Platform Behavior under Market Shocks:
A Simulation Framework and Reinforcement-Learning Based Study*

Xintong Wang[1], Gary Qiurui Ma[1], Alon Eden[1], Clara Li[1], Alexander Trott[2], Stephan Zheng[2], and David C. Parkes[1]

[1]Harvard University
[1]{xintongw,qiurui_ma,aloneden}@seas.harvard.edu, clarali@college.harvard.edu,
parkes@eecs.harvard.edu
[2]Salesforce Research
[2]{atrott,stephan.zheng}@salesforce.com

Abstract

We study the behavior of an economic platform (e.g., Amazon, Uber Eats, Instacart) under shocks, such as COVID-19 lockdowns, and the effect of different regulation considerations imposed on a platform. To this end, we develop a multi-agent Gym environment of a platform economy in a dynamic, multi-period setting, with the possible occurrence of economic shocks. Buyers and sellers are modeled as economically-motivated agents, choosing whether or not to pay corresponding fees to use the platform. We formulate the platform's problem as a *partially observable Markov decision process*, and use deep reinforcement learning to model its fee setting and matching behavior. We consider two major types of regulation frameworks: (1) taxation policies and (2) platform fee restrictions, and offer extensive simulated experiments to characterize regulatory tradeoffs under optimal platform responses. Our results show that while many interventions are ineffective with a sophisticated platform actor, we identify a particular kind of regulation—fixing fees to optimal, pre-shock fees while still allowing a platform to choose how to match buyer demands to sellers—as promoting the efficiency, seller diversity, and resilience of the overall economic system.

1 Introduction

Market-driven platforms, such as Amazon, DoorDash, Uber, and TaskRabbit, play an increasingly important role in today's economy, bringing together parties to facilitate trade and presenting new ways to create value. First, they reduce *search cost* by introducing potential matches that were not known before, and second, they reduce *fulfillment cost* by taking care of service or product delivery, thus reducing the effort made to complete transactions. The importance of the platform-based economy became even more stark during the COVID-19 pandemic, and especially for the restaurant industry. Stay-at-home orders, together with the closure of dine-in channels and caution in regard to visiting brick-and-mortar businesses, tremendously increased the fulfillment cost of consumers

transacting in physical locations. This promoted the role of platforms, for example leading an increasing number of users and restaurants to adopt food-delivery platforms. One study of Uber Eats from February through May 2020 showed a surge in both demand and supply after the shelter-in-place guidance was issued in several states in the U.S. [Raj et al., 2021].

At the same time, this new prominence has provided platforms with increased market power. As a demonstration, restaurant commission fees are being set by some platforms to as high as 30% per order, leaving traditional restaurants, many of whom may no longer have dine-in revenue, with low or even negative margins [Ahuja et al., 2021]. In December 2020, the National Restaurant Association reported that more than 110,000 U.S. restaurants—one in six—have permanently closed down since the start of pandemic [National Restaurant Association, 2020]. To support restaurants, many states such as New York and California have imposed commission caps, and yet platforms have responded with countermeasures. As an example, the day after Jersey City enforced a 10% cap on fees, Uber Eats added a $3 delivery fee to customers and reduced the delivery radius for restaurants.[1] This speaks to the complexity of the ecosystem, and regulatory policies on third-party delivery platforms have been continuously proposed and debated as the pandemic calms down.

In this work, we use the methods of AI to study a platform-based economy under market shocks through a multi-agent simulation framework, with this as a first step towards reproducing phenomena observed in the real-world platform economy and as a tool for conducting counterfactual analysis to understand platform behavior in response to different regulations. We develop a multi-agent Gym environment to capture major aspects of a platform economy in a multi-period setting, with key modeling choices based on the existing economic literature (e.g., epoch-based decision making, user behavior inertia, and the discrete-logit choice model). Our model captures a full cycle of market shock, designed to represent the pre-, during, and post-crisis periods.

We formulate the platform's problem as a *partially observable Markov decision process* (POMDP), with both commonly observable components (e.g., shocks) and private elements (e.g., buyers' knowledge about sellers and transactions made off the platform), and model the platform as a rational agent that uses *reinforcement learning* (RL) to set fees and match buyer queries to sellers. Buyers and sellers decide whether or not to join the platform, considering both fees and their search and fulfillment costs; on-platform buyers further decide their preferred channel to make transactions, i.e., either with a recommended platform seller or with a known seller off the platform. We conduct extensive simulated experiments to explore a range of settings, these differing in market structures (i.e., locations of buyers and sellers in the product and preference space), the knowledge levels of buyers about sellers, and the cost of off-platform fulfillment. We demonstrate how to successfully use RL to model the optimal behavior of a platform under different regulatory considerations, and study the effect of a platform on the efficiency and resilience of the overall economic system.

In the absence of any regulation, we find that a revenue-maximizing platform, even while helping to facilitate trades during the market shock, tends to leverage its increased market power during the shock to raise fees and extract surplus from buyers and sellers, leading to seller shutdowns and lower post-shock economic welfare when compared with the pre-shock economic welfare.

As a first kind of regulatory intervention, we consider a class of *taxation policies* that impose tax rates on different categories of profit (e.g., revenue made from user subscriptions or revenue made from transaction referrals), and study the platform response and its effect on the economy. We further study regulations that *cap platform fees* in a particular way. Our first main result shows that either taxation or capping to some subset of fees simply leads the platform to transfer incremental

---

[1]https://www.protocol.com/delivery-commission-caps-uber-eats-grubhub

costs to users by adjusting other fees, demonstrating the power of an autonomous, strategic platform agent that is represented by the learned RL policies in modeling rational behavior. On the other hand, we show that capping all fees (i.e., not just a subset) has a moderately positive effect on protecting sellers from bankruptcy and promoting a resilient ecosystem. In practice, this requires detailed domain knowledge on the part of a regulator in regard to how to effectively set these caps. Another limitation is that, for reasons of simulation tractability and interpretability, we only study this intervention in the presence of a platform that follows a myopically optimal matching policy.

In this light, our second main result is especially interesting, as in these settings, we study a platform that is required to keep the same fee structure that it chooses to use in the absence of shocks but has the flexibility in choosing how to match buyer demands to sellers. Thus, it requires no special knowledge on the part of a regulator and still gives the platform full flexibility in regard to parts of its behavior that are usually proprietary and not easily regulatable. We show that a revenue-maximizing platform learns, under this intervention, to use matching to retain a more diverse set of sellers on platform, so as to increase its long-term revenue from user registrations, while also helping to promote the efficiency, resilience, and seller diversity of the overall economy.

We aim that the present framework, which combines a multi-agent Gym environment with the use of RL to derive optimal platform responses, provides a tool to use in understanding regulations and platform economies, complementing economic theory that often becomes analytically intractable in complex and dynamic environments and pure data-driven approaches that cannot answer questions about changing market and agent designs. We return to a critical discussion of the opportunities and outstanding challenges with this kind of AI-based approach at the end of the paper.

1.1   Related Work

Platform Models.   There is an extensive economic literature that focuses on how to establish network effects through platform fee-setting or subsidizing one side of the market under various forms of platform competition (e.g., single- vs. multi-homing) [Caillaud and Jullien, 2003; Rochet and Tirole, 2003; Armstrong, 2006]. Caillaud and Jullien [2003] study competition among platforms that act as matchmakers for homogeneous buyers and sellers and use both registration and transaction fees. In a two-stage model where platforms first set fees and then users simultaneously choose which platforms to register, the authors show that dominant firms are better off charging transactions rather than registrations and that competition is more intense when platforms cannot deter multi-homing. Rochet and Tirole [2003] generalize the setting to heterogeneous users and analyze a model in the context of credit card market with charges imposed only on a per-transaction basis. They show that the volume of transactions and the profit of a platform depend not only on the total fees charged to the parties to the transaction, but also on its decomposition. Armstrong [2006] further investigates such fee-setting behavior by varying the magnitude of *cross-group externalities*, through considering a monopoly platform, a model of competing platforms with single-homing agents, and a model of "competitive bottlenecks" with one group of users joining all platforms. To facilitate equilibrium analysis, these models often rely on simplified assumptions for tractability, e.g., restricting to a single round of platform fee-setting and agent subscribing, adopting a fixed platform matching policy, and assuming homogeneous non-platform actors.

We extend economic models of the above pioneering articles to multi-period, dynamic settings for studying platform behavior under market shocks, where (1) a shock may occur to change agents' price elasticity of demand in using the platform, (2) agents can flexibly join or quit the platform in response to shocks and fees, and (3) transactions can be completed via the platform or in the

brick-and-mortar traditional market. Many of these features are essential to our problem at hand, and may not be removed or easily stylized for tractability. For such reasons, we consider a platform that uses RL to set fees and adjust matching strategies dynamically.

Besides related literature that characterizes platform behavior, there are also works on complexity results in regard to setting optimal fees [Papadimitriou et al., 2016], optimally matching buy queries to sellers [Mladenov et al., 2020], and designing a platform for each agent state [Papadimitriou et al., 2021]. Papadimitriou et al. [2016] show that setting optimal fees by a revenue-maximizing mechanism is NP-hard, even for a market with one buyer and two periods. Optimally matching buy queries to sellers has also been proved computationally hard for a single-period setting with all non-platform actors fixed and no platform fees [Mladenov et al., 2020]. In a different setting, Papadimitriou et al. [2021] explore the design of a platform for each of the agent's activities or states, under which the agent's transition probabilities of the Markov chain are affected and the designer aims to maximize engagement. It is proven that the designer's optimization problem is NP-hard to approximate within any finite ratio. Our setting differs in that we focus on a market-based platform with network effects among agents (buyers and sellers).

Existing literature on the role and behavior of economic platforms under market shocks is fairly limited. Empirical studies have been conducted to characterize the impact of COVID-19 on the supply and demand of food-delivery platforms [Raj et al., 2021], the effect of commission fee caps on on-demand services [Li and Wang, 2021], as well as the extent to which pandemic has persistently changed consumer's purchasing behavior, even after the shock calms down due to habit formation around delivery [Oblander and McCarthy, 2022]. To the best of our knowledge, we are not aware of any theoretical or simulation studies that model or investigate the strategic behavior of platform in response to market shocks and various forms of regulations.

Using Reinforcement Learning for Economic System Design. There has been considerable recent interest in making use of methods of AI, and especially RL, for understanding the design of complex economic systems, though we are not aware of previous work on the use of RL or AI for the modeling of market-based platforms. Existing work on the use of RL for economic system design includes the problem of dynamically setting reserve prices in auctions [Shen et al., 2020], learning how to sell user impressions to advertisers [Tang, 2017], the design of tax policies in multi-period economies [Zheng et al., 2022], optimizing user satisfaction for multi-stage recommender systems [Chen et al., 2019; Zhan et al., 2021], and the design of sequential price mechanisms [Brero et al., 2021].

Many of these works make use of agent-based simulations to model actors in the economic system and conduct counterfactual analysis through the dynamic interactions of agents. For example, Zheng et al. [2022] use RL to model a social planner who designs income taxes in multi-period, simulated spatial economies. Similar to ours is the work of Zhan et al. [2021], which is built on the *RecSim environment* [Ie et al., 2019] and adopts RL to optimize the long-term social welfare of both users and content providers in a dynamic recommender system. Besides the presence of economic shocks, our setting differs in the use of platform fees, the existence of an alternate sales channel (i.e., off-platform transactions), and the implication that agents can choose to join or quit the platform.

2   Motivating Example: A Simple Platform Economy

Before introducing the full model, we use a simple economic system under market shocks to illustrate that a revenue-maximizing platform with typical myopic matching policy may be suboptimal from
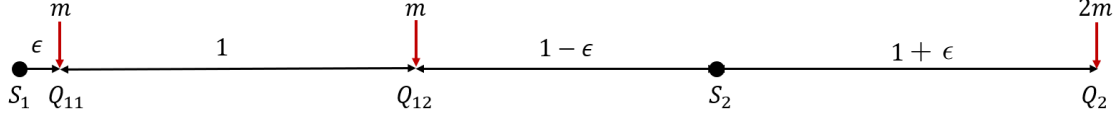
Figure 1: A simple platform economy with two buyers and two sellers located in a one-dimensional latent structure. Buyer 1 has m queries at each of the locations Q_{11} and Q_{12} and knows Seller 2. Buyer 2 has $2m$ queries at location Q_2 and knows both sellers.

the viewpoint of overall economic efficiency, while a platform that (i) uses more sophisticated matching of buyer demands to sellers or (ii) is surplus-aware in optimizing for a combination of its own revenue and the on-platform buyer and seller surplus may help to improve market outcomes.

We model an economy with two buyers and two sellers, where the sellers differ in the kinds of product they offer and the buyers differ in the kinds of product they are interested in. In particular, we represent the features of sellers and buyer queries in a one-dimensional *latent space* on the real line from $[-2, 1 + \epsilon]$, treating S_2 as the origin (see Figure 1). Each buyer generates demand for different kinds of products through submitting a *query*, which is a statement of interest for a product with a particular feature. When matched with a seller s, a query q gives the buyer a *matching utility* of $u(q, s)$, with this value being inversely proportional to the distance between q and s to reflect matching quality. Each seller is always willing to complete transactions with buyers off-platform, but this is only possible with a buyer who has knowledge of the seller. In this example, we assume that Buyer 1 only knows Seller 2, while Buyer 2 knows both Seller 1 and Seller 2. The effect is that Buyer 1 can only transact with Seller 1 if both of them are on the platform.

Buyers and sellers make decisions at the start of each *epoch* (e.g., a month) about whether or not to join the platform, incurring an associated registration fee in the case of joining. Within an epoch, we model each buyer as issuing $2m$ queries, for $m > 0$, with these queries distributed as shown in Figure 1. Buyer 1 issues m queries at location Q_{11} (with a distance of ϵ from Seller 1 and $2 - \epsilon$ from Seller 2) and m queries at location Q_{12} (with a distance of $1 + \epsilon$ from Seller 1 and $1 - \epsilon$ from Seller 2). Buyer 2 issues $2m$ queries at location Q_2 (with a distance of $3 + \epsilon$ from Seller 1 and $1 + \epsilon$ from Seller 2).

The platform sets a *per-epoch, buyer registration fee*, $P_{\mathcal{B}} \geq 0$, and a *per-epoch, seller registration fee*, $P_{\mathcal{S}}$, and recommends an on-platform seller for each query from a buyer that is on the platform.[2] A buyer can choose among the options of (i) transacting with a known seller off-platform, (ii) if on platform, transacting with the platform-recommended seller, and (3) not to transact with any seller.

