2*), this dimension holds significant economic importance, as it directly influences search friction and competition among sellers (Moraga-González et al. 2023).

To assess the impact of recommendation size and address our *RQ2*, we conduct numerical experiments with $K = 1, 2, 3$ under both Top-K utility maximization and Top-K revenue maximization recommender systems. For each scenario, we perform 50 simulation runs and record the average equilibrium price, revenue, and utility in Tables 4.2 and 4.3.

Contrary to conventional wisdom and the choice modeling literature, increasing the number of displayed products does not consistently improve the recommender system's objective. Under utility maximization, as illustrated in the third row of Table 4.2, consumer utility first rises and then declines as the recommendation set expands (0.5533 $\rightarrow$ 0.6336 $\rightarrow$ 0.6023).

[8]The grid increment is 0.01, since each weight requires a separate simulation run that takes more than 10 hours. A finer grid would demand prohibitive computational resources.

[9]For this and all subsequent comparisons, the first value corresponds to the weighted recommendation scenario, while the second corresponds to the baseline.

This seemingly counterintuitive result arises from two competing mechanisms. On the one hand, displaying more products provides consumers with greater opportunities to identify products that offer higher utility, potentially increasing their overall surplus. On the other hand, a larger number of displayed products diminishes the platform's ability to effectively influence prices through recommendations. High-priced products continue to receive exposure, and some consumers purchase them based on their horizontal preferences. Consequently, higher-priced products experience relatively smaller losses in their rewards, leading to relatively higher equilibrium prices after repeated interactions. This effect is reflected in the first row of Table 4.2 ($0.7489 \rightarrow 1.0078 \rightarrow 1.3633$). When the latter mechanism predominates, recommending more products can be detrimental from the consumer's perspective.

A parallel "more is less" phenomenon arises under revenue maximization through a similar mechanism. As shown in Table 4.3, platform revenue increases from $K = 1$ to $K = 2$ ($0.1890 \rightarrow 0.2681$) but then declines at $K = 3$ (0.2246). On the one hand, displaying more products expands sales opportunities within the platform relative to the outside option, which tends to increase revenue. On the other hand, the revenue-maximizing recommender system facilitates tacit collusion and helps sustain higher equilibrium prices. However, this collusive leverage weakens as the recommendation set grows, since a larger set of alternatives inevitably draws consumer attention regardless of the system's steering. The resulting decline in equilibrium prices (1.3942 at $K = 3$) negates the incremental demand gained from broader product exposure, ultimately reducing total revenue.

Table 4.2: Comparison Across Recommendation Sizes (Utility Maximization, $\mu = 0.25$)

	K = 1	**K = 2**	**K = 3**
Prices	0.7489 (0.0418)	1.0078 (0.0509)	1.3633 (0.0264)
Revenue	0.0765 (0.0018)	0.1278 (0.0043)	0.2108 (0.0028)
Utility	0.5533 (0.0032)	0.6336 (0.0070)	0.6023 (0.0044)

Table 4.3: Comparison Across Recommendation Sizes (Revenue Maximization, $\mu = 0.25$)

	K = 1	**K = 2**	**K = 3**
Prices	2.2690 (0.0316)	2.5657 (0.0159)	1.3942 (0.0246)
Revenue	0.1890 (0.0004)	0.2681 (0.0006)	0.2246 (0.0026)
Utility	0.3160 (0.0020)	0.3273 (0.0016)	0.5581 (0.0044)

4.4.3 Horizontal Differentiation Levels as the Moderator

Building on the mechanisms discussed in *RQ2*, horizontal differentiation can directly moderate the recommendation size effect by altering the intensity of competition. Moreover, differentiation itself constitutes a key dimension of sellers' strategic decision-making, as firms adjust their positioning in response to the platform's recommendations and competitive pressure from rival sellers. To investigate this moderating role and thereby address *RQ3*, we conduct a series of simulations by varying μ and examining scenarios where $K = 1, 2, 3$. We replicate the same analyses as in the previous subsection, presenting two sets of results: one under utility maximization in Table 4.4 and another under revenue maximization in Table 4.5. Each table contains two panels corresponding to lower ($\mu = 0$) and higher ($\mu = 0.5$) horizontal differentiation, respectively.

Surprisingly, our findings uncover contrasting patterns depending on the recommender system's objective. Under utility maximization, the "more is less" phenomenon emerges when horizontal differentiation is high (see Panel B in Table 4.4), whereas increasing the number of recommended products consistently enhances consumer utility when horizontal differentiation is low (see Panel A in Table 4.4). Under revenue maximization, however, the pattern reverses: the "more is less" effect appears when horizontal differentiation is low (see Panel A in Table 4.5), while it largely disappears when horizontal differentiation is high (see Panel B in Table 4.5).

Table 4.4: Comparison Across Recommendation Sizes (Utility Maximization)

Panel A (Low Horizontal Differentiation): $\mu = 0$			
	K = 1	**K = 2**	**K = 3**
Prices	0.7926 (0.0431)	0.9616 (0.0457)	1.0708 (0.0509)
Revenue	0.0770 (0.0021)	0.1109 (0.0032)	0.1351 (0.0043)
Utility	0.5289 (0.0037)	0.6051 (0.0052)	0.6176 (0.0074)
Panel B (High Horizontal Differentiation): $\mu = 0.5$			
	K = 1	**K = 2**	**K = 3**
Prices	0.8753 (0.0449)	1.1462 (0.0604)	1.8106 (0.0166)
Revenue	0.0796 (0.0014)	0.1514 (0.0060)	0.2841 (0.0020)
Utility	0.6012 (0.0026)	0.7108 (0.0093)	0.6431 (0.0032)

Table 4.5: Comparison Across Recommendation Sizes (Revenue Maximization)

Panel A (Low Horizontal Differentiation): $\mu = 0$			
	K = 1	**K = 2**	**K = 3**
Prices	2.3084 (0.0271)	2.0622 (0.0231)	1.1000 (0.0548)
Revenue	0.1826 (0.0006)	0.2316 (0.0008)	0.1444 (0.0053)
Utility	0.2971 (0.0024)	0.3812 (0.0029)	0.5722 (0.0081)
Panel B (High Horizontal Differentiation): $\mu = 0.5$			
	K = 1	**K = 2**	**K = 3**
Prices	2.3851 (0.0234)	2.5665 (0.0154)	1.8500 (0.0169)
Revenue	0.2123 (0.0006)	0.3061 (0.0006)	0.3002 (0.0020)
Utility	0.3367 (0.0018)	0.4063 (0.0014)	0.5999 (0.0032)

These outcomes can be understood through the distinct competitive dynamics induced by each objective. Under utility maximization, the recommender system strengthens price competition by steering consumers toward products that maximize their surplus. When horizontal differentiation is low, competition among firms is already organically intense regardless of recommendation size, so expanding the recommendation set further benefits consumers

by providing a wider consideration set. When horizontal differentiation is high, however, competition is inherently weaker because consumers' choices become more preference-driven rather than price-driven, which facilitates tacit collusion. In this regime, the recommender system's ability to discipline prices is more critical but undermined by a larger recommendation set. Although higher horizontal differentiation can also make searching among a larger recommendation set more beneficial to consumer utility, this gain may not offset the utility loss due to higher collusion levels. Consequently, the "more is less" effect under utility maximization manifests when horizontal differentiation is high.

Under revenue maximization, the recommender system instead softens competition by steering consumers toward higher-margin products, thereby facilitating collusion. When horizontal differentiation is high, this collusive dynamic is naturally strong, allowing firms to sustain elevated prices even with a larger recommendation size. Moreover, a larger recommendation set translates into additional sales opportunities, especially when products are highly differentiated. These two mechanisms collectively help attenuate the "more is less" effect. In contrast, when horizontal differentiation is low, price competition among sellers is inherently intense, making the recommender system's ability to facilitate collusion more critical. As the recommendation size expands, this capacity diminishes and hurts revenue. Meanwhile, low horizontal differentiation also makes the additional sales from a larger recommendation set less salient. As a result, the "more is less" effect under revenue maximization emerges when horizontal differentiation is low.

4.4.4 Framework Comparison and Assumption Relaxation

To incorporate recommender systems and address our research questions, we develop a repeated game framework that builds upon, yet significantly differs from, the framework proposed by Calvano et al. (2020b). We nonetheless retain several assumptions from Calvano et al. (2020b), and therefore conduct a series of extended analyses to examine these assumptions individually, providing a more comprehensive perspective. All comparisons and extended analyses are summarized in Table 4.6 and Appendix B.3, which collectively

strengthen the robustness of our findings.

Table 4.6: Framework Comparison and Summary of Extensions

	Calvano et al. (2020b)	This Work
Demand Model: Model Form	Reduced-form Models	Structural Models (more in Appendix B.4, B.9, B.10)
Demand Model: Data Usage	No data	Real-world dataset (more in Appendix B.11, B.12)
Demand Model: Choice Set	All the products	Recommended Products
Demand Model: Position Bias	Not Applicable	Yes
Pricing Algorithm: Sufficient Exploration	Yes	Relaxed (more in Appendix B.13)
Pricing Algorithm: Forward-Looking	Yes	Relaxed (more in Appendix B.14)
Pricing Algorithm: Objective	Revenue	Relaxed (more in Appendix B.15)
Game Sequence	Simultaneous	Relaxed (more in Appendix B.16)
Recommender Systems	No	Yes (More in Appendix B.17 and B.18)
Objective Misalignment	Not Applicable	Yes

4.5 Implications and Conclusions

We build a repeated game framework to incorporate the AI-AI interaction between recommender systems and pricing algorithms, and demonstrate how different recommender system configurations lead to distinct dynamics and final equilibria. Specifically, we document how varying objectives, from emphasizing consumer utility to revenue, shift the equilibrium price from a competitive level to a collusive one. In this process, we identify two weight configurations comparable to the baseline scenario along different dimensions. Further, we uncover a

"more is less" effect, where increasing the recommendation size with the intention of further improving the objective may actually reduce the objective gain. The emergence of this effect also depends on the level of horizontal differentiation. Building on these findings, we discuss managerial implications for regulators, platforms, and sellers in Section 4.5.1. In Section 4.5.2, we further situate our work within the literature, discuss its limitations, and outline future directions for deepening our understanding of the economics of AI-AI interactions and related research streams.

4.5.1 Managerial and Policy Implications

For regulators, we demonstrate that the objectives of recommender systems significantly influence price dynamics, collusion levels among sellers, and the resulting welfare allocation across market participants. Given the pivotal role of recommender systems, when antitrust legislation seeks to curb supracompetitive pricing, it may not be sufficient to hold only AI pricing providers responsible, as current practice often does (Robertson 2022, Colangelo 2021). Regulators should also ensure that platforms' recommender systems are designed responsibly, given that platforms possess more comprehensive data and greater capacity for market manipulation compared with individual sellers.

To regulate the recommender systems, new criteria are needed to evaluate their fairness in the presence of pricing algorithms. Based on our grid search across weighted objectives, we highlight two notable configurations. The first leads to a similar price and seller revenue compared with the no-recommendation baseline, but yields higher consumer utility; we define this as "price-equivalence" fairness. The second leads to similar consumer utility compared with the baseline, but results in higher prices and seller revenue; we define this as "utility-equivalence" fairness. This distinction is analogous to the trade-off between "equal treatment" and "equal opportunity" in the AI fairness literature (Fu et al. 2022). From the perspective of regulating platforms to protect consumer utility, price equivalence operates at a surface level but is in fact a stricter criterion than utility equivalence. Under utility equivalence, the focus shifts to the outcome that antitrust regulations typically prioritize

(i.e., consumer protection), but it permits higher prices, greater seller revenue, and higher platform commission.

As for the platform, it possesses the most information to implement this framework. The platform can directly observe high-frequency price changes and identify the usage of pricing algorithms. It also has access to consumer clickstream data for modeling consumer behavior and designing recommender systems accordingly. Therefore, from a technical perspective, regardless of what objective the platform seeks to achieve, it is worthwhile to consider the impact of recommendations on pricing algorithms' learning. For example, when designing recommendation algorithms that incorporate long-term rewards, the state space and transition dynamics should account for the influence of recommendations on sellers' algorithmic learning: current recommendations alter not only immediate consumer responses but also the future pricing environment.

Further, if new regulatory policies are enacted or the platform chooses to self-regulate to protect consumer utility, our results still offer actionable insights. From an economic perspective, the weighted objective analyses reveal a Pareto improvement space between the "price equivalence" and "utility equivalence" thresholds. Within this range, sellers can earn higher revenue and platforms can collect higher commission fees, without diminishing consumer utility. This represents a mutually beneficial outcome for platforms, sellers, and consumers alike.

Finally, sellers have the opportunity to jointly optimize product design (a low-frequency decision with adjustment costs) alongside their pricing algorithms (a high-frequency decision). Specifically, in the utility-maximization case, higher horizontal differentiation amplifies the "more is less" effect. As the recommender system emphasizes utility, displaying fewer products becomes more aligned with the platform's objective. However, when the platform displays fewer products under high horizontal differentiation, sellers' revenue may actually be lower than in a scenario where more products are displayed under low horizontal differentiation. This creates an incentive for sellers to strategically reduce their differentiation from one another. The dynamics differ, however, under revenue maximization. With a revenue-

maximizing platform, regardless of whether the "more is less" effect is present, the platform's choices also align with sellers' preferences. Consequently, sellers have no incentive to deviate in order to manipulate the platform's recommendations, and increasing horizontal differentiation from competitors can improve their revenue. The optimal level of differentiation is thus primarily constrained by the cost of product development.

4.5.2 Conclusions

Our study also identifies several directions for future exploration by both researchers and practitioners. First, this study is situated within the field of the economics of AI. As more types of AI emerge and adoption becomes more widespread, it is increasingly important to consider the economic and societal impacts arising from interactions among various AI systems. Our work represents an initial step in examining AI-AI interactions across different types of AI within a unified framework.

Focusing on our specific topic, we study the interactions between pricing algorithms and recommender systems, and highlight that the underlying mechanisms originate from the recommender system's ability to reallocate consumer attention in accordance with the platform's objectives. From a theoretical perspective, our analyses are generalizable to settings where: i) consumers face information friction in accessing products, ii) a platform can guide consumers' attention in pursuit of its objective, and iii) the platform achieves this objective based on consumers' choices. In certain cases, our framework and findings may be directly applicable with differences in magnitude, such as when the platform's product display algorithms draw from the assortment optimization literature (Feldman et al. 2022), or when the platform provides a shopping agent with similarly specified objectives. In other cases, our framework would require adjustment. For example, shopping agents might be provided by a third party or designed by consumers themselves, in which case the attention allocation is no longer governed by the platform. Besides, if we consider advertisement auctions on the platform, sellers must pay for impressions, and the platform benefits monetarily in a more direct manner. Overall, there are numerous exciting research directions when we consider

the broader attention economics and how AI-AI interactions reshape it.

Furthermore, based on our findings, we document the potential benefits for sellers to jointly optimize pricing and product design, particularly by horizontally differentiating from competitors. There are further strategic responses that sellers can consider, such as adjusting vertical differentiation (e.g., improving quality through investment) and participating in advertising auctions. Incorporating these dimensions would present valuable opportunities for future research.

In addition, we focus on the scenario where consumers are presented with the recommended set and make their search and purchase decisions solely among these recommended products. In reality, some consumers may exhibit strong loyalty to specific products and purchase them irrespective of pricing competition or the behavior of recommender systems. Since these consumers do not influence the relative rewards across different pricing decisions, they do not affect pricing dynamics and can be safely excluded from our analyses. Other consumers may already be knowledgeable about specific products, for whom recommender systems serve solely as tools for additional information discovery. As the proportion of such informed consumers increases, the impact of recommender systems diminishes, and in the extreme case where all consumers are aware of all available products beforehand, recommender systems would lose their effectiveness. However, informed consumers affect only the magnitude, rather than the qualitative direction, of our findings. For clarity and ease of interpretation, we focus on the setting where consumers view and select from the recommendation set. Future research could explore the heterogeneous welfare impacts of recommender systems and pricing strategies on various types of consumers.

It is also worth noting that translating our findings into real-world implementation requires additional effort (Li and Tuzhilin 2024b, Wang et al. 2024, Bi et al. 2024, Li et al. 2023). Although this endeavor is beyond the scope of the present study, we encourage algorithm designers in both academia and industry to recognize and account for the strategic pricing decisions enabled by AI-driven pricing algorithms. By doing so, they can develop algorithms that enhance both overall welfare and the equitable distribution of benefits across

market participants.

Chapter 5

DEPERSONALIZING SEARCH ON SOCIAL MEDIA: A LARGE-SCALE FIELD EXPERIMENT

5.1 Introduction

Social media has grown into a central part of modern digital life. As of 2025, over 5.4 billion people worldwide, representing approximately 65% of the global population, actively use social media[1]. The global social media market was valued at over $208 billion in 2025 and is projected to exceed $389 billion by 2030[2]. These platforms host a wide variety of content, including text, images, short-form and long-form video, live streams, and interactive media. The content is produced and consumed by billions of users whose preferences and information needs differ substantially (Zhang et al. 2019).

The volume and diversity of available content make information overload a persistent challenge for users (Pariser 2011). **Personalization** has thus become essential for helping users navigate online platforms. It takes two primary forms. The first is search personalization, where platforms tailor the ranking of search results to individual users based on their past behavior and inferred preferences (Yoganarasimhan 2020, Ghose et al. 2014). The second is content recommendation, where platforms proactively suggest items that a user is likely to find relevant (Donnelly et al. 2024, Tam and Ho 2006). Both mechanisms aim to reduce the cost of finding relevant information and improve the match between users and items (Mao et al. 2023). Further, a more accurate match can increase user engagement and generate greater revenue (Kleinberg et al. 2024).

Two forces, however, increasingly push platforms to reduce the degree of personalization

[1]See https://datareportal.com/social-media-users.

[2]See https://www.researchandmarkets.com/reports/5781298/social-media-market-report.

they provide. The first is privacy. Growing public concern over data collection has prompted regulations such as the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), which grant users greater control over their personal data and limit the extent to which platforms can profile user behavior (Goldberg et al. 2024, Ke and Sudhir 2023). These constraints reduce the data available for personalization and can lower the quality of personalized services (Jia et al. 2021). The second is computational cost. Personalizing content for each individual user requires learning from and processing large volumes of behavioral data in real time, and these costs scale with the size of the user base and the content library (Wu et al. 2022, Gupta et al. 2020). When platforms respond to these pressures by scaling back personalization, search is a natural choice. This is because search results are anchored by the user's query, which provides a relevance signal even without personalization. Recommendation, by contrast, is personalized by nature. Without personalization, the platform has no basis for selecting which content to surface, and the recommendation section loses its core function. Therefore, in this paper, we *examine the impact of search depersonalization on a social media platform.*

Although depersonalization has been studied in other contexts, existing work focuses primarily on e-commerce settings (Yuan et al. 2025, Chen et al. 2023). Social media platforms differ from e-commerce platforms in important ways. First, social media users typically browse and consume (i.e., view) a large volume of content in a single session, and the content they view tends to be topically related. In e-commerce, by contrast, purchase decisions are higher-stakes, and users compare multiple products before buying; viewing a product is merely an intermediate step toward a transaction. Second, social media platforms support a different monetization channel through paid content, where content producers make some of their content available only to paying viewers. Moreover, paid content on social media appears organically alongside ordinary content in both the search and the recommendation channels.

These differences limit the applicability of both the theoretical models (the search models) and empirical findings from e-commerce to the social media setting. To fill this gap, we study

the impact of search depersonalization on social media along two granular dimensions. *First, how does search depersonalization affect user browsing and viewing behavior for both ordinary and paid content within the search channel of social media? Second, how does the effect of search depersonalization spill over to the recommendation channel of social media?*

Neither question can be answered from existing theory alone, because for each of them plausible mechanisms point in opposite directions. Consider first behavior within the search channel. When personalization is removed, a user with a clear target in mind receives a ranking less tailored to her tastes, so she may have to browse more results to find what she is looking for, and search sessions would lengthen. Yet the same degradation could just as well shorten them: if the first several results fit poorly, the user may take this as a signal that the platform cannot serve her query, forgo further effort on the focal platform, and stop the browsing process altogether. The relative impact on paid versus ordinary content within the search channel is equally unclear. Because paid content demands a monetary outlay on top of attention, a less tailored ranking might erode interest in paid content more severely, as users hesitate to pay for content whose fit they can no longer trust. At the same time, the price of paid content can itself signal quality, and precisely when well-matched content becomes harder to come across, that assurance may make paid content relatively more valuable than the ordinary alternatives surrounding it.

The direction of the cross-channel spillover is similarly ambiguous a priori. On the one hand, search and recommendation are parts of a single product, and users need not compartmentalize their experiences between them. A degraded search experience could lower a user's overall impression of the entire platform and breed a broader disengagement, depressing browsing and clicking in the recommendation channel as well. On the other hand, the recommendation channel remains fully personalized while search does not, and users may judge the content in one channel against what the other has recently offered. By lowering the criteria users bring to the recommendation channel, a worse search experience can make recommended content appear comparatively more attractive, leading users to browse more of it and click more often. The two accounts predict spillovers of opposite signs, and which

one prevails is an empirical matter.

Heterogeneity across users adds a further layer of ambiguity. One might expect the platform's most active users to be the least affected: their usage is habitual, their loyalty to the platform is established, and ingrained routines could carry their behavior through a change in ranking quality largely untouched. The opposite prediction is just as defensible. Active users have supplied the platform with the richest interaction histories, so they enjoy the most accurate personalization to begin with and stand to lose the most when it is removed; their search experience could deteriorate the furthest. Whether habit insulates active users or their data exposure magnifies the treatment effect cannot be settled in advance. These competing predictions, for each research question and for the heterogeneity beneath them, are what motivate the randomized field experiment at the center of this chapter.

We proceed in three steps. We first build a stylized model to formalize these intuitions. We then partner with a leading social media platform and conduct a large-scale randomized field experiment, in which roughly 71,000 users were randomly assigned to either the default search ranking or a depersonalized one, with the recommendation algorithm left untouched for everyone. Finally, we estimate the causal effects of depersonalization on browsing depth and on the click-through rates of ordinary and paid content in both channels, overall and separately by users' pre-treatment activity tiers.

The results discipline the competing intuitions in an informative way. Within the search channel, depersonalization makes search harder rather than driving users away: for each query, treated users scan 1.56% more content, while their click-through rate on ordinary content falls by 5.1% relative to the control mean and their clicking on paid content barely moves. The spillover results are more surprising. Rather than souring users on the platform as a whole, a worse search experience makes the untouched recommendation channel more engaging: browsing there rises by 3.19% and clicks on ordinary content become 14.5% more frequent, the opposite sign of the effect in the channel actually treated. The heterogeneity analysis uncovers a second reversal: the direct effects in search are strongest for the most active users, who have the most personalization to lose, whereas the positive spillover to the

recommendation channel is strongest for the least active users, whose expectations are least anchored in that channel itself. Finally, the paid click-through rate in the recommendation channel falls significantly only among the most active users, exactly the group whose spending is plausibly near its ceiling, so that longer sessions spread an essentially fixed amount of paid clicking over more impressions.

This chapter contributes to two literatures. The first is the literature on social media, which has examined how platforms shape information consumption, well-being, and market outcomes (Aridor 2025, Braghieri et al. 2022, Zhang et al. 2019, Pariser 2011). We add large-scale causal evidence on a design lever that this literature has largely left unexamined: how much personalization to build into search, and how its removal propagates through the rest of the platform. Our finding that the two discovery channels are linked through a behavioral standard implies that social media platforms cannot evaluate algorithmic changes channel by channel, because the experience delivered in one channel resets the bar against which the other is judged. The second is the literature on the economics of personalization, which has quantified the value of personalized search and recommendation (Donnelly et al. 2024, Yoganarasimhan 2020, Ghose et al. 2014, Fleder and Hosanagar 2009) and, more recently, the costs of privacy regulations that constrain it (Goldberg et al. 2024, Ke and Sudhir 2023, Jia et al. 2021). To this work we contribute three findings: the value of search personalization is heterogeneous and rises with the data a user has supplied; the cost of removing it is partly offset by engagement gains in the adjacent channel, so that single-channel evaluations overstate the loss; and the burden of depersonalization falls disproportionately on monetized content among the platform's most valuable users. Together, these results give platforms weighing privacy compliance and computational savings a fuller account of what scaling back personalization actually costs.

5.2 Institutional Background

We ran our field experiment on one of China's largest knowledge-sharing and social media platforms. Launched in the 2010s, the platform centers on a question-and-answer community.

By late 2024, it had roughly 81 million monthly active users, more than 71 million cumulative content creators, and a corpus of over 870 million pieces of user-generated content. What makes it particularly well-suited to our study is that users find content through two coexisting channels, illustrated in Figure 5.1[3].

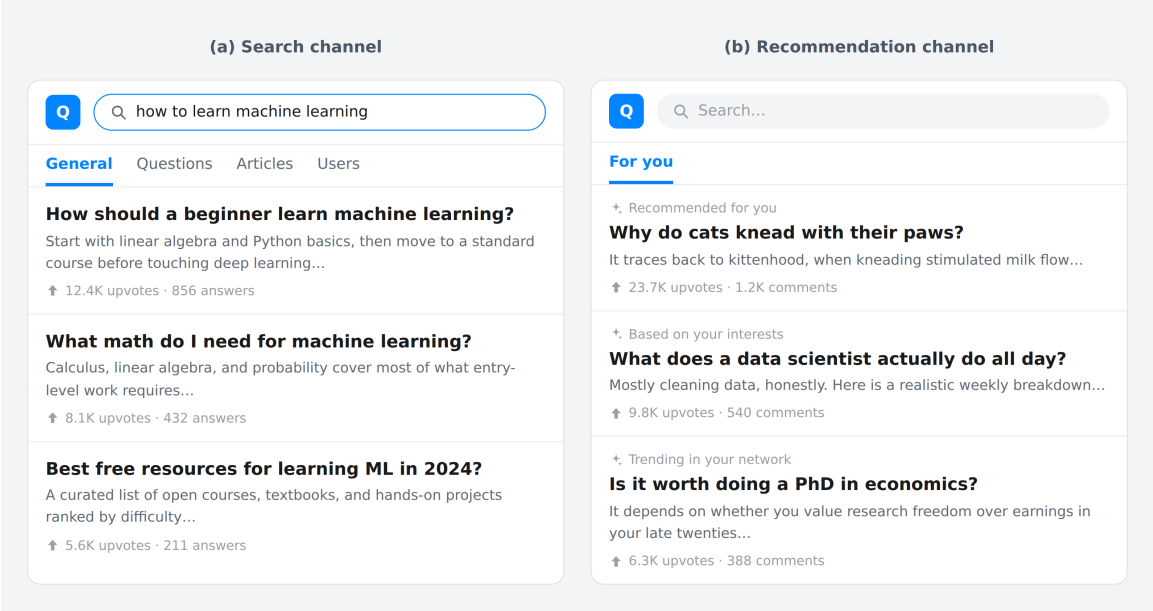


Figure 5.1: The Two Content-Discovery Channels on the Platform.

In the *search* channel (Panel a), a user types a query into the search bar and scrolls through a ranked list of results, much as she would on a general-purpose search engine; the results can be filtered by content type (questions, articles, or user profiles), and each entry displays a title, a snippet of the leading answer, and engagement statistics such as upvote and answer counts. The ranking reflects both the relevance of the content to the query and, under the platform's default configuration, the user's own history on the platform. In the *recommendation* channel (Panel b), accessed through the "For you" tab on the home page, the user submits no query at all. The platform instead assembles a personalized stream of

[3]This is an artificial figure to keep anonymity of the cooperating platform.

content from her past browsing and interaction records, her inferred interests, and activity trending within her network. The two channels therefore differ in who initiates the match between user and content: in search, the user states her intent and the platform responds; in recommendation, the platform proposes and the user decides whether to engage. Both channels carry organic alongside sponsored content, and a typical session moves freely between them. This dual-channel setup offers a natural laboratory for examining how depersonalizing the ranking algorithm in one channel shapes user behavior not only within that channel but also, through spillovers, in the other.

5.3 A Simple Stylized Framework

Before turning to the experiment, we sketch a deliberately simple framework. Its goal is not to provide a complete theoretical account, but to build intuition for how depersonalizing the search channel changes user behavior there and how that change spills over to the recommendation channel. The two channels are linked by a single object: the standard that the user carries from one channel into the other. All results below hold for a standard class of taste distributions without imposing any parametric form, and the formal proofs are collected in Appendix C.1.

5.3.1 Basic Environment

A representative user divides her attention between two channels: a *search* channel, which she enters with a goal in mind, and a *recommendation* channel, which she browses for its own sake. In each channel she scans a ranked list of items from the top down and decides whether to click on each. An item at rank n in channel j delivers value

$$v = \underbrace{m_j(n)}_{\text{predicted match}} + \underbrace{\eta}_{\text{private taste}} + \underbrace{q_\tau}_{\text{quality}}, \qquad j \in \{S, R\}. \tag{5.1}$$

The platform observes the match $m_j(n)$ and ranks items on it. The taste η is private to the user and varies from item to item, and the quality q_τ depends on the item's type τ. Items

are of two types: ordinary items are free ($\kappa_o = 0$), while paid items are sponsored and cost the user $\kappa_p = \kappa > 0$. A share $\pi \in (\frac{1}{2}, 1)$ of items is ordinary in each channel.

The platform's predicted match is highest at the top of the list and fades down the ranking,

$$m_j(n) = \rho_j(\bar{a} - \gamma n), \qquad \rho_j > 0, \;\; \gamma > 0, \tag{5.2}$$

where ρ_j measures how aggressively channel j personalizes. Our experiment corresponds to a depersonalization of search, that is, a fall in ρ_S. The recommendation channel's personalization ρ_R stays unchanged, so this channel reacts only through the standard that the user carries over, which we describe below. The private taste η is drawn independently across items from a distribution with cdf G, survival function $\bar{G} = 1 - G$, and density g.

Assumption 1 (Taste)**.** g is symmetric about 0, strictly positive, and log-concave (the standard shape restriction; Bagnoli and Bergstrom, 2005), with $\mathbb{E}|\eta| < \infty$. In particular, g is single-peaked at 0 and strictly decreasing on $(0, \infty)$.

Assumption 2 (Quality and price)**.** $q_p > q_o > 0$ and $0 < q_p - \kappa < q_o$. Equivalently, paid content must clear a higher bar to be clicked than ordinary content: the click thresholds defined below satisfy $\xi_{j,o} < \xi_{j,p}$.

This setup is the standard random-utility formulation of a binary click decision (Anderson, de Palma, and Thisse, 1992). All results are distribution-free within the log-concave class: the proofs use only the facts that g is strictly positive and falls away from its mode, and never rely on a parametric form.

5.3.2 How the User Behaves

In the search channel. The user enters the search channel with a goal-driven reservation value r_S and clicks an item when its value clears this reservation, which happens with probability

$$\phi_{S,\tau}(n) = \bar{G}\big(\xi_{S,\tau} - m_S(n)\big), \qquad \xi_{S,\tau} = r_S + \kappa_\tau - q_\tau. \tag{5.3}$$

She keeps scanning until she accumulates a target number of clicks k. This target pins down how deep she goes, namely the search depth b_S:

$$\int_0^{b_S} \Phi_S(n)\, dn = k, \qquad \Phi_S(n) = \pi\, \phi_{S,o}(n) + (1-\pi)\, \phi_{S,p}(n). \tag{5.4}$$

We track two outcomes. The first is the click-through rate of each content type, defined as clicks on type-τ content per type-τ impression, where an item is ordinary with probability $\pi_o = \pi$ and paid with probability $\pi_p = 1 - \pi$:

$$\mathrm{CTR}_{S,\tau} = \frac{\displaystyle\int_0^{b_S} \pi_\tau\, \phi_{S,\tau}(n)\, dn}{\displaystyle\int_0^{b_S} \pi_\tau\, dn} = \frac{\pi_\tau \displaystyle\int_0^{b_S} \phi_{S,\tau}(n)\, dn}{\pi_\tau\, b_S} = \frac{1}{b_S} \int_0^{b_S} \phi_{S,\tau}(n)\, dn, \qquad \tau \in \{o, p\}. \tag{5.5}$$

The second is the user's per-item surplus from the search experience, W_S/b_S, where

$$W_S = \int_0^{b_S} \Psi_S(m_S(n))\, dn - \mu_S\, b_S, \qquad \Psi_S(m) = \sum_\tau \pi_\tau\, \Psi_{S,\tau}(m), \tag{5.6}$$

$\Psi_{S,\tau}(m) = \mathbb{E}_\eta[\max\{m + \eta + q_\tau - \kappa_\tau - r_S,\, 0\}]$ is the expected click surplus and μ_S is the per-item attention cost.