Besides the benefit of being matched to a new seller, the platform also reduces the fulfillment cost when a transaction is completed on platform. We model this through the *world transaction friction, $\mu > 0$*, capturing the effort required by a buyer to transact off platform, such as visiting a restaurant. Therefore, the surplus for a buyer with query q that transacts with seller s is $r(q, s, \mathcal{I}) = u(q, s) - \mu \cdot \mathcal{I}_w$, where $\mathcal{I}_w \in \{0, 1\}$ indicates whether a transaction is completed off-platform ($\mathcal{I}_w = 1$) or via the platform ($\mathcal{I}_w = 0$). Each seller is modeled as receiving a surplus of 1 from each completed transaction, regardless of the matched buyer, and has a *fixed cost* of m per epoch. A seller whose cost is not covered by surplus received from serving buyer queries will shut down and can no longer be matched to future queries.

We study this simple economy in an environment where the matching utility function is $u(q, s) = 2 - |q - s|$, with a high and constant world transaction friction of $\mu = 1$. This can be

---

[2]In the full model, we further include a per-transaction seller referral rate that can be set and charged by a platform.

considered to model the COVID pandemic lockdowns.[3] Beyond setting fees, the platform also decides how it will match any queries executed on platform with sellers. One typical choice is myopic matching, where the platform always matches a query q to the seller who gives the highest matching utility, i.e., $s_q^* = \text{argmax}_{s \in \mathcal{S}} u(q, s)$, considering the set $\mathcal{S}$ of on-platform sellers. More generally, in the case of having both sellers on the platform, the platform can decide which fraction of queries in which locations to match to each seller.

In studying the simple economy, we compute the *Stackelberg equilibrium* where the platform makes decisions, in regard to registration fees and perhaps also matching policies, for each epoch. Buyers and sellers then best respond and form an equilibrium, where the decision of a buyer or seller is whether or not to join the platform (and subsequently how to transact on the basis of off-platform and on-platform opportunities). The policy of a platform forms a Stackelberg equilibrium when the platform cannot increase its objective value by choosing to commit to another policy.

In what follows, we consider the baseline of the absence of a platform, as well as three different cases, which vary in regard to the matching policy of the platform and whether or not the surplus to buyers and sellers is part of the platform's objective (in addition to its own revenue). We provide a high-level description of each case and summarize key results in Table 1. Complete proofs of the equilibrium structure are deferred to Appendix A.

- **Case 1: No platform.** In the absence of a platform, the market is very inefficient under shocks: transactions happened only between the Q_{12}-type queries from Buyer 2 and Seller 2. Seller 1 gets no transaction and shuts down. The per-epoch welfare of the system (i.e., the sum of buyer and seller surplus) is ϵm.

- **Case 2: A revenue-maximizing platform with myopic matching.** This case considers a platform making an equilibrium choice about registration fees, while fixing the matching policy to be myopic optimal. In effect, the platform chooses which of the buyers and sellers to bring to the platform, setting fees to maximize revenue while keeping these agents on the platform. In the Stackelberg equilibrium, the fees are set to $P_\mathcal{B} = (1 + \epsilon)m$ and $P_\mathcal{S} = 3m$, and the two buyers and Seller 2 join the platform, with all queries directed to Seller 2. The effect is that Buyer 1 gets a per-epoch surplus of ϵm ($= r(Q_{11}, S_2, 0) \cdot m + r(Q_{12}, S_2, 0) \cdot m - P_\mathcal{B}$), and Buyer 2 gets a per-epoch surplus of $(1 - 3\epsilon)m$ ($= r(Q_2, S_2, 0) \cdot 2m - P_\mathcal{B}$). Since $P_\mathcal{S}$ is set to $3m$ and each seller has a fixed cost of m, Seller 1 stays off-platform and goes bankrupt, whereas Seller 2 is matched to $4m$ queries but receives zero surplus. The platform's per-epoch revenue is $(5 + 2\epsilon)m$ ($= 2P_\mathcal{B} + P_\mathcal{S}$) and the per-epoch economic welfare (i.e., the sum of platform revenue and buyer and seller surplus) is $6m$, which is much larger than the case without a platform.

- **Case 3: A revenue-maximizing platform that rationally sets fees and matching policy.** This case allows the platform to both set fees and choose to deviate from a myopic matching policy, for example directing more queries than just the Q_{11} queries from Buyer 1 to Seller 1. In the Stackelberg equilibrium, the platform brings both sellers and both buyers to join the platform, achieving this by setting $P_\mathcal{B} = (2 - 2\epsilon)m, P_\mathcal{S} = m$, and matching all queries from Buyer 1 to Seller 1 and all queries from Buyer 2 to Seller 2. Buyer 1 gets a per-epoch surplus of m ($= r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_1, 0) \cdot m - P_\mathcal{B}$), and Buyer 2 gets a per-epoch surplus of 0 ($= r(Q_2, S_2, 0) \cdot 2m - P_\mathcal{B}$). With $P_\mathcal{S}$ set to m, each seller is matched to $2m$ queries and has zero

6

		No platf.	Revenue-maximizing	Rational matching	Surplus-aware (α)	Ideal
SystemWelfare		ϵm	$6m$	$(7-4\epsilon)m$	$(7-2\epsilon)m$	$(7-2\epsilon)m$
B1	Surplus	ϵm	ϵm	m	$(1+2\epsilon)m$	$3m$
	Matches	$Q_{12} \to S_2$	$Q_{11}, Q_{12} \to S_2$	$Q_{11}, Q_{12} \to S_1$	$Q_{11} \to S_1, Q_{12} \to S_2$	$Q_{11} \to S_1, Q_{12} \to S_2$
B2	Surplus	0	$(1-3\epsilon)m$	0	0	$(2-2\epsilon)m$
	Matches	None	$Q_2 \to S_2$	$Q_2 \to S_2$	$Q_2 \to S_2$	$Q_2 \to S_2$
S1	Surplus	0	0	0	0	0
	State	Bankrupt	Bankrupt	$2m$ queries	m queries	m queries
S2	Surplus	0	0	0	$2m$	$2m$
	State	m queries	$4m$ queries	$2m$ queries	$3m$ queries	$3m$ queries
Platf.	Revenue	0	$(5+2\epsilon)m$	$(6-4\epsilon)m$	$(4-4\epsilon)m$	0
	Fees	N/A	$P_{\mathcal{B}} = (1+\epsilon)m, P_{\mathcal{S}} = 3m$	$P_{\mathcal{B}} = (2-2\epsilon)m, P_{\mathcal{S}} = m$	$P_{\mathcal{B}} = (2-2\epsilon)m, P_{\mathcal{S}} = 0$	$P_{\mathcal{B}} = 0, P_{\mathcal{S}} = 0$

Table 1: Total system welfare, each buyer or seller agent surplus, and platform revenue achieved in the Stackelberg equilibrium of the simple economy with no platform (Case 1), a revenue-maximizing platform (Case 2), a rational-matching platform (Case 3), a surplus-aware platform (Case 4), and an ideal world where each buyer knows all sellers and there is no fee or transaction friction. The second row of the entry for each agent describes buyer query matching, seller agent states, or fees set by the platform.

surplus. The platform revenue is $(6-4\epsilon)m$ $(= 2P_{\mathcal{B}} + 2P_{\mathcal{S}})$, and the total welfare of the system is $(7-4\epsilon)m$. In this case, the platform is able to keep both sellers from going bankrupt through the use of non-myopic matching, with the effect of improving its own revenue as well the system welfare, compared with Case 2 of a revenue-maximizing platform that uses myopic matching.

- **Case 4: A surplus-aware platform with myopic matching.** In this case, we consider a platform that is motivated to optimize for a combination of its own revenue and the surplus of on-platform buyers and sellers, i.e., rev $+ \alpha \cdot$ surplus, where $\alpha < 1$ is the weight placed on user surplus in the objective. This can be motivated either as a simple model of the effect of regulation or the motivations of a more socially conscious platform. When $\alpha > 0.5$, the platform will choose fees $P_{\mathcal{B}} = (2-2\epsilon)m$ and $P_{\mathcal{S}} = 0$ and all agents join the platform with Q_{11}-type queries from Buyer 1 matched to Seller 1 and the rest of queries to Seller 2. Under such fees, Buyer 1 has a per-epoch surplus of $(1+2\epsilon)m$ $(= r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_2, 0) \cdot m - P_{\mathcal{B}})$, and Buyer 2 has a per-epoch surplus of 0 $(= r(Q_2, S_2, 0) \cdot 2m - P_{\mathcal{B}})$. Seller 1 gets m queries and has zero surplus whereas Seller 2 gets $3m$ queries and a per-epoch surplus of $2m$. The per-epoch platform revenue is $(4-4\epsilon)m$ $(= 2P_{\mathcal{B}} + 2P_{\mathcal{S}})$, and the per-epoch overall welfare is $(7-2\epsilon)m$. We note that, in this case, all queries are matched to their ideal seller, and the system achieves the maximum possible welfare.

This simple economy is helpful in demonstrating the potential effectiveness of adopting non-myopic matching and imposing market regulation (modeled simplistically in this case in the form of adding a surplus term to the platform's objective) in improving the efficiency of the ecosystem and also avoiding seller bankruptcy. Next, we extend to a more general multi-agent platform model, with the addition of a per-transaction referral rate, complex market structures (i.e., latent locations of buyers and sellers) and buyers' knowledge about sellers, agent behavior inertia, changing world transaction friction to represent full cycles of market shock, and different forms of regulations. Given the complexity and intractability of the dynamic environment, we study a platform that uses RL to respectively learn its fee-setting policy and matching policy.

3 A Multi-Agent Platform Model

This section formulates the dynamic, multi-agent model, with which we simulate a platform-based economy. We defer details of learning the platform's fee-setting and matching policy to Section 4.

3.1 Market Environment and Agent Dynamics

Consider a market populated with a set of heterogeneous buyers $\mathcal{B}$, a set of heterogeneous sellers $\mathcal{S}$, and a single platform agent p. We follow the embedding-based representations in recommender systems [Salakhutdinov and Mnih, 2007] and use a two-dimensional *latent space* to represent each buyer or seller agent, i.e., $v_b, v_s \in \mathcal{V} \subseteq [0,1]^2$. The first dimension of an agent's *location*, denoted v^0, describes product features (e.g., in the case of food, the type of cuisine, Italian, Mediterranean or Japanese, the taste of dishes, spicy or not), and the second dimension, denoted v^1, the normalized price level (e.g., \$ ... \$\$\$\$). A seller s offers food at a price v_s^1, of which an ω_s fraction is the per-unit production cost. Each buyer b knows a subset of sellers $\mathcal{S}_b \subseteq \mathcal{S}$ and may transact with $s \in \mathcal{S}_b$ without using a platform. Buyers who use the platform are also introduced to additional sellers.

We formulate an *epoch-based* decision problem for agents, similar to the recommendation problem in Mladenov et al. [2020]. Within an episode, there are multiple epochs, indexed k, and each epoch has a fixed length of T time steps (e.g., a month of 30 days). We consider a macro-level *world transaction friction*, denoted $\mu_k > 0$, which varies by epoch and represents the cost of buyers completing a transaction off the platform (see Equation (1)). We model *shocks*, corresponding to changes in this friction. For example, during a pandemic, transaction frictions for the food-service industry were extremely high, due to fears of sharing indoor spaces and the absence of dine-in options. Below we describe agent dynamics within an epoch.

At the start of an epoch k. The platform sets fees, including the buyer and seller *subscription fees*, denoted $P_{\mathcal{B},k} \geq 0$ and $P_{\mathcal{S},k} \geq 0$ respectively, and a per-transaction seller *referral rate* $P_{R,k} \in [0,1]$, the fraction of prices as transaction fee paid by the seller to the platform. We discuss the platform policy in setting fees in Section 4.1.

Buyers and sellers observe the fees and the world transaction friction μ_k, and decide whether to pay the subscription fee to use the platform for epoch k. We denote the sets of subscribed buyers and sellers in epoch k as $\mathcal{B}_k$ and $\mathcal{S}_k$. We assume that the platform knows the locations of buyer queries and on-platform sellers in the latent space. This reflects that platforms tend to have good data on the market-relevant properties of sellers and that the combination of a search interface and historical buyer information gives good information on the current demand context of a buyer. Each buyer has a *per-epoch budget constraint*, denoted $\psi_b > 0$, which is linearly proportional to the buyer's price preference v_b^1. This controls the number of transactions that a buyer will undertake in a given epoch. We discuss how buyers and sellers decide whether or not to join the platform for the upcoming epoch in Section 3.4.

Beyond fees, the platform matches queries from platform buyers to platform sellers, with a matching policy learned through RL (see Section 4.2).

For each time step t in an epoch k. We follow the sequence of "query, match, and transact":

(i) *Query*: a buyer $b \in \mathcal{B}$ is randomly selected to submit a query according to their taste and price preferences (e.g., ordering \$\$\$\$ sushi or \$\$ pizza), $q_{b,t} \sim \mathcal{N}(v_b, \sigma_b^2)$, where σ_b specifies the query variance of b around their latent location.
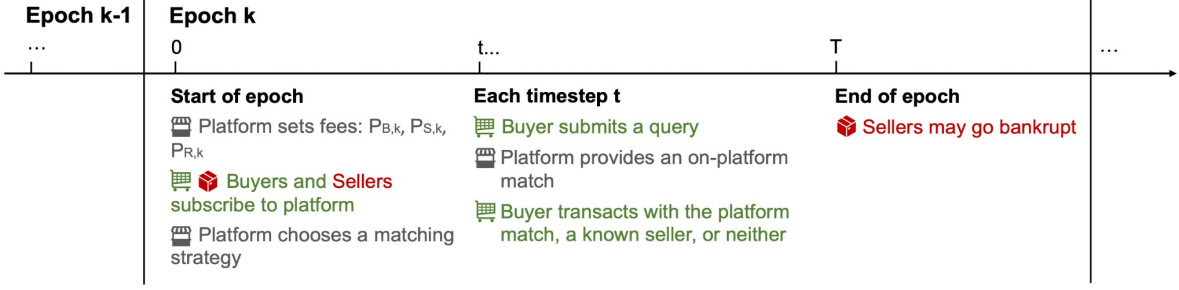
Figure 2: An illustration of the epoch-based decision making for buyers, sellers, and the platform.

(ii) *Match*: for an on-platform buyer b, the platform observes $q_{b,t}$ and matches it to an on-platform seller, denoted $s_{p,t} \in \mathcal{S}_k$.

(iii) *Transact*: the buyer b can pick a seller $s \in \{s_{p,t}\} \bigcup \mathcal{S}_b$ if b is on-platform, and $s \in \mathcal{S}_b$ otherwise. A buyer can also choose not to transact if they are not satisfied with any seller, i.e., the matching surplus is negative (details in Section 3.2).

We refer to a transaction that is matched via the platform as *a platform transaction*, and otherwise *a world transaction*. We note that an on-platform buyer may choose to complete a world transaction if the platform suggests an inferior match. In addition, a buyer may have a world transaction with a seller known to the buyer, whether or not the seller is on the platform, and by such, a seller on the platform is also able to complete off-platform transactions. For each world transaction, the buyer suffers a fulfillment cost of μ_k, while for each platform transaction, the seller pays a *referral fee*, which is a fraction $P_{R,k}$ of the seller's price.