The spillover. The quality of the search experience sets the standard that the user carries into the recommendation channel. When search becomes less rewarding per item, that is, when W_S/b_S falls, she enters the recommendation channel with a lower standard,

$$\omega = \bar{\omega} + \lambda\, \frac{W_S}{b_S}, \qquad \lambda > 0. \tag{5.7}$$

In the recommendation channel. The user clicks an ordinary item when its value clears her standard ω, which happens with probability

$$\phi_{R,o}(n) = \bar{G}\big(\omega - q_o - m_R(n)\big). \tag{5.8}$$

Paid content is different for two reasons: she must pay κ for it, and she has a fixed budget $B > 0$ for the session. She clicks paid items as they clear her standard until her cumulative

paid clicks reach B. After that point she spends nothing more and attends only to ordinary content. While the budget lasts, she clicks a paid item at rank n with probability

$$\phi_{R,p}(n) = \bar{G}\big(\omega + \kappa - q_p - m_R(n)\big), \tag{5.9}$$

Since only a share $1 - \pi$ of items is paid, her expected cumulative paid clicks are $P(n) = (1-\pi)\int_0^n \phi_{R,p}$. The budget is exhausted at the saturation rank n_B that solves $P(n_B) = B$, beyond which $\phi_{R,p} \equiv 0$. Because the match is strongest at the top, paid clicking is front-loaded and the budget is spent early in the session. We record the paid and ordinary click-through rates $\mathrm{CTR}_{R,\tau} = b_R^{-1}\int_0^{b_R}\phi_{R,\tau}$, similar to (5.5).

The user scrolls mainly for the free stream. She continues as long as the next item is worth her attention, and she stops at the depth b_R where the expected surplus of an ordinary item, $\Psi_{R,o}(m,\omega) = \mathbb{E}_\eta[\max\{m + \eta + q_o - \omega,\, 0\}]$, has fallen to her attention cost:

$$\Psi_{R,o}(m_R(b_R), \omega) = \mu_R. \tag{5.10}$$

Because the match fades down the ranking, the left side of (5.10) declines strictly with depth. Provided the top item clears the cost, the stopping rule has a unique interior solution $b_R > 0$, and a lower standard ω pushes the stopping point deeper.

5.3.3 Results and Intuition

We now follow a fall in ρ_S through the search channel, across the spillover, and into the recommendation channel. The analysis relies on two mild restrictions. The first bounds how far the user browses in the search channel.

Condition 1 (Browsing)**.** For every personalization level in the experiment range, $b_S \le n^* \equiv \bar{a}/\gamma$: the user does not routinely browse far past the rank where personalization stops helping. Because b_S is largest at the lowest intensity, it suffices to verify this condition at the lowest ρ_S considered.

The second restriction, used only for the differential effect in Proposition 2, requires personalization to be moderate.

Assumption 3 (Moderate personalization)**.** $\rho_S \bar{a} < \xi_{S,o}$, which keeps every search threshold argument $\xi_{S,\tau} - m_S(n)$ strictly positive on the browsed range, so that both arguments lie where g is decreasing. Depersonalization only slackens this restriction.

Proposition 1 (Search browsing deepens)**.** *Under Condition 1, $\partial b_S / \partial \rho_S < 0$: a fall in ρ_S raises the search depth b_S.*

When personalization weakens, the highly ranked items are less likely to be exactly what the user wants, so each one is less likely to be clicked. To still reach her target of k clicks, she has to look through more items, and her search sessions lengthen.

Proposition 2 (Search click-through rates fall, ordinary by more)**.** *Under Condition 1, depersonalizing search lowers the click-through rate of both content types. Adding Assumption 3, it lowers the ordinary rate by more than the paid rate:*

$$\frac{\partial \, \mathrm{CTR}_{S,o}}{\partial \rho_S} \; > \; \frac{\partial \, \mathrm{CTR}_{S,p}}{\partial \rho_S} \; > \; 0. \tag{5.11}$$

Weaker personalization dulls the match advantage of the top ranks, so a smaller fraction of the items shown gets clicked and both rates fall. The ordinary rate falls by more because ordinary content sits in the responsive middle of the user's taste range, where a change in match moves the click probability substantially. Paid content must already clear a high bar, so it is clicked rarely to begin with and its rate barely reacts to personalization. Note that this ordering is in levels (percentage points): because the paid rate starts from a small base, its smaller absolute movement need not be smaller in proportional terms.

Lemma 1 (The spillover)**.** *Under Condition 1, depersonalizing search lowers the per-item value of the search experience, so the user arrives at the recommendation channel with a lower standard ω and browses it more deeply:*

$$\frac{d}{d\rho_S}\left(\frac{W_S}{b_S}\right) > 0, \qquad \frac{\partial b_R}{\partial \omega} = -\frac{1}{\rho_R \gamma} < 0. \tag{5.12}$$

Personalization concentrates the good matches at the top of the list, where the user spends her clicks. Removing it spreads the match quality more thinly and lowers the value she

obtains per item. This in turn lowers the standard she carries into the recommendation channel, where the lower bar lets the free stream hold her attention for longer.

Proposition 3 (Recommendation channel: the ordinary click-through rate rises)**.** *As the user arrives with a lower standard, her ordinary click-through rate rises and she browses more deeply:* $d\,\mathrm{CTR}_{R,o}/d\omega < 0$ *and* $\partial b_R/\partial\omega < 0$*, so the lower* ω *of Lemma 1 raises* $\mathrm{CTR}_{R,o}$*.*

A lower standard raises the probability that the user clicks an ordinary item at every rank. Although she also browses more deeply, the higher click propensity dominates, so she clicks a larger share of the ordinary items she sees and her ordinary click-through rate rises.

Proposition 4 (Recommendation channel: the paid click-through rate could fall under budget constraint)**.** *(i) When the paid budget binds within the session, that is, when* $n_B \le b_R$*, depersonalizing search lowers the paid click-through rate even as it raises the ordinary one. The paid rate is*

$$\mathrm{CTR}_{R,p} = \frac{B}{(1-\pi)\,b_R}, \qquad \frac{d\,\mathrm{CTR}_{R,p}}{d\omega} = \frac{B}{(1-\pi)\,b_R^2\,\rho_R\gamma} > 0. \tag{5.13}$$

(ii) When the budget never binds, that is, when $n_B > b_R$ *both before and after the change, the paid click-through rate instead behaves like the ordinary one and rises:*

$$\frac{d\,\mathrm{CTR}_{R,p}}{d\omega} = \frac{\mathrm{CTR}_{R,p} - \phi_{R,p}(0)}{b_R\,\rho_R\gamma} < 0. \tag{5.14}$$

Two competing forces act on the paid click-through rate. On the one hand, the lower standard makes the user more willing to click any given paid item; this is the same force that raises the ordinary rate. On the other hand, the longer session spreads her paid clicks over more impressions, which dilutes the rate. Which force prevails depends on the budget. When the budget binds, the first force is shut down: paid clicking is front-loaded because the match is strongest at the top, the user spends her budget early, and the total number of paid clicks equals B no matter how far she browses. Only the dilution force remains, so the paid rate, which is proportional to $1/b_R$, falls even as the ordinary rate rises. When the budget never binds, no cap restrains the first force, paid content behaves just like ordinary

content, and both rates rise. Whether the paid rate moves with or against the ordinary rate is therefore an empirical question about whether users' paid budgets bind.

In sum, a single intervention produces opposite movements in the two channels. In the search channel, sessions lengthen but click-through rates fall. In the recommendation channel, sessions also lengthen and the ordinary click-through rate rises, while the paid click-through rate falls if paid budgets bind, since the early-spent paid clicks are diluted by the deeper browsing, and rises alongside the ordinary rate otherwise.

5.4 Experimental Design and Data

To empirically test the impact of depersonalizing search, we leverage a randomized field experiment on the partner platform. Section 5.4.1 describes the experimental design, and Section 5.4.2 details the variable construction process.

5.4.1 Experimental Design

On the platform, both channels rely on algorithmic ranking but differ in how content is retrieved and ordered. The search channel retrieves relevant content from the platform's full corpus based on the user's query and then ranks the candidates, whereas the recommendation channel retrieves and ranks content based on the user's recent interaction history on the platform. Our experiment ran from October 16 to October 27, 2025. We randomize at the user level: each user is randomly assigned to either the treatment or the control group at the start of the experiment and remains in that group throughout the entire experimental period. For users in the treatment group, the degree of personalization in the search channel's ranking process is reduced, while the recommendation channel's algorithm is left unchanged. For users in the control group, the personalization levels of both channels remain unchanged. This design isolates the effect of depersonalizing search while holding the recommendation channel fixed, allowing us to attribute the behavioral changes in both channels to the search-side intervention. Table 5.1 summarizes the experimental design.

Table 5.1: Experimental Design: Personalization by Group and Channel

	Search channel	Recommendation channel
Treatment group	Personalization reduced	Unchanged
Control group	Unchanged	Unchanged

5.4.2 Data and Variable Construction

Our sample consists of 70,930 users who opened the search channel and entered at least one query during the experimental period. These users were randomly assigned to the treatment group, in which search personalization was reduced, or to the control group, in which they would counterfactually have received the same intervention. We obtain each user's group assignment from the platform's A/B testing system, basic user characteristics from the user database, and behavioral records from the log database. All data are anonymized and linked through an anonymized user identifier.

Based on this dataset, we construct the variables for our analyses. $Treat$ indicates whether a user belongs to the treatment group. *Active* captures the user's pre-treatment activity level on the platform, computed directly by the platform using pre-treatment data and classified into high, medium, and low tiers. We use superscripts s and r to denote behaviors in the search and recommendation channels, respectively. For the search channel, we construct $\text{Log(Browse/Query)}^s$, the log-transformed average numbers of content browsed per query, as well as $CTR^s_{Ordinary}$ and CTR^s_{Paid}, the click-through rates on ordinary (organic) and paid content in the search results. For the recommendation channel, we analogously construct Log(Browse)^r, $CTR^r_{Ordinary}$, and CTR^r_{Paid} based on users' browsing and clicking behavior in that channel. The search-channel variables measure how depersonalization affects behavior within the treated channel, while the recommendation-channel variables allow us to detect spillover effects in the untreated channel. Table 5.2 presents the summary statistics of these variables.

Table 5.2: Summary Statistics

Variable	Description	N	Mean	Std. Dev.	Median	Min	Max
$Treat$	Treatment group indicator	70,930	0.5005	0.5000	1.0000	0.0000	1.0000
$Active$	Pre-treatment activity tier (1 = high, 2 = medium, 3 = low)	70,930	2.2042	0.7928	2.0000	1.0000	3.0000
$\text{Log(Browse/Query)}^s$	Log of average content browsed per query in the search channel	70,930	2.0729	0.6298	1.9924	0.6931	5.3230
$CTR^s_{Ordinary}$	Click-through rate on ordinary content in the search channel	70,930	0.1549	0.1396	0.1429	0.0000	1.0000
CTR^s_{Paid}	Click-through rate on paid content in the search channel	39,791	0.0782	0.1956	0.0000	0.0000	1.0000
Log(Browse)^r	Log of content browsed in the recommendation channel	69,625	4.3390	1.8764	4.1744	0.6931	10.2415
CTR^r_{Paid}	Click-through rate on paid content in the recommendation channel	53,501	0.1000	0.1499	0.0250	0.0000	1.0000
$CTR^r_{Ordinary}$	Click-through rate on ordinary content in the recommendation channel	69,609	0.1182	0.1148	0.1000	0.0000	1.0000

Several patterns in Table 5.2 merit comment. The mean of $Treat$ is 0.5005, consistent with the random assignment of users to the two experimental groups in equal proportions. On average, users browse about eight pieces of content per query in the search channel ($e^{2.0729} \approx 7.9$), and click-through rates are higher for ordinary content than for paid content in both channels, in line with the higher bar that sponsored content must clear before users click. Note that the number of observations varies across variables because each behavioral measure is defined only for users who generate the corresponding behavior. While all 70,930 users in our sample entered the search channel by construction, a small fraction never opened the recommendation channel during the experimental period and therefore contribute no

observations on the recommendation side. Similarly, the click-through rate for a given content type is defined only for users who were exposed to at least one piece of that type of content in the corresponding channel; because paid content accounts for a minority of impressions, the paid click-through rates are observed for fewer users than their ordinary counterparts in both channels. These differences in sample size reflect the structure of user activity rather than data loss, and the treatment and control groups remain balanced within each estimation sample.

5.5 Empirical Results

Before turning to the estimates, we lay out the econometric specification. The platform randomizes the depersonalization intervention at the user level, so we conduct all analyses at the user level as well, keeping the unit of analysis consistent with the unit of randomization (Xu et al. 2026). Each behavioral outcome is accordingly aggregated over the experimental window into one observation per user. For each outcome, we estimate the following model by ordinary least squares:

$$Y_i = \beta_0 + \beta_1 \, Treat_i + \varepsilon_i, \tag{5.15}$$

where Y_i denotes one of the six outcomes defined in Section 5.4.2, namely $\text{Log(Browse/Query)}^s$, $CTR^s_{Ordinary}$, and CTR^s_{Paid} for the search channel and Log(Browse)^r, $CTR^r_{Ordinary}$, and CTR^r_{Paid} for the recommendation channel; $Treat_i$ equals one if user i is assigned to the depersonalized search ranking and zero otherwise; and ε_i is the error term. Because assignment is random, $Treat_i$ is independent of users' potential outcomes, so β_1 identifies the average treatment effect without further controls, and the constant β_0 recovers the control-group mean of the outcome. The simplicity of Equation (5.15) is thus a feature of the design rather than a limitation: any covariates would serve only to improve precision, not to remove bias. Standard errors are reported in parentheses in all tables. For the heterogeneity analyses, we estimate Equation (5.15) separately within each pre-treatment usage-frequency tier defined by *Active*; because the tiers are constructed from pre-treatment data, splitting the sample on them does not compromise the randomization within each tier.

5.5.1 Effect on the Search Channel

Average Effect on the Search Channel

We begin with the search channel, where the intervention takes place. Because group assignment is randomized, we estimate the average treatment effects by regressing each outcome on the treatment indicator. Table 5.3 reports the results.

Column (1) of Table 5.3 shows that depersonalization significantly lengthens search sessions: treated users browse 1.56% more content per query than control users ($\beta = 0.0156$, $p < 0.01$). This pattern matches Proposition 1. Once the ranking no longer exploits the user's interaction history, the top-ranked results are less likely to be exactly what she is looking for, so she must scan further down the list to collect the same amount of useful content, and her sessions lengthen.

Table 5.3: Average Effect of Depersonalization on the Search Channel

	$Log(Browse/Query)^s$	$CTR^s_{Ordinary}$	CTR^s_{Paid}
	(1)	(2)	(3)
$Treat$	0.01558***	-0.00811***	0.00074
	(0.00473)	(0.00105)	(0.00196)
Constant	2.06508***	0.15898***	0.07781***
	(0.00333)	(0.00075)	(0.00138)
Observations	70,930	70,930	39,791

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Column (2) reports the corresponding decline in clicking. The click-through rate on ordinary content falls by 0.81 percentage points ($\beta = -0.0081$, $p < 0.01$), a 5.1% reduction relative to the control-group mean of 15.9%. Treated users are shown content that matches their tastes less well, and a smaller share of what they see is worth a click. In contrast, Column (3) shows that the click-through rate on paid content is essentially unchanged: the estimate is small and statistically insignificant ($\beta = 0.0007$, $p > 0.1$). This asymmetry is the

ordering predicted by Proposition 2. Ordinary content sits in the responsive middle of the user's taste range, where a weaker match moves the click decision substantially, whereas paid content must clear a much higher bar and is clicked rarely to begin with, so its rate barely responds to the quality of personalization. Taken together, the three columns describe a search experience that becomes uniformly less rewarding per item viewed: users work harder and click less. In the language of the framework, the per-item value of search falls, which is precisely the object that transmits the intervention to the recommendation channel in Section 5.5.2.

Heterogeneous Effect on the Search Channel

We next examine how these effects vary with users' activity on the platform, estimating the same specifications separately for the high, medium, and low usage-frequency tiers. Tables 5.4, 5.5, and 5.6 report the results.

The estimates display a clear monotone gradient. For high-frequency users, depersonalization raises browsing per query by 2.72% ($\beta = 0.0272$, $p < 0.01$) and lowers the ordinary click-through rate by 1.16 percentage points ($\beta = -0.0116$, $p < 0.01$), a 6.6% reduction relative to their control-group mean of 17.6%. For medium-frequency users, both effects shrink: browsing rises by 1.51% ($\beta = 0.0151$, $p < 0.1$) and the ordinary rate falls by 0.77 percentage points ($\beta = -0.0077$, $p < 0.01$), a 4.9% relative decline. For low-frequency users, the browsing effect is smaller still and statistically insignificant ($\beta = 0.0103$, $p > 0.1$), while the ordinary rate falls by 0.63 percentage points ($\beta = -0.0063$, $p < 0.01$), a 4.1% relative decline. The paid click-through rate is unaffected in every tier, mirroring the average results.

This gradient is informative about the mechanism. The performance of a personalized ranking rests on the interaction data the user has supplied: the more a user searches, clicks, and browses, the more accurately the algorithm can predict which results she will value. For highly active users, the personalized ranking therefore differs sharply from its depersonalized counterpart, and removing personalization destroys a great deal of predictive power. For infrequent users, the platform holds thin histories, the personalized and depersonalized

rankings are close to begin with, and the same intervention changes little. In the framework's terms, the experimental fall in ρ_S is effectively larger for active users, and Propositions 1 and 2 then imply exactly the ordering we observe: the deepening of browsing and the decline in ordinary clicking are strongest in the high tier and fade across the medium and low tiers. It is worth noting that even in the lowest tier the ordinary click-through rate falls significantly, which suggests that even limited interaction histories carry real predictive value. These results also speak to the economics of data: the cost of withholding personal data from the ranking algorithm is not uniform but increases with the very usage that generates the data.

Table 5.4: Effect of Depersonalization on the Search Channel: High Usage Frequency

	$Log(Browse/Query)^s$	$CTR^s_{Ordinary}$	CTR^s_{Paid}
	(1)	(2)	(3)
$Treat$	0.02718***	-0.01163***	0.00063
	(0.00904)	(0.00197)	(0.00355)
Constant	2.06393***	0.17556***	0.07386***
	(0.00633)	(0.00144)	(0.00246)
Observations	16,527	16,527	10,221

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table 5.5: Effect of Depersonalization on the Search Channel: Medium Usage Frequency

	$Log(Browse/Query)^s$	$CTR^s_{Ordinary}$	CTR^s_{Paid}
	(1)	(2)	(3)
$Treat$	0.01511*	-0.00770***	0.00137
	(0.00817)	(0.00184)	(0.00325)
Constant	2.02481***	0.15557***	0.07389***
	(0.00577)	(0.00133)	(0.00230)
Observations	23,394	23,394	14,010

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table 5.6: Effect of Depersonalization on the Search Channel: Low Usage Frequency

	$Log(Browse/Query)^s$	$CTR^s_{Ordinary}$	CTR^s_{Paid}
	(1)	(2)	(3)
Treat	0.01027	-0.00631***	0.00028
	(0.00746)	(0.00164)	(0.00334)
Constant	2.09586***	0.15258***	0.08392***
	(0.00525)	(0.00117)	(0.00235)
Observations	31,009	31,009	15,560

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

5.5.2 Effect on the Recommendation Channel

Average Effect on the Recommendation Channel

We now turn to the recommendation channel. The experiment leaves the recommendation algorithm untouched, so any treatment effect in this channel is a spillover from the degraded search experience, the path formalized in Lemma 1. We estimate Equation (5.15) on the recommendation-channel outcomes; Table 5.7 reports the results.

Table 5.7: Average Effect of Depersonalization on the Recommendation Channel

	$Log(Browse)^r$	CTR^r_{Paid}	$CTR^r_{Ordinary}$
	(1)	(2)	(3)
Treat	0.03191**	-0.00158	0.01595***
	(0.01422)	(0.00130)	(0.00087)
Constant	4.32306***	0.10074***	0.11022***
	(0.01008)	(0.00093)	(0.00061)
Observations	69,625	53,501	69,609

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Column (1) of Table 5.7 shows that recommendation sessions lengthen even though noth-

ing in this channel was altered: treated users browse 3.19% more content in the recommendation channel than control users ($\beta = 0.0319$, $p < 0.05$). This is the spillover predicted by Lemma 1. When search becomes a less rewarding way to spend time, users lower the bar for what is worth their attention, and they carry that lower bar into the recommendation channel, where it keeps them scrolling longer.

Column (3) reports the corresponding change in clicking on ordinary content. The ordinary click-through rate rises by 1.60 percentage points ($\beta = 0.0160$, $p < 0.01$), a 14.5% increase over the control-group mean of 11.0%, consistent with Proposition 3: users arrive at the recommendation channel less selective, so a larger share of what they see now looks worth clicking. The contrast with the search channel deserves emphasis. The same intervention lowers the ordinary click-through rate in the channel where it is applied and raises it in the channel it never touches. A uniform shock to engagement could not generate this sign reversal, whereas a standard carried from one channel into the other produces it naturally.

Column (2) shows that the paid click-through rate is, on average, essentially unchanged ($\beta = -0.0016$, $p > 0.1$). Proposition 4 explains why the average hides more than it reveals. Clicking paid content costs money, not just attention, so two forces pull in opposite directions: the lower bar makes paid content look more attractive, while the reluctance to spend more puts a brake on how far paid clicking can actually rise, and longer sessions stretch whatever paid clicking remains over more impressions. Which force dominates depends on how close a user already is to the limit of what she is willing to spend, so the aggregate effect can easily wash out. The heterogeneity analysis below shows that this is exactly what happens.

Heterogeneous Effect on the Recommendation Channel

We next estimate the spillover effects separately for the three usage-frequency tiers, with the tier-level estimates presented in Tables 5.8, 5.9, and 5.10.

Two patterns stand out. First, the rise in the ordinary click-through rate appears in every tier but is largest among the least active users, reversing the gradient observed in the search channel. The effect is 0.70 percentage points for high-frequency users ($\beta = 0.0070$, $p < 0.01$,

or 4.0% of their control mean of 17.5%), 1.89 percentage points for medium-frequency users ($\beta = 0.0189$, $p < 0.01$, or 16.8% of 11.2%), and 1.94 percentage points for low-frequency users ($\beta = 0.0194$, $p < 0.01$, or 26.9% of 7.2%). Browsing depth tilts the same way: it rises by 6.51% for medium-frequency users ($\beta = 0.0651$, $p < 0.01$) and by 3.79% for low-frequency users ($\beta = 0.0379$, $p < 0.05$), while the estimate for high-frequency users is positive but insignificant ($\beta = 0.0253$, $p > 0.1$). Second, the paid click-through rate falls only in the high tier, by 0.49 percentage points ($\beta = -0.0049$, $p < 0.05$), a 3.8% decline relative to their control mean of 12.9%; the estimates for the medium and low tiers are small and insignificant ($\beta = -0.0012$ and $\beta = 0.0016$, both $p > 0.1$).

Table 5.8: Effect of Depersonalization on the Recommendation Channel: High Usage Frequency

	$Log(Browse)^r$	CTR^r_{Paid}	$CTR^r_{Ordinary}$
	(1)	(2)	(3)
$Treat$	0.02528	-0.00487**	0.00697***
	(0.02025)	(0.00229)	(0.00151)
Constant	6.41397***	0.12876***	0.17531***
	(0.01414)	(0.00163)	(0.00106)
Observations	16,518	15,250	16,518

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

The reversal of the activity gradient is consistent with how the cross-channel standard is formed. The spillover works through a reference point: users judge the recommendation channel against the standard their recent search experience has set. Highly active users know the recommendation channel well, and their sense of what it should offer rests on a long history with the channel itself, so a few days of worse search move it only modestly. Less active users have not built up such expectations, so they lean more heavily on the search experience as their point of comparison, and the same decline in search quality pulls their

standard down further and lifts their ordinary clicking more. The search and recommendation gradients thus run in opposite directions for different reasons: activity strengthens the direct effect because personalization has more data to lose, but it weakens the spillover because the expectations of experienced users are harder to move.

Table 5.9: Effect of Depersonalization on the Recommendation Channel: Medium Usage Frequency

	$Log(Browse)^r$	CTR^r_{Paid}	$CTR^r_{Ordinary}$
	(1)	(2)	(3)
$Treat$	0.06506***	-0.00117	0.01885***
	(0.01873)	(0.00228)	(0.00145)
Constant	4.26926***	0.10439***	0.11245***
	(0.01320)	(0.00164)	(0.00103)
Observations	23,240	18,487	23,239

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

The paid results reflect the two competing forces in Proposition 4. High-frequency users already browse and click the most paid content, so they sit closest to the ceiling of what they are willing to spend. For them, there is little room for paid clicking to grow; their longer sessions mostly stretch a roughly unchanged amount of paid clicking over more impressions, and the paid rate falls even as the ordinary rate rises. The divergence between the two rates in the high tier, $+0.70$ versus -0.49 percentage points, is what a binding budget looks like in the data. Medium- and low-frequency users start from much lower spending. The lower standard makes paid content more attractive to them, as in part (ii) of the proposition, but money is not attention: each additional paid click runs into a growing reluctance to pay more, so the budget in practice acts less like a fixed allowance and more like a rising resistance to further spending. For these users the two forces pull against each other, the inclination to click more paid content and the unwillingness to spend much more, and the net effect is

too small to detect. Seen this way, the heterogeneity completes the picture: the most active users have effectively hit the ceiling of their willingness to pay, while for everyone else the appeal of paid content under a lower bar and the resistance to spending more roughly cancel.

Table 5.10: Effect of Depersonalization on the Recommendation Channel: Low Usage Frequency

	$Log(Browse)^r$	CTR^r_{Paid}	$CTR^r_{Ordinary}$
	(1)	(2)	(3)
$Treat$	0.03793**	0.00156	0.01941***
	(0.01639)	(0.00211)	(0.00129)
Constant	3.19430***	0.07520***	0.07205***
	(0.01168)	(0.00149)	(0.00089)
Observations	29,867	19,764	29,852

Note: Standard errors are reported in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

5.6 Discussion and Conclusion

5.6.1 Summary of Key Findings

In this study, we examine how depersonalizing the search channel of a major social media platform changes user behavior, both within the treated channel and beyond it. We first develop a stylized framework in which users carry a behavioral standard from the search channel into the recommendation channel, and we then test its predictions with a large-scale randomized field experiment covering roughly 71,000 users. Within the search channel, depersonalization makes search measurably harder: treated users browse more content per query yet click a smaller share of the ordinary content they see, while their clicking on paid content barely moves. The effects spill over to the untouched recommendation channel with the opposite sign: treated users browse more of the recommendation channel and click ordinary content there more often. Heterogeneity analyses reveal two reversals that discipline the interpretation. Within search, the burden of depersonalization is heaviest for

the most active users, since their rich interaction histories are what gave the personalized ranking its predictive power. The spillover gains, by contrast, accrue mainly to infrequent users, whose benchmark for the recommendation channel rests largely on what search has recently delivered. And only within the most active tier does paid clicking decline, a pattern consistent with a spending limit that binds for the heaviest spenders and stays slack for everyone else.

5.6.2 Implications for Platforms

Our findings carry several practical implications for platform operators. First, algorithmic changes cannot be evaluated channel by channel. The two discovery channels are linked through the standard users carry between them, so an A/B test that tracks only the treated channel will systematically misjudge the change: in our setting, a manager looking solely at search metrics would see longer sessions and lower click-through rates, and would miss both the engagement gains and the monetization losses materializing in the recommendation channel. Platform experimentation pipelines should therefore report outcomes across all major surfaces of the product, even for interventions that appear local. Second, engagement metrics are not welfare metrics. The increases in browsing depth and recommendation-channel clicking in our experiment arise from a degraded search experience, not an improved product: users scroll more because each item is worth less to them, and they click more in the recommendation channel because their standard has fallen. A platform that optimizes for time spent or click-through rates alone may thus mistake deterioration for success, a concern increasingly emphasized in the literature on engagement-based optimization (Kleinberg et al. 2024, Donnelly et al. 2024). Third, the costs of scaling back personalization are heterogeneous in a way that platforms can exploit. Because the burden falls most heavily on the most active users, while infrequent users experience little change in their search experience, a platform pressed by computational costs could depersonalize selectively, preserving personalized rankings for heavy users whose experience and paid consumption are most at risk while economizing on the long tail. Finally, the divergence in paid click-through rates warns

that monetization through paid content is exposed precisely where it matters most: among the platform's most active users, whose willingness to spend is already near its ceiling and whose paid clicking is diluted rather than expanded by longer sessions.

5.6.3 Implications for Policymakers

The results also speak to regulators weighing privacy rules that constrain behavioral data use (Goldberg et al. 2024, Ke and Sudhir 2023, Jia et al. 2021). First, the costs of restricting personalization are real but uneven. Depersonalizing search degrades the experience most for the users who have supplied the most data, so the incidence of privacy regulation falls disproportionately on a platform's heaviest users; analyses that examine only the average user will understate the burden on this group. Second, aggregate engagement is a misleading indicator of regulatory harm. In our experiment, total browsing rose in both channels even as the quality of the search experience fell; a regulator who reads stable or rising engagement as evidence that a data restriction is costless would draw the wrong conclusion. Third, channel-specific rules have cross-channel consequences. Restricting data use in search pushed users toward the recommendation channel, which remains the most data-intensive and algorithmically curated part of the platform. A privacy intervention aimed at one surface can thus increase reliance on another that raises the same concerns more acutely, an unintended substitution that policy design should anticipate. Regulations drafted at the level of data practices, rather than individual product surfaces, are less vulnerable to this kind of leakage.

5.6.4 Limitations and Future Directions

Several limitations of this study point to promising directions for future inquiry. First, our experimental window spans roughly two weeks. Over longer horizons, users may recalibrate their expectations, develop new search habits, or migrate to competing platforms, so the spillover we document could strengthen, fade, or reverse as the reference standard adapts. Future research could implement longer experiments, or exploit staggered rollouts of depersonalization, to trace these dynamics and to assess effects on retention and churn that a

short window cannot capture.

Second, our evidence comes from a single knowledge-sharing platform with a particular dual-channel architecture. Platforms that differ in how prominently search and recommendation are positioned, or in how paid content is integrated, may exhibit spillovers of different magnitudes or even different signs. Replicating the design across platform types, including short-video and e-commerce settings, would establish the boundary conditions of the cross-channel mechanism.

Third, our outcomes are behavioral. Browsing and clicking reveal how users reallocate attention, but they do not directly measure satisfaction or welfare, and our framework highlights exactly why the two can diverge. Combining experimental variation with survey measures of satisfaction, willingness-to-pay elicitations, or long-run retention would allow researchers to quantify the welfare consequences that our behavioral estimates can only bound. A structural estimation of our framework, which would recover the taste distribution and the spillover sensitivity from the experimental moments, is a natural next step in this direction.

Fourth, while the reference-standard mechanism is consistent with the full pattern of our results, we do not observe the standard itself. Richer designs could measure it directly, for example by eliciting users' stated expectations, by varying the intensity of depersonalization to trace out a dose-response curve, or by randomizing the order in which users encounter the two channels within a session. Finally, our analysis holds the supply side fixed. Over longer horizons, content creators and advertisers will respond to shifted click patterns by adjusting what they produce and how they price paid content, and the platform itself may rebalance its ranking objectives. Embedding the user-side mechanism we document in an equilibrium model of the content market is a promising avenue for understanding the full consequences of depersonalization.

Chapter 6

CONCLUDING REMARKS

This dissertation offers a multifaceted exploration of the economics of human-AI interaction and AI-AI interaction, contributing to a deeper understanding of how AI reshapes digital markets and how the participants in those markets respond. By examining the strategic adaptation of workers to generative AI in online labor markets, the role of platform recommender systems in shaping competition among autonomous pricing algorithms, and the consequences of depersonalizing search on a social media platform, the three essays trace the influence of AI from the moment it enters a market, through its interactions with other algorithms, to the platform's choice of how much algorithmic mediation to provide.

Chapter 3 examines the labor market consequences of LLM-based generative AI. Drawing on comprehensive data from a leading online labor platform and the launch of ChatGPT as an exogenous shock, it documents a substantial displacement of demand in the submarkets most exposed to the technology, a slower contraction of supply, and a resulting intensification of competition. It further shows that freelancers respond strategically rather than passively, transitioning toward programming-intensive work, and that this transition is led by high-skilled workers with meaningful consequences for earnings. The essay demonstrates that the effects of a general-purpose technology on work are mediated by workers' own strategic choices, and it highlights the capacity to absorb new technology as the asset that separates those who adapt from those who are displaced, a message relevant to workers, platforms, and policymakers alike.

Chapter 4 turns from human responses to the interactions among algorithms themselves. Using simulations of reinforcement-learning pricing agents grounded in a structural search model estimated from field data, it shows that the platform's recommender system is a de-

cisive force in algorithmic competition: the objective the system pursues and the size of the recommendation set jointly determine whether pricing algorithms settle into collusive or competitive outcomes, and how surplus is allocated among sellers, consumers, and the platform. It also uncovers a "more is less" effect: enlarging the recommendation set can undermine the platform's own objective, because a wider set loosens the recommender's grip on the dynamic rewards that shape what pricing algorithms learn. The essay establishes that information intermediation is an instrument of market governance rather than a neutral conduit. Further, it suggests that platform recommendation can itself generate the price correlation that empirical studies of algorithmic pricing have documented, and it offers guidance for platforms and antitrust authorities tasked with overseeing markets where pricing has been handed to machines.

Chapter 5 investigates what happens when a platform deliberately dials back the personalization of its AI, a choice increasingly forced by privacy regulation and computational cost. In a randomized field experiment conducted at scale with a major social media platform, it shows that depersonalizing search makes search harder within the treated channel, yet raises engagement in the untouched recommendation channel as users carry a lowered standard across channels, with the burden of the direct effect concentrated among the most active users and the paid-content losses among the heaviest spenders. The essay demonstrates that the value of personalization rises with the data a user has supplied, that the consequences of algorithmic changes travel across the surfaces of a platform, and that engagement metrics can rise even as the underlying experience deteriorates, lessons that matter equally for platform experimentation and for regulatory assessment.