At the end of an epoch k. Buyers and sellers evaluate their surplus from transactions and fees paid in order to guide future subscription decisions. The platform evaluates the revenue made through subscription and referral fees and can make adjustments to fees or its matching policy (see Section 4). Each seller has a *shutdown threshold*, $\lambda_s \in \mathbb{N}_{>0}$, and will go bankrupt if they do not obtain enough number of transactions or surplus to meet the threshold. Once bankrupt, a seller is unable to engage in transactions in future epochs.

3.2 Transaction-Level Decisions

A buyer with query q who transacts with seller s receives a *matching utility*, $u_{\mathcal{B}}(q, s)$, reflecting the matching quality. A buyer with a choice of transactions prefers the one that maximizes immediate *matching surplus*, defined as matching utility minus any transaction friction (i.e., the buyer is myopic without considering how their current choice will affect sellers and the availability of future options).

Choice in the world. A buyer b with query $q_{b,t}$ can choose from known sellers whose prices (denoted by v_s^1 for seller s) are within their epoch-budget left at t, denoted $\psi_{b,t}$, i.e., $\mathcal{S}_{b,t} := \{s \in \mathcal{S}_b : v_s^1 \leq \psi_{b,t}\}$. For $\mathcal{S}_{b,t} \neq \emptyset$, the best world choice is $s_w^* := \operatorname{argmax}_{s \in \mathcal{S}_{b,t}} u_{\mathcal{B}}(q_{b,t}, s)$. Since the world transaction friction μ_k could be high enough to prevent a buyer from transacting, we let $s_{b,t}^w$ be s_w^* if $u_{\mathcal{B}}(q_{b,t}, s_w^*) > \mu_k$, and ϕ otherwise, to denote no preferred world seller. Writing $u_{\mathcal{B}}(q_{b,t}, \phi) = 0$,

9

the *world surplus* to buyer b at time t is

$$u_{b,t}^w = \max(u_\mathcal{B}(q_{b,t}, s_{b,t}^w) - \mu_k, 0). \tag{1}$$

Choice on the platform. For an on-platform buyer b, the *platform surplus*, $u_{b,t}^p$, at time t is $u_{b,t}^p = u_\mathcal{B}(q_{b,t}, s_{p,t})$ in the case that the platform-recommended seller, $s_{p,t}$, is within the remaining budget $\psi_{b,t}$, and 0 otherwise. For an off-platform buyer b, we set $u_{b,t}^p = 0$.

Overall choice. If no seller provides a positive surplus, the buyer will choose not to transact. Otherwise, a buyer b chooses $s_{b,t}^w$ if it is off-platform, and the most preferred of $s_{b,t}^w$ and $s_{p,t}$ when it is on-platform.[4] We write $s_{b,t}$ to denote the choice of the buyer at time t, and denote a buyer's query, off- and on-platform seller options, and the chosen seller for transaction as the 4-tuple $(q_{b,t}, s_{b,t}^w, s_{p,t}, s_{b,t})$, where $s_{p,t}$ is ϕ in case of an off-platform buyer.

Surplus from a transaction. The buyer surplus is $r_{b,t} = \max\{u_{b,t}^w, u_{b,t}^p\}$. We define the *world matching surplus* and the *platform matching surplus* respectively as

$$r_{b,t}^w = u_{b,t}^w \cdot \mathcal{I}_{b,t}^w \ \ \text{and} \ \ r_{b,t}^p = u_{b,t}^p \cdot (1 - \mathcal{I}_{b,t}^w),$$

where $\mathcal{I}_{b,t}^w$ is an indicator of whether buyer b transacted in the world or not at time t. A seller when chosen by a buyer cannot decline a transaction, and the surplus (i.e., net profit) of seller s is

$$r_{s,t} = \begin{cases} v_s^1(1 - \omega_s - P_{R,k}) & \text{for a platform transaction,} \\ v_s^1(1 - \omega_s) & \text{for a world transaction,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We denote $n_{s,k}^p$ the number of transactions completed by seller s via the platform during epoch k, and $n_{s,k}^w$ the number of transactions completed by the seller in the world.

3.3 Epoch Surplus and Platform Revenue

Buyer b's *epoch surplus*, $r_{b,k}$, is their total surplus from matching minus any subscription fee paid,

$$r_{b,k} = \sum_{t \in k} r_{b,t} - \mathcal{I}_{b,k}^p P_{\mathcal{B},k} = \underbrace{\sum_{t \in k} r_{b,t}^w}_{r_{b,k}^w} + \underbrace{\sum_{t \in k} r_{b,t}^p - \mathcal{I}_{b,k}^p P_{\mathcal{B},k}}_{r_{b,k}^p}, \tag{3}$$

where $\mathcal{I}_{b,k}^p \in \{0, 1\}$ indicates whether the the buyer is on or off the platform during epoch k. This surplus decomposes into surplus from world transactions $r_{b,k}^w$ and surplus from platform transactions $r_{b,k}^p$, and we define the *total buyer surplus generated by the platform* in epoch k as $r_{\mathcal{B},k}^p = \sum_{b \in \mathcal{B}} r_{b,k}^p$, and the *total buyer surplus from the world* in epoch k as $r_{\mathcal{B},k}^w = \sum_{b \in \mathcal{B}} r_{b,k}^w$.

Similarly, a seller s's *epoch surplus*, $r_{s,k}$, is their total net profit from transactions minus any fee,

$$r_{s,k} = \sum_{t \in k} r_{s,t} - \mathcal{I}_{s,k}^p P_{\mathcal{S},k} = \underbrace{n_{s,k}^w v_s^1(1 - \omega_s)}_{r_{s,k}^w} + \underbrace{n_{s,k}^p v_s^1(1 - \omega_s - P_{R,k}) - \mathcal{I}_{s,k}^p P_{\mathcal{S},k}}_{r_{s,k}^p}, \tag{4}$$

---

[4] We note that $s_{p,t}$ and $s_{b,t}^w$ may be the same seller, in which case the buyer will choose to transact via the platform since the world transaction friction is always positive.

10

where $\mathcal{I}_{s,k}^p \in \{0,1\}$ is an indicator to denote whether the the seller is on or off the platform during epoch k. We define the *total seller surplus from platform transactions* in epoch k as $r_{\mathcal{S},k}^p = \sum_{s \in \mathcal{S}} r_{s,k}^p$ and the *total seller surplus from world transactions* in epoch k as $r_{\mathcal{S},k}^w = \sum_{s \in \mathcal{S}} r_{s,k}^w$.

The *total platform revenue* in epoch k is the sum of the subscription and referral fees it charges,

$$r_{p,k} = \sum_{b \in \mathcal{B}} \mathcal{I}_{b,k}^p P_{\mathcal{B},k} + \sum_{s \in \mathcal{S}} \left(\mathcal{I}_{s,k}^p P_{\mathcal{S},k} + n_{s,k}^p v_s^1 P_{R,k} \right). \tag{5}$$

The *total welfare* of the economy in epoch k is the sum of all buyer and seller surplus and platform revenue, i.e., $\sum_{b \in \mathcal{B}} r_{b,k} + \sum_{s \in \mathcal{S}} r_{s,k} + r_{p,k}$.

3.4 Subscription-Level Decisions

At the start of each epoch, each buyer and seller "wakes up" with some probability to reevaluate their current state and decide whether or not to subscribe to the platform. This decision is utility-theoretic and depends on (1) estimating the effect of a subscription decision (i.e., the surplus from joining vs. operating off platform) and (2) an agent-specific inertia that captures the extent to which the agent becomes affiliated with a particular transaction channel. We provide in the sequel a high-level description as to how buyers and sellers make such decisions, and defer detailed mathematical expressions to Appendix B.

Estimating the effect of a subscription decision. Each agent, whether a buyer or seller, subscribes by comparing the estimated surplus on and off platform, under the newly proposed platform fees and the observable world transaction friction. We assume that agents do not coordinate among themselves and thus conduct such comparisons assuming a unilateral change, i.e., being the only one who wakes up and makes a different subscription decision, and that queries from the buyer agent itself as well as others remain the same as in the previous epoch. For buyers or sellers who were on platform in the past epoch, this estimate means to re-evaluate their transaction decisions under new fees and friction. For agents who were off platform in the past epoch, we assume that the platform can provide information (honestly) to facilitate this estimation. In practice such estimates could occur through a trial period offered by the platform or through the platform providing an estimate of costs and benefits based on recent history.

Agent-specific decision inertia. We incorporate *behavior inertia* in agent choice model to capture an agent's tendency to stick with their current state or decision. This kind of inertia has been empirically observed in platform adoption decisions post-pandemic [Oblander and McCarthy, 2022], along with many other settings, including choosing consumer packaged goods [Shum, 2004; Dubé et al., 2010] and selecting health and automobile insurance [Handel, 2013; Honka, 2014].

Following prior models [Dubé et al., 2009; MacKay and Remer, 2021; Farrell and Shapiro, 1988], we include inertia, denoted χ, as an additive term to an agent's surplus from the current decision. Specifically, we define inertia as a term with its value growing logarithmically in the number of consecutive epochs for which an agent has made the same choice (either on or off the platform), and "reset" this inertia upon a change to the subscription decision. Intuitively, this reflects the fact that the longer an agent has stick to a certain option, the less likely they will switch to alternative options. Based on this adjusted surplus, both buyer and seller agents decide whether or not to pay the fees and join the platform according to probabilities calculated from the standard *discrete-choice logit model* [Dubé et al., 2009; MacKay and Remer, 2021] (see Appendix B Equation (8)).

4 The Platform's Decision Model

In this section, we formulate the platform's problem as a POMDP [Kaelbling et al., 1998], with buyers' knowledge about sellers and transactions completed in the world as private information and thus not observable to the platform. The platform learns a fee-setting policy and a matching policy based on observations of the activities of on-platform buyers and sellers. In our current work, for reasons of computational tractability and interpretability, we choose to study platforms that either follow a typical, myopic optimal matching policy and learn to set fees or follow fixed, regulated fees and learn how to match. That said, one of our regulatory interventions is to fix platform fees to a certain regime (e.g., those adopted by a platform in an environment without shocks), studying how the platform subsequently chooses how to match.

4.1 Learning to Set Fees under Myopic Query Matching

We study the sequential decision making problem of a platform who learns to set fees while using *myopic query matching* (i.e., recommending the on-platform seller that yields the highest matching utility to the buyer). A POMDP [Kaelbling et al., 1998] can be formally described as a 6-tuple $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O})$, with the state space $\mathcal{X}$, action space $\mathcal{A}$, a Markovian state-action-state transition probability function $\mathcal{P}$, and a reward function $\mathcal{R}$. Instead of seeing the true state $x \in \mathcal{X}$, the agent receives an observation $o \in \Omega$, generated from the underlying state according to the probability distribution $o \sim \mathcal{O}(x)$. We describe each component of the POMDP and define the platform's fee-setting policy. Here, the platform is making a decision in regard to fees for the upcoming epoch k based on its experience from epoch $k-1$:

- The *state* $x_k \in \mathcal{X}$ at the start of epoch k (i.e., before the platform sets fees) is comprised of

    (i) buyer attributes: the latent location, epoch budget, query distribution, and knowledge of world sellers,

    (ii) seller attributes: the latent location, cost fraction, and shutdown threshold,

    (iii) agent subscription states: either on- or off-platform for the past epoch, $\mathcal{I}^p_{\mathcal{B},k-1}$ and $\mathcal{I}^p_{\mathcal{S},k-1}$,

    (iv) the agent inertia levels: $\chi_{b,k-1}$ and $\chi_{s,k-1}$,

    (v) a sequence of query, seller candidates, and buyer's choices in previous epoch: $Q_{k-1} = \{(q_{b,t}, s^p_{b,t}, s^w_{b,t}, s_{b,t})\}_{t \in k-1,\ b \in \mathcal{B}}$,

    (vi) the shutdown states for sellers: whether a seller has shut down at the end of epoch $k-1$, $\mathcal{I}_{\mathcal{S},k-1}$,

    (vii) the platform fees for the past epoch: $P_{\mathcal{B},k-1}, P_{\mathcal{S},k-1}, P_{R,k-1}$,

    (viii) the world transaction friction for the current epoch: μ_k.

- An *action* $a_k = (P_{\mathcal{B},k}, P_{\mathcal{S},k}, P_{R,k})$ defines the fees for the upcoming epoch k. We model a discrete action space $\mathcal{A}$ where fees take discrete values at integer multiples of a tick (or percentage) size.

- For the *state transition* $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$, agent attributes (e.g., latent locations, knowledge of sellers) remain the same across epochs. Buyers and (viable) sellers follow their choice model to subscribe to the platform (Section 3.4), leading to new subscription states and inertia levels. For each time step $t \in k$, we follow the "query, match, transact" dynamics (Section 3.1), which gives a full sequence of Q_k. Each viable seller may shut down based on the surplus in epoch k and their

shutdown threshold. Fees follow from the actions taken, and the world transaction friction evolves according to a defined Markov process. Altogether, this gives a new state $x_{k+1} \sim \mathcal{P}(x_k, a_k)$.

- A *reward* $r_k \sim \mathcal{R}(x_k, a_k)$ is provided to the platform at the end of epoch k, when agent subscription and transaction outcomes are available. The reward can be set to model different platform objectives, integrating considerations that come from regulation.

- The platform's *observation* $o_k \in \Omega$ consists of the world transaction friction μ_k, agent subscription states in the past epoch, and the sequence of queries generated by on-platform buyers, as well as their decisions on whether or not to transact via the platform, i.e., $Q_{k-1}^p := \{(q_{b,t}, s_{b,t}^p, \mathbb{I}\{s_{b,t} = s_{b,t}^p\})\}_{t \in k-1, \ b \in \mathcal{B}_{k-1}}$ (and not counterparties in off-platform transactions).

4.2 Learning to Match Queries under Fixed, Regulated Fees

In a second setting, we study the sequential decision making problem of a platform who learns how to match buyer queries with on-platform sellers when the platform fees are fixed, for example due to regulation. Typical myopic matching, as described above, favors the buyer side of the market, by directing a query to the buyer's utility-maximizing, on-platform seller. To complement this, we model matching strategies that can choose to benefit other parties in the economy, for example sellers or the platform itself. To facilitate interpretability, we define a *matching strategy* by two parameters: (1) a *matching utility threshold* $\eta \in [0, 1]$, that specifies the minimum utility that a recommended seller should provide to the buyer, as a fraction of utility from the myopically-optimal match, and (2) a *matching rule*, which directs how to pick a seller amongst those that meet this utility threshold. We consider two rules:

- *The seller-aware matching rule*: Among sellers who meet the utility threshold, match a query to the seller who has the lowest surplus on the platform so far during the epoch,[5] breaking ties in the buyer's favor.

- *The profit-driven matching rule*: Among sellers who meet the utility threshold, match a query to the seller who brings the largest revenue to the platform, breaking ties in the buyer's favor (given referral fees in our setting, this is the most expensive, on-platform seller).

As a special case, myopic query matching corresponds to setting the matching threshold $\eta = 1$ (along with either rule, since they have no choice as to how to further select a seller).

Intuitively, the seller-aware rule may be useful in promoting a more diverse set of sellers, by increasing sales to those who have been benefiting less from the platform, whereas the profit-driven rule is at the other end of the spectrum, aiming to maximize the platform's myopic transaction revenue without considering longer-term effects. We detail the implementation of a matching strategy, given the matching rule and a utility threshold in Appendix C Algorithm 1.

The goal of the platform is to learn a *matching policy* that chooses a matching strategy for *each epoch*—a utility threshold and a matching rule, based on an observation as to which buyers and sellers choose to subscribe for the *upcoming* epoch. To model this, we make several adjustments to the fee-setting POMDP in defining a *matching POMDP*, as in this case, the platform observation needs to be defined *after* agents make subscription decisions. We defer detailed adjustments to Appendix C.2.

---

[5] We assume for simplicity that the platform knows a seller's production cost, and thus can calculate its surplus from transactions. In practice, this idealized seller-aware rule could be modified and defined in terms of the number of platform transactions completed by a seller, or other inferred quantities about seller surplus.

4.3 Finding the Optimal Policy under Regulatory Interventions

Interactions between the platform and buyers and sellers can be considered as a *Stackelberg game*: the platform agent is the *leader*, choosing the fee or matching policies, and buyers and sellers are the *followers*, responding to platform policies. In our environment, the buyer and seller strategies are a fixed mapping from prior matching experience, platform fees, and world transaction friction to decisions in regard to joining or exiting the platform, and we can handle this Stackelberg structure by modeling the agents within the POMDP of the platform (i.e., as part of the transition model).

Depending on the setting, the platform learns a fee policy or a matching policy, denoted $\pi(a|o_k)$, to maximize its discounted cumulative reward across different episodes. An optimal policy in a POMDP requires the action to be taken depending on the entire history of observations, which we denote $h_k := \{o_0, a_0, ..., a_{k-1}, o_k\}$. Following the success of using representation learning to solve POMDPs [Heess et al., 2015; Wierstra et al., 2007; Hausknecht and Stone, 2015], we use neural networks to learn sufficient statistics of the history, denoted $T(h_k)$. Altogether, we use deep RL to learn the platform policy $\pi(a|o_k; \boldsymbol{\theta})$, specifically parameters $\boldsymbol{\theta}$ of a neural network that extract $T(h_k)$ and map to actions to maximize the platform objective,

$$\max_{\pi} \quad \mathbb{E}_{a \sim \pi, x \sim \mathcal{P}} \left[\sum_{k=0}^{K} \gamma^k r_k \right], \tag{6}$$

where $\gamma \in (0, 1)$ is the discount factor, $K = |\tau|$ is the total number of epochs in an episode, and r_k is based on Equation (5) and whose precise value can further depend on a regulatory structure, for example a taxation policy. We discuss the algorithm that we use to solve this POMDP and give additional implementation details in Section 5.

5 Platform Behavior under Shocks and Regulations

We study platform behavior and its effect on the economic system under the following regulatory interventions: (i) taxation policies that directly change the reward to the platform, (ii) fee caps that are enacted through restricting the platform's action space, and (iii) fee freezes that are studied along with a platform who can continue to change its matching policy. We first provide specifics on the configuration of the simulation settings for our experiments and present main experimental results.

5.1 Environment Settings of the Platform-Based Economy

Market structure and dynamics. We follow Section 3 in specifying a range of different market environments. We consider three types of *market structures*, corresponding to distinct latent locations of buyers and sellers:

- *Uniform:* Buyers and sellers are uniformly distributed, i.e., $v \sim U[0,1]^2$, representing an economy with diverse buyer interests and seller attributes.

- *Core-and-Niche:* There is a propensity for buyers and sellers to locate towards the center, i.e., $v \sim$ truncated Gaussian$(\mu, \sigma, 0, 1)$ with $\mu = [0.5, 0.4]$ and $\sigma = 0.2$, representing a "core" of agents around μ as well as "niche" agents located away from the market center. We choose the distribution to be slightly biased towards lower prices, representing the majority.
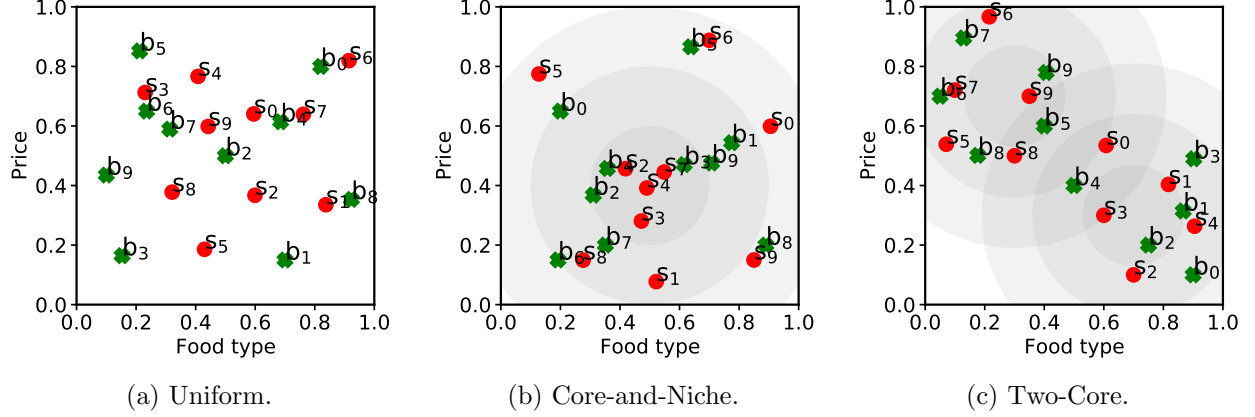
(a) Uniform.   (b) Core-and-Niche.   (c) Two-Core.

Figure 3: An illustration of latent locations of buyers and sellers for each market structure.

- *Two-Core:* One group $v \sim$ truncated Gaussian$(\mu_1, \sigma_1, 0, 1)$ with $\mu_1 = [0.7, 0.3]$ and $\sigma_1 = 0.17$, and another group $v \sim$ truncated Gaussian$(\mu_2, \sigma_2, 0, 1)$ with $\mu_2 = [0.3, 0.7]$ and $\sigma_2 = 0.17$. The first group of buyers and sellers is centered around relatively cheap options, and the second around more expensive ones.

Across all market structures, we consider environments with 10 buyers and 10 sellers, an episode that consists of $K = 12$ epochs, with each epoch containing $T = 100$ time steps. Figure 3 provides a visualization for each market structure. For each latent location, we also consider markets that vary in regard to the buyer's knowledge level about sellers, denoted ρ. In particular, each buyer b samples i.i.d. $Bern(\rho)$ for each seller, to generate its set of known sellers $\mathcal{S}_b$.

Agent attributes. Buyers arrive in a round robin fashion. A buyer b who arrives at t submits a query around the buyer's latent location, $q_{b,t} \sim \mathcal{N}(v_b, \sigma_b^2)$, with $\sigma_b^2 = 0.02$. When query q is fulfilled by seller s, the buyer receives a matching utility of $u_{\mathcal{B}}(q, s) = \exp(-c\|q - v_s\|_2)$, where we choose $c = 2$ to have matching utilities span $[0, 1]$. We set a buyer's *per-epoch budget* ψ_b to be v_b^1 (their price preferences) times the number of arrivals within an epoch. Each seller's fractional cost is randomly drawn from $\omega_s \sim U[0.2, 0.4]$. We set the shutdown threshold to $\lambda = 2$ for all sellers, meaning that a seller will go bankrupt after two consecutive epochs without any (positive) surplus. Each agent, whether a buyer or seller, is initialized randomly to stay in the world or join the platform.

Shocks. We vary the world transaction friction, μ_k, across epochs to model the *pre-, during,* and *post-shock* stages. To facilitate the pre- and post-shock comparison, we set the pre- and post-shock stages to each last for at least three epochs and with low world friction, $\mu_k = 0.1$. The shock stage is controlled by a *shock intensity,* $I \sim U[I_{\min}, I_{\max}]$, specifying the largest value that can be attained. We sample $\mu_k \sim$ Lognormal$(\mu = 0, \sigma = 0.5)$, and multiply the values by the intensity I. Figure 4 (red line) shows the average shock schedule across a hundred simulated episodes with $I \sim U[0.8, 1]$.
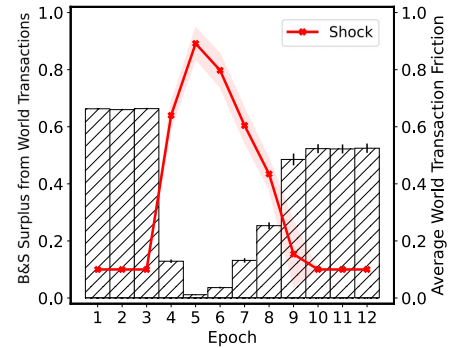

Figure 4: Normalized agent surplus (bars) in markets with no platform under a full cycle of shock.

15

Platform action space. For a platform agent that sets fees, its action space is divided into three subspaces, one for each fee type. Without caps or regulations on fees, the registration fees, $P_{\mathcal{B},k}$ and $P_{\mathcal{S},k}$, range from 0 to 10, with discrete levels at intervals of 0.2. The seller referral rate, $P_{R,k}$, ranges from 0 to 1, with discrete levels at intervals of 0.1.

For a platform agent that decides how to match queries to sellers, the action space is a combination of the choice of matching rule (seller-aware or profit-driven) and a matching utility threshold, ranging from 0 to 1 and discretized at intervals of 0.1.

Implementation details. We sample initial states with a *warm-up epoch* (in addition to the twelve epochs), where the friction is set to $\mu_0 = 0.1$, the platform charges no fees, and buyers and sellers join the platform based on their initialization. The experience gained by buyers and sellers in this warm-up epoch provides a basis for the platform to choose actions, and for the buyers and sellers to form estimates to guide their subscription decisions.

Following the platform POMDP formulation (Section 4), we make the following observations available to the platform:

(i) On-platform buyers and sellers, represented by two binary vectors indicating their subscription states, i.e., $\mathcal{I}_{\mathcal{B}}^p$ and $\mathcal{I}_{\mathcal{S}}^p$,

(ii) Summary statistics of on-platform agents, including the number of platform transactions and the surplus accumulated on platform so far in the epoch,

(iii) The platform matching and transaction matrix between *on-platform* buyers and on-platform sellers for the epoch,

(iv) The platform fees, the matching rule and utility threshold (if the platform learns matching), and the current epoch's world friction.

We use the *Advantage Actor-Critic (A2C)* algorithm [Sutton and Barto, 1998; Degris et al., 2012] to learn the optimal platform policy. Based on preliminary explorations, we design the actor and critic to share a fully-connected layer, LSTM cells of size 128, and again a fully-connected layer to recover sufficient statistics of the history, using this to in effect infer the knowledge structure of buyers about sellers and the demand elasticity of agents to platform fees under shocks. Each network also has its own two fully-connected layers. The critic outputs the value $V_\psi(o)$ of an observation o, and the actor gives policy π_θ for an observation o. For the fee-setting actor, this includes three separate output layers, with each returning a vector of probabilities for one type of platform fee. For the matching actor, this is a vector of probabilities over the matching utility thresholds that are applicable to both matching rules. Figure 5 illustrates the neural network structure we implement.

Besides policy gradient loss, we apply *entropy regularization* to the policy network to encourage exploration [Mnih et al., 2016; Williams and Peng, 1991]. The respective losses for the policy network and the value network are,

$$\mathcal{L}_\pi = -\log \pi(a_k|o_k; \boldsymbol{\theta})(R_k - V_\psi(o_k)) - \beta \mathcal{H}(\pi(A_k|o_k; \boldsymbol{\theta})) \quad \text{and} \quad \mathcal{L}_V = (R_k - V_\psi(o_k))^2,$$

where R_k is the observed return and $\mathcal{H}$ denotes the entropy over learned action probabilities.

We tune the platform policy with various combinations of learning rates $\{0.0001, 0.0005, 0.001\}$, batch sizes $\{4, 16, 32, 64, 128\}$, and entropy weights $\{0.001, 0.01, 0.05\}$, and select hyperparameters that maximize the objective function. In experiments, we present the average performance across a hundred test episodes from two models trained with different torch seeds. We report the detailed training parameters in Appendix D.
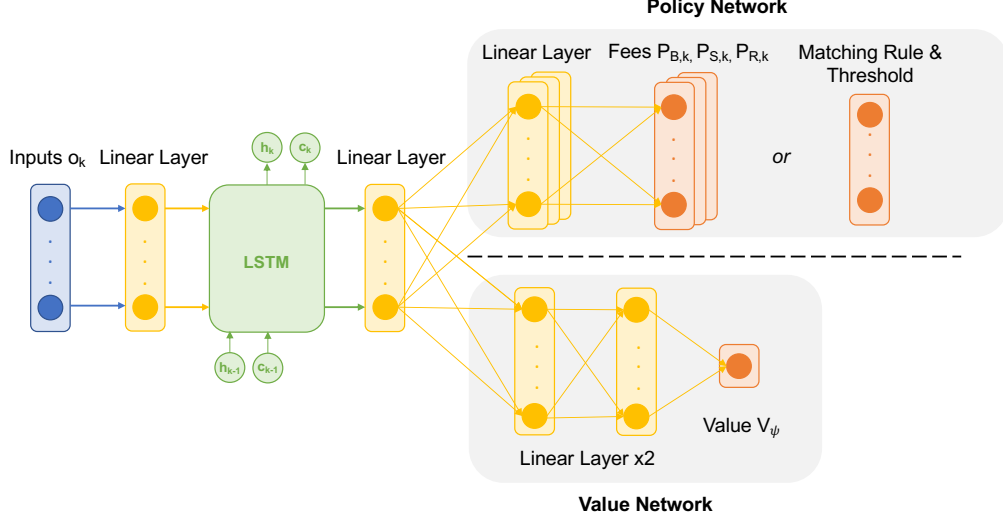
16

Figure 5: Neural network structure for the platform fee-setting and matching policies.