Together, the essays converge on a central insight: the economic consequences of AI are properties of interactions rather than of algorithms in isolation. A technology that displaces demand also reshapes the incentives of those who supply it; a population of pricing algorithms produces different market outcomes under different recommendation designs; an intervention in one channel reverberates through another via the expectations users carry between them. Design choices that appear technical, such as a recommender's objective or

a ranking's personalization level, are in fact economic levers with first-order consequences for competition, welfare, and the distribution of surplus. Evaluating AI therefore requires looking at the whole market and the whole platform, not the algorithm alone.

This dissertation also opens several avenues for future research. The labor market analysis could be extended with longer horizons, direct measures of AI adoption, and settings beyond online platforms, to trace how skill transitions and earnings consequences evolve as generative AI matures. The collusion framework could incorporate other forms of intermediation, such as advertising auctions and emerging AI shopping agents, and its predictions could be tested empirically as data on algorithmic pricing in recommendation-mediated markets becomes available. The depersonalization findings invite welfare quantification through structural estimation, direct measurement of the cross-channel standard, and equilibrium analyses that allow content creators and advertisers to respond. Beyond each essay, a broader frontier is emerging where the two themes of this dissertation meet: as workers deploy their own AI agents and consumers delegate search and purchasing to algorithmic assistants, human-AI and AI-AI interactions will intertwine, and markets will feature layered algorithmic intermediation whose economics remain largely unexplored.

Answering such questions will require the methodological breadth this dissertation has sought to exemplify. The essays draw on causal inference, structural modeling, multi-agent simulation, randomized field experimentation, and analytical models, and they borrow from information systems, economics, marketing, and computer science. As AI systems grow more capable and more interconnected, no single method or discipline will suffice; cross-disciplinary collaboration and continued investment in both experimental access and computational tools will be essential for research to keep pace.

Ultimately, how AI reshapes markets will be determined by the choices of the businesses that deploy it, the platforms that govern it, the policymakers who regulate it, and the individuals who work and consume alongside it. This dissertation shows that those choices interact, and that good outcomes depend on understanding the interactions rather than the technology alone. By continuing to study how humans adapt to algorithms, how algorithms

respond to one another, and how platforms calibrate the systems in between, researchers can help ensure that increasingly algorithmic markets remain competitive, transparent, and beneficial to the people they serve.

BIBLIOGRAPHY

Abraham, K., Haltiwanger, J., Sandusky, K., and Spletzer, J. (2017). Measuring the gig economy: Current knowledge and open issues. *Measuring and Accounting for Innovation in the 21st Century*.

Acemoglu, D. and Autor, D. (2011). Skills, tasks and technologies: Implications for employment and earnings. In *Handbook of labor economics*, volume 4, pages 1043–1171. Elsevier.

Acemoglu, D. and Restrepo, P. (2019). Automation and new tasks: How technology displaces and reinstates labor. *Journal of economic perspectives*, 33(2):3–30.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Agrawal, A., Gans, J., and Goldfarb, A. (2019). *The economics of artificial intelligence: an agenda*. University of Chicago Press.

Akşin, Z., Ata, B., Emadi, S. M., and Su, C.-L. (2013). Structural estimation of callers' delay sensitivity in call centers. *Management Science*, 59(12):2727–2746.

Altonji, J. G., Elder, T. E., and Taber, C. R. (2005). Selection on observed and unobserved variables: Assessing the effectiveness of catholic schools. *Journal of political economy*, 113(1):151–184.

Angrist, J. D. and Pischke, J.-S. (2009). *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

Aouad, A. and Saban, D. (2023). Online assortment optimization for two-sided matching platforms. *Management Science*, 69(4):2069–2087.

Aridor, G. (2025). Measuring substitution patterns in the attention economy: An experimental approach. *The RAND Journal of Economics*, 56(3):302–324.

Aridor, G., Gonçalves, D., Kluver, D., Kong, R., and Konstan, J. (2022). The informational role of online recommendations: Evidence from a field experiment. *arXiv preprint arXiv:2211.14219*.

Assad, S., Clark, R., Ershov, D., and Xu, L. (2020). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market.

Autor, D. H. and Dorn, D. (2013). The growth of low-skill service jobs and the polarization of the us labor market. *American economic review*, 103(5):1553–1597.

Autor, D. H., Levy, F., and Murnane, R. J. (2003). The skill content of recent technological change: An empirical exploration. *The Quarterly journal of economics*, 118(4):1279–1333.

Bai, J., Gao, Q., Goes, P., and Lin, M. (2025). All that glitters is not gold: The impact of certification test costs in online labor markets. *Information Systems Research*.

Banchio, M. and Skrzypacz, A. (2022). Artificial intelligence and auction design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 30–31.

Barach, M. A., Golden, J. M., and Horton, J. J. (2020). Steering in online markets: the role of platform incentives and credibility. *Management Science*, 66(9):4047–4070.

Becker, G. S. (1962). Investment in human capital: A theoretical analysis. *Journal of political economy*, 70(5, Part 2):9–49.

Ben-Porat, O., Goren, G., Rosenberg, I., and Tennenholtz, M. (2019). From recommendation systems to facility location games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1772–1779.

Benndorf, V., Müller, S., and Rau, H. A. (2024). The effects of betrayal aversion on effort provision when incentives are fragile. *Management Science*, 70(11):7750–7769.

Benson, A., Sojourner, A., and Umyarov, A. (2020). Can reputation discipline the gig economy? experimental evidence from an online labor market. *Management Science*, 66(5):1802–1825.

Bi, X., Yang, M., and Adomavicius, G. (2024). Consumer acquisition for recommender systems: A theoretical framework and empirical evaluations. *Information Systems Research*, 35(1):339–362.

Braghieri, L., Levy, R., and Makarin, A. (2022). Social media and mental health. *American Economic Review*, 112(11):3660–3693.

Brynjolfsson, E., Li, D., and Raymond, L. (2025). Generative ai at work. *The Quarterly Journal of Economics*, page qjae044.

Brynjolfsson, E., Li, D., and Raymond, L. R. (2023). Generative ai at work. Technical report, National Bureau of Economic Research.

Brynjolfsson, E. and McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & company.

Burtch, G., Carnahan, S., and Greenwood, B. N. (2018). Can you gig it? an empirical examination of the gig economy and entrepreneurial activity. *Management Science*, 64(12):5497–5520.

Calder-Wang, S. and Kim, G. H. (2023). Coordinated vs efficient prices: The impact of algorithmic pricing on multifamily rental markets. *Available at SSRN 4403058*.

Callaway, B., Goodman-Bacon, A., and Sant'Anna, P. H. (2024). Difference-in-differences with a continuous treatment. Technical report, National Bureau of Economic Research.

Calvano, E., Calzolari, G., Denicolò, V., Harrington Jr, J. E., and Pastorello, S. (2020a). Protecting consumers from collusive prices due to ai. *Science*, 370(6520):1040–1042.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2019). Algorithmic pricing what implications for competition policy? *Review of industrial organization*, 55:155–171.

Calvano, E., Calzolari, G., Denicolo, V., and Pastorello, S. (2020b). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297.

Campbell, D. T. and Stanley, J. C. (2015). *Experimental and quasi-experimental designs for research*. Ravenio books.

Cassano, F., Gouwar, J., Nguyen, D., Nguyen, S., Phipps-Costin, L., Pinckney, D., Yee, M.-H., Zi, Y., Anderson, C. J., Feldman, M. Q., et al. (2022). Multipl-e: A scalable and extensible approach to benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*.

Che, Y.-K. and Hörner, J. (2018). Recommender systems as mechanisms for social learning. *The Quarterly Journal of Economics*, 133(2):871–925.

Chen, D. L. and Horton, J. J. (2016). Research note—are online labor markets spot markets for

tasks? a field experiment on the behavioral response to wage cuts. *Information Systems Research*, 27(2):403–423.

Chen, L., Mislove, A., and Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th international conference on World Wide Web*, pages 1339–1349.

Chen, M. K., Rossi, P. E., Chevalier, J. A., and Oehlsen, E. (2019). The value of flexible work: Evidence from uber drivers. *Journal of political economy*, 127(6):2735–2794.

Chen, Y., Yuan, Z., Sun, T., and Chen, A. (2023). Understanding the impacts of de-personalization in search algorithm on consumer behavior: A field experiment with a large online retail platform.

Chung, J. H., Chintagunta, P., and Misra, S. (2024). Simulated maximum likelihood estimation of the sequential search model. *Quantitative Marketing and Economics*, pages 1–60.

Cinelli, C., Forney, A., and Pearl, J. (2022). A crash course in good and bad controls. *Sociological Methods & Research*, page 00491241221099552.

Cohen, W. M. and Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative science quarterly*, 35(1):128–152.

Colangelo, G. (2021). Artificial intelligence and anticompetitive collusion: From the 'meeting of minds' towards the 'meeting of algorithms'? *TTLF Stanford Law School Working Paper*.

Colliard, J.-E., Foucault, T., and Lovo, S. (2022). Algorithmic pricing and liquidity in securities markets. *HEC Paris Research Paper*.

Compiani, G., Lewis, G., Peng, S., and Wang, P. (2024). Online search and optimal product rankings: An empirical framework. *Marketing Science*, 43(3):615–636.

Daviss, C. and Leung, M. D. (2025). Reactions, revisions, and rehiring: Changes in employers' gendered preferences in online labor markets. *Organization Science*.

De los Santos, B. and Koulayev, S. (2017). Optimizing click-through in online rankings with endogenous search refinement. *Marketing Science*, 36(4):542–564.

Dell'Acqua, F., McFowland III, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., and Lakhani, K. R. (2023). Navigating the jagged technological

frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper*, (24-013).

Derakhshan, M., Golrezaei, N., Manshadi, V., and Mirrokni, V. (2022). Product ranking on online platforms. *Management Science*, 68(6):4024–4041.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Donnelly, R., Kanodia, A., and Morozov, I. (2024). Welfare effects of personalized rankings. *Marketing Science*, 43(1):92–113.

Dou, W. W., Goldstein, I., and Ji, Y. (2024). Ai-powered trading, algorithmic collusion, and price efficiency. *Jacobs Levy Equity Management Center for Quantitative Financial Research Paper*.

Eloundou, T., Manning, S., Mishkin, P., and Rock, D. (2024). Gpts are gpts: Labor market impact potential of llms. *Science*, 384(6702):1306–1308.

Faccio, M. and O'Brien, W. J. (2021). Business groups and employment. *Management Science*, 67(6):3468–3491.

Feldman, J., Zhang, D. J., Liu, X., and Zhang, N. (2022). Customer choice models vs. machine learning: Finding optimal product displays on alibaba. *Operations Research*, 70(1):309–328.

Felten, E., Raj, M., and Seamans, R. (2021). Occupational, industry, and geographic exposure to artificial intelligence: A novel dataset and its potential uses. *Strategic Management Journal*, 42(12):2195–2217.

Felten, E. W., Raj, M., and Seamans, R. (2023). Occupational heterogeneity in exposure to generative ai. *Available at SSRN 4414065*.

Filippas, A., Horton, J., Parasurama, P., and Urraca, D. (2024). Costly capacity signaling increases matching efficiency: Evidence from a field experiment. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 414–415.

Filippas, A., Horton, J. J., and Golden, J. (2018). Reputation inflation. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 483–484.

Fleder, D. and Hosanagar, K. (2009). Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712.

Fu, R., Aseri, M., Singh, P. V., and Srinivasan, K. (2022). "un" fair machine learning algorithms. *Management Science*, 68(6):4173–4195.

Gallego, A. and Kurer, T. (2022). Automation, digitalization, and artificial intelligence in the workplace: implications for political behavior. *Annual Review of Political Science*, 25:463–484.

Galor, O. and Özak, Ö. (2016). The agricultural origins of time preference. *American economic review*, 106(10):3064–3103.

Garcia, D., Tolvanen, J., and Wagner, A. K. (2022). Demand estimation using managerial responses to automated price recommendations. *Management Science*, 68(11):7918–7939.

Garg, N. and Johari, R. (2021). Designing informative rating systems: Evidence from an online labor market. *Manufacturing & Service Operations Management*, 23(3):589–605.

Gathmann, C. and Schönberg, U. (2010). How general is human capital? a task-based approach. *Journal of Labor Economics*, 28(1):1–49.

Ghose, A., Ipeirotis, P. G., and Li, B. (2014). Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science*, 60(7):1632–1654.

Goldberg, S. G., Johnson, G. A., and Shriver, S. K. (2024). Regulating privacy online: An economic evaluation of the GDPR. *American Economic Journal: Economic Policy*, 16(1):325–358.

Guo, X., Cheng, Z., and Pavlou, P. A. (2025). Skill-biased technical change, again? online gig platforms and local employment. *Information Systems Research*, 36(3):1354–1374.

Gupta, U., Wu, C.-J., Wang, X., Naumov, M., Reagen, B., Brooks, D., Cottel, B., Hazelwood, K., Hempstead, M., Jia, B., et al. (2020). The architectural implications of Facebook's DNN-based personalized recommendation. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 488–501. IEEE.

Gussek, L. and Wiesche, M. (2024). Understanding the careers of freelancers on digital labor platforms: The case of it work. *Information Systems Journal*, 34(5):1664–1702.

Handa, K., Tamkin, A., McCain, M., Huang, S., Durmus, E., Heck, S., Mueller, J., Hong, J., Ritchie, S., Belonax, T., et al. (2025). Which economic tasks are performed with ai? evidence from millions of claude conversations. *arXiv preprint arXiv:2503.04761*.

Hashim, M. J. and Bockstedt, J. C. (2024). Real-effort incentives in online labor markets: Punishments and rewards for individuals and groups. *MIS Quarterly*, 48(1).

Hatch, N. W. and Dyer, J. H. (2004). Human capital and learning as a source of sustainable competitive advantage. *Strategic management journal*, 25(12):1155–1178.

Hong, Y., Wang, C., and Pavlou, P. A. (2016). Comparing open and sealed bid auctions: Evidence from online labor markets. *Information Systems Research*, 27(1):49–69.

Horton, J. J. (2019). Buyer uncertainty about seller capacity: Causes, consequences, and a partial solution. *Management Science*, 65(8):3518–3540.

Huang, N., Burtch, G., Hong, Y., and Pavlou, P. A. (2020). Unemployment and worker participation in the gig economy: Evidence from an online labor market. *Information Systems Research*, 31(2):431–448.

Jensen, R. and Miller, N. H. (2018). Market integration, demand, and the growth of firms: Evidence from a natural experiment in india. *American Economic Review*, 108(12):3583–3625.

Jia, J., Jin, G. Z., and Wagman, L. (2021). The short-run effects of the General Data Protection Regulation on technology venture investment. *Marketing Science*, 40(4):661–684.

Kanat, I., Hong, Y., and Raghu, T. (2018). Surviving in global online labor markets for it services: A geo-economic analysis. *Information systems research*, 29(4):893–909.

Ke, T. T. and Sudhir, K. (2023). Privacy rights and data security: GDPR and personal data markets. *Management Science*, 69(8):4389–4412.

Kleinberg, J., Mullainathan, S., and Raghavan, M. (2024). The challenge of understanding what users want: Inconsistent preferences and engagement optimization. *Management Science*, 70(9):6336–6355.

Kleiner, M. and Xu, M. (2025). Occupational licensing and labor market fluidity. *Journal of Labor Economics*.

Kocetkov, D., Li, R., Allal, L. B., Li, J., Mou, C., Ferrandis, C. M., Jernite, Y., Mitchell, M., Hughes, S., Wolf, T., et al. (2022). The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Kokkodis, M. and Ipeirotis, P. G. (2023). The good, the bad, and the unhirable: Recommending job applicants in online labor markets. *Management Science*.

Kokkodis, M. and Ransbotham, S. (2023). Learning to successfully hire in online labor markets. *Management Science*, 69(3):1597–1614.

Kontopantelis, E., Doran, T., Springate, D. A., Buchan, I., and Reeves, D. (2015). Regression based quasi-experimental approach when randomisation is not an option: interrupted time series analysis. *bmj*, 350.

Koulayev, S. (2014). Search for differentiated products: identification and estimation. *The RAND Journal of Economics*, 45(3):553–575.

Laitenberger, U., Viete, S., Slivko, O., Kummer, M. E., Borchert, K., and Hirth, M. (2022). Unemployment and online labor: evidence from microtasking. *ZEW Discussion Papers*, 18.

Lee, D. and Hosanagar, K. (2019). How do recommender systems affect sales diversity? a cross-category investigation via randomized field experiment. *Information Systems Research*, 30(1):239–259.

Lee, D. and Hosanagar, K. (2021). How do product attributes and reviews moderate the impact of recommender systems through purchase stages? *Management Science*, 67(1):524–546.

Li, L., Chen, J., and Raghunathan, S. (2018). Recommender system rethink: Implications for an electronic marketplace with competing manufacturers. *Information Systems Research*, 29(4):1003–1023.

Li, L., Chen, J., and Raghunathan, S. (2020). Informative role of recommender systems in electronic marketplaces: A boon or a bane for competing sellers. *MIS Quarterly*, 44(4).

Li, P. and Tuzhilin, A. (2024a). Deep pareto reinforcement learning for multi-objective recommender systems. *arXiv preprint arXiv:2407.03580*.

Li, P. and Tuzhilin, A. (2024b). When variety seeking meets unexpectedness: Incorporating variety-seeking behaviors into design of unexpected recommender systems. *Information Systems Research*, 35(3):1257–1273.

Li, P., Wang, Y., Chi, E. H., and Chen, M. (2023). Hierarchical reinforcement learning for modeling user novelty-seeking intent in recommender systems. *arXiv preprint arXiv:2306.01476*.

Li, S. E. (2023). *Reinforcement learning for sequential decision and optimal control*. Springer.

Li, X., Grahl, J., and Hinz, O. (2022). How do recommender systems lead to consumer purchases? a causal mediation analysis of a field experiment. *Information Systems Research*, 33(2):620–637.

Li, Z., Liang, C., Peng, J., and Yin, M. (2024). The value, benefits, and concerns of generative ai-powered assistance in writing. In *Proceedings of the 2024 CHI conference on human factors in computing systems*, pages 1–25.

Liang, C., Hong, Y., Chen, P.-Y., and Shao, B. B. (2022). The screening role of design parameters for service procurement auctions in online service outsourcing platforms. *Information Systems Research*, 33(4):1324–1343.

Liang, C., Hong, Y., and Gu, B. (2024). Monitoring and the cold start problem in digital platforms: Theory and evidence from online labor markets. *Information Systems Research*.

Lin, S., Shi, Z. J., and Sun, X. (2025). Towards intelligent shopping assistant: How can llm-based chatbots transform online shopping process? *HKUST Business School Research Paper*, (2025-196).

Liu, Y., Lou, B., Zhao, X., and Li, X. (2024). Unintended consequences of advances in matching technologies: Information revelation and strategic participation on gig-economy platforms. *Management Science*, 70(3):1729–1754.

Lokkila, E., Christopoulos, A., and Laakso, M.-J. (2023). A data-driven approach to compare the syntactic difficulty of programming languages. *Journal of Information Systems Education*, 34(1):84–93.

Manh, D. N., Hai, N. L., Dau, A. T., Nguyen, A. M., Nghiem, K., Guo, J., and Bui, N. D.

(2023). The vault: A comprehensive multilingual dataset for advancing code understanding and generation. *arXiv preprint arXiv:2305.06156*.

Mao, S., Dewan, S., and Ho, Y.-J. (2023). Personalized ranking at a mobile app distribution platform. *Information Systems Research*, 34(3):811–827.

Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, J. C., Parli, V., et al. (2023). Artificial intelligence index report 2023. *arXiv preprint arXiv:2310.03715*.

McInnes, L., Healy, J., Astels, S., et al. (2017). hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205.

Miklós-Thal, J. and Tucker, C. (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science*, 65(4):1552–1561.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Moraga-González, J. L., Sándor, Z., and Wildenbeest, M. R. (2023). Consumer search and prices in the automobile market. *The Review of Economic Studies*, 90(3):1394–1440.

Morozov, I., Seiler, S., Dong, X., and Hou, L. (2021). Estimation of preference heterogeneity in markets with costly search. *Marketing Science*, 40(5):871–899.

Morris, M. R., Sohl-dickstein, J., Fiedel, N., Warkentin, T., Dafoe, A., Faust, A., Farabet, C., and Legg, S. (2023). Levels of agi: Operationalizing progress on the path to agi. *arXiv preprint arXiv:2311.02462*.

Mortensen, D. T. (2011). Markets with search friction and the dmp model. *American Economic Review*, 101(4):1073–1091.

Mousavi, R. and Gu, B. (2024). Resilience messaging: The effect of governors' social media communications on community compliance during a public health crisis. *Information Systems Research*, 35(2):505–527.

Narahari, Y., Garg, D., Narayanam, R., and Prakash, H. (2009). *Game theoretic problems in network economics and mechanism design solutions*. Springer Science & Business Media.

Neal, D. (1995). Industry-specific human capital: Evidence from displaced workers. *Journal of labor Economics*, 13(4):653–677.

Nguyen, N., Johnson, J., and Tsiros, M. (2024). Unlimited testing: Let's test your emails with ai. *Marketing Science*, 43(2):419–439.

Noy, S. and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Available at SSRN 4375283*.

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

Ousterhout, J. K. (1998). Scripting: Higher level programming for the 21st century. *Computer*, 31(3):23–30.

Paolillo, A., Colella, F., Nosengo, N., Schiano, F., Stewart, W., Zambrano, D., Chappuis, I., Lalive, R., and Floreano, D. (2022). How to compete with robots by assessing job automation risks and resilient alternatives. *Science Robotics*, 7(65):eabg5561.

Pariser, E. (2011). *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press.

Perez-Truglia, R. (2020). The effects of income transparency on well-being: Evidence from a natural experiment. *American Economic Review*, 110(4):1019–1054.

Petrova, M., Sen, A., and Yildirim, P. (2023). New technologies and political competition: the impact of social media communication on political contributions. *Petrova, M., Sen, A., Yildirim, P. New technologies and political competition: The impact of social media communication on political contributions in'The Political Economy of Social Media,'Campante, F, R Durante and A Tesei (eds)(2023). CEPR Press, Paris, London. https://cepr. org/publications/boo*.

Polanyi, M. (2009). The tacit dimension. In *Knowledge in organisations*, pages 135–146. Routledge.

Poldrack, R. A., Lu, T., and Beguš, G. (2023). Ai-assisted coding: Experiments with gpt-4. *arXiv preprint arXiv:2304.13187*.

Prechelt, L. (2000). An empirical comparison of seven programming languages. *Computer*, 33(10):23–29.

Ran, Z., Rau, P. R., and Ziegler, T. (2025). Sometimes, always, never: Regulatory clarity and the development of digital financing. *Management Science*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Robertson, V. H. (2022). Antitrust law and digital markets: a guide to the european competition law experience in the digital economy.

Rocher, L., Tournier, A. J., and de Montjoye, Y.-A. (2023). Adversarial competition and collusion in algorithmic markets. *Nature Machine Intelligence*, 5(5):497–504.

Rosen, S. (1983). Specialization and human capital. *Journal of Labor Economics*, 1(1):43–49.

Salcedo, B. (2015). Pricing algorithms and tacit collusion. *Manuscript, Pennsylvania State University*.

Schmidheiny, K. and Siegloch, S. (2023). On event studies and distributed-lags in two-way fixed effects models: Identification, equivalence, and generalization. *Journal of Applied Econometrics*, 38(5):695–713.

Sen, A., Grad, T., Ferreira, P., and Claussen, J. (2024). (how) does user-generated content impact content generated by professionals? evidence from local news. *Management Science*, 70(9):6045–6068.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Spann, M., Bertini, M., Koenigsberg, O., Zeithammer, R., Aparicio, D., Chen, Y., Fantini, F., Jin, G. Z., Morwitz, V., Leszczyc, P. P., et al. (2024). Algorithmic pricing: Implications for consumers, managers, and regulators. Technical report, National Bureau of Economic Research.

Stanton, C. T. and Thomas, C. (2025). Who benefits from online gig economy platforms? *American Economic Review*, 115(6):1857–1895.

Stone, J. V. (2013). Bayes' rule: a tutorial introduction to bayesian analysis.

Sundararajan, A. (2017). *The sharing economy: The end of employment and the rise of crowd-based capitalism*. MIT press.

Tafti, A. and Shmueli, G. (2020). Beyond overall treatment effects: Leveraging covariates in randomized experiments guided by causal structure. *Information Systems Research*, 31(4):1183–1199.

Tam, K. Y. and Ho, S. Y. (2006). Understanding the impact of web personalization on user information processing and decision outcomes. *MIS Quarterly*, 30(4):865–890.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Ursu, R., Seiler, S., and Honka, E. (2023). The sequential search model: A framework for empirical research. *Available at SSRN 4236738*.

Ursu, R. M. (2018). The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science*, 37(4):530–552.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Vee, A. (2013). Understanding computer programming as a literacy.

Wan, X., Kumar, A., and Li, X. (2024). How do product recommendations help consumers search? evidence from a field experiment. *Management Science*, 70(9):5776–5794.

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., et al. (2023a). Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60.

Wang, Q., Huang, Y., Singh, P. V., and Srinivasan, K. (2023b). Algorithms, artificial intelligence and simple rule based pricing. *Available at SSRN 4144905*.

Wang, W., Li, B., Luo, X., and Wang, X. (2023c). Deep reinforcement learning for sequential targeting. *Management Science*, 69(9):5439–5460.

Wang, Y., Tao, L., and Zhang, X. X. (2024). Recommending for a multi-sided marketplace: A multi-objective hierarchical approach. *Marketing Science*.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Weitzman, M. (1979). Optimal search for the best alternative. *Econometrica*, 47(3):641–654.

Wieting, M. and Sapi, G. (2021). Algorithms in the marketplace: An empirical analysis of automated pricing in e-commerce. *Available at SSRN 3945137*.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., et al. (2022). Sustainable AI: Environmental implications, challenges and opportunities. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 4, pages 795–813.

Xu, X., Wang, Q., Liu, Y., and Qiu, L. (2026). Social audience size as a reference point: Evidence from a field experiment. *Management Science*, 72(5):4298–4318.

Ye, T. and Shankar, V. (2025). Close a store to open a pandora's box? the effects of store closure on sales, omnichannel shopping, and mobile app usage. *Marketing Science*, 44(4):820–837.

Yoganarasimhan, H. (2020). Search personalization using machine learning. *Management Science*, 66(3):1045–1070.

Yuan, Z., Chen, A. Y., Wang, Y., and Sun, T. (2025). How recommendation affects customer search: A field experiment. *Information Systems Research*, 36(1):84–106.

Zahra, S. A. and George, G. (2002). Absorptive capacity: A review, reconceptualization, and extension. *Academy of management review*, 27(2):185–203.

Zhang, J., Yang, M., Bi, X., and Wei, Q. (2024). Algorithmic governance via recommender systems: The case of short video platform. *Available at SSRN 4572513*.

Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Zhang, X., Ferreira, P., Godinho de Matos, M., and Belo, R. (2021). Welfare properties of profit maximizing recommender systems: Theory and results from a randomized experiment. *MIS Quarterly*, 45(1).

Zhou, M., Zhang, J., and Adomavicius, G. (2023). Longitudinal impact of preference biases on recommender systems' performance. *Information Systems Research*.

Appendix A

APPENDIX TO "GENERATE" THE FUTURE OF WORK THROUGH AI: EMPIRICAL EVIDENCE FROM ONLINE LABOR MARKETS

A.1 Platform Information and Matching Process

The focal platform, a publicly traded entity, stands as one of the preeminent online labor marketplaces, boasting millions of visits each month. As of the close of 2023, it had amassed over 70 million registered users and hosted upwards of 20 million job postings. The platform primarily utilizes a client-driven matching mechanism, which can be elucidated through the accompanying flowchart:

1. **Project Posting**: A client (denoted as u) posts a job (project) on the platform.

2. **Bid Submission**: Freelancers can browse the available project list and potentially bid on this project. Should no bids be submitted, the process for this project terminates.

3. **Freelancer Selection**: In cases where one or more freelancers place bids, the client u evaluates these bids and may select a freelancer (denoted as v) for the project.

4. **Project Execution**: The chosen freelancer v undertakes the project. If v does not complete the project, the procedure concludes prematurely.

5. **Payment and Conclusion**: Upon successful project completion, the client compensates the freelancer, effectively finalizing the transaction.

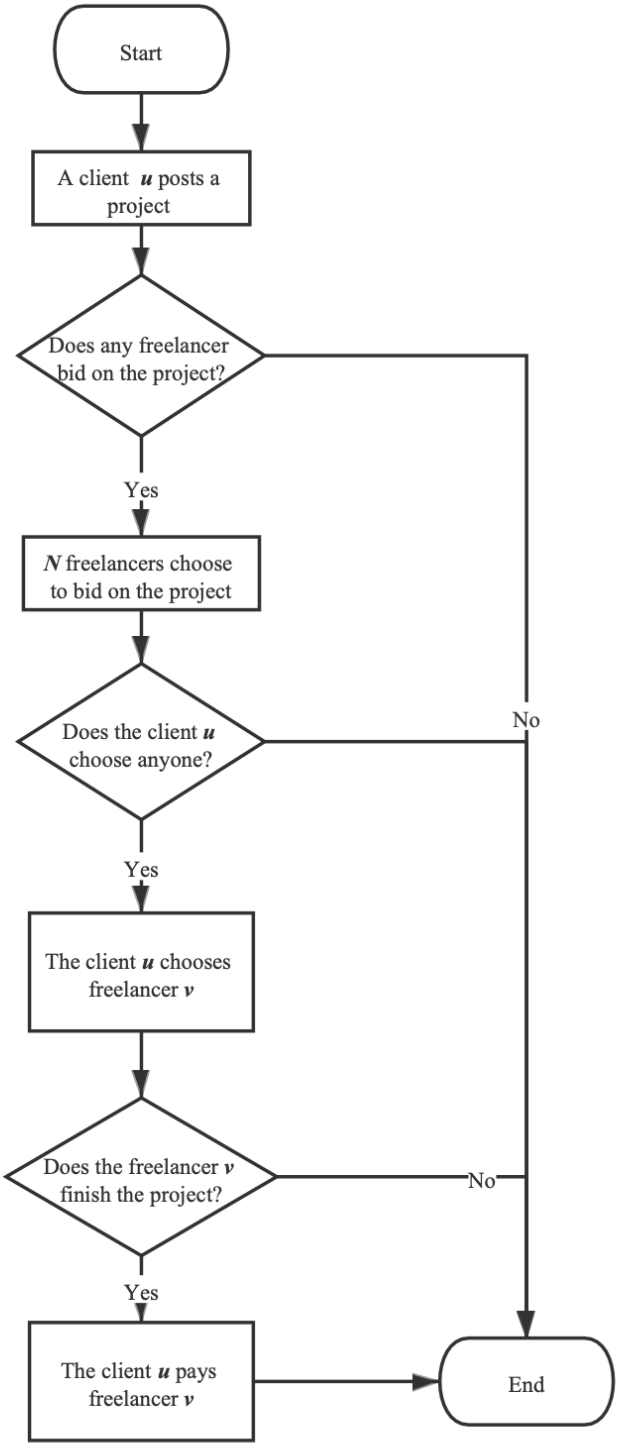


Figure A.1: Matching Process

A.2 Detailed Job Classification, Submarket Construction Process, and LM-AIOE-Based Treatment Definition

In this appendix, we provide a detailed description of the job classification procedure and the construction of submarkets. Our raw dataset includes all job postings along with their associated skill tags (JT), with many jobs sharing identical sets of skill tags. Jobs that share the same skill set are assumed to belong to a common demand pool and therefore compete for the same group of freelancers. Furthermore, many skill sets exhibit hierarchical or subset relationships. For example, Skill Set A may include [skill tag 1, skill tag 2, skill tag 3, ..., skill tag 7], while Skill Set B consists of only [skill tag 1, skill tag 2, skill tag 3]. Since Skill Set B is a subset of Skill Set A, both are considered part of the same demand pool and are presumed to compete for a similar pool of freelance labor. Building on this logic, we also treat skill tag sets that exhibit a high degree of overlap or semantic similarity as belonging to the same demand pool, assuming they compete for comparable freelancer resources. These skill sets, which share a common demand pool and freelancer pool, can be collectively regarded as a group that we define as a single submarket on the platform.

To classify all jobs based on their associated skill tag sets and subsequently construct submarkets, we first implement a multi-step clustering procedure to aggregate similar skill sets into distinct groups. Each job is then assigned to a specific group based on its associated skill set. Each resulting group represents a collection of jobs that require similar skill sets, thereby reflecting a common type of labor demand and forming the basis of a submarket. The relationships among jobs, skill sets, skill tag clusters, and submarkets are illustrated in Figure A.2.

After clustering, we identify 1,082 distinct submarkets from 485,370 unique skill sets, which serve as the units of our market-level analysis. Since each job is associated with a specific skill set, and each skill set is mapped to a single cluster, we are able to assign every job to a corresponding submarket. This mapping provides us with a complete classification of jobs into submarkets. Using these submarkets, we proceed to define treatment and control

groups at both the market and freelancer levels. Specifically, we leverage the LM-AIOE, a variant of the AI Occupational Exposure (AIOE) Index developed by Felten et al. (2023, 2021), which quantifies the degree to which various occupations are exposed to AI tools based on large language models. We begin by calculating the semantic similarity between each of the platform's 2,719 skill tags and the 774 occupations for which LM-AIOE scores are available. Each skill tag is then assigned an exposure score (LM-AIOE value) based on the most semantically aligned occupation. Next, we compute a weighted average LM-AIOE score for each submarket, where the weights reflect the distribution of skill tags within the submarket. We then adopt a conservative method by seting a threshold to define a binary treatment variable, classifying submarkets into treated and control groups. The details of this process are summarized in the pseudocode presented in Algorithm 1, and the complete procedure for the multi-step clustering and the construction of treatment and control submarkets is outlined below:

1. **Skill Set Embedding**: We use the Sentence Transformer model (SBERT) [1] to generate vector representations for each skill tag set ST.