5.2 Characterizing the Environment: Value Created by the Platform

We start by building some intuition about the basic economics of our simulation environments. For this, we characterize the value generated by a revenue-maximizing platform across a range of *single-epoch, no-shock* environments that respectively vary in (1) buyers' knowledge level about sellers ρ and (2) the world transaction friction μ. The warm-up epoch is retained to facilitate agents' subscription decisions. For each market structure, we generate three samples of latent locations of buyer and seller agents, and for each latent sample and knowledge level, ρ, we sample ten knowledge structures, specifying which sellers are known by each buyer. For these single-epoch baseline settings, we can use *Bayesian Optimization* (BO) [Nogueira, 2014] in place of RL to find platform fees that maximize the platform's revenue, and conduct controlled experiments with and without a platform.
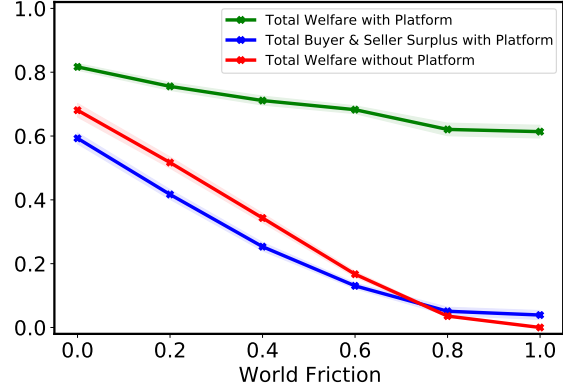
Figure 6 shows the buyer and seller surplus, as well as the total welfare of the Core-and-Niche markets.[6] In the case where there is no platform, the total welfare is equal to the total buyer and seller surplus. We normalize surplus by the total *ideal welfare* achieved in a world where each buyer knows all sellers and there is no world transaction friction (or this is equivalent to a benevolent platform that charges no fee). We validate the simulator by confirming that under no platform scenarios (Figure 6, red lines), total welfare increases as buyers' knowledge level about sellers increases and the world friction decreases.

Across all environments varying in ρ and μ, a revenue-maximizing platform consistently increases total welfare relative to the absence of a platform, creating value by reducing search costs (i.e., matching buyers to unknown sellers) and facilitating transactions (i.e., reducing off-platform fulfilment costs). We can also confirm that the revenue a platform can extract (i.e., differences between green and blue lines) increases as buyers have less knowledge about sellers and as the world transaction friction increases. For scenarios when ρ is extremely low or μ is very high, the platform possesses large market power and can extract all surplus from buyers and sellers.

---

[6]We defer results for the other two market structures to Appendix D.1, which are qualitatively similar to the presented Core-and-Niche market.

17
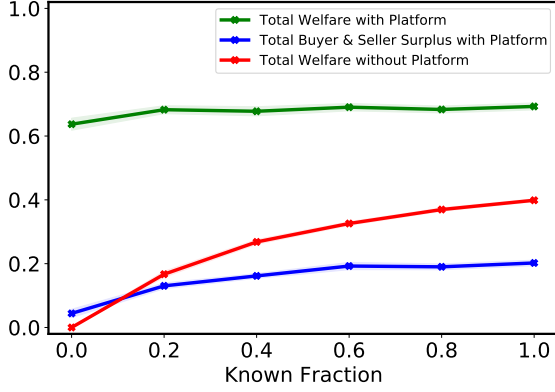
(a) Varying ρ buyers' knowledge level about sellers, with fixed $\mu = 0.6$.

(b) Varying μ the world transaction friction, with fixed $\rho = 0.2$.

Figure 6: Total welfare and buyer and seller surplus achieved in environments that vary in ρ and μ, with and without a platform under the Core-and-Niche market structure.

5.3 Platform Responses to Interventions under Market Shocks

We now present our main results, analyzing in the presence of market shocks, a no-intervention case as well as three different kinds of interventions. The headline result is that, whereas taxes, and caps imposed on some subset of fees, are largely ineffective, there are two forms of interventions— (1) capping all fees while leaving the platform the flexibility to continue to set fees subject to these caps and (2) fixing fees to those that a self-interested platform chooses in an environment *without* shocks while allowing the platform to continue to optimize its matching policy—that can successfully incentivize a revenue-maximizing platform to promote the seller diversity, efficiency, and resilience of the overall economy. We find the second kind of successful intervention especially interesting because it seems of particular relevance to practice (the knowledge of how to set fees comes from the platform's own behavior, and the platform can continue to flexibly make matching decisions). More broadly, we aim to bring the strategic aspects of platform matching to attention in considering different regulatory interventions and evaluating their effectiveness in incentive alignment.

Case 1: Platform fee setting in a laissez-fair system. We first consider a platform that is free from any form of economic interventions: it adopts *myopic query matching* while learning to set fees to maximize its revenue. In Figure 7, we first illustrate the similarity between our simulated increase in the number of platform transactions as an economic shock occurs to the empirically-observed demand surge in Atlanta after the COVID-19 shelter-in-place launched in late March, which is representative of changes in other cities (i.e., on-platform transactions, or demand, increasing by around 50% during the shock). This builds a basic validation of the economic dynamics in our model.

Figure 8, columns 1-2, compares the outcomes achieved in markets without and with a platform. As a first observation, from Figure 8a, we see that the platform generally improves the overall economic welfare, considering the sum of the revenue to the platform and the surplus to buyers and sellers (i.e., the total heights of the bars). This is especially salient during the shock stage (i.e., central bars), when the world transaction friction becomes high and very few transactions can take place and generate surplus in the absence of a platform.
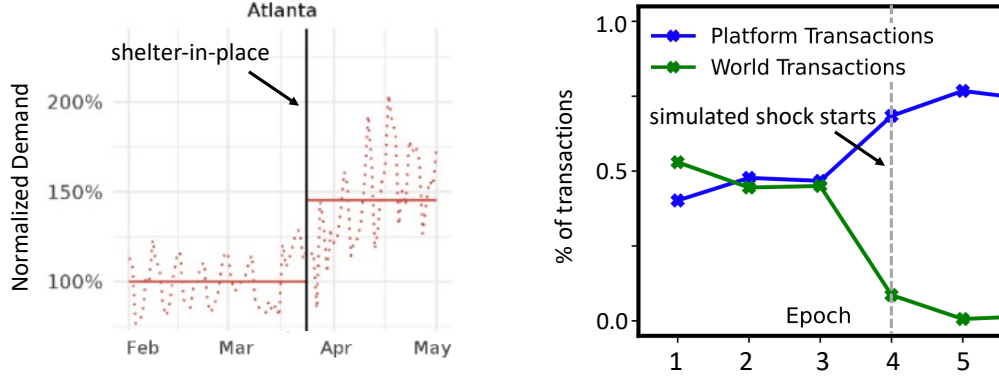
Figure 7: A comparison of empirically-observed demand surge after the shelter-in-place order as shown in Raj et al. [2021] (Left) to the increase in the number of on-platform transactions induced by our simulated economic shock (Right).

At the same time, we see two less beneficial outcomes. First, by comparing pre- and post-shock periods (i.e., the left and right bars of the same column), we find that in both settings with and without a platform, the overall economic welfare falls after the shock. This arises as a result of sellers going bankrupt during the shock, and thus buyer queries can only be matched to less-preferred sellers afterwards. As further verified in Figure 8c, columns 1-2, we see that the number of bankrupt sellers in markets with a platform is almost the same as that in markets without a platform, indicating the ineffectiveness of adopting a self-interested, revenue-maximizing platform in helping businesses to survive an economic shock. In addition, by comparing the surplus to buyers and sellers with and without the platform, we find that the platform tends to reduce total surplus to buyers and sellers in the post-shock stage relative to the case without a platform.

We further characterize the type of seller that is more likely to go bankrupt under markets with and without a platform, classifying sellers into three groups: *core sellers* that are located within one standard deviation of the center and with at least two buyers nearby, *niche sellers* that are beyond two standard deviations from the center with at most one buyer nearby, and *cheap sellers* with prices in the lower quartile. As shown in Table 2, cheap sellers are much more likely to go bankrupt in markets with a platform, whereas niche ones have higher bankrupt probability in settings without a platform. This suggests that a rational platform learns to raise fees as much as possible during the shock (Figure 8b), causing sellers who have lower profit margins and cannot afford the fees to quit the market. As a result, buyers who prefer cheap options are forced to match with expensive sellers.

A surplus-aware platform. As an idealized baseline for regulatory interventions, we can consider the effect of a *surplus-aware platform*, that sets fees to optimize some combination of its own revenue and the on-platform user (buyer and seller) surplus, i.e., $r_{p,k} + \alpha(r_{b,k}^p + r_{s,k}^p)$ and here we choose $\alpha = 0.4$. As shown in Figure 8 (column 3), this platform would in effect restore a comparable level of overall economic welfare after the shock, and lead to fewer bankrupt sellers (see Table 2).

In Figure 9, we capture snapshots of agent states and top-matched sellers in the last epoch (i.e., post-shock) of markets with no platform, a revenue-maximizing platform, and the surplus-aware platform. Controlled experiments are conducted for these settings, so that any change in agent behavior is caused by the presence or different fee-setting policies of the platform. As one can see,

19

for the economic system without a platform, niche sellers go bankrupt, whereas for the revenue-maximizing platform, more cheap sellers go bankrupt, resulting in buyer queries to be matched to less-preferred sellers (e.g., queries from b_0 and b_8 are matched to s_2). When the platform also cares about user surplus, fewer sellers, often those with less demand, shut down.
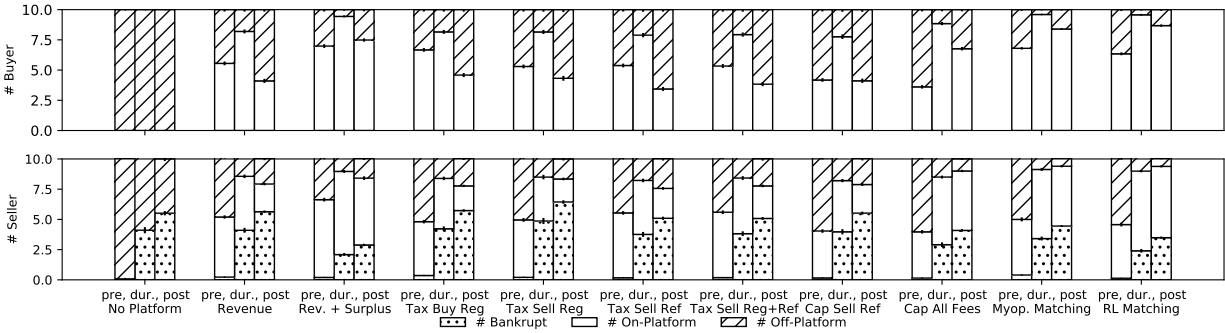
In the sequel, we consider more practical interventions and evaluate their effectiveness under optimal platform responses.
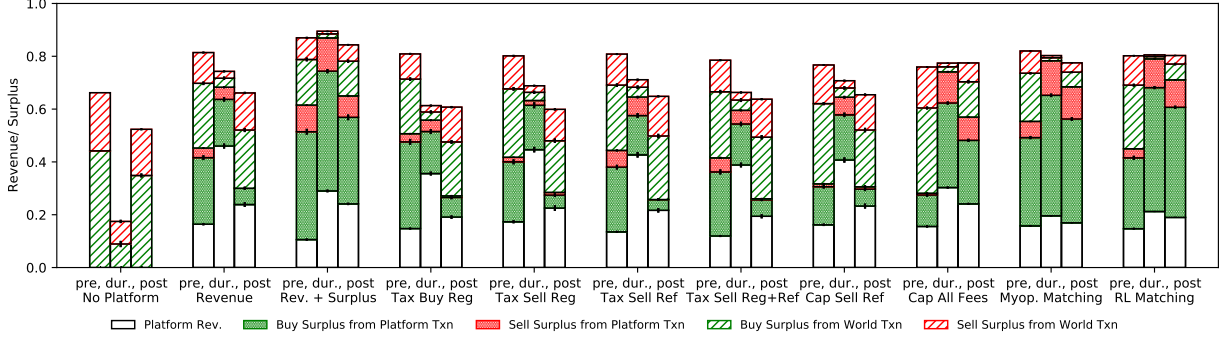
(a) Welfare decomposition achieved by different learned fee-setting policies in each shock stage.
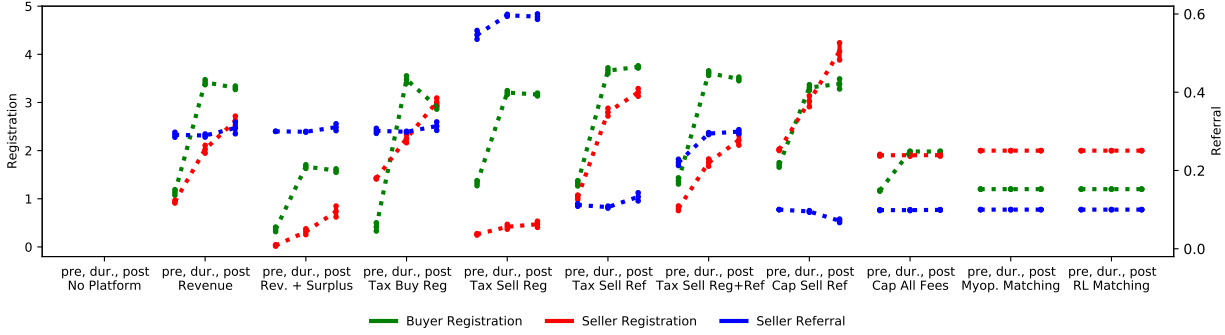
(b) Platform fees set for each shock stage.

(c) The number of on- or off-platform agents and bankrupt sellers induced by different regulatory interventions.

Figure 8: The welfare decomposition, platform fees, and agent states induced by different design objectives and considerations coming from regulation under the Core-and-Niche market structure.

Type of seller	No platform	Revenue-maximizing	Surplus-aware
All sellers	0.55 (0.02)	0.57 (0.01)	0.29 (0.01)
Core sellers	0.33 (0.03)	0.46 (0.02)	0.28 (0.02)
Niche sellers	0.88 (0.08)	0.45 (0.03)	0.17 (0.02)
Cheap sellers	0.52 (0.04)	0.81 (0.03)	0.43 (0.03)

Table 2: Seller bankrupt frequencies in markets with no platform, a revenue-maximizing platform, and a surplus-aware platform.
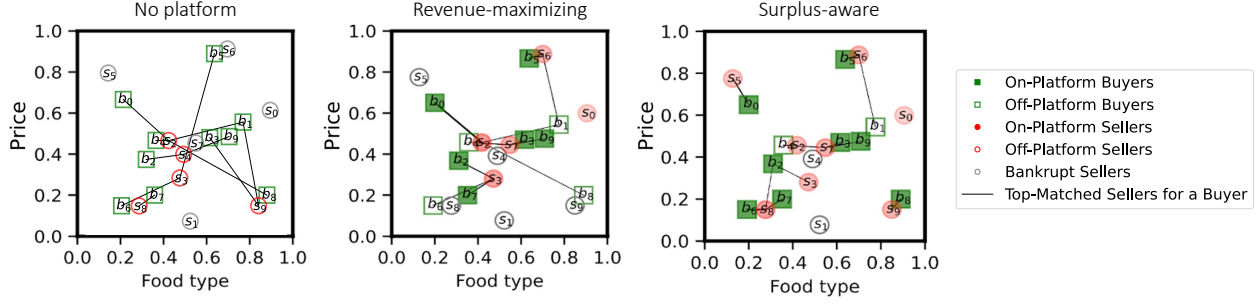

Figure 9: Snapshots of the post-shock market scenarios of economic systems with no platform, a revenue-maximizing platform, and a surplus-aware platform.