2. **Clustering of Skill Tag Sets Based on Embedding Representations**: We cluster the skill tag sets ST based on their embedding representations using the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm.

3. **LM-AIOE Value Assignment for Skill Tags**: We leverage the 774 occupations OT in the LM-AIOE data, each associated with a specific LM-AIOE score, to assign exposure values to skill tags. Specifically, we compute the cosine similarity between each of the platform's 2,719 skill tags ST and each occupation.Then each skill tag is matched to the most semantically similar occupation, and its corresponding LM-AIOE value is assigned as the LM-AIOE value of this skill tag.

[1]The Sentence Transformer model (SBERT) is a state-of-the-art neural architecture that generates semantically meaningful sentence or text embeddings. Reimers and Gurevych (2019)

4. **Submarket LM-AIOE Values Calculation**: We use the LM-AIOE values of skill tags to compute the weighted LM-AIOE score for each submarket. Specifically, each submarket consists of multiple skill sets, and each skill set is composed of multiple skill tags. Within each submarket, skill tags occur with different frequencies. We define the frequency of a skill tag as $\frac{n_{\text{tag ST}}}{\sum n_{\text{all tags}}}$ and use this value as its weight. The submarket's LM-AIOE score is then calculated as the weighted average of the LM-AIOE values of its skill tags.

5. **Submarket Classification Based on a Defined LM-AIOE Threshold**: We define a binary treatment variable $Treat$ for each submarket based on a predefined LM-AIOE threshold TH. A submarket is assigned $Treat = 1$ if its LM-AIOE value exceeds the threshold TH, and $Treat = 0$ otherwise.

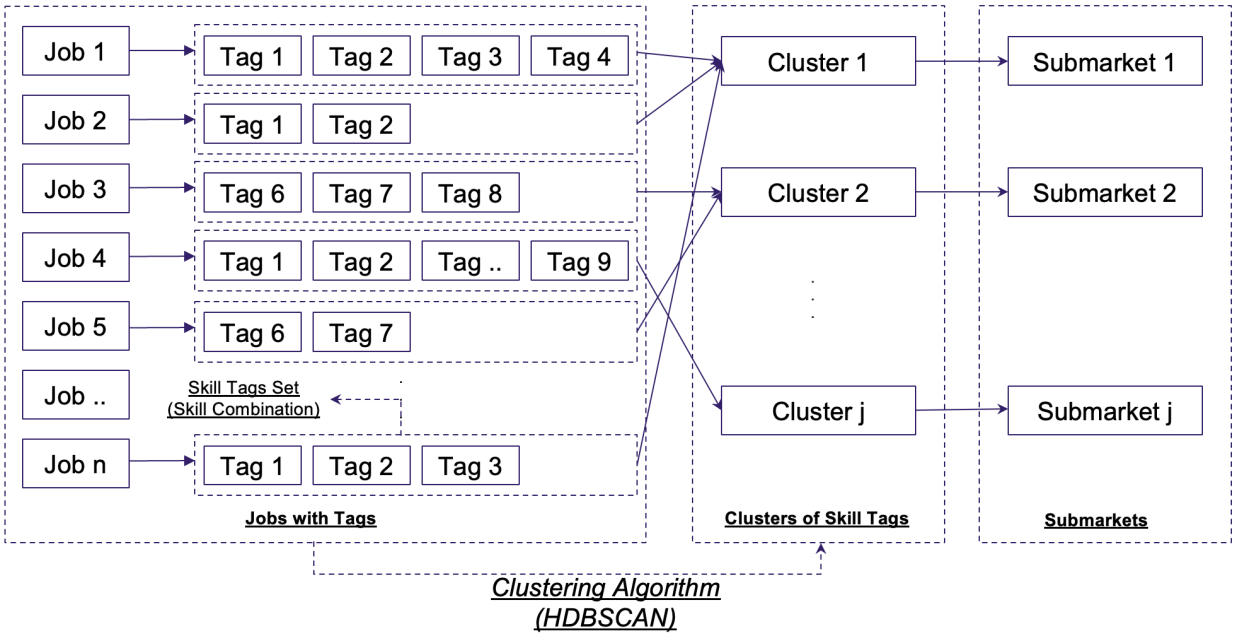


Figure A.2: The Relationship between Jobs, Skill Sets, Skill Tags Clusters, and Submarkets.

Algorithm 1 Submarket Construction and LM-AIOE-Based Treatment Definition

Require: Skill Tag Sets ST, LM-AIOE Dataset Occupations OT, Skill Tags List

1. Skill Set Embedding Generation

for each skill tag set $st \in ST$ **do**

Compute embedding E_{st} using Sentence Transformer (SBERT)[2].

end for

2. Skill Tag Set Clustering

Cluster all embedding vectors E_{st} using HDBSCAN algorithm.

Obtain clusters as distinct submarkets.

3. Assign LM-AIOE Values to Skill Tags

for each skill tag $t \in$ Skill Tags List **do**

for each occupation $ot \in OT$ **do**

Compute cosine similarity $sim(t, ot)$

end for

Assign the LM-AIOE value of the most similar occupation to tag t

end for

4. Calculate Submarket LM-AIOE Values

for each submarket s **do**

Initialize submarket LM-AIOE score $LM\text{-}AIOE_s$

Count frequency of each skill tag ST_s within submarket s

for each skill tag $t \in ST_s$ **do**

Calculate tag frequency weight $w_t = \frac{n_t}{\sum_{t' \in ST_s} n_{t'}}$

Update submarket LM-AIOE: $LM\text{-}AIOE_s \mathrel{+}= w_t \times LM\text{-}AIOE_t$

end for

end for

Algorithm 1 Submarket Construction and AIOE-Based Treatment Definition (Continued)

5. Define Treatment Variable for Submarkets

for each submarket s **do**

  if $LM\text{-}AIOE_s > TH$ **then**

    Set $Treat_s = 1$

  else

    Set $Treat_s = 0$

  end if

end for

A.3 Variable Summary Statistics

This section reports summary statistics for the primary variables of interest. Because each submarket appears on the platform at different times, the number of observations varies by submarket. We define a submarket's inception as the month of its first job posting; for example, if the first posting occurs in January 2022, we set January 2022 as the submarket's start date. In each subsequent month, we record the count of job postings, assigning zero to months with no activity. Prior to January 2022, there are no observations for that submarket. As for freelancers, we treat their registration time as the moment they first enter the market.

It is notable that the variables exhibit substantial positive skewness. After applying the natural logarithm, the means and standard deviations become comparable in scale, rendering conventional count models unsuitable. The same pattern arises in the freelancer-level data: while some active freelancers submit bids to hundreds of jobs, many place only a few bids due to the cold-start problem. Accordingly, we apply log transformations to all variables except *Programming%*, *Rating*, and *Developed*, whose limited ranges obviate the need for transformation. It is noteworthy that 301 occupations in our dataset are associated with at least one skill tag, and that the median LM-AIOE score for these occupations is 0.4. Using this value as a threshold, approximately 85% of submarkets qualify as "treated." This result accords with our theoretical framework and empirical observations: a substantial share of online labor–market tasks produce text or code as their primary outputs, and are therefore highly related to the capabilities of LLM.

Additionally, several data imbalances warrant attention. For the freelancer-month dataset, both *Programming%* and $ln(Avg\_Bids\_job + 1)$ are defined as ratios with *Bids* in the denominator, and *Bids* may be zero. Moreover, *Rating* values are missing for freelancers who have not yet received client reviews; in our DDD analyses, we impute these missing ratings as zero. Finally, $Tenure$ has some small number of missing data and these points are omitted from the DDD analyses. For those with a missing value in $Tenure$, we will regard their first bidding time as the beginning of the panel data.

Table A.1: Variable Summary Statistics

Variable	Observation	Mean	Std	Min	Max
Submarket-Month-Level					
$ln(Jobs+1)$	25,833	2.9487	1.2439	0	6.9256
$ln(Bids+1)$	25,833	5.3525	1.8577	0	10.6361
$ln(Transaction+1)$	25,833	5.5241	3.0187	0	12.3863
$ln(Avg\_Bids\_job+1)$	25,833	2.1322	0.9989	0	5.1180
Freelancer-Month-Level					
$Programming\%$	711,738	0.3201	0.3410	0	1
$ln(Bids+1)$	1,705,081	0.6631	1.1072	0	7.8399
$ln(Transaction+1)$	1,705,081	0.1920	0.9962	0	11.7073
$ln(Avg\_Earn\_Bid+1)$	711,738	0.2129	0.8279	0	11.7073
Freelancer-Level					
$Treat$	132,260	0.8790	0.2752	0	1
$Developed$	132,260	0.0850	0.2788	0	1
$ln(Avg\_Earn\_Bid\_Pre+1)$	132,260	0.2219	0.7734	0	11.0141
$ln(Tenure+1)$	130,686	5.9539	1.5158	0.0029	8.4988
$Rating$	34,325	4.8158	0.6950	0	5
Submarket-Level					
$Treat$	1,082	0.8475	0.3597	0	1
$Programming$	1,082	0.3900	0.4880	0	1

A.4 Notes on Difference-in-difference-in-differences Specification

In Section 3.5.2, we employ a Triple Difference (DDD) specification that includes interactions among $Treat$, $Post$, and $Programming$. However, upon incorporating two-way fixed effects, we are left with only two coefficients to examine, as shown in Equation 3.2. This phenomenon arises because, in our analysis, any submarket categorized under $Programming = 1$ is simultaneously considered a treated submarket ($Treat = 1$), due to its high relatedness to large language models and correspondingly elevated LM-AIOE. In other words, programming-intensive submarkets are a subset of the treated submarkets. Consequently, the interaction term $Treat * Programming$ is equivalent to $Programming$ and is also absorbed by the unit fixed effect. Similarly, $Programming * Post$ is indistinguishable from $Treat * Programming * Post$, leading to the omission of one variable. After omitting these two interaction terms, Equation 3.2 is formulated, wherein $Treat * Post$ captures the post-ChatGPT changes in non-programming-intensive treated submarkets relative to the control submarkets. Meanwhile, $Treat * Programming * Post$ captures additional differences in programming-intensive submarkets compared to non-programming-intensive submarkets, highlighting the heterogeneity of interest.

A.5 Interrupted Time Series Analysis of Freelancer-level Skill Transition

As a supplement to Section 3.5.2, we further examined skill-transition effects at the freelancer level using an Interrupted Time Series Analysis (ITSA). We estimate the ITSA model using the specification described below, with the results presented in Table A.2.

$$Programming\%_{it} = \beta_0 + \beta_1 * Time_t + \beta_2 * Post_t + \beta_3 * PostTime_t + u_i + \epsilon_{it} \quad \text{(A.1)}$$

Here, $Programming\%_{it}$ denotes the average programming intensity of bids submitted by freelancer i in month t. $Time_t$ represents the number of months elapsed since the start of the observation period. $Post_t$ is a binary indicator equal to 0 during the pre-treatment period and 1 during the post-treatment period. $PostTime_t$ equals 0 before the treatment and increases by one in each subsequent post-treatment month. We also include u_i to control for freelancer-level fixed effects, while the error term ϵ_{it} captures unobserved factors that may influence the outcome. To ensure that the skill-transition process is observable, we restrict the sample to freelancers who submitted at least one bid in both the pre- and post-treatment periods. We find a statistically significant post-treatment increase in $Programming\%$ ($\beta_2 = 0.0500$, $p < 0.01$), indicating that freelancers indeed transitioned toward programming jobs following the introduction of ChatGPT.

Table A.2: Results for Interrupted Time Series Analysis

	$Programming\%$
$Time$	0.0010***
	(0.0001)
$Post$	0.0500***
	(0.0009)
$Post\_Time$	-0.0001***
	(0.0003)
Freelancer FE	Yes
Observations	432,558

Note: Robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.6 Freelancer-level Heterogeneity in Skill Transition: DDD and Event-study Evidence

To complement the analysis in Section 3.5.3, we investigate heterogeneity in freelancers' skill-transition behavior by estimating a difference-in-differences-in-differences (DDD) model. The specification is as follows, with results reported in Table A.3:

$$\begin{aligned} Programming\%_{it} = \beta_0 + \beta_1 * Treat_i * Post_t * Moderator_i + \beta_2 * Treat_i * Post_t \\ + \beta_3 * Post_t * Moderator_i + u_i + T_t + \epsilon_{it} \end{aligned} \tag{A.2}$$

In this specification, $Treat_i$ is a continuous variable ranging from 0 to 1, representing the proportion of each freelancer's pre-treatment bids submitted in treated submarkets. This variable captures the degree of initial skill relatedness to ChatGPT's capabilities. $Moderator_i$ reflects freelancer characteristics, operationalized in three ways across the columns: freelancer rating (Column (1)), the logarithm of pre-treatment earning efficiency (Column (2)), and a binary indicator for whether the freelancer is located in a developed country (Column (3)).

The estimates in Columns (1) and (2) are consistent with those from the main analysis: higher-skilled freelancers—proxied by higher ratings or by greater earning efficiency—are significantly more likely to shift into programming work ($\beta_1 = 0.0256$ and 0.0328, respectively; $p < 0.01$). Column (3) reveals no statistically significant difference in transition behaviour between freelancers in developed and developing countries ($\beta_1 = 0.0049$; $p > 0.10$). Finally, Column (4) indicates that freelancers with longer platform tenure are more inclined to move into programming-intensive jobs ($\beta_1 = 0.027$; $p < 0.01$).

Furthermore, we estimate an event-study (relative-time) specification to test the parallel-trends assumption. As shown in Figure A.3, the pre-treatment coefficients cluster around zero and are statistically insignificant, indicating no violation of the parallel-trends assumption.

$$\begin{aligned} Outcome_{it} &= \sum_{k \neq -1} \beta_k \, D_{i,t=k} + u_i + T_t + \epsilon_{it} \\ &= \sum_{k \neq -1} \beta_k \left(D_i \cdot \mathbf{1}\{t - \tau_i = k\} \right) + u_i + T_t + \epsilon_{it}, \end{aligned} \tag{A.3}$$

Table A.3: DDD Analyses about Skill-Transition Effects

	Programming%			
	(1)	(2)	(3)	(4)
$Treat \times Post \times Moderator$	0.0256***	0.0328***	0.0040	0.0271***
	(0.0006)	(0.0013)	(0.0045)	(0.0010)
$Treat \times Post$	-0.0366***	0.0084***	0.0226***	-0.1530***
	(0.0022)	(0.0017)	(0.0016)	(0.0068)
$Post \times Moderator$	-0.0165***	-0.0212***	-0.0193***	-0.0155***
	(0.0005)	(0.0012)	(0.0043)	(0.0009)
Constant	0.3275***	0.3158***	0.3127***	0.3482***
	(0.0008)	(0.0006)	(0.0006)	(0.0024)
Moderator	*Rating*	$ln(Earning\_Bid\_Pre + 1)$	*Developed*	$ln(Tenure + 1)$
Freelancer FE	Yes	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes	Yes
Observations	711,738	711,738	711,738	697,908
R-squared	0.8584	0.8578	0.8576	0.8572

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

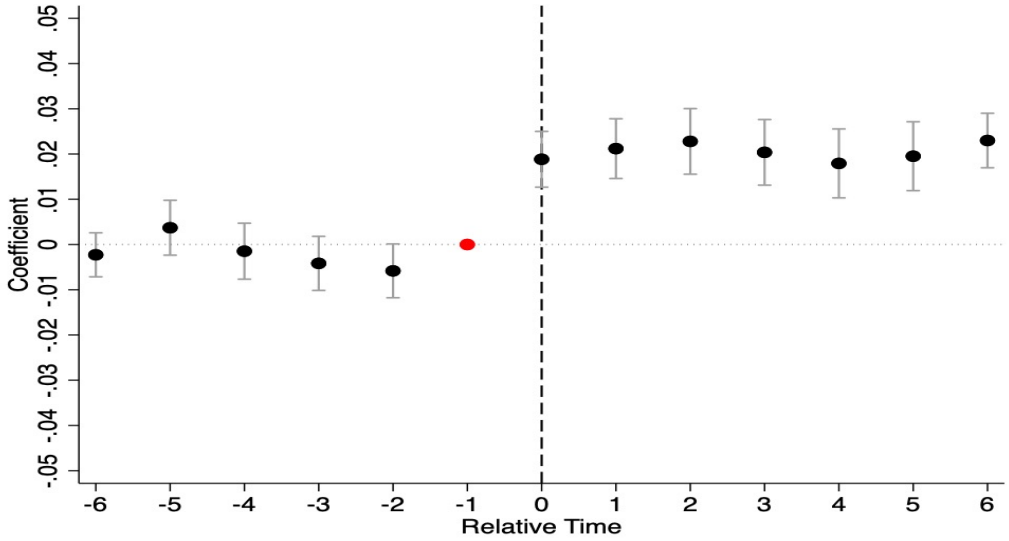


Figure A.3: Pre-Trends and Temporal Dynamics of Freelancer's Programming Proportion

A.7 Additional Details on Freelancer-Level Earnings

Similar to our market-level analysis, we examine the effect of ChatGPT on freelancer-level earnings. Using the specification in Equation 3.1, we estimate the impact on freelancer's total transaction value and the mechanisms driving the changes, including changes in bidding behavior and earning efficiency.

Since freelancers who did not receive any payment during their pre-treatment period had no potential for income loss, we first focus on freelancers who were awarded at least one job and had prior earning activity, as shown in Table A.4. We examine how the magnitude of change in transactions is affected by ChatGPT, which varies across different freelancers. Here, $Treat$ is a continuous variable ranging from 0 to 1, representing the proportion of bids submitted to treated submarkets. A higher value of $Treat$ indicates that the freelancer is more likely to be affected by ChatGPT's demand displacement effect. We find a significant decline in transaction value ($\beta_1 = -0.1162$, $p < 0.05$), which is primarily driven by a substantial reduction in the number of bids submitted ($\beta_1 = -0.0870$, $p < 0.01$) and a small, statistically insignificant decrease in earning efficiency ($\beta_1 = -0.0041$, $p > 0.1$).

In Tables A.5 and A.6, we separately analyze freelancers who made a skill transition and those who did not. Freelancers who transitioned exhibit no significant decline in transaction value; their bidding activity remains stable, and their earning efficiency significantly improves. In contrast, those who did not transition experience substantial reductions in both transaction value and bid volume, along with a significant decline in earning efficiency.

For completeness, Tables A.7, A.8 and A.9 replicate these analyses including all freelancers (even those with zero pre-treatment earnings). Unsurprisingly, the estimated effects are attenuated because most freelancers report zero earnings before treatment, which precludes substantial further reduction, but the qualitative patterns are consistent.

Table A.4: Impact on Freelancers' Earnings

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	-0.1162**	-0.0870***	-0.0041
	(0.0467)	(0.0265)	(0.0351)
Constant	1.0045***	1.7269***	0.6292***
	(0.0152)	(0.0086)	(0.0110)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	310,771	310,771	218,716
R^2	0.2787	0.6704	0.2437

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.5: Impact on Freelancers' Earnings (Transition)

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	0.0695	0.0483	0.1175***
	(0.0523)	(0.0322)	(0.0452)
Constant	1.0545***	1.8877***	0.5926***
	(0.0175)	(0.0108)	(0.0149)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	217,637	217,637	160,571
R^2	0.2903	0.6704	0.2458

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.6: Impact on Freelancers' Earnings (Non-Transition)

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	-0.2081***	-0.1460***	-0.1555***
	(0.0540)	(0.0340)	(0.0496)
Constant	0.7619***	1.2117***	0.6828***
	(0.0157)	(0.0099)	(0.0131)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	98,504	98,504	60,573
R^2	0.2299	0.6089	0.2647

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.7: Impact on Freelancers' Earnings, Including Those with Zero Pre-Treatment Earnings

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	-0.0228***	-0.0425***	-0.0140
	(0.0072)	(0.0057)	(0.0094)
Constant	0.2000***	0.6779***	0.2177***
	(0.0025)	(0.0020)	(0.0033)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	1,705,081	1,705,081	711,738
R^2	0.3608	0.6710	0.3215

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.8: Impact on Freelancers' Earnings, Including Those with Zero Pre-Treatment Earnings (Transition)

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	0.0109	0.0229***	0.0084
	(0.0087)	(0.0073)	(0.0121)
Constant	0.2678***	0.8227***	0.2542***
	(0.0030)	(0.0025)	(0.0043)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	932,942	932,942	430,712
R^2	0.3763	0.6990	0.3178

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.9: Impact on Freelancers' Earnings, Including Those with Zero Pre-Treatment Earnings (Non-Transition)

	$\ln(y+1)$		
	Transaction	*Bids*	*Avg_Earn_Bid*
	(1)	(2)	(3)
$Treat \times Post$	-0.0117	0.0033	-0.0179
	(0.0077)	(0.0067)	(0.0118)
Constant	0.0984***	0.4487***	0.1542***
	(0.0026)	(0.0023)	(0.0039)
Freelancer FE	Yes	Yes	Yes
Year–Month FE	Yes	Yes	Yes
Observations	814,668	814,668	294,952
R^2	0.2908	0.5388	0.3407

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.8 The Heterogeneous Effects of Generative AI Across Programming Domains

In our market-level analysis, we find that although programming-intensive and non-programming-intensive submarkets exhibit no consistent differences in labor demand, the contraction in labor supply is significantly smaller in programming-intensive submarkets. We aim to take one step further in exploring whether any heterogeneity exists among programming jobs. Large language models support numerous programming languages since they have been trained on multi-language code corpora (Manh et al. 2023, Kocetkov et al. 2022) and evaluated on benchmarks spanning a wide range of programming languages (Cassano et al. 2022). Broadly, programming languages can be categorized into two distinct types: scripting languages and compiled languages (Prechelt 2000, Ousterhout 1998). These categories differ in syntactic complexity and learning cost: scripting languages typically feature simpler syntax and lower learning barriers, while compiled languages are generally more complex and demanding (Lokkila et al. 2023), potentially influencing how different languages mediate the impact of generative AI.

In light of this, we further divide the programming-intensive submarkets into two segments based on the predominant type of programming language, in order to examine the heterogeneous impact on scripting-language-dominated versus compiled-language-dominated submarkets. Specifically, we classify each submarket according to the relative proportions of scripting and compiled languages used. If the proportion of scripting languages exceeds that of compiled languages within a submarket, it is categorized as scripting-language-dominated; otherwise, it is considered compiled-language-dominated. A small number of submarkets, however, do not predominantly feature either language type and instead rely on alternative paradigms, such as the declarative language SQL. As there are only four such submarkets, we exclude them from the analysis.

Subsequently, following the model specifications in Section 3.4, we separately compare the demand, supply, transaction, and competition in scripting-language-dominated and

compiled-language-dominated submarkets with those in the control submarkets. The estimation results are reported in Table A.10. Panel A compares scripting-language-dominated submarkets with the control group, while Panel B compares compiled-language-dominated submarkets with the control group. Consistent with the main results, both types of submarkets show a notable decline in labor demand ($\beta_1 = -0.1880$ or -0.2340, $p < 0.01$) and total transaction volume ($\beta_1 = -0.2830$, $p < 0.01$ or $\beta_1 = -0.2680$, $p < 0.05$) relative to the control group. However, it is worth noting that on the supply side, measured by the total number of bids, scripting-language-dominated submarkets do not exhibit a significant decline relative to the control group ($\beta_1 = -0.0640$, $p > 0.1$). In contrast, compiled-language-dominated submarkets still experience a significant reduction in labor supply ($\beta_1 = -0.1310$, $p < 0.05$). This result may reflect the lower barrier to entry associated with scripting languages compared to compiled languages, which generally require more formal training. Such suggestive evidence indicates that the skill-transition effect is inherently constrained: freelancers cannot move freely into every role without encountering substantive limitations.

Table A.10: The Heterogeneous Effects of Generative AI Across Different Types of Programming Submarkets

	Panel A			
	Programming Type: Scripting Language VS Control			
	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.1880***	-0.0640	-0.2830***	0.2130***
	(0.0348)	(0.0524)	(0.1050)	(0.0408)
Constant	2.8630***	5.1990***	5.4670***	2.0950***
	(0.0082)	(0.0123)	(0.0246)	(0.0096)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	10,268	10,268	10,268	10,268
R^2	0.8950	0.8480	0.6360	0.5440
	Panel B			
	Programming Type: Compiled Language VS Control			
	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2340***	-0.1310**	-0.2680**	0.1810***
	(0.0371)	(0.0560)	(0.1030)	(0.0421)
Constant	2.7590***	4.8870***	5.1510***	1.9330***
	(0.0067)	(0.0102)	(0.0187)	(0.0076)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	7,496	7,496	7,496	7,496
R^2	0.8980	0.8470	0.6450	0.5240

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.9 The Impact of Generative AI on Market Entry Decisions of New Freelancers

Given that some freelancers can be active in online labor markets for an extended period and work for diverse jobs, several potential mechanisms may explain the average and heterogeneous impacts on the labor supply side, as documented in Sections 3.5.1 and 3.5.2. Utilizing the market-level data, we explore the influx of new freelancers as one such mechanism. To this end, we introduce a new outcome variable, $New\_Freelancer$, defined as the monthly count of freelancers who newly register and submit at least one bid within specific submarkets. We employ the same econometric specifications to perform DiD and DDD analyses, with the results presented in Table A.11. The estimates suggest that the decline in new participants is one contributing mechanism to the observed decrease in average labor supply across treatment submarkets ($\beta_1 = -0.1450$, $p < 0.01$). However, there is no significant heterogeneity in the number of new freelancers between programming-intensive and non-programming-intensive submarkets ($\beta_2 = -0.0430$, $p > 0.1$), effectively ruling out differences in new participation as the driving mechanism for heterogeneous effects on labor supply. We also test the parallel trend assumption and examine whether any differential trends existed prior to the introduction of ChatGPT. The results show that the confidence intervals for all pre-treatment periods encompass zero, suggesting an absence of systematic differences before the intervention.

Table A.11: The Impact of Generative AI on Market Entry Decisions of New Freelancers

	$ln(New\_Freelancer + 1)$	
	(1)	(2)
$Treat * Post$	-0.1450***	-0.1250***
	(0.0371)	(0.0411)
$Treat * Post *$		-0.0430
$Programming$		
		(0.0283)
Constant	3.5910***	3.5910***
	(0.0119)	(0.0119)
Submarket FE	Yes	Yes
Year-Month FE	Yes	Yes
Observations	25,833	25,833
R^2	0.9000	0.9010

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

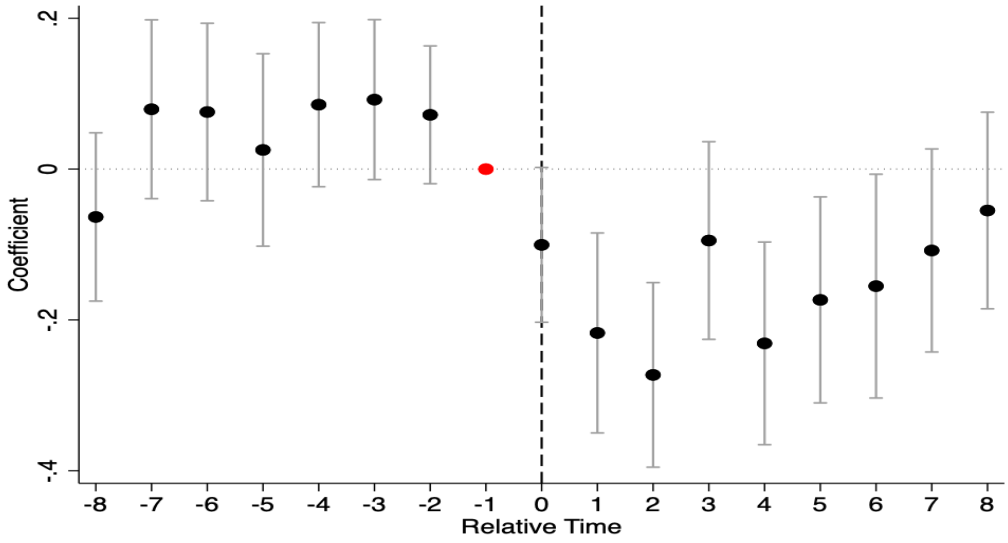


Figure A.4: Pre-Trends and Temporal Dynamics of New Freelancers

A.10 The Impacts on Project Budgets

To better understand the underlying motivations behind freelancers' transitions, we further examine how the introduction of ChatGPT affects job budgets. As demonstrated in Sections 3.5.2, freelancers shift toward bidding on programming-intensive jobs. While these jobs have traditionally commanded higher wages than text-related ones, ChatGPT may have further widened this wage gap. This widening gap likely strengthens freelancers' motivations to transit to programming-intensive submarkets as a way to mitigate Chatgpt's negative impact. Therefore, we examine the wages clients are willing to offer for completing a given task, as reflected in job budgets, and analyze how these budgets have changed differentially between programming-intensive and text-based submarkets following the introduction of ChatGPT.

Specifically, we define a new dependent variable, *Budget*, as the average project budget of all jobs posted within a submarket during a given month. We then follow the DDD model specification outlined in Section 3.5.2 to investigate the heterogeneous treatment effects (HTE) on job budgets. The corresponding estimation results are presented in Table A.12. The negative and statistically insignificant coefficients of the interaction term $Treat \times Post \times Programming$ suggest that the wage gap was not widened by the introduction of ChatGPT. Therefore, the absence of a significant widening in the wage gap allows us to rule out this explanation.

Table A.12: The Impact of Generative AI on Project Budgets

	$ln(Budget + 1)$	
	(1)	(2)
$Treat * Post$	0.0575*	0.0727**
	(0.0327)	(0.0349)
$Treat * Post *$		-0.0327
$Programming$		
		(0.0223)
Constant	5.6690***	5.6690***
	(0.0105)	(0.0105)
Submarket FE	Yes	Yes
Year-Month FE	Yes	Yes
Observations	25,086	25,086
R^2	0.3810	0.3810

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.11 Additional Evidence on Demand-Side Market Dynamics

To provide additional insight into market demand dynamics, we visualize the temporal evolution of three key indicators within the treated submarkets: the proportion of programming-intensive jobs, the average number of required skills per job, and the average job budget. Specifically, we calculate the monthly proportion of programming-intensive jobs in the overall market and plot its trend over time. The results are presented in Figure A.5, where the horizontal axis represents the number of months relative to the treatment point, and the vertical axis indicates the proportion of programming-intensive jobs. The blue dots show the observed monthly values, while the red line illustrates the fitted trend estimated through linear regression. Similarly, we calculate the monthly average number of required skills per job and average job budget across the entire market and plot its changes over time in Figure A.6 and Figure A.7.

Based on the results presented in Figure A.5, the proportion of programming-intensive jobs remains largely stable over the observation period, exhibiting only a slight decline. The overall variation is minimal, and the fitted trend line is nearly flat. This suggests that, on the demand side, clients have not demonstrated any notable shift in preference or a growing tendency toward programming-intensive tasks. Figure A.6 presents the temporal trend in the number of required skills per job. The fitted trend line appears relatively flat with a slight upward slope, indicating a gradual increase in the number of skills requested by clients. Figure A.7 illustrates the monthly evolution of the average job budget, revealing a modest upward trend over time, particularly following the introduction of ChatGPT. This pattern suggests a gradual rise in job budgets within the treated submarkets and aligns with the observed increase in skill requirements. We re-estimate the DiD model separately to compare high- and low-budget submarkets with the control group. The results show a higher demand displacement effect in low-budget submarkets.