Case 2: platform fee setting under taxation policies. We study the effect of taxation policies imposed on platform profits made from different categories of fees, with tax schemes that charge a 20% tax on profits made from buyer registration fees, seller registration fees, seller referral fees, and all fees from sellers respectively.[7] As shown in Figure 8, columns 4-7, our main finding in this regard is that taxation policies in general lead to similar outcomes as those observed in the laissez-fair system (column 2): the overall economic welfare decreases after the shock due to bankrupt sellers. Specifically, charging taxes on a particular kind of fee leads a rational platform to increase other fees, transferring loss to another user group. For example, we compare fees set by a platform under which taxes are imposed on seller referral profits (Figure 8b, column 6) to platform fees set in a laissez-fair system: the platform learns to decrease seller-referral rates while raising registration fees during and after the shock. As a result, on-platform sellers achieve higher surplus (see Figure 8a, column 6, solid red) relative to the laissez-fair case, however, at the expense of lower on-platform buyer surplus (solid green). In another case, where tax is charged on seller registration revenue (Figure 8b, column 5), the platform chooses to significantly increase seller referral rates, leading to even lower surplus to platform sellers.

Case 3: platform fee setting under fee caps. As a third case, we first study the introduction of a 10% cap on the referral fee (the type of intervention that has been launched in several states in the U.S.). Similar to the case of taxes imposed on referral profits, the platform learns to respond by raising the registration fees (Figure 8b, column 8). This leads more buyers and sellers to stay off the platform, especially before the shock, and thus reduces the on-platform surplus.

---

[7]In experiments, we have tried a range of tax rates (i.e., 20%, 40%, 60%). Despite differences in the absolute fee values, they lead to qualitatively similar trends.

As an example of a more positive kind of intervention, we further consider the case where *every* kind of fee is capped, here with $P_{\mathcal{B},k} \leq 2$, $P_{\mathcal{S},k} \leq 2$, and $P_{R,k} \leq 0.1$. As shown in Figure 8b, column 9, under these fee caps, the platform chooses to set the maximum possible fees, except for a lower value in the case of the pre-shock buyer registration. We find that such caps in effect may induce platform fee-setting policies that benefit the economic system: the overall welfare is not affect by the shock, with more viable sellers after the shock offering good query matches. This leads to the intervention that we study in Case 4, which operates without needing special knowledge of how to cap fees and still provides the platform with flexibility in regard to how to match.

Case 4: platform matching under fixed fees. As an extension of imposing caps on all fees, we consider a regulatory policy where the platform is required to keep the same fee structure as it picks, optimally, in a world without an economic shock, but the platform may adapt its matching policy in response to shocks. This is of interest because, unlike platform fees which are observable, matching algorithms used by a platform are often proprietary and hard to regulate, bringing additional challenges in aligning platform incentives with broader societal considerations.

In particular, we follow Section 4.2 and consider a platform that uses RL to learn a matching policy under the fixed, optimal fees (i.e., $P_{\mathcal{B},k} = 1.2$, $P_{\mathcal{S},k} = 2.0$, and $P_{R,k} = 0.1$) that are chosen by a revenue-optimizing platform based on the use of Bayesian optimization in the no-shock setting. As shown in Figure 8a, column 10, even together with strategic matching, the effect of the platform is to help the system preserve a similar level of economic welfare post-shock and result in fewer bankrupt sellers (Figure 8c, column 10).
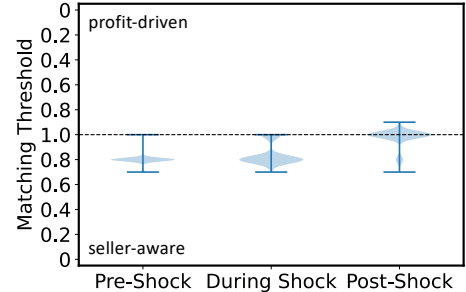

Figure 10: Probability density of the matching rule and the matching utility threshold chosen by the platform in the pre-, during, and post-shock stages.

Figure 10 visualizes the learned matching policy. We can see that to maximize revenue, the platform learns to generally adopt the seller-aware rule, which matches to sellers with lower on-platform surplus, both before and during the shock. This can be explained as follows: (1) the regulated, relatively low fees motivate the platform to retain a larger number of sellers through matching, so as to generate more revenue from registration fees, and (2) with the shock affecting world transactions, the platform has more flexibility in matching, and can afford to compromise a bit on matching quality during the shock without losing many buyers. As the shock decays, the platform tends to increase the matching utility threshold to provide buyers with better matches, and sometimes makes use of the profit-driven rule to extract more revenue from transaction fees paid by high-price sellers.

To further understand the role of the platform, we compare the use of RL matching with *myopic query matching* (Figure 8, column 9) under the same set of fixed fees. Overall, as shown in Table 3, the RL-matching platform achieves both higher revenue and total welfare compared to myopic-matching; it also substantially reduces the probability of cheap and niche sellers going bankrupt due to the shocks.

22

	Myopic matching	RL matching
Welfare	973.12 (5.29)	989.44 (4.40)
Revenue	241.63 (3.06)	261.28 (1.59)
Seller shutdown freq.	0.44 (0.02)	0.34 (0.01)
Core seller shutdown freq.	0.30 (0.01)	0.21 (0.01)
Niche seller shutdown freq.	0.23 (0.02)	0.10 (0.02)
Cheap seller shutdown freq.	0.86 (0.02)	0.78 (0.02)

Table 3: Welfare, platform revenue, and seller bankrupt frequencies of markets mediated by a myopic-matching platform and a RL-matching platform.

6   Discussion

In this paper, we have introduced a multi-agent Gym environment with which to study a platform-mediated economy in the presence of market shocks, making use of RL to model the decision making of a rational, self-interested platform. This is an important new application of AI techniques given the prominence of platforms in today's economic systems and the complexity of these systems. Through this framework, we study and interpret the effect of several regulatory considerations on the efficiency and resilience of the overall economic system under optimal platform responses, confirming caution in regard to some interventions and suggesting a new kind of intervention.

There are several interesting extensions that can be made to the current framework, including platform policies that combine fee-setting and matching, platforms with incomplete information about agent queries and locations, allowing buyer preferences or seller offerings to adapt over time, allowing buyers to learn from on-platform matches about sellers to match with off platform, and achieving economic systems at larger scale.

While we think the present approach holds great promise, we note that there also remain a number of challenges that need to be addressed before AI-based simulation frameworks can be used to guide the design of real-world economic systems. For example, we need principled ways, together with data collection from real-world, platform-based markets, to better inform modeling choices and calibrate the developed simulation framework. It is also important to interpret the learned policies in order to understand such factors as the fairness implications of platform behavior on all parties and in providing transparency to all stakeholders. We leave these aspects for future research, with the goal that the multi-agent platform model developed here offers a constructive basis for researchers and policymakers to better evaluate regulatory interventions and understand their interplay with different parties of agents.

References

Kabir Ahuja, Vishwa Chandra, Victoria Lord, and Curtis Peens. 2021. Ordering in: The rapid evolution of food delivery. *McKinsey & Company* (2021).

Mark Armstrong. 2006. Competition in Two-Sided Markets. *The RAND Journal of Economics* 37, 3 (2006), 668–691. http://www.jstor.org/stable/25046266

Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David C. Parkes, and Duncan Rheingans-Yoo.

2021. Reinforcement Learning of Sequential Price Mechanisms. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. 5219–5227.

Bernard Caillaud and Bruno Jullien. 2003. Chicken & Egg: Competition among Intermediation Service Providers. *The RAND Journal of Economics* 34, 2 (2003), 309–328. http://www.jstor.org/stable/1593720

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 456–464.

Thomas Degris, Patrick M. Pilarski, and Richard S. Sutton. 2012. Model-Free reinforcement learning with continuous action in practice. In *2012 American Control Conference (ACC)*. 2177–2182.

Jean-Pierre Dubé, Günter J Hitsch, and Peter E Rossi. 2009. Do switching costs make markets less competitive? *Journal of Marketing research* 46, 4 (2009), 435–445.

Jean-Pierre Dubé, Günter J Hitsch, and Peter E Rossi. 2010. State dependence and alternative explanations for consumer inertia. *The RAND Journal of Economics* 41, 3 (2010), 417–445.

Joseph Farrell and Carl Shapiro. 1988. Dynamic competition with switching costs. *The RAND Journal of Economics* (1988), 123–137.

Benjamin R Handel. 2013. Adverse selection and inertia in health insurance markets: When nudging hurts. *American Economic Review* 103, 7 (2013), 2643–82.

Matthew J. Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. *CoRR* abs/1507.06527 (2015).

Nicolas Heess, Jonathan J. Hunt, Timothy P. Lillicrap, and David Silver. 2015. Memory-based control with recurrent neural networks. *CoRR* abs/1512.04455 (2015).

Elisabeth Honka. 2014. Quantifying search and switching costs in the US auto insurance industry. *The RAND Journal of Economics* 45, 4 (2014), 847–884.

Eugene Ie, Chih-Wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. 2019. RecSim: A Configurable Simulation Platform for Recommender Systems. *CoRR* abs/1909.04847 (2019).

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101, 1 (1998), 99–134.

Zhuoxin Li and Gang Wang. 2021. Regulating Powerful Platforms: Evidence from Commission Fee Caps in On-Demand Services. https://ssrn.com/abstract=3871514

Alexander MacKay and Marc Remer. 2021. Consumer Inertia and Market Power. https://ssrn.com/abstract=3380390

Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard S. Zemel, and Craig Boutilier. 2020. Optimizing Long-term Social Welfare in Recommender Systems: A Constrained Matching Approach. In *Proceedings of the 27th International Conference on Machine Learning*. 6987–6998.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48. 1928–1937.

National Restaurant Association. 2020. Restaurant Industry in Free Fall; 10,000 Close in Three Months. (2020).

Fernando Nogueira. 2014. Bayesian Optimization: Open source constrained global optimization tool for Python. https://github.com/fmfn/BayesianOptimization

E. Shin Oblander and Daniel McCarthy. 2022. Persistence of Consumer Lifestyle Choices: Evidence from Restaurant Delivery During COVID-19. https://ssrn.com/abstract=3836262

Christos Papadimitriou, Kiran Vodrahalli, and Mihalis Yannakakis. 2021. The Platform Design Problem. In *Web and Internet Economics (WINE)*. 317–333.

Christos H. Papadimitriou, George Pierrakos, Christos-Alexandros Psomas, and Aviad Rubinstein. 2016. On the Complexity of Dynamic Mechanism Design. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, Robert Krauthgamer (Ed.). SIAM, 1458–1475.

Manav Raj, Arun Sundararajan, and Calum You. 2021. COVID-19 and Digital Resilience: Evidence from Uber Eats. https://ssrn.com/abstract=3625638

Jean-Charles Rochet and Jean Tirole. 2003. Platform Competition in Two-Sided Markets. *Journal of the European Economic Association* 1, 4 (06 2003), 990–1029.

Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*. 1257–1264.

Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. 2020. Reinforcement Mechanism Design: With Applications to Dynamic Pricing in Sponsored Search Auctions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2236–2243.

Matthew Shum. 2004. Does advertising overcome brand loyalty? Evidence from the breakfast-cereals market. *Journal of Economics & Management Strategy* 13, 2 (2004), 241–272.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: an Introduction*. MIT Press.

Pingzhong Tang. 2017. Reinforcement mechanism design. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 5146–5150.

Daan Wierstra, Alexander Foerster, Jan Peters, and Jürgen Schmidhuber. 2007. Solving Deep Memory POMDPs with Recurrent Policy Gradients. In *Artificial Neural Networks – ICANN 2007*, Joaquim Marques de Sá, Luís A. Alexandre, Włodzisław Duch, and Danilo Mandic (Eds.). 697–706.

Ronald J. Williams and Jing Peng. 1991. Function Optimization using Connectionist Reinforcement Learning Algorithms. *Connection Science* 3 (1991), 241–268.

Ruohan Zhan, Konstantina Christakopoulou, Ya Le, Jayden Ooi, Martin Mladenov, Alex Beutel, Craig Boutilier, Ed Chi, and Minmin Chen. 2021. Towards Content Provider Aware Recommender Systems: A Simulation Study on the Interplay between User and Provider Utilities. In *Proceedings of the Web Conference 2021*. 3872–3883.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. 2022. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances* 8, 18 (2022), eabk2607.

A Deferred Materials from Section 2

A.1 Stackelberg Equilibrium Definition for Section 2

The *Stackelberg equilibrium* for the platform economy in Section 2 is:

1. The platform first decides upon a buyers' fee $P_\mathcal{B}$ and a sellers' fee $P_\mathcal{S}$. If the platform adopts non-myopic matching, it also decides upon a matching policy.

2. Given the platform's fees and matching policy, buyers and sellers decide to join the platform by comparing surpluses achieved on- and off-platform. That is, agents are playing an equilibrium where no agent can improve their surplus by making a different decision in regard to joining the platform.

3. The platform can not increase its objective (e.g., revenue) by changing their fees or matching policy and allowing the agents to adapt their behavior.

A.2 Proofs of Section 2

Proof of Case 1. When there is no platform, Buyer 1 does not know Seller 1 and can only match their queries to Seller 2. Because of the world transaction friction $\mu = 1$, matching queries of type Q_{11} to Seller 2 provides the buyer with negative surplus, i.e., $r(Q_{11}, S_2, 1) = 2 - |Q_{11} - S_2| - 1 \cdot \mu = \epsilon - 1 < 0$. Therefore, Buyer 1 will only match queries of type Q_{12}, giving a total surplus of

$$r(Q_{12}, S_2, 1) \cdot m = (2 - |Q_{12} - S_2| - 1) \cdot m = \epsilon m.$$

Similarly, with the high friction, Buyer 2 will get negative surplus from matching its queries to either seller, and thus will choose not to transact. As for the sellers, Seller 1 does not get any transaction to cover its fixed cost, and has a negative surplus of $-m$, causing the seller to go bankrupt. Seller 2 gets to transact with m queries, obtaining a zero surplus. The total welfare of the economic system is ϵm. □

Proof of Case 2. With myopic optimal matching, the platform reasons about which agents to keep on the platform, such that in equilibrium, the maximal fees it can set maximizes revenue while retaining the agents to be on-platform. We consider the following options:

1. *All agents join the platform.* In this case, the myopic matching ensures that Q_{11}-type queries are matched to Seller 1, and Q_{12}- and Q_2-type queries are matched to Seller 2 via the platform, i.e., $\mathcal{I}_w = 0$. Since Seller 1 gets zero surplus considering the fixed cost of m, the platform must set $P_{\mathcal{S}} = 0$ in order to keep this seller on-platform. Assuming no buyer registration fee, Buyer 1's surplus is

$$r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_2, 0) \cdot m = (2 - |Q_{11} - S_1|) \cdot m + (2 - |Q_{12} - S_2|) \cdot m = 3m,$$

   whereas with no platform (Case 1), this buyer has surplus ϵm. Similarly, Buyer 2's surplus is $r(Q_2, S_2, 0) \cdot m = (2 - 2\epsilon)m$, while without a platform, its surplus is zero. Therefore, the platform can set a maximal fee $P_{\mathcal{B}} = \min\{3m - \epsilon m, (2 - 2\epsilon)m\} = (2 - 2\epsilon)m$ in order to keep both buyers on-platform. The platform revenue in this case is $2P_{\mathcal{B}} + 2P_{\mathcal{S}} = (4 - 4\epsilon)m$.