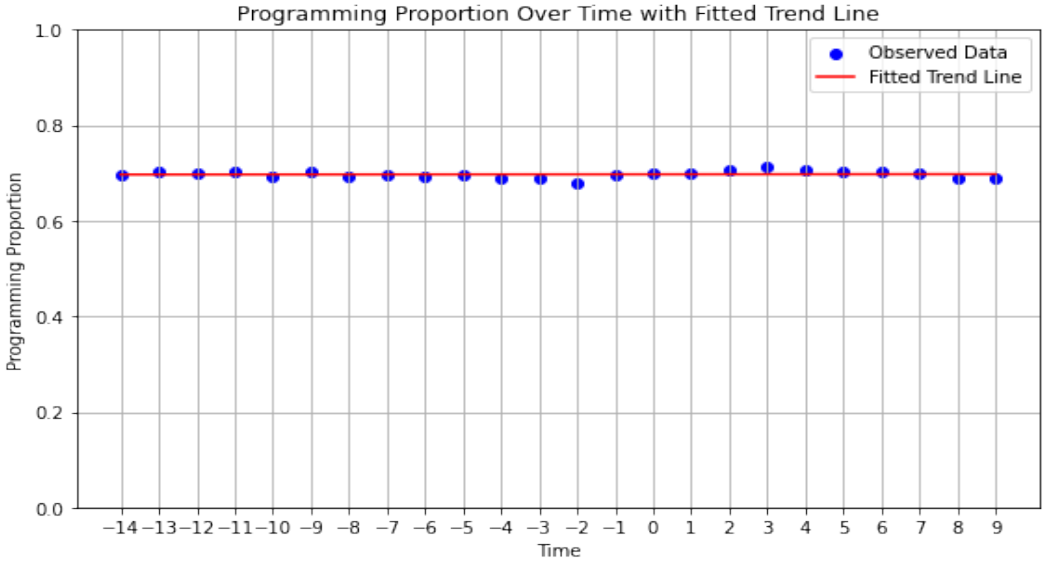


Figure A.5: Temporal Trends in Proportion of Programming Jobs

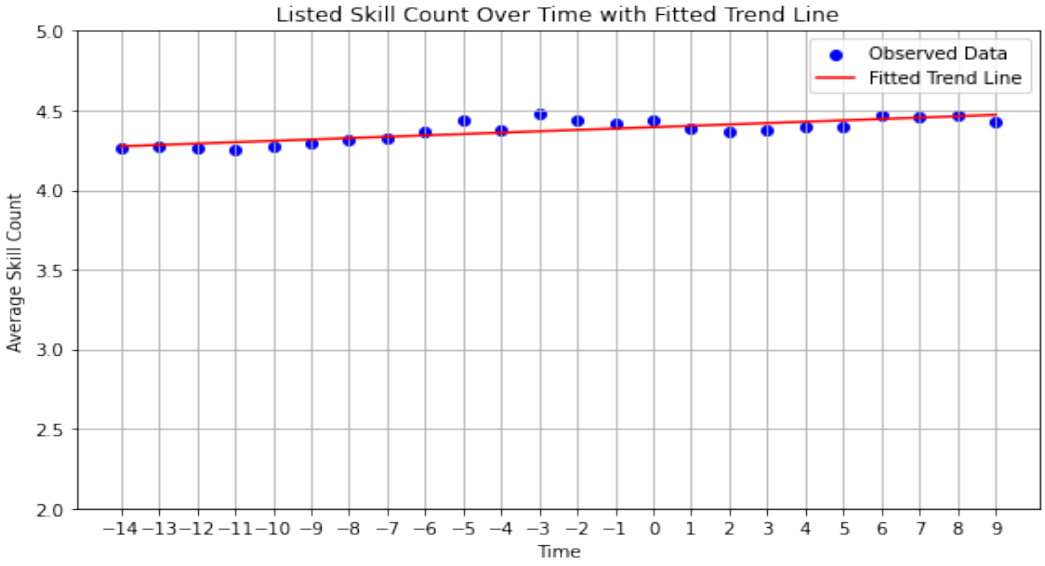


Figure A.6: Temporal Trends in Average Skill Count

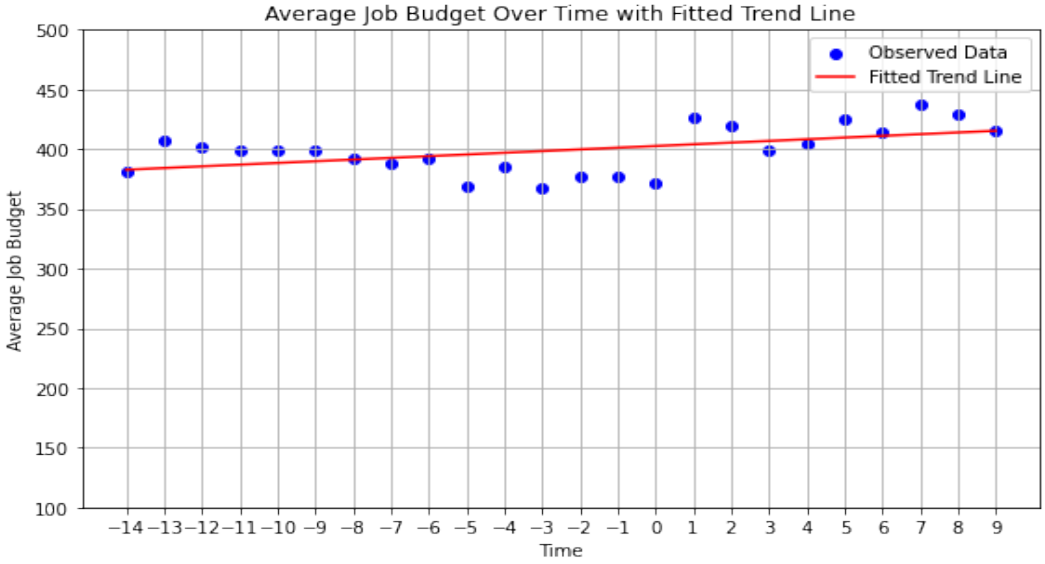


Figure A.7: Temporal Trends in Average Budget

Table A.13: The Heterogeneous Effects of Generative AI Across Sub-markets with Different Budget Levels

	Panel A			
	High-Budget Sub-market			
	ln(y+1)			
	Jobs	Bids	Transaction	Avg_Bids_job
	(1)	(2)	(3)	(4)
Treat × Post	–0.1850***	–0.1560***	–0.2330**	0.0886**
	(0.0381)	(0.0566)	(0.1100)	(0.0400)
Constant	3.0780***	5.6150***	5.7130***	2.2720***
	(0.0121)	(0.0181)	(0.0351)	(0.0128)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	17,421	17,421	17,421	17,421
R^2	0.8890	0.8360	0.5970	0.5010
	Panel B			
	Low-Budget Sub-market			
	ln(y+1)			
	Jobs	Bids	Transaction	Avg_Bids_job
	(1)	(2)	(3)	(4)
Treat × Post	–0.2950***	–0.1920**	–0.4890***	0.2430***
	(0.0574)	(0.0873)	(0.1540)	(0.0713)
Constant	2.8970***	4.9720***	5.4430***	1.7070***
	(0.0183)	(0.0279)	(0.0493)	(0.0228)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	8,412	8,412	8,412	8,412
R^2	0.8950	0.8380	0.6430	0.5320

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.12 The Impacts on Image-Related Jobs

We further investigate how image-related work is affected by the emergence of generative AI. Specifically, we consider two key events: the release of the first publicly available text-to-image AI model, and the subsequent launch of ChatGPT. To analyze their respective impacts on the image-related submarket, we estimate two regression specifications. Consistent with our earlier submarket construction procedure, we apply the same clustering approach to the skill sets of image-related jobs to define the corresponding submarkets. In this setup, control submarkets serve as the baseline, while the treatment group comprises the image-related submarkets.

For the first specification, we use $Post\_image$ to denote the the first publicly available version as the treatment time, set at July 2022 on a monthly level. For the second specification, we adopt the same $Post$ variable from our main analyses, using the ChatGPT launch date as the start of the treatment period. We define $Treat_{image,i}$ as an indicator variable equal to 1 if unit i corresponds to an image-related submarket and 0 otherwise.

$$Outcome_{it} = \beta_0 + \beta_1 * Treat_{image,i} * Post_{image,t} + u_i + T_t + \epsilon_{it} \tag{A.4}$$

$$Outcome_{it} = \beta_0 + \beta_1 * Treat_{image,i} * Post_t + u_i + T_t + \epsilon_{it} \tag{A.5}$$

Equation A.4 examines changes in demand, supply, and matching outcomes in image-related submarkets relative to control submarkets, in response to the introduction of text-to-image generative AI. This analysis includes data only up to November 2022, thereby excluding the post-ChatGPT period, with the corresponding results reported in Table A.14. In the second analysis, we explore whether the widespread attention generated by ChatGPT led to increased interest in text-to-image AI applications, potentially encouraging broader adoption and influencing labor market outcomes. For this purpose, we extend the observation window to include the post-ChatGPT period and estimate relative changes in image-related submarkets. The results of this spillover analysis are presented in Table A.15.

Both sets of analyses show insignificant results across all measurements. This suggests

that current text-to-image generative AI may not be general-purpose or powerful enough to influence a wide range of jobs in online labor markets.

Table A.14: The Effect of Generative AI on Image-related Labor Market

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat_{image} * Post_{image}$	-0.0021	0.0854	-0.1500	0.0233
	(0.0402)	(0.0571)	(0.1390)	(0.0600)
Constant	3.0470***	5.4600***	5.1680***	2.1920***
	(0.0075)	(0.0107)	(0.0261)	(0.0113)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	5,461	5,461	5,461	5,461
R^2	0.9250	0.9130	0.6600	0.5740

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.15: The Spillover Effect of ChatGPT's Success on Image-related Labor Market

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat_{image} * Post$	-0.0536	-0.0619	-0.0775	0.0683
	(0.0337)	(0.0470)	(0.0959)	(0.0421)
Constant	2.9830***	5.5100***	5.0150***	2.3030***
	(0.0071)	(0.0099)	(0.0201)	(0.0088)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	8,792	8,792	8,792	8,792
R^2	0.9190	0.9040	0.6600	0.5730

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.13 Robustness Check - Alternative Treatment/Control Group Definition Thresholds

Given that our submarket classification is based on a conservative approach using the median value of LM-AIOE to distinguish between submarkets with relatively high and low LM-AIOE levels, we acknowledge that observations near the threshold may introduce sensitivity into the estimation. Varying the threshold can alter submarket classifications, particularly for those with LM-AIOE values close to the cutoff. For example, lowering the threshold may lead to some submarkets that were previously assigned to the control group being reclassified into the treatment group.

To address this concern and enhance the robustness of our findings, we implement an alternative thresholding strategy to ensure that the results are not driven by the specific choice of cutoff. Specifically, we assess the sensitivity of our results by adjusting the original median-based threshold by ± 0.05 units. The lower threshold is defined as the median minus 0.05, while the higher threshold is defined as the median plus 0.05. This approach enables us to assess whether submarkets classified as high or low LM-AIOE continue to exhibit consistent patterns that are not influenced by the classification criteria. The estimation results based on these alternative thresholds are presented in Table A.16 and Table A.17. Table A.16 reports the results using the lower threshold, while Table A.17 presents the results using the higher threshold.

Interestingly, we find that although the estimated coefficients exhibit slight fluctuations compared to those reported in Table 3.2, they remain statistically significant. Specifically, the outcomes for demand, supply, transaction value consistently show a moderate decline, indicating stable and coherent patterns of heterogeneous treatment effects. These results confirm the robustness and consistency of our findings under alternative threshold specifications.

Table A.16: Main Effects Replication with Low Threshold

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2200***	-0.1830***	-0.3320***	0.1300***
	(0.0359)	(0.0535)	(0.1010)	(0.0397)
$Treat * Post *$	0.0207	0.1330***	0.0523	0.1090***
Programming				
	(0.0224)	(0.0346)	(0.0667)	(0.0267)
Constant	3.0180***	5.3930***	5.6250***	2.0740***
	(0.0109)	(0.0162)	(0.0311)	(0.0121)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8900	0.8410	0.6120	0.5410

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.17: Main Effects Replication with High Threshold

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.1980***	-0.1950***	-0.2990***	0.0862**
	(0.0339)	(0.0513)	(0.0946)	(0.0372)
$Treat * Post *$	0.0294	0.1490***	0.0657	0.1120***
Programming				
	(0.0229)	(0.0352)	(0.0680)	(0.0273)
Constant	3.0060***	5.3910***	5.6080***	2.0890***
	(0.0096)	(0.0145)	(0.0272)	(0.0106)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8900	0.8410	0.6120	0.5410

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.14 Robustness Check - Fake Treatment Timing (Placebo Test)

Since the launch of ChatGPT occurred at the end of the year, holidays and seasonal effects might influence the observed post-shock differences. For instance, these differences could be attributed to an alternative mechanism where treated submarkets naturally experience a more pronounced downward trend around the end of the year compared to control submarkets. To address this concern, we conduct another placebo test using November 30, 2021, the same day in the previous year, as a fake shock date. We maintain the time period from September 2021 to August 2022 and rerun all regressions. As shown in Table A.18, our main findings remain robust, suggesting that they are not driven by end-of-year effects or seasonal factors.

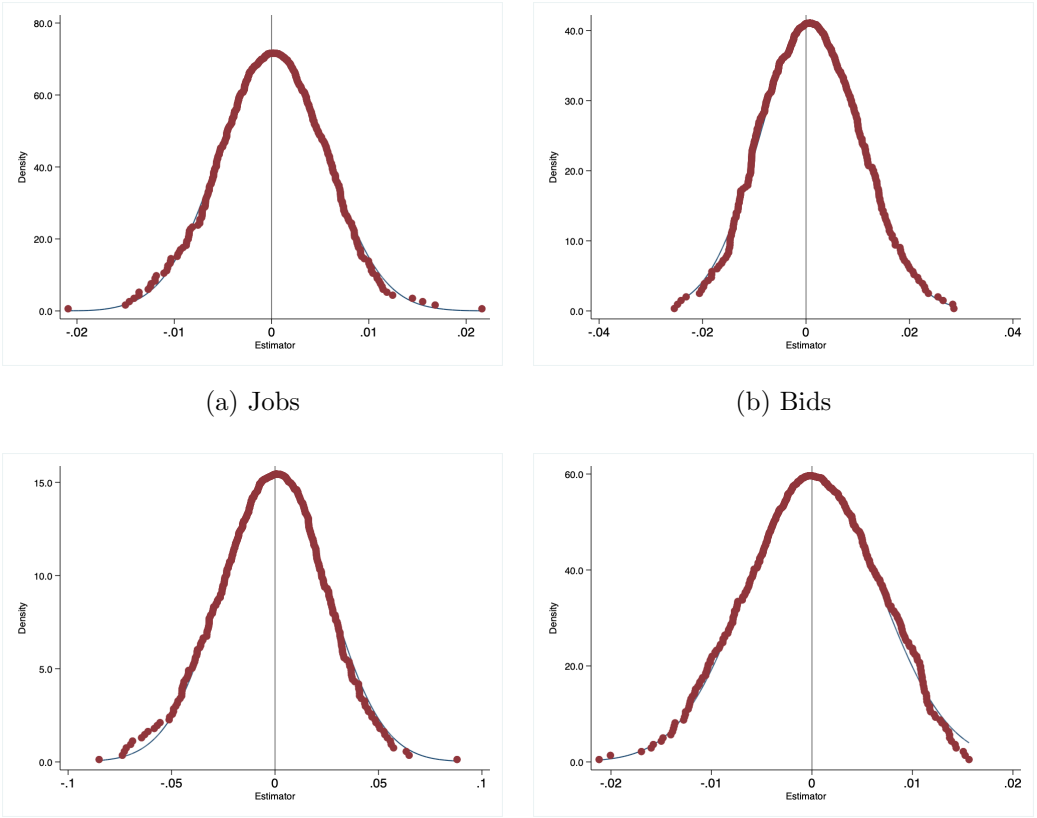
Table A.18: Placebo Test with a Fake Shock Time

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.0203	0.0229	0.2020	0.0706
	(0.0315)	(0.0540)	(0.1280)	(0.0476)
Constant	3.1200***	5.3450***	5.6140***	1.9060***
	(0.0196)	(0.0336)	(0.0796)	(0.0296)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	11,769	11,769	11,769	11,769
R^2	0.9280	0.8730	0.6660	0.6060

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.15 Robustness Check - Treatment Reassignment (Placebo Tests)

Statistically, there is a possibility that the observed effects are driven by random variations across observations. To address this, we conduct a placebo test by randomly reassigning the treatment ($Treat * Post$) within our sample. We simulate this permutation procedure 1,000 times and capture the distribution of the placebo treatment effects for all six main outcome variables based on random treatment assignment. The distribution of placebo treatment effects resembles a normal distribution, as shown in Figure A.8, and our estimated coefficient lies far in the tail of this distribution. This placebo test indicates that the treatment effects are not driven by chance, further reinforcing the robustness of our findings.



(a) Jobs

(b) Bids

(c) Transaction

(d) Average_job_bids

Figure A.8: Placebo Tests

A.16 Robustness Check - Pretrends and Temporal Dynamics

We use a DiD specification for most analyses to uncover how generative AI introduces different trends for treated submarkets versus control submarkets. To attribute post-shock differences to the shock itself, we first need to examine whether these two sets of submarkets follow similar trends before the shock (parallel trend assumption). Additionally, we are interested in whether the observed effects appear primarily at the beginning of the shock (due to novelty effects, etc.) and attenuate over time, or whether these differences strengthen over time. With these objectives in mind, we employ the relative time model to obtain per-period effect estimates, following previous scholarly investigations (Braghieri et al. 2022, Perez-Truglia 2020).

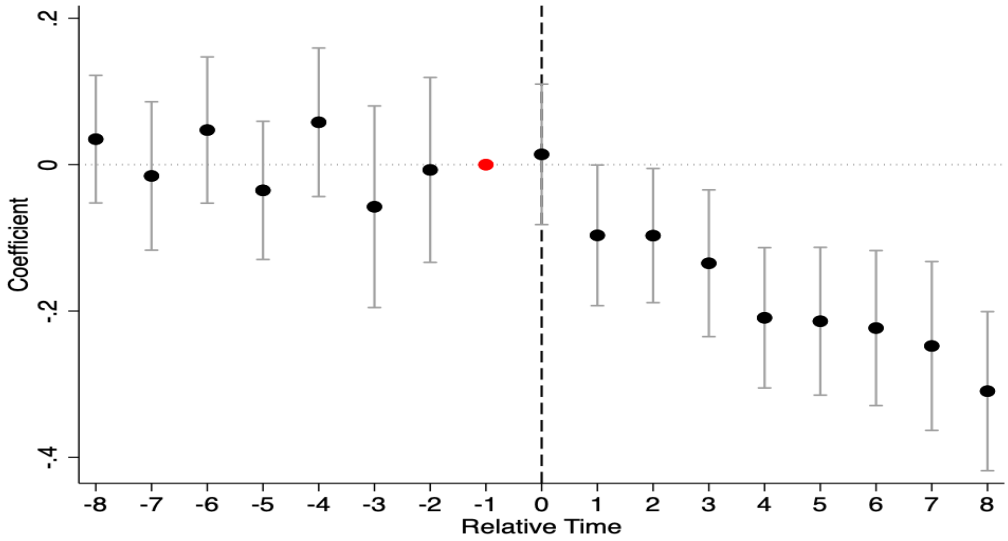
$$\begin{aligned} Outcome_{it} &= \sum_{k \neq -1} \beta_k \, D_{i,t=k} + u_i + T_t + \epsilon_{it} \\ &= \sum_{k \neq -1} \beta_k \left(D_i \cdot \mathbf{1}\{t - \tau_i = k\} \right) + u_i + T_t + \epsilon_{it}, \end{aligned} \tag{A.6}$$

where $D_{i,t=k}$ equals 1 if unit i is treated and period t is k periods relative to unit i's event (adoption) time τ_i, and 0 otherwise; the reference period $k = -1$ is omitted. Equivalently, $D_{i,t=k}$ can be written as $D_i \cdot \mathbf{1}\{t - \tau_i = k\}$, where D_i indicates whether unit i is ever treated and $\mathbf{1}\{\cdot\}$ is the indicator function. u_i and T_t denote unit and time fixed effects, respectively, and ϵ_{it} is the idiosyncratic error term.

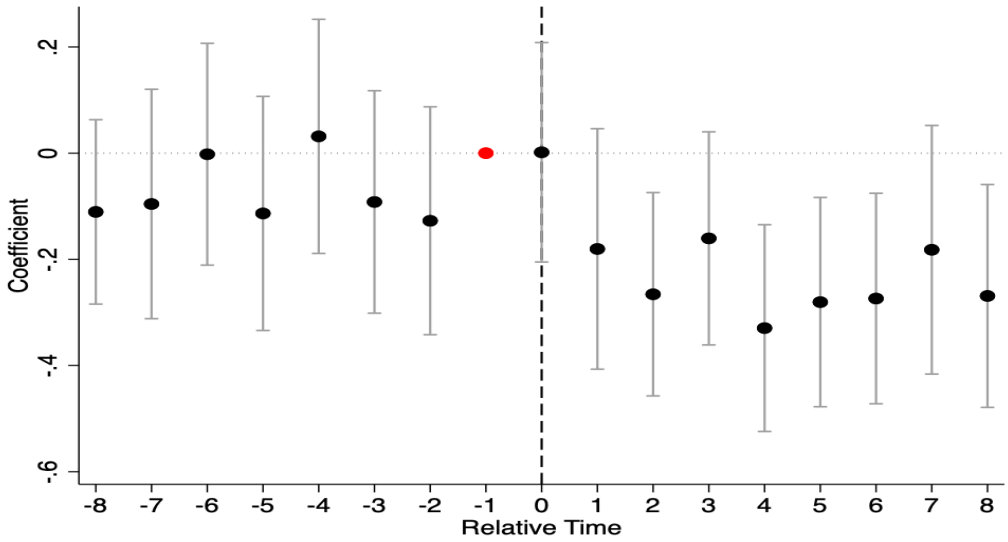
To estimate the relative time model, we utilize the full panel of data. Following common practice in event study analyses (Schmidheiny and Siegloch 2023), we merge all periods prior to event time $k = -8$ into a single bin at $k = -8$, and here $k = +8$ is the final time period. This binning approach reduces noise and ensures a more balanced panel. Importantly, our main findings remain robust without this binning: even when all relative time periods are included without aggregation, we continue to find no evidence of differential pre-trends.

The results are shown in Figures A.9 and A.10. First, the confidence intervals for the periods leading up to the launch of ChatGPT all include zero, indicating no significant pre-existing trend differences. This finding strengthens the credibility of attributing the observed

post-shock differences to the launch of LLM-based generative AI. Second, the figure reveals an increasing impact over time in the post-treatment periods, suggesting that the influence of these technologies is not a one-time event but deepens with increasing adoption.

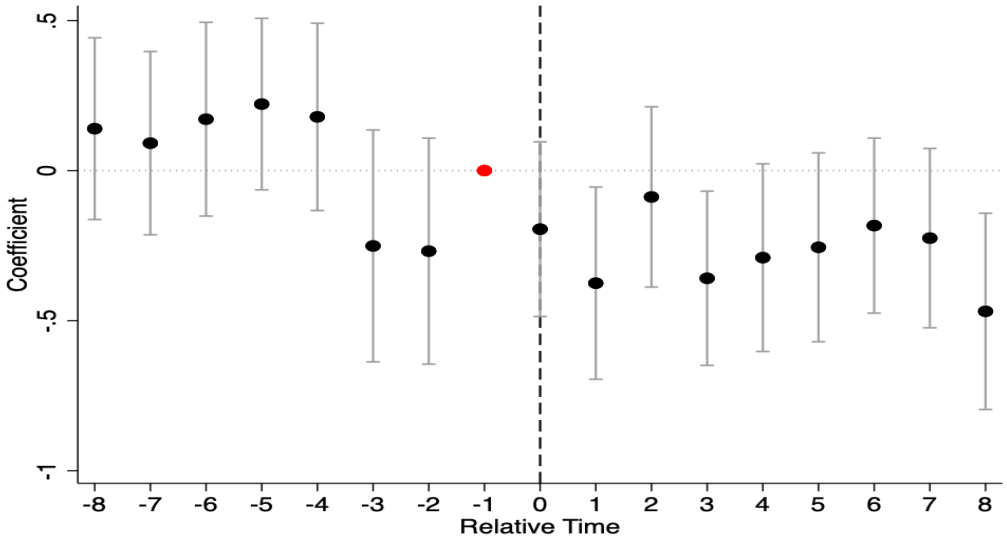


(a) Jobs



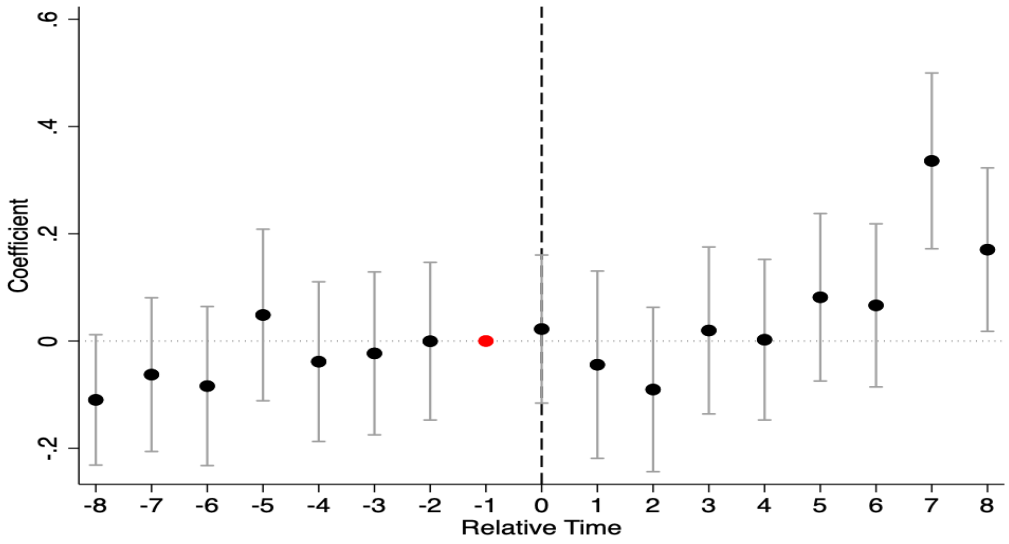
(b) Bids

Figure A.9: Pretrends and Temporal Dynamics in Demand and Supply



h

(a) Transaction



(b) Avg_Bids_job

Figure A.10: Pretrends and Temporal Dynamics in Matching Outcomes

A.17 Robustness Check - Propensity Score Matching

Although the treated and control submarkets follow similar pre-shock trends, as demonstrated in Appendix A.16, they may still differ across various dimensions such as market size prior to the treatment. To address this concern and enhance the comparability between the two sets of submarkets, we re-estimate our analysis using a propensity score matched sample of treated and control submarkets. Specifically, we estimate a probit selection model for each submarket to predict the probability of receiving treatment. To improve the comparability of the matched submarkets, we include all relevant pre-treatment variables (*Jobs*, *Clients*, *Bids*, *Bidders*, *Transaction*, *Matches*, and *Avg_Bids_job*) as covariates in the matching process. Based on the predicted propensity scores, we apply a nearest-neighbor matching procedure following standard practices in the literature (Ran et al. 2025, Ye and Shankar 2025, Faccio and O'Brien 2021) to identify the closest control submarket for each treated submarket. The resulting matched sample consists of 952 submarkets.

Next, we evaluate the quality of the matches by comparing the treated and control submarkets in terms of the covariates used in the matching procedure. Table A.19 presents the mean values of the matching variables. In contrast to the full sample results shown in columns (2)–(4), columns (5)–(7) reveal no statistically significant differences between the treated and control submarkets. This indicates that the propensity score matching procedure effectively mitigates key pre-treatment differences between the two groups. Additionally, Figures A.11 and A.12 illustrates the distribution of propensity scores before and after matching, demonstrating that the matched sample is substantially more balanced.

In Table A.20, we re-estimate our main regression model using the matched submarket sample. Following the specification outlined in Section 3.4, the coefficient on the interaction term $Treat \times Post$ (β_1) remains negative and statistically significant. These results are consistent with our baseline findings. Overall, the robustness checks suggest that our main results are not driven by insufficient matching quality.

Table A.19: Statistical Validation of Pre- and Post-Matching Balance

	Before-Matching			After-Matching		
Variables	Control	Treat	P-Value	Control	Treat	P-Value
Jobs	24.38	47.49	0.000	28.65	35.68	0.112
Clients	21.30	41.17	0.000	25.03	31.04	0.110
Bids	406.14	774.44	0.002	490.80	591.41	0.349
Bidders	234.65	443.35	0.001	281.71	363.47	0.189
Matches	3.28	6.35	0.000	3.85	4.71	0.199
Transaction	1677.13	3175.91	0.000	1995.10	2366.75	0.286
Avg_Bids_job	8.30	10.00	0.004	9.42	10.00	0.375

Table A.20: Effects of Generative AI on Labor Market Outcomes (After PSM)

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.213***	-0.159**	-0.362***	0.112**
	(0.0471)	(0.0694)	(0.1170)	(0.0443)
Constant	2.8070***	5.1620***	5.2750***	2.1290***
	(0.0089)	(0.0131)	(0.0221)	(0.0084)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	38,962	38,962	38,962	38,962
R^2	0.8710	0.7990	0.5840	0.4740

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

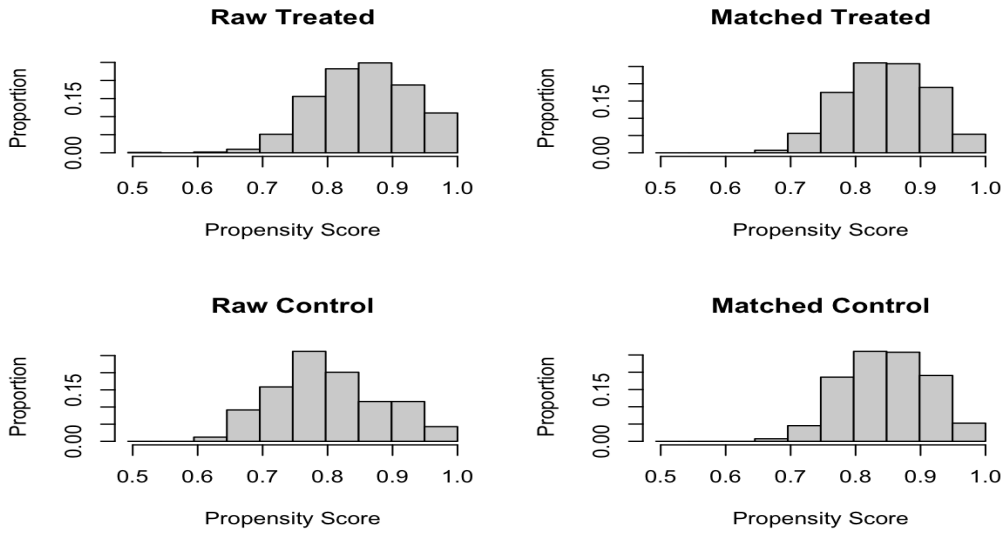


Figure A.11: Propensity Score Distribution Pre- and Post-Matching

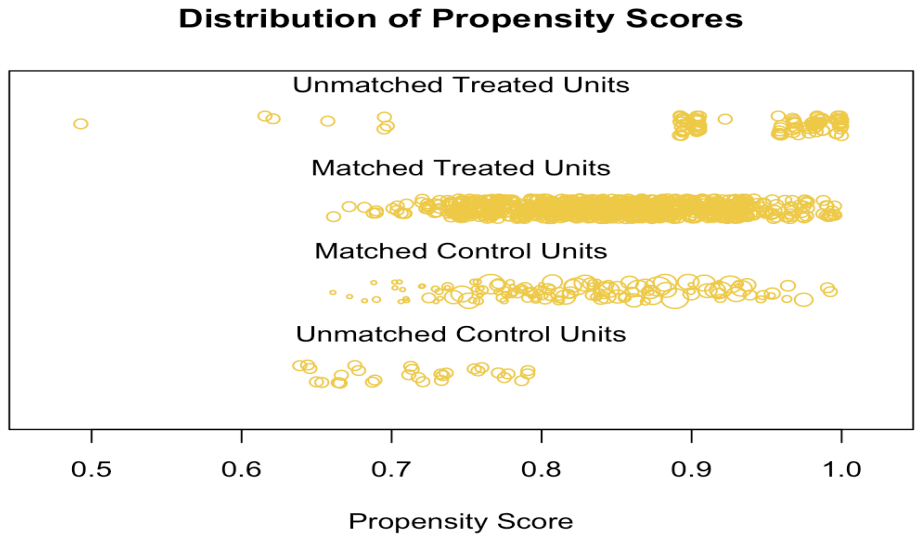


Figure A.12: Comparison Between Matched and Unmatched Units in Propensity Score Matching

A.18 Robustness Check - Interrupted Time Series (ITS) Analysis

We further validate our main results with an Interrupted Time-Series Analysis (ITSA), which is a quasi-experimental approach that projects the pre-intervention trend forward to form a counterfactual baseline, and then measures the intervention's effect as the deviation of actual post-intervention outcomes from that projected path (Kontopantelis et al. 2015, Campbell and Stanley 2015). ITSA offers an additional robustness check because it operates on a single series: treatment effects are identified from the structural break itself rather than from cross-group comparisons. Accordingly, this method is well suited to our empirical context, as it allows us to focus exclusively on the treated submarkets without having to account for pre-treatment differences between treated and control groups. By relying on within-series dynamics, ITSA also relaxes the stringent parallel-trends requirement inherent in our earlier difference-in-differences design. We estimate the following regression model using a sample restricted to treated submarkets:

$$Outcome_{it} = \beta_0 + \beta_1 * Time_t + \beta_2 * Post_t + \beta_3 * PostTime_t + u_i + \epsilon_{it} \tag{A.7}$$

Here, $Outcome_{it}$ denotes the dependent variable ($Jobs$, $Bids$, $Transaction$, or Avg_Bids_job) for submarket i in month t. $Time_t$ represents the time elapsed since the beginning of the observation period. $Post_t$ is a binary indicator equal to 0 during the pre-treatment period and 1 during the post-treatment period. $PostTime_t$ equals 0 prior to the treatment (i.e., the emergence of ChatGPT) and increases by one for each subsequent time period following the treatment. Furthermore, we incorporate u_i for the unit-level fixed effect. The error term ϵ_{it} encapsulates unobserved factors that influence the outcome. Our primary interest lies in estimating the coefficient on $Post_t$. A statistically significant coefficient indicates a level change in the outcome variable attributable to the treatment, namely, the emergence of ChatGPT. We are also interested in the post-treatment trend, captured by the coefficient on $PostTime_t$, which reflects changes in the slope of the outcome trajectory following the intervention.

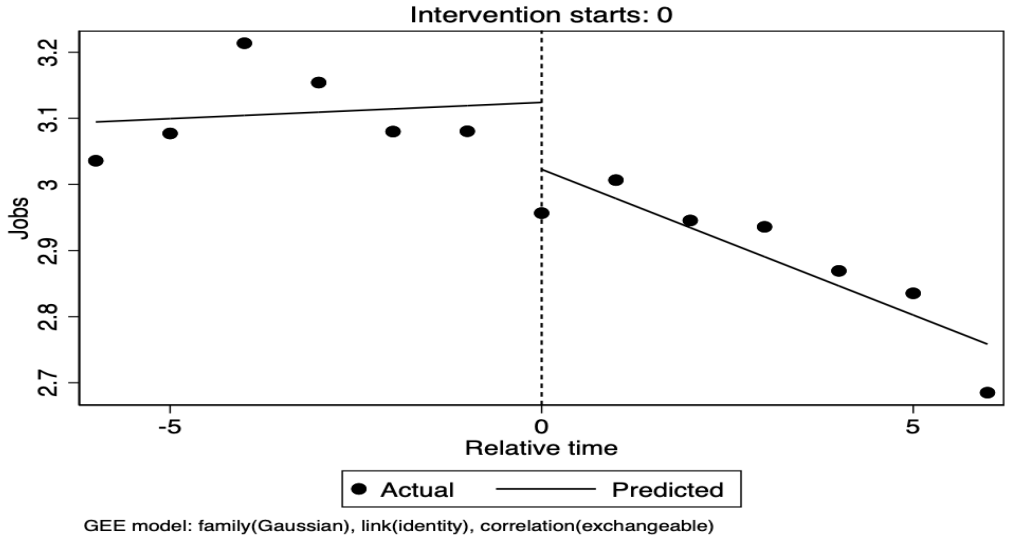
We estimate the model specified in Equation A.7 using only observations from the treated

submarkets and report the results in Table A.21. The findings are closely aligned with those from the DID estimation. Specifically, demand (measured by $Jobs$) and market transaction volume ($Transaction$) show a significant immediate decline following the treatment ($\beta_2 = -0.101$ and -0.287, respectively; $p < 0.01$), while the supply side exhibits no statistically significant immediate response ($\beta_2 = -0.0055$, $p > 0.1$). As a result, competition within the submarket intensifies, as indicated by a notable increase in the average number of bids per job (Avg_Bids_job; $\beta_2 = 0.0726$, $p < 0.01$). This shift persists into the post-treatment period, as reflected in the trend coefficients. Notably, the supply side begins to show a significant downward trend over time following the intervention ($\beta_3 = -0.0361$, $p < 0.01$), suggesting a sustained and dynamic adjustment in market dynamics. Figures A.13 and A.14 provide a more detailed and visually intuitive representation of these trends. To further validate our findings, we also implement a regression discontinuity design (RDD) to estimate the immediate effect at the shock point, the results, are consistent with those from the ITS analysis.

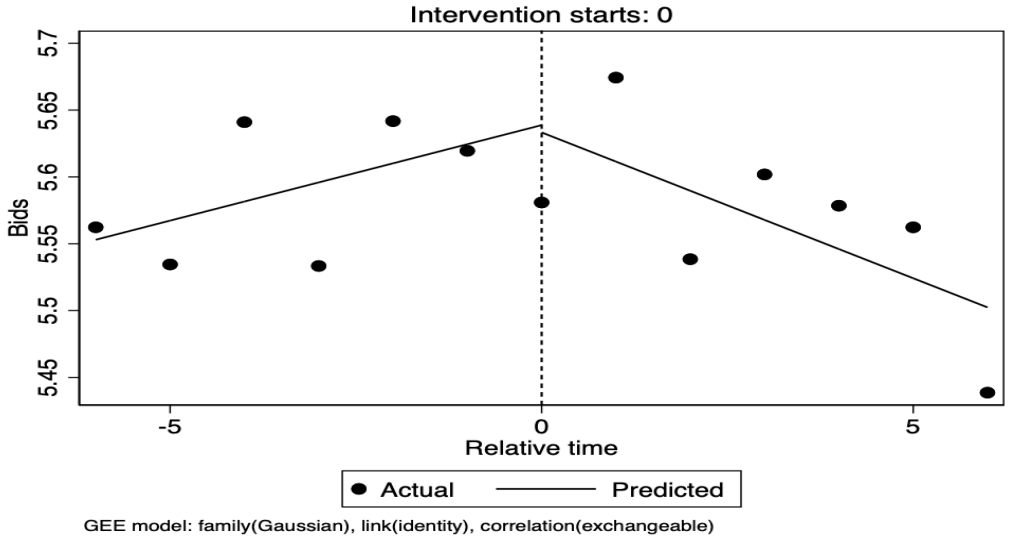
Table A.21: Results for Interrupted Time Series Analysis

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
Time	0.0049*	0.0143***	-0.0193	0.0132**
	(0.0028)	(0.0055)	(0.0144)	(0.0053)
Post	-0.1010***	-0.0055	-0.2870***	0.0726***
	(0.0128)	(0.0248)	(0.0715)	(0.0240)
PostTime	-0.0490***	-0.0361***	-0.0177	0.0456***
	(0.0040)	(0.0075)	(0.0184)	(0.0076)
Constant	3.0950***	5.5530***	5.9970***	2.0840***
	(0.0356)	(0.0496)	(0.0795)	(0.0282)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	11,921	11,921	11,921	11,921

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

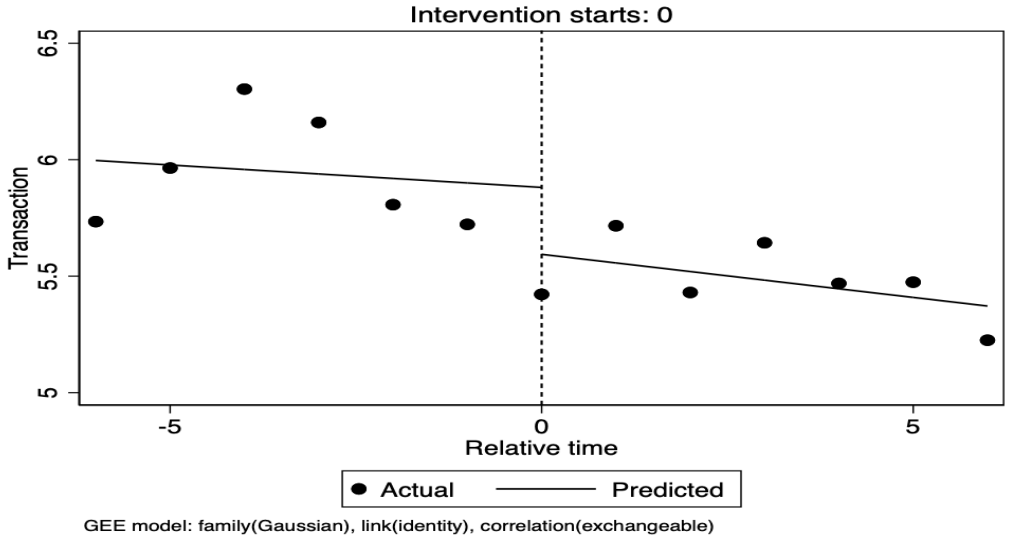


(a) Jobs

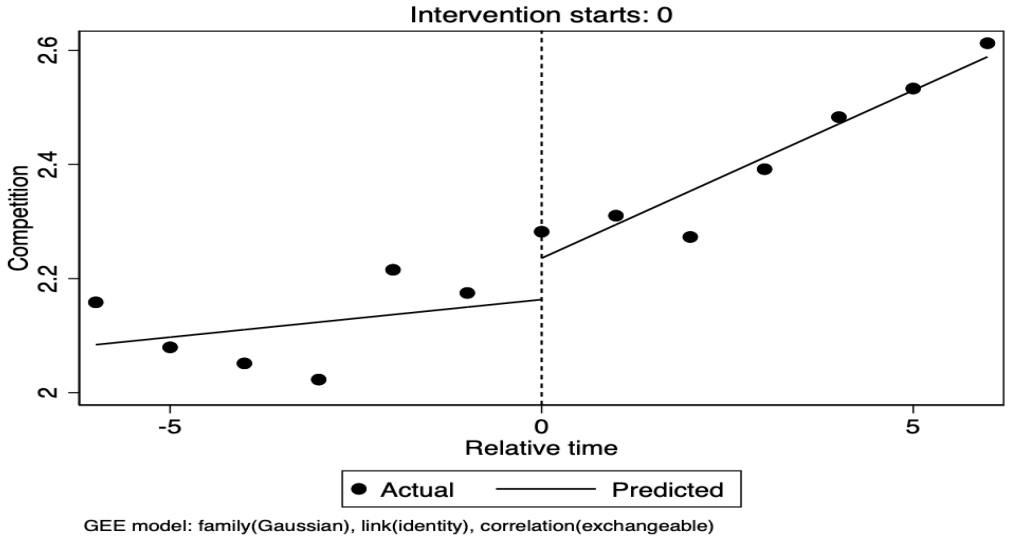


(b) Bids

Figure A.13: Interrupted Time Series (ITS) Analysis



(a) Transaction



(b) Average_job_bids

Figure A.14: Interrupted Time Series (ITS) Analysis

A.19 Robustness Check - More Details on Control Variable Inclusion and Omitted Variable Sensitivity Analyses

In this section, we incorporate control variables that capture both external conditions and internal market characteristics to obtain a conservative estimate of ChatGPT's impact and to conduct a sensitivity analysis.

Since online labor markets may be affected by external factors such as trends in the broader employment landscape, we account for these influences by collecting monthly Google Trends data for each skill, which reflects the relative popularity of each skill in a given month. Using the skill composition of each submarket, we construct a weighted monthly Google Trends index for each submarket to capture and control for fluctuations in the external environment. In addition, within the labor market, demand and supply interact dynamically. Demand in one period may influence supply in the subsequent period, while supply may, in turn, affect future demand. These two forces jointly determine the market transaction value and competition in the periods that follow. Therefore, when estimating the effect on demand ($Jobs$), we include lagged supply (Lag_Bids) as a control variable. Similarly, when estimating the effect on supply ($Bids$), we control for lagged demand (Lag_Jobs). In the estimation of transaction value and competition, both lagged demand and lagged supply (Lag_Jobs and Lag_Bids) are included as control variables to account for their joint influence. The estimation results are presented in Table A.22 and remain consistent with our main findings.

After incorporating these observable controls, the estimated coefficients β_1 change from -0.221 to -0.222 for $Jobs$, from -0.168 to -0.141 for $Bids$, from -0.315 to -0.267 for $Transaction$, and from 0.138 to 0.111 for Avg_Bids_job.

Including post-treatment covariates is likely to attenuate the estimated treatment effect. A consistent estimator should capture the total influence of ChatGPT on the outcome, that is, it should capture the cumulative effect of all underlying mechanisms. In practice, ChatGPT may first alter demand and subsequently affect labor supply and transaction volume. Competition, in particular, can rise both because the number of available jobs

falls and because changes in job mix propagate through bidding behavior before they are reflected in competitive metrics. Variables such as *Jobs* or *Bids* therefore lie on the causal pathway from the treatment to the final outcome, making them endogenous to ChatGPT adoption. Conditioning on such mediators severs part of the causal chain and introduces the classic "bad-control" problem, which mechanically biases the treatment coefficient toward zero. Hence, our reported estimates should be interpreted as conservative lower bounds; even so, they remain both economically meaningful and statistically significant.

Furthermore, to evaluate the risk of omitted variable bias, we conduct a sensitivity analysis following the approach proposed by Altonji et al. (2005), which has been widely applied in prior empirical studies (Sen et al. 2024, Petrova et al. 2023, Galor and Özak 2016). Specifically, we assess the relative sensitivity of unobserved factors by incorporating these additional control variables, thereby generating an alternative set of estimates. We then compare the new coefficient to that obtained from the regression without the additional controls, using the ratio defined as $\frac{\beta_{original}}{\beta_{original} - \beta_{control}}$. The rationale behind this ratio is straightforward: a smaller difference between the uncontrolled and controlled estimates suggests that the parameter of interest is less affected by selection on observables, thus requiring a stronger degree of selection on unobservables relative to observables to fully explain the observed effect. In contrast, a larger numerator implies that a greater portion of the effect may be driven by selection on unobservables, resulting in a higher value of the ratio. In our estimated result, the similarity of the coefficients, suggests that selection on unobservables would need to be at least 5 times (Oster 2019) stronger than selection on observables to fully account for the observed effect. This substantially alleviates concerns about omitted variable bias.

Table A.22: Effects of Generative AI on Demand, Supply, and Matching Outcomes

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2220***	-0.1410***	-0.2670***	0.1110***
	(0.0316)	(0.0469)	(0.0886)	(0.0355)
Constant	2.2340***	3.9580***	2.9720***	1.9040***
	(0.1810)	(0.3090)	(0.6510)	(0.2250)
External Confounder Controls	Yes	Yes	Yes	Yes
Internal Confounder Controls	Yes	Yes	Yes	Yes
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8920	0.8420	0.6140	0.5450

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.20 Robustness Check - Alternative Model for Count Variables

It is noteworthy that several of our variables, including *Jobs* and *Bids*, are count variables, taking on non-negative integer values. As shown in Appendix A.3, the variables exhibit substantial skewness, rendering count models not perfectly appropriate. Since the Negative Binomial model is relatively better suited for fitting such data distributions compared to the Poisson model (Stone 2013), we employ it to re-estimate the impact on these count variables. Tables A.23 present the results of this analysis. The estimation results are qualitatively aligned with our main findings, corroborating the robustness of our conclusions. Specifically, we observe a significant decrease in labor demand in programming-intensive or text-related submarkets, with no heterogeneity observed between these two types of treated submarkets. On the labor supply side, we still observe a significant decrease in text-related submarkets, while the programming-intensive submarkets incur a significantly smaller supply loss, particularly in terms of the number of bids. The estimation results are still consistent with our main findings.

Table A.23: Alternative Model for Count Variables

	Negative Binominal Regressions	
	Jobs	*Bids*
	(1)	(2)
$Treat * Post$	-0.1890***	-0.2070***
	(0.0437)	(0.0500)
$Treat * Post *$ *Programming*	0.0070	0.1430***
	(0.0332)	(0.0391)
Constant	3.7520***	5.9370***
	(0.0106)	(0.0173)
Submarket FE	Yes	Yes
Year-Month FE	Yes	Yes
Observations	25,833	25,833

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.21 Robustness Check - Alternative Aggregation Level (Week)

Leveraging the two-year dataset, our primary analyses use variables aggregated at the monthly level to smooth trends, particularly for smaller submarkets. In this section, we aggregate all variables to the weekly level and rerun the estimations. The outcomes of this analysis are presented in Table A.24. The estimation results remain qualitatively consistent with the main findings, confirming the robustness and reliability of our conclusions across different levels of data granularity. However, it is important to note that at the weekly level, there will be a substantial number of zero observations, which makes it less appropriate to directly interpret the coefficients as proportional changes.

Table A.24: Effects of Generative AI on Labor Market Outcomes (Week-level)

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.1810***	-0.1270***	-0.2940***	0.1060***
	(0.0225)	(0.0388)	(0.0520)	(0.0281)
Constant	1.7300***	3.7060***	3.1770***	1.8950***
	(0.0073)	(0.0126)	(0.0169)	(0.0091)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	112,744	112,744	112,744	112,744
R^2	0.7990	0.7250	0.4850	0.3740

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.22 Robustness Check - Alternative Fixed Effect

To more effectively account for seasonal effects, we also adopt an alternative specification that includes separate fixed effects for year (Y_t) and month (M_t), allowing us to control for both annual variations and recurring monthly patterns. This approach enhances interpretability by decomposing temporal variation into distinct components. The definitions and constructions of the remaining variables remain consistent with those outlined in Section 3.4. The model specification is presented as follows:

$$Outcome_{it} = \beta_0 + \beta_1 * Treat_i * Post_t + \beta_2 * Post_t + u_i + Y_t + M_t + \epsilon_{it} \tag{A.8}$$

Using this model specification, we re-estimate the market-level outcomes, with the results reported in Table A.25. The findings suggest that the alternative fixed effects structure yields results that are broadly consistent with those of the main specification.

Table A.25: Effects of Generative AI on Demand, Supply, and Matching Outcomes (Alternative Fixed Effects)

	$ln(y+1)$			
	$Jobs$	$Bids$	$Transaction$	Avg_Bids_job
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2200***	-0.1680***	-0.3140***	0.1370***
	(0.0318)	(0.0476)	(0.0896)	(0.0356)
$Post$	0.1470***	0.2110***	-0.1080	0.0343
	(0.0327)	(0.0557)	(0.1210)	(0.0467)
Constant	2.9640***	5.3270***	5.6650***	2.0750***
	(0.0065)	(0.0129)	(0.0358)	(0.0120)
Submarket FE	Yes	Yes	Yes	Yes
Year FE	Yes	Yes	Yes	Yes
Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8900	0.8400	0.6100	0.5380

Note: Cluster-robust standard errors are reported at the freelancer level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.23 Robustness Check - Alternative Outcomes

In Section 3.5, we have analyzed the average post-ChatGPT changes in labor demand, supply, and matching outcomes for treated submarkets relative to control submarkets, with a primary focus on changes in the overall volume of demand, supply, and transactions. To further substantiate these findings, we introduce three additional count variables corresponding to the demand side, supply side, and matching outcomes. These variables serve to both complement and validate the aggregate analysis by capturing changes in the number of platform participants and successfully matched jobs. Similar to other market-level variables used in our analysis, these are constructed at the submarket-month level to ensure consistency in temporal and unit resolution across all outcome measures. Specifically, on the demand side, we introduce the variable $Clients$, denoting the number of unique clients who post jobs within the submarket in a given month. On the supply side, we adopt the variable $Freelancers$, representing the number of distinct freelancers who have placed at least one bid within a particular submarket during a month. To complement the analysis of market matching outcomes, we construct the variable $Matches$, epitomizing the number of successful matches made (i.e., successfully completed jobs) within a submarket in a given month.

Following the same procedure as in the main analysis presented in Section 3.5, we estimate the results using Equation 3.1, as specified in Section 3.4. The findings further confirm the observed contractions in demand, supply, and matching outcomes in the online labor market. Specifically, The results indicate a marginal decrease in the number of participating clients ($\beta_1 = -0.1990$, $p < 0.01$), active freelancers ($\beta_1 = -0.1410$, $p < 0.01$), and successful matches ($\beta_1 = -0.1020$, $p < 0.01$) in the treatment submarkets relative to the control submarkets following the launch of ChatGPT. These additional analyses provide further support for the validity of our findings.

Summary statistics for these additional variables are reported in Table A.26. The corresponding estimation results, based on our empirical specification, are presented in Tables A.27 and A.28. These results provide additional support for the robustness of our findings.

Furthermore, the parallel trends assumption is supported for these variables, as illustrated in Figure A.15.

Table A.26: Variable Summary Statistics

Variable	Observation	Mean	Std	Min	Max
$ln(Clients + 1)$	25,833	2.8488	1.2107	0	6.8977
$ln(Freelancers + 1)$	25,833	5.0859	1.6808	0	10.3477
$ln(Matches + 1)$	25,833	1.2865	1.0445	0	5.2781

Table A.27: Effects of Generative AI on Other Market Outcomes

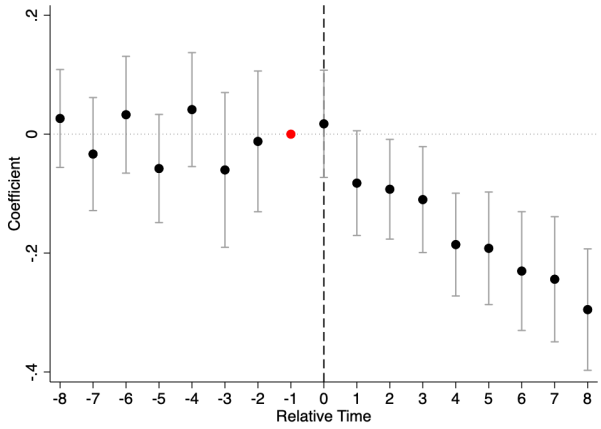
	Clients	*Freelancers*	*Matches*
	(1)	(2)	(3)
$Treat * Post$	-0.1990***	-0.1410***	-0.1020***
	(0.0292)	(0.0456)	(0.0252)
Constant	2.9120***	5.1310***	1.3190***
	(0.0093)	(0.0146)	(0.0081)
Submarket FE	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes
Observations	25,833	25,833	25,833
R^2	0.8990	0.8200	0.7860

Robust standard errors in parentheses *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

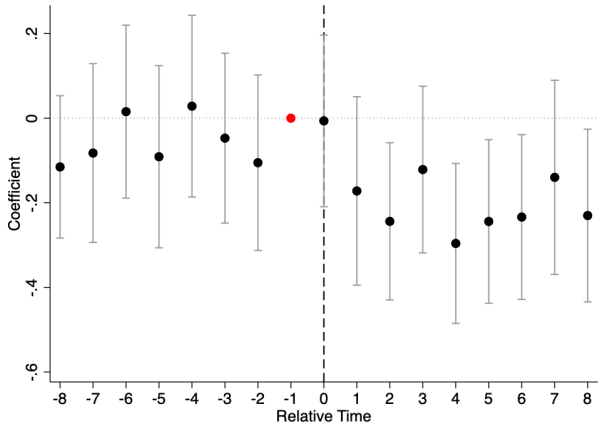
Table A.28: Effects of Generative AI on Other Market Outcomes

	Clients	*Freelancers*	*Matches*
	(1)	(2)	(3)
$Treat * Post$	-0.2070***	-0.1940***	-0.1010***
	(0.0317)	(0.0495)	(0.0278)
$Treat * Post *$ *Programming*	0.0183	0.1170***	-0.0007
	(0.0210)	(0.0325)	(0.0216)
Constant	2.9120***	5.1310***	1.3190***
	(0.0093)	(0.0146)	(0.0081)
Submarket FE	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes
Observations	25,833	25,833	25,833
R^2	0.8990	0.8200	0.7860

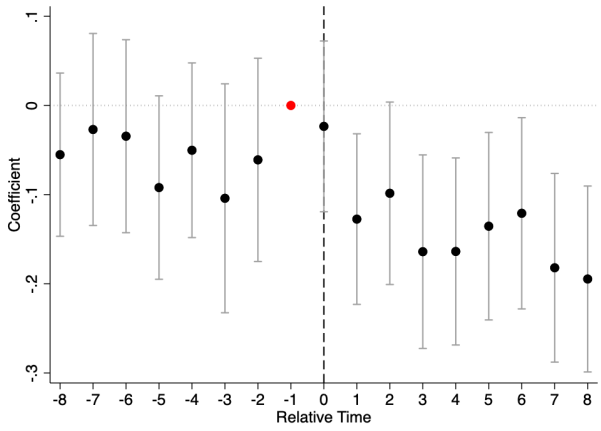
Robust standard errors in parentheses *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$



(a) Clients



(b) Freelancers



(c) Matches

Figure A.15: Pretrends and Temporal Dynamics

A.24 Robustness Check - Removing Outliers

Given that submarkets can vary significantly in size, one might question whether the estimates are disproportionately affected by the largest and the smallest submarkets. To address this concern, we first exclude the top 1% of submarkets, followed by the removal of submarkets with an average job post count of less than one. The re-estimated model results, as presented in Table A.29 and A.30, remain consistent with our main findings. This indicates that our conclusions are not driven by a few large submarkets.

Table A.29: The Impact of Generative AI after Excluding Outliers (Drop Top Submarkets)

	$ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2220***	-0.1730***	-0.3180***	0.1350***
	(0.0319)	(0.0479)	(0.0902)	(0.0358)
Constant	2.9890***	5.3710***	5.5800***	2.0860***
	(0.0102)	(0.0153)	(0.0288)	(0.0114)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,569	25,569	25,569	25,569
R^2	0.8840	0.8350	0.6030	0.5400

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.30: The Impact of Generative AI after Excluding Outliers (Drop Bottom Submarkets)

	$ln(y+1)$			
	$Jobs$	$Bids$	$Transaction$	Avg_Bids_job
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2190***	-0.1600***	-0.2990***	0.1370***
	(0.0339)	(0.0480)	(0.0960)	(0.0348)
Constant	3.0740***	5.4950***	5.7270***	2.1150***
	(0.0110)	(0.0155)	(0.0310)	(0.0112)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,306	25,306	25,306	25,306
R^2	0.8820	0.8290	0.5890	0.5400

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.25 Robustness Checks - Continues Treatment

To further verify the robustness of our main results, we re-estimate the main models using each sub-market's LM-AIOE score as a continuous treatment measure, replacing the median-split binary classification. Specifically, we estimate the following specification:

$$Outcome_{it} = \beta_0 + \beta_1 * LM\_AIOE_i * Post_t + u_i + T_t + \epsilon_{it} \tag{A.9}$$

where LM_AIOE_i denotes the continuous LM-AIOE score of submarket i, capturing its degree of exposure to LLM-based technologies, and all other variables are defined the same as before. The coefficient β_1 captures the differential effect of ChatGPT across submarkets with varying levels of LM-AIOE exposure. The estimation results are reported in Tables A.31 and A.32. Across both outcomes, the estimates remain consistent with our baseline findings in terms of direction and statistical significance, further supporting the robustness of our main conclusions.

Table A.31: The Impact of Generative AI at the Market Level (Continues Treatment)

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
LM_AIOE * *Post*	-0.2920***	-0.2210***	-0.5250***	0.1870***
	(0.0357)	(0.0535)	(0.1140)	(0.0410)
Constant	3.0300***	5.4140***	5.6710***	2.0800***
	(0.0100)	(0.0150)	(0.0318)	(0.0114)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5410

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.32: The Heterogeneous Impact of Generative AI at the Market Level (Continues Treatment)

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$LM\_AIOE * Post$	-0.2960***	-0.2640***	-0.5360***	0.1460***
	(0.0363)	(0.0540)	(0.1160)	(0.0419)
$LM\_AIOE * Post *$ *Programming*	0.0144	0.1590***	0.0413	0.1530***
	(0.0266)	(0.0413)	(0.0798)	(0.0316)
Constant	3.0300***	5.4070***	5.6690***	2.0730***
	(0.0100)	(0.0151)	(0.0320)	(0.0114)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5420

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.26 Robustness Checks - Multi-group Analysis

To further assess the robustness of our findings to the choice of treatment categorization, we replace the median-split binary classification with a tercile-based approach, dividing submarkets into three groups (high, medium, and low exposure) based on their LM-AIOE values. We use the low-exposure group as the baseline and estimate separate treatment effects for the high- and medium-exposure groups. Tables A.33 and A.34 report the results for the main analysis and heterogeneity analyses, respectively.

As shown in Table A.33, the results remain consistent with our baseline findings: relative to the low-exposure group, both the high- and medium-exposure groups experience significant declines in jobs, bids, and transactions following ChatGPT's introduction, with stronger effects for the high-exposure group. Results in Table A.34 also show a broadly similar heterogeneity pattern with main analysis. Together, these results further support the robustness of our main findings.

Table A.33: The Impact of Generative AI at the Market Level (Three Group)

Panel A				
	High vs Low			
	ln(y+1)			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2580***	-0.2010***	-0.4210***	0.1670***
	(0.0345)	(0.0518)	(0.0982)	(0.0387)
Constant	3.0070***	5.3820***	5.5340***	2.0460***
	(0.0096)	(0.0144)	(0.0272)	(0.0107)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	14,855	14,855	14,855	14,855
R^2	0.8860	0.8370	0.6140	0.5260
Panel B				
	Medium vs Low			
	ln(y+1)			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.1830***	-0.1350***	-0.2090**	0.1090***
	(0.0331)	(0.0499)	(0.0934)	(0.0378)
Constant	2.8250***	5.0880***	5.2990***	2.0550***
	(0.0092)	(0.0139)	(0.0260)	(0.0105)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	14,850	14,850	14,850	14,850
R^2	0.8940	0.8450	0.6310	0.5430

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.34: The Heterogeneous Impact of Generative AI at the Market Level (Three Group)

Panel A				
	High vs Low			
	ln(y+1)			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
Treat ∗ *Post*	-0.2820***	-0.2890***	-0.4610***	0.0982**
	(0.0378)	(0.0567)	(0.1070)	(0.0422)
Treat ∗ *Post* ∗ *Programming*	0.0649*	0.2340***	0.1070	0.1850***
	(0.0348)	(0.0530)	(0.1070)	(0.0395)
Constant	3.0070***	5.3820***	5.5340***	2.0460***
	(0.0096)	(0.0143)	(0.0272)	(0.0107)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	14,855	14,855	14,855	14,855
R^2	0.8860	0.8370	0.6140	0.5270
Panel B				
	Medium vs Low			
	ln(y+1)			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
Treat ∗ *Post*	-0.1690***	-0.1690***	-0.1860*	0.0667
	(0.0393)	(0.0604)	(0.1080)	(0.0450)
Treat ∗ *Post* ∗ *Programming*	-0.0251	0.0624	-0.0426	0.0776**
	(0.0305)	(0.0483)	(0.0871)	(0.0379)
Constant	2.8250***	5.0880***	5.2990***	2.0550***
	(0.0092)	(0.0139)	(0.0260)	(0.0105)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	14,850	14,850	14,850	14,850
R^2	0.8940	0.8450	0.6310	0.5430

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.27 Robustness Checks - Alternative Measures of Gen AI Exposure

To further examine the robustness of our results, we adopt an alternative peer-reviewed measure of occupational AI exposure from Eloundou et al. (2024), who assess the labor market impact potential of large language models across occupations. We reconstruct the treatment classification and re-estimate the models. Tables A.35 and A.36 present the corresponding estimation results. The findings remain largely consistent with the main analysis, further supporting the robustness of our conclusions.

Table A.35: The Impact of Generative AI at the Market Level (Alternative Measures of Gen AI Exposure)

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2330***	-0.1170**	-0.3390***	0.2190***
	(0.0317)	(0.0508)	(0.0890)	(0.0354)
Constant	3.0200***	5.3880***	5.6290***	2.0650***
	(0.0098)	(0.0157)	(0.0274)	(0.0109)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5420

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table A.36: The Heterogeneous Impact of Generative AI at the Market Level (Alternative Measures of Gen AI Exposure)

	$\ln(y+1)$			
	Jobs	*Bids*	*Transaction*	*Avg_Bids_job*
	(1)	(2)	(3)	(4)
$Treat * Post$	-0.2460***	-0.1710***	-0.3670***	0.1830***
	(0.0344)	(0.0548)	(0.0958)	(0.0382)
$Treat * Post *$ *Programming*	0.0297	0.1180***	0.0607	0.0765***
	(0.0221)	(0.0338)	(0.0670)	(0.0266)
Constant	3.0200***	5.3880***	5.6290***	2.0650***
	(0.0098)	(0.0156)	(0.0274)	(0.0109)
Submarket FE	Yes	Yes	Yes	Yes
Year-Month FE	Yes	Yes	Yes	Yes
Observations	25,833	25,833	25,833	25,833
R^2	0.8910	0.8410	0.6120	0.5420

Note: Cluster-robust standard errors are reported at the submarket level. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Appendix B

APPENDIX TO ALGORITHMIC COLLUSION OR COMPETITION: THE ROLE OF PLATFORMS' RECOMMENDER SYSTEMS

B.1 Summary of Notations

Table B.1: Summary of Notations for the Main Model (Part 1)

Notation	Explanation
t	Time period t
J	The number of products available on the market
K	The number of products shown to each consumer
j	A product j (with $j = 0$ representing the outside option)
p_j^t	The price of product j at period t
$\boldsymbol{X_j}$	The other attributes of product j which are stable across periods
D_j^t	Demand of product j at period t
π_j^t	Revenue of product j at period t
i	A Consumer i (the index is reordered every period to simplify the notation)
$\boldsymbol{A_i}$	The consumer i's attributes
$\boldsymbol{B_i}$	A set of products shown to the consumer i

Table B.2: Summary of Notations for the Main Model (Part 2)

Notation	Explanation
β_i	The consumer's price sensitivity
$\mathbf{\Gamma}$	Other parameters linking product' attributes to utility
$\mathbf{\Theta}$	Other preference parameters linking consumer's attributes to utility
$pos_{i,j}$	Ranking (Position) of product j seen by consumer i
$c_{i,j}$	The search cost of product j for consumer i
τ_i	The constant term of consumer i's search cost
φ	The ranking effect parameter of the search cost
$u_{i,j}$	The consumption utility of product j for consumer i
$v_{i,j}$	The expected pre-search utility of product j for consumer i
$\xi_{i,j}$	The post-search utility shock of product j for consumer i
$z_{i,j}$	The reservation utility of product j for consumer i
μ	The horizontal differentiation level on the market
S_t	State of the system at the period t, which is the price vector at $t-1$
Q_j	Product j's pricing strategy (characterized by a Q matrix in Q-learning)
TU^t	Total Consumer Utility at period t
TD^t	Total Demand at period t
Π^t	Total Revenue at period t
r_j^t	Product j's reward at period t
m^t	Platform's reward at period t
α	Learning rate of the pricing algorithm
δ	Discount rate of the pricing algorithm
ϵ	Exploration rate of the pricing algorithm
ω	Decay rate of exploration rate

B.2 Repeated Game Framework Pseudocode

Algorithm 2 General Repeated Game Process

Require: Initial price p_1^0,..., p_J^0; product attribute $\boldsymbol{X_1}$,...,$\boldsymbol{X_J}$; initial pricing strategies Q_1,..., Q_J; consumer attribute $\boldsymbol{A_1^t}$,...,$\boldsymbol{A_N^t}$; consumer decision model CM; initial recommendation strategy RS.

while $t \leq t_{max}$ **do**
  for each seller j: **do**
    p_j^t is made based on Q_j.
  end for
  for each consumer i: **do**
    Consumer i sees a list of products with ranking $\boldsymbol{B_i}$ which is determined based on p_1^0,..., p_J^0, $\boldsymbol{X_1}$,...,$\boldsymbol{X_J}$, and $\boldsymbol{A_i^t}$ by RS^t.
    Consumer i makes the search and purchase decisions following the decision model M.
  end for
  for each seller j: **do**
    Demand D_j^t and Profits R_j^t are realized.
  end for
  Total utility TU, total demand TD^t, and total Revenue Π^t on the platform are realized.
  Step 3: Algorithm Update
  for each seller j: **do**
    Pricing strategy Q_j is updated based on the seller's objective r_j^t.
  end for
  Platform also realizes the reward m^t.
  $t \leftarrow t + 1$.
end while

B.3 Summary of Assumption Relaxation and Robustness Checks

With respect to the demand model, most studies on algorithmic collusion follow Calvano et al. (2020b) by employing a reduced-form model in which parameters are directly specified to generate demand responses. In contrast, our main analyses seek to more accurately capture consumer behavior in the presence of recommended products and to systematically characterize the underlying utility functions. To this end, we develop a sequential search model that incorporates heterogeneous consumer preferences and search costs, as detailed in Section 4.2.2 and Appendix B.4. We estimate the model parameters using the procedure outlined in Section 4.3 and a real-world dataset described in Appendix B.11, enabling us to conduct counterfactual simulations and derive empirically grounded insights. Further distinguishing our approach, Calvano et al. (2020b) assume that all products are fully visible to consumers, providing them with complete information. In contrast, and to more closely reflect the information overload often encountered in online markets, our model restricts the consumer choice set primarily to recommended products and introduces search costs for additional information. We also incorporate position bias into the sequential search model, allowing for differential consideration of products based on their placement within the recommendation set. Although this sequential search model offers several advantages, we additionally employ a canonical logit demand model for robustness checks. As shown in Appendices B.9 and B.10, the core insights remain robust even when using a logit demand model with or without heterogeneity in the outside option.