2. *All but Seller 1 join the platform.* The myopic policy matches all $4m$ queries to Seller 2 (recall that Buyer 1 cannot transact with Seller 1 without the platform's recommendation). Without fees, Seller 2 has a surplus of $3m$, compared with a surplus of zero without joining, so the platform can set $P_{\mathcal{S}} = 3m$. Since Seller 1 receives no query, it goes bankrupt. Without fees, Buyer 1's surplus on platform is

$$r(Q_{11}, S_2, 0) \cdot m + r(Q_{12}, S_2, 0) \cdot m = \epsilon m + (1 + \epsilon)m = (1 + 2\epsilon)m,$$

   compared with ϵm from not joining. Similarly, Buyer 2's surplus is $(2 - 2\epsilon)m$, while the off-platform surplus is zero. Therefore, to have both buyers join the platform, the platform can set $P_{\mathcal{B}} = \min\{(1 + 2\epsilon)m - \epsilon m, (2 - 2\epsilon)m\} = (1 + \epsilon)m$ with infinitesimal ϵ. This give a total revenue of $2P_{\mathcal{B}} + P_{\mathcal{S}} = 2(1 + \epsilon)m + 3m = (5 + 2\epsilon)m$.

3. *Seller 2 does not join the platform.* In this case, Buyer 2 cannot get a positive surplus either on- or off-platform, and will also stay off the platform. Without fees, Buyer 1's surplus on platform is

$$r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_1, 0) \cdot m = (2 - \epsilon)m + (1 - \epsilon)m = (3 - 2\epsilon)m,$$

   compared with ϵm off platform, and the platform can set $P_{\mathcal{B}} = (3 - 3\epsilon)m$. Without fees, Seller 1's surplus is m, being matched to $2m$ queries, while its off-platform surplus is 0. The platform sets $P_{\mathcal{S}} = m$, and in this case, the platform revenue is $P_{\mathcal{B}} + P_{\mathcal{S}} = (4 - 3\epsilon)m$.

Comparing all options, a revenue-maximizing platform will choose the second one, with all agents but Seller 1 joining the platform. □

Proof of Case 3. In this case, the platform needs to decide (1) who joins the platform and (2) how many queries of each type to match to which on-platform seller. We first note that if there is only one seller joining the platform, the platform's matching policy is necessarily myopic, which reduces to Case 2 and the maximum revenue is $(5 + 2\epsilon)m < 6m$. Consider the case of a single buyer joining the platform. Since there are only $2m$ queries to be matched, the sellers' total surplus is at most m and the buyer's surplus is at most $4m$, therefore the revenue cannot be greater than $5m$.

We consider the case when all agents join the platform. Note that all queries of Q_2 must be matched to Seller 2, otherwise Buyer 2 will reject the matches. Given this, the platform aims to increase Seller 1's surplus from matching, in order to increase the fee that can be charged to sellers.

To this end, the platform considers to divert $x < m$ queries of type Q_{12} to Seller 1, with the rest $m - x$ matched to Seller 2. We note that even for Q_{12}-type queries, Buyer 1 prefers transacting on-platform with Seller 1 to transacting off-platform with Seller 2, due to the world transaction friction. Without fees, the relative surplus of each agents on- and off-platform is:

- *Buyer 1.* On-platform: $r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_1, 0) \cdot x + r(Q_{12}, S_2, 0) \cdot (m - x) = 3m + 2x$, and off-platform: ϵm.

- *Buyer 2.* On-platform: $(2 - 2\epsilon)m$, and off-platform: 0.

- *Seller 1.* On-platform: $m + x - m = x$, and off-platform: 0.

- *Seller 2.* On-platform: $2m + (m - x) - m = 2m - x$, and off-platform: 0.

Based on this, the platform can set $P_\mathcal{B} = \min\{3m + 2x - \epsilon m, (2 - 2\epsilon)m\} = (2 - 2\epsilon)m$ to keep both buyers and $P_\mathcal{S} = \min\{x, 2m - x\} = m$ under the optimal choice of $x = m$. The platform revenue is $2P_\mathcal{B} + 2P_\mathcal{S} = (6 - 4\epsilon)m$, with all agents joining the platform. □

Proof of Case 4. Consider a fee $P \geq 0$ for an on-platform agent, which is set to some value strictly less than the difference between their (without fee) on-platform surplus and off-platform surplus. Increasing P to $P + \delta$ for some infinitesimal $\delta > 0$ increases the platform's objective by $(1 - \alpha)\delta$, which is strictly larger than 0 with $\alpha < 1$. Thus, for whichever agents join the platform, the platform should set fees to maximize its own revenue (under the constraint that those agents do not leave). Similar to Case 2, the platform again needs to consider different sets of agents who might join the platform. We consider the following cases:

1. *All agents join the platform.* To maximize its revenue, the platform sets fees $P_\mathcal{B} = (2 - 2\epsilon)m$ and $P_\mathcal{S} = 0$ while keeping all agents on the platform. Under such fees, Buyer 1 has a per-epoch surplus of $r(Q_{11}, S_1, 0) \cdot m + r(Q_{12}, S_2, 0) \cdot m - P_\mathcal{B} = (1 + 2\epsilon)m$, and Buyer 2 has a per-epoch surplus of $r(Q_2, S_2, 0) \cdot 2m - P_\mathcal{B} = 0$. Seller 1 gets to match with m queries and has a zero surplus, and Seller 2 gets to match with $3m$ queries and has an epoch surplus of $2m$. Therefore, the platform's objective value is $(4 - 4\epsilon)m + \alpha \cdot (3 + 2\epsilon)m$.

2. *All but Seller 1 join the platform.* This is the same as Case 2. The platform sets $P_\mathcal{B} = (1 + \epsilon)m$ and $P_\mathcal{S} = 3m$. The platform's objective value is $(5 + 2\epsilon)m + \alpha \cdot (1 - 2\epsilon)m$.

3. *Seller 2 does not join the platform.* By the same scenario of Case 2, both the on-platform user surplus and the revenue are lower than the previous scenario where all but Seller 1 join the platform, and so is the objective value.

Since $(4 - 4\epsilon)m + \alpha \cdot (3 + 2\epsilon)m > (5 + 2\epsilon)m + \alpha \cdot (1 - 2\epsilon)m$, whenever $\alpha > 1/2 + 2\epsilon/(1 + 2\epsilon)$, the best option is for all buyers and sellers to join the platform, leading no seller to go bankrupt and a total welfare of $(7 - 2\epsilon)m$. □

B   Subscription-Level Decision

Calculations for Subscription Effect Estimation   We first describe the subscription effect estimates for each type of the agents. To recap, these estimates, denoted ξ_{k+1}, are conducted based on the new platform fees and world transaction friction, under the same set of queries submitted or received in the last epoch, i.e., epoch k, and the same subscription decisions made by other agents.

- **A buyer b, on-platform in epoch k.** We denote the estimate of epoch surplus if buyer b does not subscribe to the platform as $\xi^w_{b,k+1}$. For this, given each query of b in epoch k, the choice in the world under new friction μ_{k+1} (Section 3.2) is reevaluated

$$\xi^w_{b,k+1} = \sum_{t\in k} u^w_{b,t}(\mu_{k+1}) = \sum_{t\in k} \max\{u_\mathcal{B}(q_{b,t}, s^*_w) - \mu_{k+1}, 0\},$$

where we denote $u^w_{b,t}(\mu)$ the world matching surplus for buyer b at t under the world friction μ, i.e., $u^w_{b,t}(\mu) = \max\{u_\mathcal{B}(q_{b,t}, s^*_w) - \mu, 0\}$. Similarly, we denote $\xi^p_{b,k+1}$ the estimate of epoch surplus if buyer b remains on platform. The new friction may affect the choice between a platform seller and a world seller. This surplus is estimated based on updated decisions under μ_{k+1}, with

$$\xi^p_{b,k+1} = -P_{\mathcal{B},k+1} + \sum_{t\in k} \max\{u^w_{b,t}(\mu_{k+1}), u^p_{b,t}\}.$$

- **A buyer b, off-platform in epoch k.** The surplus estimate of remaining off-platform depends on the new friction, i.e., $\xi^w_{b,k+1} = \sum_{t\in k} u^w_{b,t}(\mu_{k+1})$. For $\xi^p_{b,k+1}$, we need to estimate the epoch surplus if b subscribes to the platform. We assume that buyer b can observe platform-recommended sellers, e.g., this can be from trial periods to gain platform experience or a platform's estimate of costs and benefits based on past orders. For a query sequence $\{q_{b,t}\}_{t\in k}$, denote the corresponding best platform-recommended sellers as $\{s^{p*}_{b,t}\}$. The estimated surplus if subscribing to the platform is

$$\xi^p_{b,k+1} = -P_{\mathcal{B},k+1} + \sum_{t\in k} \max\{u^w_{b,t}(\mu_{k+1}), u_\mathcal{B}(q_{b,t}, s^{p*}_{b,t})\}.$$

- **A seller s, on-platform in epoch k.** For $\xi^w_{s,k+1}$, we need to estimate the epoch surplus if s does not subscribe to the platform, by reasoning about (1) how many more world transactions would happen if s is not on the platform, and (2) how buyers' transaction decisions may be affected by the epoch $k+1$ world friction. To facilitate a precise estimate, we assume that seller s can observe the sequence of query and seller candidate tuples $\{(q_{b,t}, s, s^p_{b,t})\}_{t\in k}$ in which seller s is chosen as the best world option. Given $q_{b,t}$, $\mathcal{S}_k\backslash s$, and a fixed platform matching strategy used in epoch k, we denote the updated, best-platform seller as $s^{p*}_{b,t}$. For this modified sequence $\{(q_{b,t}, s, s^{p*}_{b,t})\}$, we consider the choices buyers will make under the new friction μ_{k+1}, and estimate the number of transactions seller s will receive without being on the platform, denoted $n^{w'}_{s,k}$. Thus, the estimated surplus if seller s is off platform is $\xi^w_{s,k+1} = n^{w'}_{s,k} v^1_s(1 - \omega_s)$.

For $\xi^p_{s,k+1}$, we reason about how the new friction affects the number of transactions. Given the sequence $\{(q_{b,t}, s^w_{b,t}, s)\}_{t\in k}$ where s is picked as the best platform seller, we consider each buyer's choice and estimate the number of platform visits under μ_{k+1}, denoted $n^{p*}_{s,k}$. Given the sequence $\{(q_{b,t}, s, s^p_{b,t})\}$ where s is picked as the best world seller, we estimate the number of world transactions under μ_{k+1} (even if seller s chooses to stay on the platform), denoted $n^{w*}_{s,k}$. Thus, the estimated surplus that seller s would receive by subscribing to the platform is

$$\xi^p_{s,k+1} = -P_{\mathcal{S},k+1} + n^{p*}_{s,k} v^1_s(1 - \omega_s - P_{R,k+1}) + n^{w*}_{s,k} v^1_s(1 - \omega_s).$$

- **A seller s, off-platform in epoch k.** For $\xi^p_{s,k+1}$, we reason about the surplus from subscribing to the platform, e.g., by asking the platform for an estimate on the number of platform transactions.

Given the sequence of queries on the platform, i.e., $\{q_{b,t}\}_{b\in\mathcal{B}_k,t\in k}$, the platform can update the matches it would suggest s also on-platform by following its matching strategy. Denote the sequence of query and updated seller matches with tuples $\{(q_{b,t}, s_{b,t}^w, s_{b,t}^{p^*})\}_{b\in\mathcal{B}_k,t\in k}$. With μ_{k+1}, we estimate the numbers of platform transactions $n_{s,k}^{p^*}$ and world transactions $n_{s,k}^{w^*}$. The estimated surplus if seller s subscribes to the platform is

$$\xi_{s,k+1}^p = -P_{\mathcal{S},k+1} + n_{s,k}^{p^*} v_s^1 (1 - \omega_s - P_{R,k+1}) + n_{s,k}^{w^*} v_s^1 (1 - \omega_s).$$

Seller s also adjusts the surplus that they would get by remaining off platform, reasoning about how buyers' transaction decisions will be affected by the new world friction. Given the sequence of query and matched seller tuples $\{(q_{b,t}, s, s_{b,t}^p)\}_{b\in\mathcal{B},t\in k}$ in which seller s was chosen as the best world seller, we re-evaluate each buyer's choice to get $n_{s,k}^{w'}$. The estimated off-platform surplus is

$$\xi_{s,k+1}^w = n_{s,k}^{w'} v_s^1 (1 - \omega_s).$$

Calculations for Agent-Specific Decision Inertia   There is a rich body of literature that establishes *decision inertia*, modeling the presence of such inertia across different markets (see Section 3.4). In our setting an agent, either a buyer or seller, in subscription state $\mathcal{I}_k^p \in \{0,1\}$, is prone to stay in the same state in epoch $k+1$, due to habit formation, loyalty, or inattention. We treat buyers and sellers in the same way and illustrate the concept with a buyer b for simplicity. Each buyer starts with an initial preference in regard to adopting the platform or not, denoted by an integer $\chi_{b,0} \sim U[-\chi, \chi]$ for some integer χ. If the initial $\chi_{b,0}$ is positive, the buyer subscribes at the warm-up epoch, otherwise they do not. The inertia $\chi_{b,k}$ maps into an additive bonus to either the surplus for joining the platform or remaining in the world through the functional form:

$$\sigma_{b,k+1}^p := \mathcal{I}_{b,k}^p \log(\chi_{b,k}), \qquad \sigma_{b,k+1}^w := (1 - \mathcal{I}_{b,k}^p) \log(-\chi_{b,k}). \tag{7}$$

That is, the longer an agent sticks to their decision, the larger the bias term gets, which increases in a concave way (logarithmically) over time. Such interpretation of decision inertia as an additive bonus is common in the literature [MacKay and Remer, 2021; Dubé et al., 2009; Farrell and Shapiro, 1988]. Based on this adjusted utility, agents decide whether to subscribe or not according to probabilities inferred by the standard *discrete-choice logit* model [MacKay and Remer, 2021; Dubé et al., 2009], where the probability of subscribing to the platform is

$$\delta_{b,k+1}^p := \frac{\exp\left(\xi_{b,k+1}^p + \sigma_{b,k+1}^p\right)}{\exp\left(\xi_{b,k+1}^p + \sigma_{b,k+1}^p\right) + \exp\left(\xi_{b,k+1}^w + \sigma_{b,k+1}^w\right)}. \tag{8}$$

The inertia is updated after each decision in the following way. If $\chi_{b,k} > 0$ and the buyer subscribes, then $\chi_{b,k+1} := \chi_{b,k} + 1$, and otherwise $\chi_{b,k+1} := -1$ (i.e., it resets for off-platform). Similarly if $\chi_{b,k} < 0$ and the buyer decides to stay off-platform, then $\chi_{b,k+1} := \chi_{b,k} - 1$, and otherwise reset $\chi_{b,k+1} := 1$.