For the pricing algorithm, we follow Calvano et al. (2020b) and implement a basic Q-learning approach. Q-learning is chosen due to its prominence in reinforcement learning, its minimal input requirements, and its nonparametric simplicity. However, Q-learning inherently embodies a forward-looking component through its state transitions. Therefore, we conduct a robustness check by relaxing the forward-looking assumption and modeling the algorithms as basic multi-armed bandits. As demonstrated in Appendix B.14, this modification does not fundamentally alter our results, indicating that forward-looking behavior is

not essential for the core mechanism underlying our findings. Moreover, we also examine alternative seller objectives in Appendix B.15 to explore the broader applicability of our findings.

Regarding the sequence of the game, Calvano et al. (2020b) assume that all sellers make their decisions simultaneously. While we retain this assumption in our primary analyses, we also relax it by allowing sellers to update their prices asynchronously. As shown in Appendix B.16, the core results remain stable under both simultaneous and asynchronous updating protocols, reinforcing the generality of our conclusions.

Finally, our framework introduces a key innovation by explicitly incorporating recommender systems. Beyond addressing the technical challenges of developing an algorithm that is both empirically relevant and theoretically tractable, this addition enables us to examine objective misalignments among various stakeholders. In the main analyses, we illustrate the impact of differing objectives by considering two polar cases. To further enhance realism, we also examine scenarios in which the platform optimizes a weighted combination of revenue and utility, with results reported in Appendix B.18. Additionally, although consumer utility is closely related to purchase probability and demand, making demand-maximization theoretically similar to utility-maximization, we nevertheless explicitly test this scenario in Appendix B.17. In this analysis, we report outcomes for sellers who optimize either demand or revenue under a demand-maximizing recommender system. Collectively, these results underscore the influential role of recommender systems in shaping product exposure, influencing seller payoffs, and guiding the evolution of the game.

B.4 Consumer Search and Purchase Process

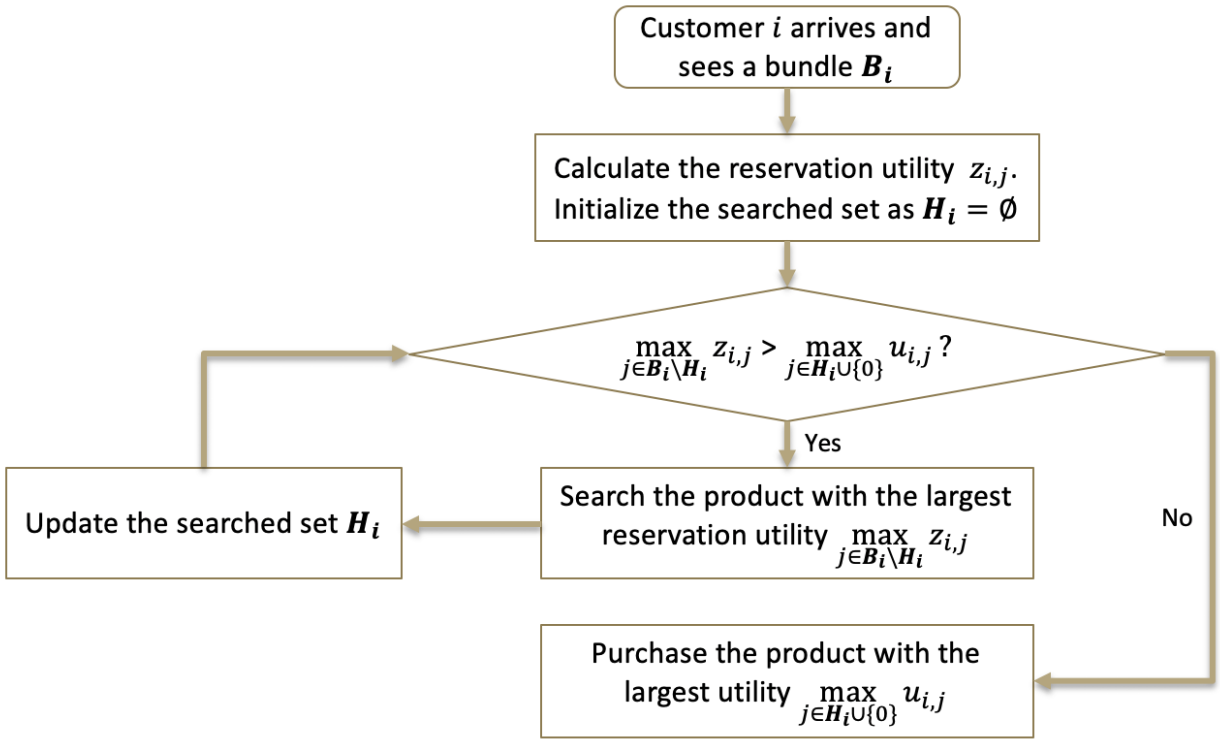


Figure B.1: Consumer Search and Purchase Process

B.5 Estimation Details

As researchers, we observe some covariates ($\boldsymbol{A_i}$, $\boldsymbol{X_j}$), consumers' search sequences $\boldsymbol{H_i} = \{\boldsymbol{H_i}(0), \ldots, \boldsymbol{H_i}(h)\}$, and their final purchase decisions k. However, the utility parameters (β_i, $\boldsymbol{\Gamma}$, $\boldsymbol{\Theta}$) and search cost parameters (τ_i, φ) are unobserved and need to be estimated from the data. Following the sequential search literature, we also do not observe the exact values of the post-search shocks $\xi_{i,j}$ but assume they are drawn from the standard normal distribution $N(0,1)$ (Ursu 2018, Weitzman 1979)[1]. From the consumers' decision rules, we can infer the following mathematical relationships:

1. A product searched later has a reservation utility higher than the utilities of products searched earlier:
$$z_{i,\boldsymbol{H_i}(q)} \geq \max_{j=0}^{q-1} u_{i,\boldsymbol{H_i}(j)}, \quad \forall q \in \{1, \ldots, h\}. \tag{B.1}$$

2. The reservation utilities of all unsearched products are lower than the maximum utility among the searched ones:
$$\max_{q \in \boldsymbol{B_i} \setminus \boldsymbol{H_i}} z_{i,\boldsymbol{H_i}(q)} \leq \max_{j \in \boldsymbol{H_i}} u_{i,j}. \tag{B.2}$$

3. The purchased product has the highest utility among all searched products:
$$u_{i,k} \geq \max_{j \in \boldsymbol{H_i}} u_{i,j}. \tag{B.3}$$

By integrating these relationships, the probability of consumers' search sequences and purchase decisions that align with the observed data can be expressed as:
$$\begin{aligned} P_{i,k,\boldsymbol{H_i}} = \Pr & \left(z_{i,\boldsymbol{H_i}(q)} \geq \max_{j=0}^{q-1} u_{i,\boldsymbol{H_i}(j)}, \quad \forall q \in \{1, \ldots, h\} \right) \cap \\ & \left(\max_{q \in \boldsymbol{B_i} \setminus \boldsymbol{H_i}} z_{i,\boldsymbol{H_i}(q)} \leq \max_{j \in \boldsymbol{H_i}} u_{i,j} \right) \cap \\ & \left(u_{i,k} \geq \max_{j \in \boldsymbol{H_i}} u_{i,j} \right) \end{aligned} \tag{B.4}$$

[1]This is a standard assumption in the sequential search literature because, in any utility maximization framework, only differences among choices matter (Ursu et al. 2023). Consequently, only the *relative* magnitudes of the search cost and the post-search variance are crucial in balancing the costs and benefits of conducting searches. This normalization allows us to fix one part while estimating the other.

To estimate the model, we address two primary challenges. First, structural search models typically lack closed-form expressions that directly relate parameters to choice probabilities. Consequently, consistent with prior literature, we employ simulated maximum likelihood (SML) estimation using a logit-smoothed accept-reject (AR) simulator (Ursu et al. 2023). Second, to enhance the model's flexibility in analyzing the pricing game under the influence of recommendations, it is essential to account for unobservable heterogeneities in price coefficients and search costs. Specifically, we assume that the price coefficient and search cost are drawn from log-normal distributions (see Equations B.5 and B.6):

$$\beta_i \sim \exp(m_\beta + \sigma_\beta \cdot y_{\beta,i}) \tag{B.5}$$

$$c_{i,j} \sim \exp(m_\tau + \sigma_\tau \cdot y_{\tau,i} + \varphi \cdot pos_{i,j}) \tag{B.6}$$

where $y_{\beta,i} \sim N(0,1)$ and $y_{\tau,i} \sim N(0,1)$.

Although SML is commonly utilized for econometric models with random coefficients (Train 2009), incorporating an additional layer of simulation to capture unobservable heterogeneities substantially increases the computational burden. To mitigate this, we innovatively integrate Gaussian Hermite quadrature into the estimation procedure (Akşin et al. 2013). To the best of our knowledge, this represents the first application of the Hermite approach to estimate a structural search model incorporating unobservable heterogeneities.

Table B.3: Structural Estimation Results

Variable	**Coefficients**	**Std**
m_β	-1.0384***	(0.0047)
σ_β	0.0067***	(0.0025)
$\Gamma_{HotelStars}$	0.4955***	(0.0197)
$\Gamma_{HotelReview}$	-0.1897***	(0.0387)
Γ_{Brand}	0.0030	(0.0235)
$\Gamma_{HotelLocationScore}$	-0.1851***	(0.0469)
Γ_{Promo}	0.0408*	(0.0247)
$\Theta_{ReqRoomNum}$	-0.1315***	(0.0197)
$\Theta_{SatNight}$	0.1045***	(0.0168)
$v_{i,0}$	0.1653	(0.3173)
m_τ	-1.1490***	(0.0172)
σ_τ	0.0050*	(0.0029)
φ	0.0050***	(0.0009)

Note: Bootstrap standard errors in parentheses; *** p<0.01, ** p<0.05, * p<0.1.

B.6 Pricing Algorithms' Details

In modeling the pricing algorithm, we employ the canonical Q-learning for investigating pricing games, consistent with prior research on algorithmic collusion (Dou et al. 2024, Wang et al. 2023b, Colliard et al. 2022, Banchio and Skrzypacz 2022). The selection of the classical Q-learning approach is motivated by several compelling factors (Calvano et al. 2020b). First, Q-learning is extensively utilized in practice and serves as the foundation for numerous state-of-the-art reinforcement learning algorithms, including DQN (Mnih et al. 2015), Double DQN (Van Hasselt et al. 2016), Double-Dueling DQN (Wang et al. 2023c), and many advanced variants (Silver et al. 2018, 2016). Consequently, studying Q-learning provides insights into a broad spectrum of algorithmic pricing models that build upon its adaptations. Second, Q-learning models the reward structure non-parametrically and requires only a minimal number of parameters. Its simplicity, relative to neural network-based algorithms, requires fewer assumptions and facilitates a clearer interpretation of the parameters' impact on game dynamics. Third, Q-learning's capacity to operate without prior knowledge of the economic environment aligns with real-world scenarios where sellers possess limited market information, thereby enhancing the practical applicability of our pricing model.

We now provide a detailed description of our Q-learning algorithm and introduce the relevant notation for our framework. Building on the foundational work on Q-learning, we define the agent's state, action, immediate reward, and objective to determine its optimal strategy within the framework of Markov decision processes (Watkins 1989). In each discrete period t, the pricing agent j (i.e., seller j on the platform) observes the current state variable s^t and selects a pricing action p_j^t from a set of feasible actions $\boldsymbol{P}$. Although the pricing agent has access to historical pricing data, to streamline calculations and maintain consistency with prior studies, we assume a one-period memory (Calvano et al. 2020b), resulting in the state being represented by the prices of all products in the previous period: $s^t = (p_1^{t-1}, \ldots, p_J^{t-1})$. Subsequently, the pricing agent receives a reward r_j^t from the environment and observes the complete set of prices transitioning to the next period, denoted by $s^{t+1} = (p_1^t, \ldots, p_J^t)$. In

the context of our pricing game, the reward corresponds to the seller's revenue generated by the demand model. The primary objective of the pricing agent j is to maximize the total discounted cumulative reward (i.e., revenue for seller j), expressed as $\mathbb{E}\left[\sum_{t=0}^{\infty}\delta^t r_j^t\right]$, where δ represents the discount factor.

Since each agent's action influences not only the immediate reward but also the subsequent state transition, it is imperative for the agent to adopt a forward-looking perspective that accounts for the value of each state s[2]. This state-specific value can be expressed as the maximum expected value obtained by summing the immediate reward from an action and the anticipated value of the ensuing state s' resulting from action p[3]. This concept is traditionally encapsulated by Bellman's value function:

$$V(s) = \max_{p\in\boldsymbol{P}}\left\{E\left[r_i \mid s,p\right] + \delta E\left[V\left(s'\right) \mid s,p\right]\right\} \tag{B.7}$$

In line with the prevalent methodology for addressing this optimization problem, we adopt the Q-function, which encapsulates the expected reward of taking a particular action in a given state, thereby circumventing the direct solution of Bellman's equations (Li 2023, Watkins 1989). The Q-function is represented as a matrix with dimensions $|\boldsymbol{S}| \times |\boldsymbol{P}|$, and each element is computed according to the following procedure:

$$Q_j(s,p) = E\left[r_j \mid s,p\right] + \delta E\left[\max_{p'\in\boldsymbol{P}} Q_j\left(s',p'\right) \mid s,p\right] \tag{B.8}$$

If the true Q-function were known, the agent could readily determine the optimal action to take in the current state. Therefore, the essence of Q-learning lies in approximating the Q-function by exploring various actions while minimizing the loss of rewards due to suboptimal choices. To estimate the Q matrix Q_j, at each period t, agent j selects an action p_j^t based on the current state s^t, observes the resulting state transition s^{t+1}, receives the corresponding reward r_j^t, and updates the Q-value for this state-action pair as follows:

$$Q_j(s=s^t, p=p_j^t) = (1-\alpha)Q_j(s,p) + \alpha\left[r_j^t + \delta\max_{p'\in\boldsymbol{P}} Q_j\left(s^{t+1},p'\right)\right] \tag{B.9}$$

[2]This assumption will be relaxed in Section 4.4.4.

[3]For notational simplicity, variables pertaining to the current period are denoted without a prime, while variables representing the subsequent period are indicated with a prime.

For all other state-action pairs where $s \neq s^t$ or $p \neq p_j^t$, the Q-values remain unchanged. Here, we introduce a hyperparameter α, representing the learning rate. This parameter determines the extent to which the updated Q-value depends on newly acquired information during the current period relative to the previous Q-value.

Beginning with minimal information and an arbitrary Q matrix, the agent must learn the value associated with each state-action pair. Given the uncertainty in the environment, particularly the unknown actions of other sellers, each element in the Q table may require numerous iterations to accurately estimate its expected value. However, uniformly exploring all actions can lead to the opportunity cost of selecting sub-optimal actions. Consequently, the Q-learning agent needs to balance exploration and exploitation. To address this, a common strategy is the ϵ-greedy approach, where, in each period, the agent selects a random action with probability ϵ (exploration) and chooses the optimal action based on the current Q matrix with probability $1 - \epsilon$ (exploitation). Moreover, acknowledging that the reliability of the Q matrix improves as the agent gathers more information, it is customary to employ a time-decaying ϵ value, thereby increasing the propensity for exploitation over time. In accordance with this methodology, we incorporate an exponential time decay for ϵ, defined as $\epsilon = \exp(-\omega t)$, where ω denotes the decay rate.

B.7 Recommender Systems' Details

In each time period, J products are available, each characterized by a vector of static attributes $\boldsymbol{X_j}$ and a price p_j set by the seller[4]. Each consumer is represented by observed attributes $\boldsymbol{A_i}$, as well as two unobserved attributes, β_i and τ_i, which are not observable to the platform. Nevertheless, the platform can infer the distributions of these parameters across the consumer base using available data. In the subsequent subsections, we outline the platform's decision-making process in presenting products to each consumer i, focusing on the two aforementioned objectives.

For comparative analysis, we also consider a baseline scenario in which the platform does not utilize any optimization techniques, opting instead to present each consumer with a randomly assorted and ranked selection of products.

The revenue-maximization recommender system aims to maximize the total revenue for all the sellers on the market (Compiani et al. 2024). This approach closely aligns with the objectives of platforms that generate income by collecting a portion of sellers' revenue as commission fees, as observed in prominent platforms such as Amazon and Tmall.

For each consumer i, the platform is privy to her observable attributes $\boldsymbol{A}_i$ as well as the distributions of unobservable heterogeneities β_i and τ_i, but not their precise values. Additionally, the exact values of the post-search match values $\xi_{i,j}$ also remain unknown to the platform; only their distributions are accessible. Consequently, for each potential recommendation set[5], the platform computes the expected revenue and identifies the sets that maximize this expected revenue for exposure through the following process (Algorithm 3).

Specifically, the recommendation generation process for consumer i proceeds as follows. For each potential recommendation set, we first sample the unobserved heterogeneities β_i

[4]For clarity and simplicity, the time index is omitted in our notation, as previously done in Section 4.2.2.

[5]For instance, consider three products (indexed as 1, 2, and 3), where the platform must recommend two to the consumer, with the ranking of the displayed products being important. In this case, there are six possible sets: {1,2}, {2,1}, {1,3}, {3,1}, {2,3}, {3,2}.

and τ_i (outer loop), followed by sampling the post-search shocks for each product, $\widehat{\xi_{i,j}}$ (inner loop). In the inner loop, the estimated structural search model is employed to simulate the consumer's purchase decisions, thereby determining the revenue for each sample. By averaging across the samples in the inner loop, we numerically integrate out the post-search shocks, obtaining the expected revenue conditional on the specific unobservable heterogeneities (β_i and τ_i). Subsequently, in the outer loop, we average over all sampled unobservable heterogeneities to derive the unconditional expected revenue for the recommendation set. This numerical integration process is widely utilized in optimization and econometrics, such as in simulated maximum likelihood estimation, and will be further detailed in Section 4.3. Finally, we compare the expected revenues across all recommendation sets, identify those with the highest revenue, and randomly recommend one of these sets to consumer i.

In contrast, utility-maximization recommender systems are designed to enhance the utility experienced by consumers on the platform (Compiani et al. 2024). Utility serves as an economic measure of consumer satisfaction, which benefits the platform by improving consumer retention rates, thereby making it a common objective for many platforms (Zhang et al. 2019, Ursu 2018). Similar to revenue-maximization recommender systems, utility-maximization systems also employ numerical integration techniques to estimate the expected consumer utility for each potential recommendation set. The primary distinction lies in the post-simulation analysis: for each sample, the estimated structural model is used to simulate consumer behavior, after which the search history and purchase decisions are utilized to calculate utility (purchase utility minus total search costs) instead of revenue. Once the expected utility for each recommendation set is derived, the platform identifies those sets that maximize utility and randomly selects one of these optimal sets to recommend to consumer i (Algorithm 4).

Algorithm 3 Revenue-Maximization Recommender Systems

Require: $K, F(\beta_i), F(\tau_i), F(\xi_{i,j}), \boldsymbol{A_i}, p_j^t,$ and $\boldsymbol{X_j}$ for $j = 1, 2, \ldots, J$

Require: Estimated Structural Search Model CM

Derive all the potential K-product recommendation sets, each represented by $\boldsymbol{B}_b$

for each $\boldsymbol{B}_b$ **do**

  for each sample of unobserved heterogenities (NS_1 samples) **do**

    Sample $\widehat{\beta_i}$ based on $F(\beta_i)$

    Sample $\widehat{\tau_i}$ based on $F(\tau_i)$

    for each sample of post-search shocks (NS_2 samples) **do**

      Sample $\widehat{\boldsymbol{\xi_i}}$, which is a vector of $\widehat{\xi_{i,j}}$, based on $F(\xi_{i,j})$

      Simulate the purchase decision $\widehat{k_i}$ following CM

      Derive the platform's award (revenue): $m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}, \widehat{\boldsymbol{\xi_i}}) = p^t_{\widehat{k_i}}$

    end for

    Approximate the conditional expected revenue:

$$m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}) = \frac{1}{NS_2} \sum m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}, \widehat{\boldsymbol{\xi_i}})$$

  end for

  Approximate the conditional expected revenue:

$$m|\boldsymbol{B}_b = \frac{1}{NS_1} \sum m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i})$$

end for

Find the $\boldsymbol{B}_b$ with the largest expected revenue $m|\boldsymbol{B}_b$

Output: Equally assign the exposure to these sets with the largest expected revenue

Algorithm 4 Utility-Maximization Recommender Systems

Require: $K, F(\beta_i), F(\tau_i), F(\xi_{i,j}), \boldsymbol{A_i}, p_j^t,$ and $\boldsymbol{X_j}$ for $j = 1, 2, \ldots, J$

Require: Estimated Structural Search Model CM

Derive all the potential K-product recommendation sets, each represented by $\boldsymbol{B}_b$

for each $\boldsymbol{B}_b$ **do**

  for each sample of unobserved heterogenities (NS_1 samples) **do**

    Sample $\widehat{\beta_i}$ based on $F(\beta_i)$

    Sample $\widehat{\tau_i}$ based on $F(\tau_i)$

    for each sample of post-search shocks (NS_2 samples) **do**

      Sample $\widehat{\boldsymbol{\xi_i}}$, which is a vector of $\widehat{\xi_{i,j}}$, based on $F(\xi_{i,j})$

      Simulate the search sequence $\widehat{\boldsymbol{H_i}}$ and purchase decision $\widehat{k_i}$ following CM

      Derive the utility:

$$m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}, \widehat{\boldsymbol{\xi_i}}) = u_{i,\widehat{k_i}} - \sum_{j \in \widehat{\boldsymbol{H_i}}} c_{i,j}$$

    end for

    Approximate the conditional expected utility:

$$m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}) = \frac{1}{NS_2} \sum m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i}, \widehat{\boldsymbol{\xi_i}})$$

  end for

  Approximate the conditional expected utility:

$$m|\boldsymbol{B}_b = \frac{1}{NS_1} \sum m|(\boldsymbol{B}_b, \widehat{\beta_i}, \widehat{\tau_i})$$

end for

Find the $\boldsymbol{B}_b$ with the largest expected utility $m|\boldsymbol{B}_b$

Output: Equally assign the exposure to these sets with the largest expected utility

B.8 Two Thresholds

Table B.4: Two Thresholds

	$w = 0.38$	$w = 0.43$	**Baseline**
Prices	1.7131 (0.0183)	1.9433 (0.0145)	1.7108 (0.0159)
Revenue	0.2288 (0.0015)	0.2398 (0.0009)	0.2228 (0.0013)
Utility	0.4884 (0.0025)	0.4673 (0.0016)	0.4668 (0.0024)

B.9 Alternative Economic Environment: Logit Demand

To align our proposed structural search model with the traditional algorithmic collusion literature (Calvano et al. 2020b), which typically employs the canonical logit model, we also evaluate our results using the logit model as a robustness check. In fact, the logit demand model can be considered a restricted version derived from the sequential search model. Specifically, from the sequential search model, we can (1) set the search cost to zero instead of allowing it to take any value estimated from data, (2) assume that all recommended products' full information is available to consumers, thereby eliminating position bias under this full information perspective (a natural consequence of (1)), and (3) change the distribution of taste shocks from a normal distribution to a type-I extreme value distribution to facilitate a closed-form expression of the choice probability as a logit function. This modification yields the logit choice model when a consumer i sees a recommendation set:

$$d_{i,j}^t = \frac{e^{\frac{v_j - p_j^t}{\mu}}}{\sum_{k \in \boldsymbol{B_i}} e^{\frac{v_k - p_k^t}{\mu}} + e^{\frac{v_0}{\mu}}} \tag{B.10}$$

Here, the parameters v_j represent representative utility for each product j, capturing vertical differentiation. The parameter μ reflects the degree of horizontal differentiation where perfect substitution is achieved as μ approaches 0. The denominator of the equation represents the exponential sum over all products' consumption utilities within the recommendation set B_i along with the outside good, characterized by the quality index v_0. Furthermore, under the assumption that all consumers are homogeneous in their observable preferences and differ solely in their idiosyncratic taste shocks, the above single-individual choice probability also characterizes the demand for product j when a unit of consumers is presented with the recommendation set $\boldsymbol{B}_i$.

In alignment with our primary framework, we continue to address a two-out-of-three recommendation scenario. In the absence of position bias within this logit demand model, product ranking does not influence consumer behavior, thereby reducing the platform's task to an assortment optimization problem. Furthermore, since consumers are perceived as

homogeneous by the platform, it is sufficient to construct a single recommendation set that either maximizes revenue or maximizes consumer utility to all consumers.

Table B.5: Price, Revenue, and Utility Comparison (Logit Demand Model)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	0.9186 (0.0069)	0.3541 (0.0247)	0.5623 (0.0087)
Revenue	0.2247 (0.0006)	0.0740 (0.0059)	0.1691 (0.0025)
Utility	0.3257 (0.0025)	0.9320 (0.0200)	0.6366 (0.0083)

The expected revenue for each recommendation set can be computed as:

$$r_{\boldsymbol{B}_i} = \sum_{j \in \boldsymbol{B}_i} p_j^t \cdot \frac{\exp\left(\frac{v_j - p_j^t}{\mu}\right)}{\sum_{k \in \boldsymbol{B}_i} \exp\left(\frac{v_k - p_k^t}{\mu}\right) + \exp\left(\frac{v_0}{\mu}\right)} \tag{B.11}$$

Conversely, the expected utility for a recommendation set is calculated using the welfare formula (Train 2009):

$$TU_{\boldsymbol{B}_i} = \sum_{k \in \boldsymbol{B}_i} \exp\left(\frac{v_k - p_k^t}{\mu}\right) + \exp\left(\frac{v_0}{\mu}\right) \tag{B.12}$$

We adhere to the game sequence outlined in Algorithm 2 of Section 4.2.1 and employ Q-learning pricing algorithms with consistent parameters. We normalize the valuations v_j of all products to 1 and set the outside option valuation v_0 to 0, allowing prices to fluctuate between 0 and 1. After conducting 50 simulation trials, each spanning 20,000,000 periods, we present the equilibrium prices, revenue, and utility in a manner consistent with Section 4.4.

As illustrated in Table B.5, the comparisons across the three cases are qualitatively similar to our main results, thereby reinforcing the robustness of our findings across different demand model specifications.

B.10 Alternative Economic Environment: Heterogeneous Logit Demand

In Appendix B.9, we employ a logit demand model that does not incorporate customer segmentation (Calvano et al. 2020b). In this appendix, we extend the economic environment by introducing heterogeneous customer types and subsequently adapt the recommender systems to optimize for different customer categories.

Specifically, we allow each consumer to possess heterogeneous outside option values. To illustrate, we assign half of the consumers a higher outside option value of $v_0 = 0.25$, while the remaining consumers have a lower outside option value of $v_0 = 0$. For each segment, we recommend the set that maximizes either expected revenue or expected utility. With all other settings held constant, we present the results in Table B.6.

Notably, the primary findings remain consistent under this modified setting. When the recommender system is configured for revenue maximization, sellers tend to engage in more collusive behavior, resulting in elevated equilibrium prices (0.8167) and increased revenue (0.1914) at the expense of consumer utility (0.4421). Conversely, when the system prioritizes the maximization of consumer utility, the equilibrium is characterized by lower prices (0.5399) and enhanced consumer utility (0.6724), alongside a reduction in seller revenue (0.1540).

Table B.6: Price, Revenue, and Utility Comparison (Heterogeneous Logit Demand Model)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	0.8167 (0.0069)	0.3393 (0.0242)	0.5399 (0.0081)
Revenue	0.1914 (0.0006)	0.0654 (0.0055)	0.1540 (0.0021)
Utility	0.4421 (0.0034)	0.9590 (0.0191)	0.6724 (0.0075)

B.11 Empirical Data Description and Summary Statistics

In this appendix, we detail the construction of our dataset for structural estimation, basically following the steps taken in Chung et al. (2024) and Ursu (2018). The original Expedia dataset contains millions of search impressions, including query information and all displayed hotels. Because the dataset is divided into training and test sets, and the test set lacks consumer decision data (search or purchase), we focus exclusively on the training set.

We apply the following filtering procedures to the training set:

1. Exclude search impressions that include any hotels with extreme prices (less than \$10 or more than \$1,000).

2. For search impressions that led to transactions, infer the tax from the total spending and displayed prices, and exclude those impressions where the inferred tax is less than \$1 or exceeds 30% of the price.

3. Drop search impressions containing any missing values (NaN) in query or hotel attributes.

4. Exclude search impressions based on Expedia's ranking rather than the random ranking.

5. Segment the dataset by travel destination, recognizing that consumers and hotels may vary significantly across destinations, and focus on the destination with the highest volume of search impressions (destination ID 8192).

6. Although most impressions have similar lengths (number of hotels displayed), there are outliers. For the selected destination, the number of hotels per impression ranges from 5 to 35, with the 25th percentile at 29. Therefore, we restrict our analysis to search

impressions with lengths greater than 29[6].

The final dataset comprises 1,067 impressions and 34,702 observations, where each observation represents a hotel option within a search impression. It includes hotel information, query information (customer attributes constant across different hotels within one search impression), and final decisions (search or purchase). Additionally, we add an outside option for each search impression, coded as a purchase decision equal to one if the customer does not buy from any displayed hotel and zero otherwise. Thus, for the structural estimation, the total number of observations is 35,769 (34,702 hotel options plus 1,067 outside options). For 34,702 hotel options, the variable descriptions and summary statistics are provided in Table B.7.

Table B.7: Variable Description and Summary Statistics

Variable	**Description**	**Mean**	**Std**	**Min**	**Max**
Price ($100)	Price of the hotel ($100)	1.4326	0.9026	0.1700	9.8400
HotelStars	The star rating of the hotel	3.9291	0.8633	2	5
HotelReview	Average customer review rating	4.0370	0.4901	3	5
Brand	A binary indicator of chained hotel	0.7872	0.4093	0	1
HotelLocationScore	A score of location desirability	4.0328	0.3011	3.0400	4.3800
Promo	A binary indicator of promotion	0.5919	0.4915	0	1
ReqRoomNum	Number of rooms required	1.1414	0.4151	1	4
SatNight	A binary indicator of Saturday night involved in travel	0.4752	0.4994	0	1
Click	A binary indicator of whether the customer search the hotel	0.0370	0.1888	0	1
Tran	A binary indicator of whether the customer book the hotel	0.0026	0.0511	0	1

[6]This is slightly less stringent than Ursu (2018), which only uses the two most frequently occurring search impression lengths.

B.12 Alternative Estimation: Handling Price Endogeneity

Although prices are not randomized, hotel revenue management systems typically do not set prices at the individual level, which would otherwise introduce first-degree price discrimination (Garcia et al. 2022, Koulayev 2014). Instead, hotel prices can be largely explained by hotel and query characteristics (Ursu 2018), mitigating the concern that prices are correlated with individual-level post-search preference shocks (De los Santos and Koulayev 2017). Moreover, prior work has shown that applying a control function approach to address potential price endogeneity generally yields similar estimates (De los Santos and Koulayev 2017). To further probe the sensitivity of our estimates to price endogeneity, we follow the existing literature and implement a control-function method to complement the search model estimation (De los Santos and Koulayev 2017). The resulting price sensitivity estimates are largely consistent with the main estimates obtained without the control function.

Table B.8: Structural Estimation Results with Control Function

Variable	**Coefficients**	**Std**
m_β	-0.8477***	(0.0002)
σ_β	0.0082***	(0.0020)
$\Gamma_{PredictedPrice}$	0.27234***	(0.0127)
$\Gamma_{HotelStars}$	0.3320***	(0.0119)
$\Gamma_{HotelReview}$	-0.2636***	(0.0133)
Γ_{Brand}	-0.0181	(0.0141)
$\Gamma_{HotelLocationScore}$	-0.0867***	(0.0054)
Γ_{Promo}	0.1072***	(0.0191)
$\Theta_{ReqRoomNum}$	-0.0822***	(0.0045)
$\Theta_{SatNight}$	0.0461***	(0.0046)
$\upsilon_{i,0}$	0.1089***	(0.0088)
m_τ	-1.2393***	(0.0071)
σ_τ	0.0066**	(0.0030)
φ	0.0059***	(0.0011)

Note: Bootstrap standard errors in parentheses; *** p<0.01, ** p<0.05, * p<0.1.

Specifically, we segment each observation (i.e., an individual hotel from the search results list) based on hotel identity, the number of rooms, and whether the stay includes a Saturday night (De los Santos and Koulayev 2017). Each segment is then treated as a fixed effect, and we regress the observed price on these segment fixed effects to account for segment-level price differences (De los Santos and Koulayev 2017). The regression yields an R-squared of 0.62, indicating that these characteristics alone possess substantial explanatory power over price variation. We then include both the predicted price, serving as the control function, and the actual price in the sequential search model (De los Santos and Koulayev 2017).

The estimation results are presented in Table B.8. Using these estimates, we conduct counterfactual simulations and obtain qualitatively consistent results, as summarized in Table B.9.

Table B.9: Price, Revenue, and Utility Comparison Across Objectives

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	1.8689 (0.0114)	0.9094 (0.0452)	1.3445 (0.0183)
Revenue	0.1453 (0.0004)	0.0828 (0.0018)	0.1231 (0.0007)
Utility	0.2161 (0.0014)	0.4071 (0.0047)	0.2992 (0.0022)

B.13 Alternative Pricing Algorithm: More Exploration

In our primary analyses, we demonstrate that prices increase (decrease) under conditions where product exposure is governed by revenue-maximizing (utility-maximizing) recommender systems. As illustrated in Figure 4.2, the algorithms' exploration process appears to alter only the magnitude of price changes rather than their direction across the three different scenarios examined. Nevertheless, it remains plausible that extended and more comprehensive exploration could yield different outcomes, such as the algorithms learning to collude even in the presence of utility-maximizing recommender systems. Therefore, in this appendix, we investigate the effects of increased exploration on price dynamics and equilibrium outcomes.

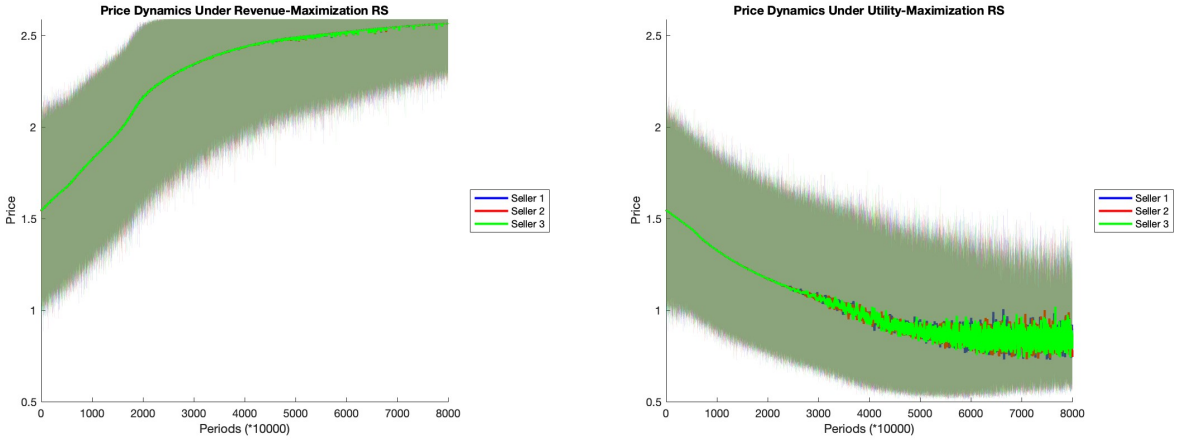
Using an ϵ-greedy algorithm, the exploration rate, denoted by ϵ, modulates the probability of undertaking a stochastic action at each temporal interval according to the equation $\epsilon = e^{-\omega t}$. This probability undergoes exponential decay, governed by both the decay rate ω and the time parameter t. Extended exploration periods enable the algorithm to obtain more accurate estimates of the optimal pricing strategy; however, they also incur higher opportunity costs due to the exploration of suboptimal pricing alternatives. In our baseline configuration, the decay rate ω is set to 2×10^{-7}. In contrast, this appendix explores a "More Exploration" scenario, wherein we employ a substantially smaller decay rate of $\omega = 5 \times 10^{-8}$, thereby significantly increasing exploratory activities. For example, after twenty million periods, the pricing algorithms in the baseline configuration retain an approximate probability of 1.11% for opting for a stochastic action. In contrast, in the "More Exploration" scenario, this probability remains above 36%.

To ensure experimental consistency, all parameters remain identical to those in the baseline configuration, with the sole modification being the exploration rate parameter ω. We conduct 20 simulation runs, each spanning 80,000,000 periods, after which the exploration probability falls below 1%. The resulting average equilibrium prices and profits are presented in Table B.10, while the temporal evolution of the learning dynamics is depicted in Figure

B.2. Notably, the equilibrium prices attained in the "More Exploration" scenario exhibit substantial congruence with those observed in the baseline scenario. Specifically, compared to the equilibrium price of 1.7537 observed in the baseline condition, the implementation of revenue-maximizing recommender systems results in an elevated price of 2.5668. Conversely, the deployment of utility-maximizing recommender systems leads to a reduced price of 0.7930. Furthermore, the learning dynamics of the pricing algorithms, influenced by the different types of recommender systems, continue to progress monotonically toward their respective equilibria, in alignment with the objectives of the respective recommender systems.

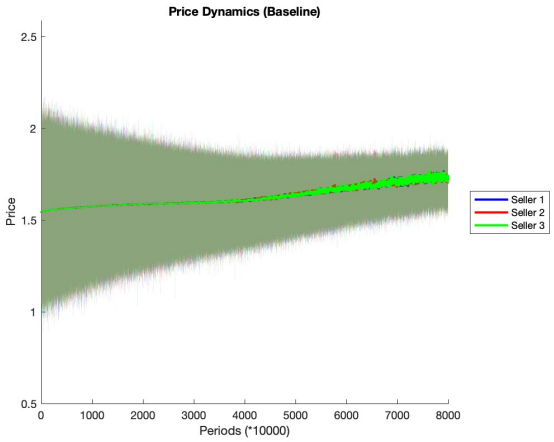
Table B.10: Price, Revenue, and Utility Comparison ($\mu = 0.25$, $\omega = 5 \times 10^{-8}$)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	2.5668 (0.0248)	0.7930 (0.0455)	1.7537 (0.0318)
Revenue	0.2681 (0.0010)	0.1132 (0.0036)	0.2245 (0.0023)
Utility	0.3271 (0.0025)	0.6606 (0.0052)	0.4612 (0.0045)



(a) Revenue-Maximization RS

(b) Utility-Maximization RS



(c) Baseline

Figure B.2: Learning Dynamics ($\mu = 0.25$, $\omega = 5 \times 10^{-8}$)

Note: The points on the continuous lines depict the moving average, with a window size of 10000, of the average price observed across 20 experiments at each t. The shaded region's upper and lower limits indicate the highest and lowest values occurring within the window.

B.14 Alternative Pricing Algorithm: Multi-armed Bandit

Q-learning inherently incorporates a forward-looking component through its state transitions. To assess the robustness of our findings, we relax the forward-looking assumption by modeling the algorithms as basic multi-armed bandits. Instead of constructing a Q-table of size $|\boldsymbol{S}| \times |\boldsymbol{P}|$ to record the rewards of various pricing decisions under different states (i.e., past pricing outcomes), we now construct a reward table of size $2 \times |\boldsymbol{P}|$, which tracks the number of times each pricing decision (arm) has been attempted and the corresponding average reward for each arm. We maintain an initially empty reward table for each pricing agent (seller) and allow each seller to learn to set prices through sampling. Consistent with our Q-learning approach, we employ an ϵ-greedy sampling strategy with the same exploration rate.

We follow the simultaneous game sequence once more and present the equilibrium results in Table B.11. Notably, the results remain largely consistent with our primary findings. This consistency arises because our core mechanism relies on the recommender system's influence on the pricing agents' rewards, making forward-looking behavior an unnecessary condition for the pricing algorithms to produce the observed outcomes.

Table B.11: Price, Revenue, and Utility Comparison Across Objectives (MAB)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	2.5522 (0.0144)	1.0619 (0.0118)	1.5509 (0.0071)
Revenue	0.2664 (0.0007)	0.1621 (0.0010)	0.2093 (0.0007)
Utility	0.3347 (0.0023)	0.5934 (0.0016)	0.4876 (0.0011)

B.15 Alternative Pricing Objective of Sellers

In Calvano et al. (2020b), sellers are assumed to optimize revenue. When sellers maximize revenue, they must balance a trade-off between demand and per-sale revenue: setting lower prices can boost demand but reduce revenue per sale, whereas setting higher prices can increase revenue per sale but suppress demand. In such contexts, sellers may engage in price undercutting, thereby intensifying price competition.

Accordingly, starting from the Bertrand game, most studies in industrial organization and traditional game theory assume that sellers aim to maximize revenue, ensuring meaningful competitive dynamics (Narahari et al. 2009). By contrast, if a seller instead sought to maximize demand (or consumer utility), it could simply set its price to zero, regardless of competitors' prices, thereby removing any incentive for strategic interactions.

Despite these considerations, we show that recommender systems retain their influence even when sellers optimize demand and willingly offer the lowest possible prices. To illustrate this, we modify the seller's objective to demand maximization and rerun our numerical experiments. Here we report the results using MAB pricing algorithms, but the results remain robust when Q-learning is used as the pricing algorithm instead. As shown in Table B.14, both the utility-maximizing and the baseline scenarios produce the lowest possible prices. Nevertheless, when the platform pursues revenue maximization, it still induces sellers to sustain relatively higher prices. This finding further emphasizes the pivotal role of recommender systems in shaping algorithmic pricing strategies.

Furthermore, when some sellers prioritize demand while others focus on revenue maximization, an asymmetric equilibrium might emerge. For example, consider a scenario in which one seller aims to maximize demand and two sellers aim to maximize revenue, with the recommender system oriented towards utility maximization. In this case, the utility-driven seller reduces its price to the lowest possible level, while the revenue-driven sellers converge to slightly higher prices, though these prices remain lower than those observed in the baseline scenario. Due to the vast number of possible combinations, we do not present all

these results here. However, from all these interesting asymmetric equilibria, we consistently observe the influential role of recommender systems across various configurations.

Table B.12: Price, Revenue, and Utility Comparison Across Objectives (Demand-Driven Sellers)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	1.8871 (0.0080)	0.5195 (0.0134)	0.5197 (0.0134)
Revenue	0.2515 (0.0004)	0.0903 (0.0002)	0.0920 (0.0007)
Utility	0.4285 (0.0005)	0.6970 (0.0006)	0.6643 (0.0016)

B.16 Alternative Game Sequence: Asynchronous Pricing Game

In our primary analyses, we adhere to the framework established by Calvano et al. (2020b), assuming that all sellers make pricing decisions simultaneously. While this assumption underpins our main analyses, we revisit and relax it in this section.

If we permit sellers to update their pricing decisions sequentially, without loss of generality, the decision-making process unfolds as follows: seller 1 updates first, followed by seller 2 and then seller 3. Given the structure of Q-learning, seller 1 observes the prices from the preceding three periods before making a decision. After seller 1's update, the environment changes over the subsequent two periods. By aggregating these three periods into a single "macro" period, the scenario reduces to the simultaneous decision-making framework. Therefore, from a theoretical standpoint, an asynchronous decision sequence does not fundamentally alter the nature of the algorithmic pricing game.

This reasoning similarly applies to settings where multi-armed bandits (MAB) serve as the pricing algorithms. For the sake of comparison, we present the results of the MAB approach under the aforementioned asynchronous game in Table B.13. As anticipated, these results are largely consistent with those reported in Appendix B.14, thereby reinforcing the robustness of our findings. The same consistency is observed in the Q-learning based pricing game.

Table B.13: Price, Revenue, and Utility Comparison Across Objectives (Asynchronous)

	Revenue-Maximization	**Utility-Maximization**	**Baseline**
Prices	2.5504 (0.0142)	1.0694 (0.0099)	1.5564 (0.0069)
Revenue	0.2662 (0.0007)	0.1635 (0.0008)	0.2098 (0.0007)
Utility	0.3356 (0.0024)	0.5910 (0.0012)	0.4868 (0.0011)

B.17 Alternative Recommendation Objective: Total Demand

It is worth noting that demand can serve as another plausible objective for the platform. Economically, demand is analogous to utility, since higher utility increases consumers' purchase probability and, consequently, overall demand. To maintain conceptual clarity and alignment with the existing literature, we primarily focus on revenue and consumer utility in our main analyses, while nonetheless examining demand as an alternative objective here. Specifically, we substitute the platform's objective with total market demand, keeping all other aspects of the recommender system unchanged.

We conduct two sets of numerical experiments, one with revenue-driven sellers and another with demand-driven sellers, both operating under a demand-maximizing recommender system. As reported in Table B.14, our rationale holds. Under the standard revenue-driven pricing framework, the equilibrium price settles at a level lower than the baseline scenario (see Table 4.1 for comparison). When sellers optimize demand, as expected, the price converges to its minimum possible level, consistent with the outcomes shown in the Column 2 of Table B.12.

Table B.14: Price, Revenue, and Utility Comparison Across Different Pricing Objectives

	Revenue-Driven Sellers	**Demand-Driven Sellers**
Prices	1.0335 (0.0114)	0.5207 (0.0134)
Revenue	0.1667 (0.0009)	0.0934 (0.0003)
Utility	0.5697 (0.0014)	0.6708 (0.0006)

B.18 Alternative Recommendation Objective: Weighted Average

It is worth noting that, in practice, platforms may adopt a mixed objective to balance the welfare of both sellers (revenue) and consumers (utility). To illustrate how varying the weight assigned to these two objectives influences the pricing dynamics, we show five different cases with weights ranging from 0 (pure utility maximization) to 1 (pure revenue maximization) and conduct corresponding numerical experiments. Our findings indicate that as the weight increases, equilibrium prices and sellers' revenues also increase, while consumer utility gradually decreases.

Table B.15: Price, Revenue, and Utility Comparison Across Objectives (Different Weights)

Weight	**0**	**0.25**	**0.5**	**0.75**	**1**
Prices	1.0078	1.0916	1.9039	2.4292	2.5657
	(0.0509)	(0.0257)	(0.0166)	(0.0193)	(0.0159)
Revenue	0.1278	0.1614	0.2411	0.2616	0.2681
	(0.0043)	(0.0026)	(0.0011)	(0.0008)	(0.0006)
Utility	0.6336	0.5975	0.4546	0.3597	0.3273
	(0.0070)	(0.0039)	(0.0026)	(0.0029)	(0.0016)

Appendix C

APPENDIX TO DEPERSONALIZING SEARCH ON SOCIAL MEDIA: A LARGE-SCALE FIELD EXPERIMENT

C.1 Proofs

C.1.1 Proof of Proposition 1

The depth b_S is defined implicitly by the click-target equation (5.4), $\int_0^{b_S} \Phi_S(n;\rho_S)\,dn = k$. Differentiate both sides with respect to ρ_S. Both the upper limit b_S and the integrand Φ_S depend on ρ_S, so the Leibniz rule gives

$$\Phi_S(b_S)\,\frac{\partial b_S}{\partial \rho_S} + \int_0^{b_S} \frac{\partial \Phi_S}{\partial \rho_S}\,dn = 0, \tag{C.1}$$

Since $\Phi_S(b_S) > 0$, solving for the depth response yields

$$\frac{\partial b_S}{\partial \rho_S} = -\frac{1}{\Phi_S(b_S)} \int_0^{b_S} \frac{\partial \Phi_S}{\partial \rho_S}\,dn. \tag{C.2}$$

To sign the integrand, differentiate the click probability (5.3), $\phi_{S,\tau}(n) = \bar{G}(\xi_{S,\tau} - m_S(n))$, with respect to ρ_S. The parameter ρ_S enters only through the match $m_S(n) = \rho_S(\bar{a} - \gamma n)$, so the chain rule, together with $\bar{G}' = -g$ and $\partial m_S/\partial \rho_S = \bar{a} - \gamma n$, gives

$$\begin{aligned} \frac{\partial \phi_{S,\tau}}{\partial \rho_S} &= \bar{G}'\big(\xi_{S,\tau} - m_S(n)\big) \cdot \Big(- \frac{\partial m_S}{\partial \rho_S}\Big) = -g\big(\xi_{S,\tau} - m_S(n)\big) \cdot \big(- (\bar{a} - \gamma n)\big) \\ &= g\big(\xi_{S,\tau} - m_S(n)\big)\,(\bar{a} - \gamma n). \end{aligned} \tag{C.3}$$

Hence $\partial \Phi_S/\partial \rho_S = \big[\pi\, g(\xi_{S,o} - m_S(n)) + (1-\pi)\, g(\xi_{S,p} - m_S(n))\big](\bar{a} - \gamma n)$, which is nonnegative for $n \le n^* = \bar{a}/\gamma$ and strictly positive on a set of positive measure. Under Condition 1, the integral in (C.2) is therefore positive and $\partial b_S/\partial \rho_S < 0$: a fall in ρ_S raises b_S. Finally, because $\Phi_S > 0$ makes the left side of (5.4) increasing in b_S, the depth is largest at the lowest personalization intensity. It therefore suffices to check the bound there, which is equivalent to $k \le \int_0^{\bar{a}/\gamma} \Phi_S(n;\rho_S)\,dn$. The argument uses only $g > 0$. □

C.1.2 Proof of Proposition 2

The proof rests on one observation, used three times here and once in Lemma 1: *for any continuously differentiable, strictly increasing f, the per-item average $\bar{f}(\rho_S) = b_S^{-1}\int_0^{b_S} f(m_S(n))\,dn$ is strictly increasing in ρ_S.* To establish it, differentiate $\bar{f}$ with respect to ρ_S. The prefactor $1/b_S$, the upper limit b_S, and the integrand $f(m_S(n))$ all depend on ρ_S, so the quotient and Leibniz rules, with $\partial m_S/\partial\rho_S = \bar{a} - \gamma n$, give

$$\begin{aligned}\frac{d\bar{f}}{d\rho_S} &= -\frac{1}{b_S^2}\frac{\partial b_S}{\partial \rho_S}\int_0^{b_S} f(m_S(n))\,dn + \frac{1}{b_S}\left[f(m_S(b_S))\frac{\partial b_S}{\partial \rho_S} + \int_0^{b_S} f'(m_S(n))\,(\bar{a}-\gamma n)\,dn\right] \\ &= \frac{\partial b_S/\partial \rho_S}{b_S}\Big(f(m_S(b_S)) - \bar{f}\Big) + \frac{1}{b_S}\int_0^{b_S} f'(m_S(n))\,(\bar{a}-\gamma n)\,dn.\end{aligned} \tag{C.4}$$

The integral term is positive because $f' > 0$ and $\bar{a} - \gamma n \geq 0$ for $n \leq n^*$. The first term is also positive: $f(m_S(\cdot))$ is strictly decreasing in n, so its endpoint value lies below its average and the bracket is negative, while $\partial b_S/\partial\rho_S < 0$ by Proposition 1. Hence $d\bar{f}/d\rho_S > 0$, and lowering ρ_S lowers $\bar{f}$.

Both rates fall. Apply (C.4) to $f(m) = \bar{G}(\xi_{S,\tau} - m)$, which is strictly increasing because $f'(m) = g(\xi_{S,\tau} - m) > 0$ everywhere (Assumption 1). Since $\mathrm{CTR}_{S,\tau} = \bar{f}$,

$$\frac{d\,\mathrm{CTR}_{S,\tau}}{d\rho_S} > 0, \qquad \tau \in \{o, p\}, \tag{C.5}$$

so lowering ρ_S lowers both rates. This part uses neither the positivity of the thresholds nor Assumption 3.

The ordinary rate falls more. Because the two rates share the single depth b_S, their gap is itself a per-item average: $\mathrm{CTR}_{S,o} - \mathrm{CTR}_{S,p} = \bar{H}$ with $H(m) = \bar{G}(\xi_{S,o} - m) - \bar{G}(\xi_{S,p} - m)$. Its derivative is

$$H'(m) = g(\xi_{S,o} - m) - g(\xi_{S,p} - m) > 0, \tag{C.6}$$

because $0 < \xi_{S,o} - m < \xi_{S,p} - m$ on the browsed range (Assumption 3) and g is strictly decreasing on $(0, \infty)$. So H is strictly increasing, and (C.4) applied to H gives $\frac{d}{d\rho_S}(\mathrm{CTR}_{S,o} -$

$\mathrm{CTR}_{S,p}) > 0$. Combining this with (C.5) at $\tau = p$ yields

$$\frac{d\,\mathrm{CTR}_{S,o}}{d\rho_S} > \frac{d\,\mathrm{CTR}_{S,p}}{d\rho_S} > 0, \tag{C.7}$$

so a fall in ρ_S reduces $\mathrm{CTR}_{S,o}$ by more. Without Assumption 3, the threshold arguments could fall on the increasing side of g, where $H' < 0$ and the ordering reverses. □

C.1.3 Proof of Lemma 1

The search standard falls. By (5.6), $W_S/b_S = \bar{\Psi}_S - \mu_S$ is the per-item average of $\Psi_S(m_S(n)) = \sum_\tau \pi_\tau \Psi_{S,\tau}(m_S(n))$ net of the constant μ_S, so its response turns on the derivative $\Psi'_{S,\tau}$. A type-τ item is clicked when $m + \eta + q_\tau - \kappa_\tau - r_S > 0$, that is, when $\eta > \xi_{S,\tau} - m$, so

$$\Psi_{S,\tau}(m) = \int_{\xi_{S,\tau}-m}^{\infty} \big(m + \eta + q_\tau - \kappa_\tau - r_S\big)\, g(\eta)\, d\eta. \tag{C.8}$$

Differentiating by the Leibniz rule gives

$$\begin{aligned}\Psi'_{S,\tau}(m) &= \underbrace{\big(m + \eta + q_\tau - \kappa_\tau - r_S\big)\, g(\eta)\,\big|_{\eta=\xi_{S,\tau}-m}}_{=0} + \int_{\xi_{S,\tau}-m}^{\infty} \underbrace{\tfrac{\partial}{\partial m}\big(m + \eta + q_\tau - \kappa_\tau - r_S\big)}_{=1}\, g(\eta)\, d\eta \\ &= \bar{G}(\xi_{S,\tau} - m).\end{aligned} \tag{C.9}$$

Summing over types, $\Psi'_S(m) = \sum_\tau \pi_\tau \bar{G}(\xi_{S,\tau} - m) > 0$, so Ψ_S is strictly increasing, and (C.4) applied to $f = \Psi_S$ gives

$$\frac{d}{d\rho_S}\left(\frac{W_S}{b_S}\right) = \frac{d\bar{\Psi}_S}{d\rho_S} > 0. \tag{C.10}$$

A fall in ρ_S therefore lowers W_S/b_S, and by (5.7) the standard ω falls.

The browsing response. The recommendation surplus in (5.10) can be written in integral form as $\Psi_{R,o}(m,\omega) = \int_{\omega-q_o-m}^{\infty} (m + \eta + q_o - \omega)\, g(\eta)\, d\eta$, where the integrand is again zero at the lower limit. Differentiating in m and in ω by the Leibniz rule gives

$$\frac{\partial \Psi_{R,o}}{\partial m} = \int_{\omega-q_o-m}^{\infty} g(\eta)\, d\eta = \phi_{R,o}, \qquad \frac{\partial \Psi_{R,o}}{\partial \omega} = -\int_{\omega-q_o-m}^{\infty} g(\eta)\, d\eta = -\phi_{R,o}. \tag{C.11}$$

Now differentiate the stopping rule (5.10), $\Psi_{R,o}(m_R(b_R), \omega) = \mu_R$, with respect to ω. The left side depends on ω directly through the second argument and indirectly through $b_R(\omega)$

in the first argument $m_R(b_R)$. The chain rule, with $m'_R(n) = \partial m_R/\partial n = -\rho_R\gamma$ and the derivatives in (C.11), gives

$$\begin{aligned} \frac{\partial \Psi_{R,o}}{\partial m}\, m'_R(b_R)\, \frac{\partial b_R}{\partial \omega} + \frac{\partial \Psi_{R,o}}{\partial \omega} &= 0 \\ \phi_{R,o}(b_R)\,(-\rho_R\gamma)\, \frac{\partial b_R}{\partial \omega} - \phi_{R,o}(b_R) &= 0 \\ \frac{\partial b_R}{\partial \omega} &= -\frac{1}{\rho_R\gamma} < 0, \end{aligned} \tag{C.12}$$

so the lower ω deepens recommendation browsing. □

C.1.4 Proof of Proposition 3

The ordinary click probability (5.8), $\phi_{R,o}(n) = \bar{G}(\omega - q_o - m_R(n))$, depends on the standard ω and on the rank n only through its argument, which we abbreviate as $u(n,\omega) = \omega - q_o - m_R(n)$. Since $m'_R(n) = -\rho_R\gamma$, the partial derivatives of the argument are $\partial u/\partial\omega = 1$ and $\partial u/\partial n = \rho_R\gamma$. Differentiating $\phi_{R,o} = \bar{G}(u)$ with respect to ω and to n by the chain rule, with $\bar{G}' = -g$, gives

$$\frac{\partial \phi_{R,o}}{\partial \omega} = \bar{G}'(u)\,\frac{\partial u}{\partial \omega} = -g(u), \qquad \frac{\partial \phi_{R,o}}{\partial n} = \bar{G}'(u)\,\frac{\partial u}{\partial n} = -g(u)\,\rho_R\gamma. \tag{C.13}$$

The two derivatives differ only by the constant factor $\rho_R\gamma$, so

$$\frac{\partial \phi_{R,o}}{\partial \omega} = \frac{1}{\rho_R\gamma}\,\frac{\partial \phi_{R,o}}{\partial n}. \tag{C.14}$$

This relation lets us convert an ω-derivative under the integral sign into boundary terms. The ordinary click-through rate is the number of ordinary clicks over the session, $A(\omega) \equiv \int_0^{b_R} \phi_{R,o}(n)\,dn$, divided by the depth b_R: $\mathrm{CTR}_{R,o} = A(\omega)/b_R$. Both the numerator A and the depth b_R depend on ω. Differentiate the numerator with respect to ω. The upper limit b_R and the integrand both move, so the Leibniz rule and then (C.14) give

$$\begin{aligned} A'(\omega) &= \phi_{R,o}(b_R)\,\frac{\partial b_R}{\partial \omega} + \int_0^{b_R} \frac{\partial \phi_{R,o}}{\partial \omega}\,dn = \phi_{R,o}(b_R)\,\frac{\partial b_R}{\partial \omega} + \frac{1}{\rho_R\gamma}\int_0^{b_R} \frac{\partial \phi_{R,o}}{\partial n}\,dn \\ &= \phi_{R,o}(b_R)\,\frac{\partial b_R}{\partial \omega} + \frac{1}{\rho_R\gamma}\big(\phi_{R,o}(b_R) - \phi_{R,o}(0)\big), \end{aligned} \tag{C.15}$$

where the last integral is evaluated by the fundamental theorem of calculus. Substituting $\partial b_R/\partial\omega = -1/(\rho_R\gamma)$ from (C.12) makes the two terms in $\phi_{R,o}(b_R)$ cancel, leaving

$$A'(\omega) = -\frac{\phi_{R,o}(0)}{\rho_R\gamma}. \tag{C.16}$$

Now differentiate $\mathrm{CTR}_{R,o} = A(\omega)/b_R$ with respect to ω by the quotient rule and substitute (C.16) and $\partial b_R/\partial\omega$:

$$\begin{aligned}\frac{d\,\mathrm{CTR}_{R,o}}{d\omega} &= \frac{A'(\omega)}{b_R} - \frac{A(\omega)}{b_R^2}\frac{\partial b_R}{\partial\omega} = \frac{A'(\omega)}{b_R} - \mathrm{CTR}_{R,o}\,\frac{\partial b_R/\partial\omega}{b_R} \\ &= \frac{1}{b_R}\left(-\frac{\phi_{R,o}(0)}{\rho_R\gamma}\right) - \mathrm{CTR}_{R,o}\left(\frac{-1/(\rho_R\gamma)}{b_R}\right) = \frac{\mathrm{CTR}_{R,o} - \phi_{R,o}(0)}{b_R\,\rho_R\gamma} < 0,\end{aligned} \tag{C.17}$$

The sign is negative because $\phi_{R,o}(n)$ is largest at the top rank, where m_R is largest, so the average $\mathrm{CTR}_{R,o}$ lies strictly below $\phi_{R,o}(0)$. Hence a fall in ω raises $\mathrm{CTR}_{R,o}$, and the fact that b_R rises is (C.12). The argument uses only $g > 0$. □

C.1.5 Proof of Proposition 4

While the budget lasts, the paid click probability (5.9), $\phi_{R,p}(n) = \bar{G}(\omega + \kappa - q_p - m_R(n))$, is strictly decreasing in n: since $m_R'(n) = -\rho_R\gamma < 0$, the argument increases in n and $\bar{G}$ is decreasing. Paid clicking is therefore highest at the top, and the budget is spent early. Only a share $1-\pi$ of items is paid, so expected cumulative paid clicks are $P(n) = (1-\pi)\int_0^n \phi_{R,p}\,dn$, and the saturation rank n_B solves $P(n_B) = B$.

Part (i): the budget binds. Suppose $n_B \le b_R$. No paid clicks occur beyond n_B, so expected paid clicks over the session equal the budget,

$$(1-\pi)\int_0^{b_R} \phi_{R,p}(n)\,dn = P(n_B) = B, \qquad \text{i.e.} \qquad \int_0^{b_R} \phi_{R,p}\,dn = \frac{B}{1-\pi}. \tag{C.18}$$

The paid click-through rate is then

$$\mathrm{CTR}_{R,p} = \frac{1}{b_R}\int_0^{b_R} \phi_{R,p}\,dn = \frac{B}{(1-\pi)\,b_R}. \tag{C.19}$$

Only b_R depends on ω here: B and π are fixed, and the integral is pinned at $B/(1-\pi)$ by the binding budget. Differentiating (C.19) with respect to ω by the chain rule and substituting

$\partial b_R/\partial\omega = -1/(\rho_R\gamma)$ from (C.12) gives

$$\frac{d\,\mathrm{CTR}_{R,p}}{d\omega} = \frac{B}{1-\pi}\,\frac{d}{d\omega}\Big(\frac{1}{b_R}\Big) = -\frac{B}{(1-\pi)\,b_R^2}\,\frac{\partial b_R}{\partial\omega} = -\frac{B}{(1-\pi)\,b_R^2}\left(-\frac{1}{\rho_R\gamma}\right) = \frac{B}{(1-\pi)\,b_R^2\,\rho_R\gamma} > 0. \tag{C.20}$$

Hence a fall in ω raises b_R and lowers $\mathrm{CTR}_{R,p}$: a fixed stock of paid clicks is spread over a longer session. This proves (5.13). The binding condition is preserved under the experiment, because a lower ω raises b_R and, by raising $\phi_{R,p}$ pointwise through the lower paid bar, weakly lowers n_B, so $n_B \le b_R$ continues to hold.

Part (ii): the budget never binds. Suppose $n_B > b_R$ both before and after the change. Unlike in Part (i), this case is not self-preserving: a sufficiently large fall in ω raises b_R and lowers n_B, so the statement is a local one, valid as long as the budget stays slack. Throughout this region the user never exhausts her budget within the session, and $\phi_{R,p}$ follows the plain decreasing schedule (5.9). The argument of $\phi_{R,p}$ then depends on ω and on n exactly as the argument of $\phi_{R,o}$ does, with partial derivatives 1 and $\rho_R\gamma$, so the proportionality (C.14) holds with $\phi_{R,p}$ in place of $\phi_{R,o}$. Repeating the steps of the proof of Proposition 3 with $\phi_{R,p}$ gives

$$\frac{d\,\mathrm{CTR}_{R,p}}{d\omega} = \frac{\mathrm{CTR}_{R,p} - \phi_{R,p}(0)}{b_R\,\rho_R\gamma} < 0, \tag{C.21}$$

which is negative because $\phi_{R,p}(n)$ is largest at the top rank, so the average $\mathrm{CTR}_{R,p}$ lies strictly below $\phi_{R,p}(0)$. This proves (5.14): a fall in ω raises the paid click-through rate, and paid content tracks ordinary content. □