C The Platform's Matching Problem

C.1 Platform Matching Strategies

Algorithm 1 The platform matching strategy with a matching rule and a utility threshold $\eta_k \in [0,1]$.

    Input: Platform fees $P_{\mathcal{B},k}$, $P_{\mathcal{S},k}$, and $P_{R,k}$. On-platform sellers $\mathcal{S}_k$ and their observable attributes. A buy query $q_{b,t}$ for $t \in k$ and $b \in \mathcal{B}_k$.
    Output: A recommended seller $s^* \in \mathcal{S}_k$.

1: Initialize a vector of estimated platform surplus for sellers $\mathbf{r}_{\mathcal{S},t}^p$ at the start of epoch k:

$$r_{s,t}^p = \begin{cases} -P_{\mathcal{S},k} & \text{if } s \in \mathcal{S}_k, \\ 0 & \text{otherwise.} \end{cases}$$

2: Given $q_{b,t}$, compute $u^* \leftarrow \max_{s \in \mathcal{S}_k} u_{\mathcal{B}}(q_{b,t}, s)$.
3: $\mathcal{S}_u \leftarrow \{s : s \in \mathcal{S}_k \text{ and } u_{\mathcal{B}}(q_{b,t}, s) \geq \eta_k u^*\}$ ▷ candidate sellers yield surplus at least $\eta_k u^*$.
4: **if** $|\mathcal{S}_u| = 1$ **then**
5:     $s^* \leftarrow s \in \mathcal{S}_u$ ▷ the same seller as myopic matching.
6: **else**
7:     **if the seller-aware rule then**
8:         $\mathcal{S}_r \leftarrow \{s : s \in \mathcal{S}_u \text{ and } r_{s,t}^p \leq 0\}$
9:         **if** $\mathcal{S}_r = \emptyset$ **then**   ▷ every candidate has break-even.
10:             $r_{\min}^p \leftarrow \min_{s \in \mathcal{S}_u} r_{s,t}^p$, $\mathcal{S}^* \leftarrow \{s : s \in \mathcal{S}_u \text{ and } r_{s,t}^p = r_{\min}^p\}$
11:             $s^* \leftarrow \operatorname{argmax}_{s \in \mathcal{S}^*} u_{\mathcal{B}}(q_{b,t}, s)$ ▷ the best candidate with lowest platform surplus.
12:         **else**
13:             $r_{\max}^p \leftarrow \max_{s \in \mathcal{S}_r} r_{s,t}^p$, $\mathcal{S}^* \leftarrow \{s : s \in \mathcal{S}_r \text{ and } r_{s,t}^p = r_{\max}^p\}$
14:             $s^* \leftarrow \operatorname{argmax}_{s \in \mathcal{S}^*} u_{\mathcal{B}}(q_{b,t}, s)$ ▷ the best candidate who is closest to break-even.
15:     **else**
16:         $s^* \leftarrow \operatorname{argmax}_{s \in \mathcal{S}^*} P_{R,k} v_s^1$ `▷ the profit-driven rule`
17: $r_{s^*,t}^p \leftarrow r_{s^*,t}^p + v_{s^*}^1(1 - \omega_{s^*} - P_{R,k})$
18: **return** s^*

C.2 The Platform Matching POMDP

Based on the fee-setting POMDP (Section 4.1), we make the following adjustments to define the matching POMDP:

- We define $x_k \in \mathcal{X}$ for epoch k as the state of the market after the platform has set fees and agents have chosen to subscribe, but still before the first query is submitted. That is, the state includes agent-subscription states $\mathcal{I}_{\mathcal{B},k}^p$ and $\mathcal{I}_{\mathcal{S},k}^p$ and platform fees, i.e., $P_{\mathcal{B},k}, P_{\mathcal{S},k}, P_{R,k}$, for the upcoming epoch k. We also include the matching utility threshold adopted in the previous epoch, η_{k-1}, to state x_k. All other elements of the state remain the same.

- Here, the platform's action a_k chooses (i) the matching utility threshold for epoch k, and (ii) the matching rule for the epoch, whether seller-aware or profit-driven. For the threshold, we consider η_k that takes discrete values in $[0,1]$.

- Different from the fee-setting POMDP, the state transition for platform matching starts with buyer queries: (1) a buy agent generates a query, (2) if the buyer is on platform, the platform recommends a seller based on η_k and its matching rule, and (3) the buyer selects a seller with whom to transact (Section 3.2). This gives the full sequence of queries, Q_k. At the end of epoch k, buyers and sellers observe the new fees $P_{\mathcal{B},k+1}$, $P_{\mathcal{S},k+1}$ and $P_{R,k+1}$, as given by a fixed fee schedule, and decide whether or not to subscribe to the platform for the next epoch.

- The reward $r_k \sim \mathcal{R}(x_k, a_k)$ of the platform for epoch k includes both the referral fees from epoch k and the subscription fees collected from buyers and sellers, reflecting the decision in regard to whether or not to join the platform for epoch $k + 1$. This avoids delayed reward for the platform, as the registration decisions, and thus registration fees for the next epoch, are influenced by the platform's matching strategy (i.e., action a_k) during epoch k.

- The platform's observability of information in the state follows the same as for the fee-setting POMDP.

D Deferred Materials for Section 5

| Parameters	Value
Observation size	282
Buyer/seller registration max	10
Seller referral max	1
Buyer/seller registration discretization	0.2
Seller referral discretization	0.1
Matching threshold discretization	0.1
Number of matching actions	21
A2C algorithm	
Optimizer	Adam
Learning rate	0.0001
Batch size (fee-setting policy)	4
Batch size (matching policy)	16
Reward discount factor	0.99
Entropy weight	0.01
Training episodes	30000
Number of epochs per episode	12
Linear layer before LSTM	256
LSTM cell size	128
Linear layers for policy and value networks	128
Bayesian Optimization	
Rounds	64
Initial points	10
Iterations in round	50

Table 4: Hyperparameters adopted for platform policy training.

D.1 Experimental Results for Uniform and Two-Core Market Structures

Supplementing Figure 6, Figure 11 shows the total welfare, as well as buyer and seller surplus, achieved in environments with and without a platform under the Uniform and Two-Core market structures. As in the previously-seen graph for the Core-and-Niche structure, we normalize welfare and surplus by the total welfare achieved in an ideal world where buyers have complete knowledge about sellers and there is no world transaction friction.
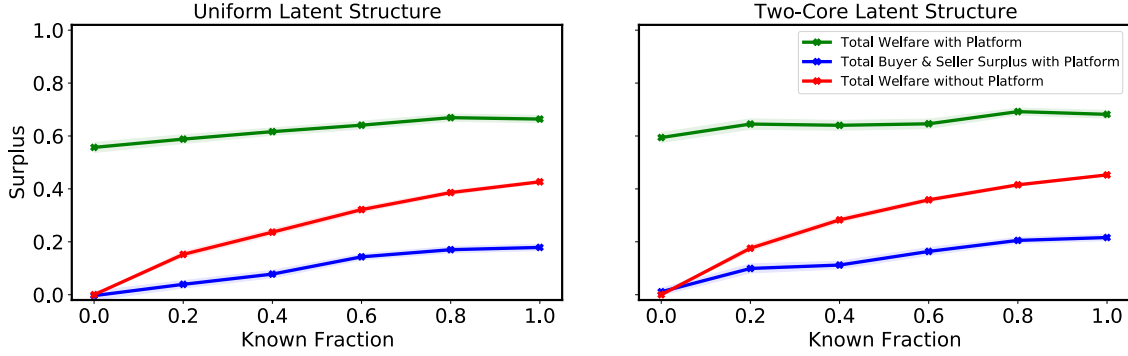

(a) Varying knowledge level about sellers ρ, with world transaction friction $\mu = 0.6$.
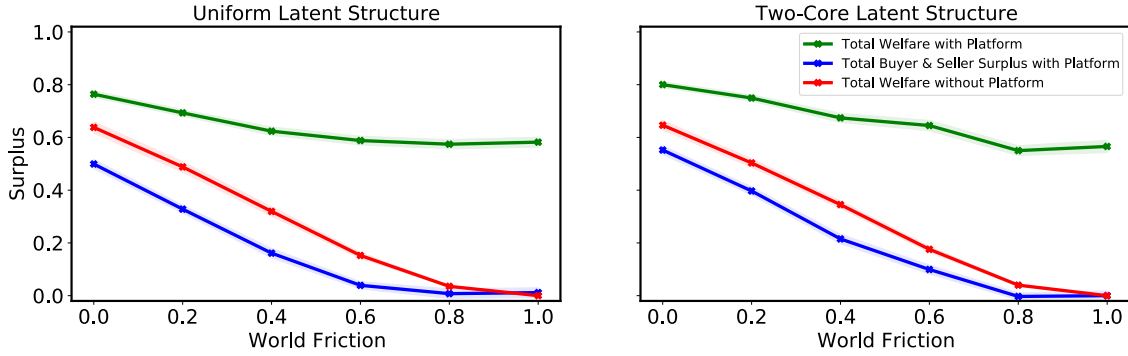
(b) Varying the world transaction friction μ, with knowledge level about sellers $\rho = 0.2$.

Figure 11: Total welfare and buyer and seller surplus, achieved in environments that vary in knowledge level and world transaction friction, with and without a platform under the *uniform* and *two-core* market structures. Results presented are the average of a hundred runs for each environment.

In the sequel, we further present our main results of the *uniform* and *two-core* market structures, analyzing in the presence of market shocks, a no-intervention case as well as three different kinds of interventions. As shown in Figure 12 and Figure 13 below, the headline results remain qualitatively consistent to those of Core-and-Niche markets: taxation policies and caps imposed on some subset of fees are largely ineffective, whereas fixing fees to those that a self-interested platform chooses in an environment *without* shocks while allowing the platform to continue to adapt its matching policy may successfully incentivize a revenue-maximizing platform to promote the seller diversity, efficiency, and resilience of the overall economy.
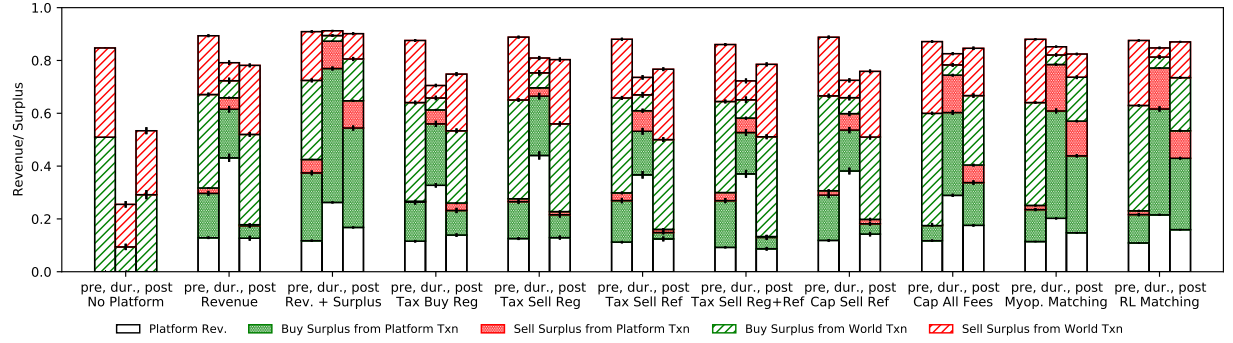
(a) Welfare decomposition achieved by different learned fee-setting policies in each shock stage.
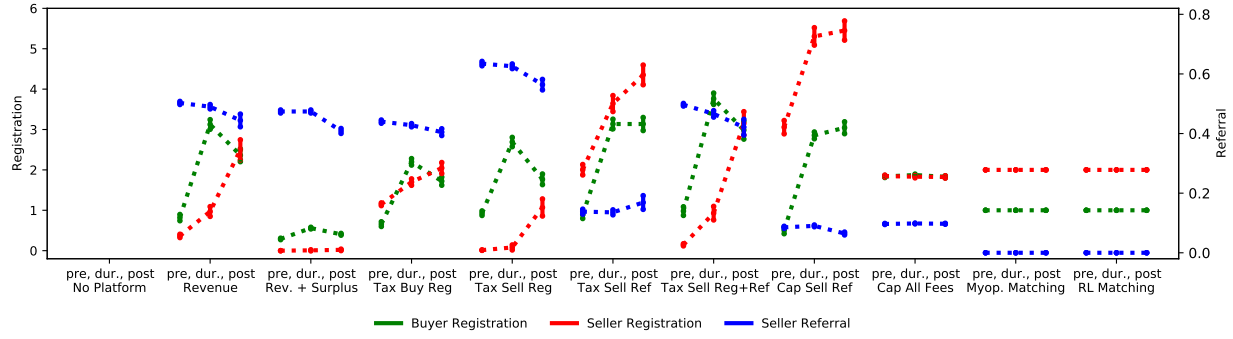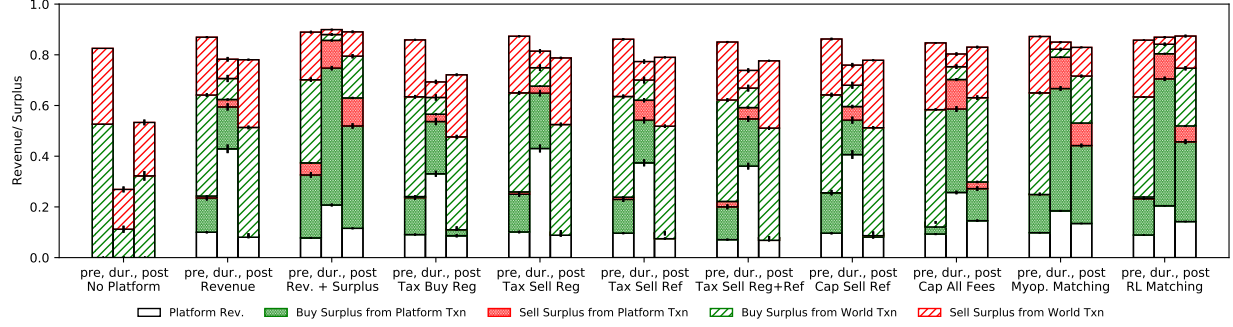

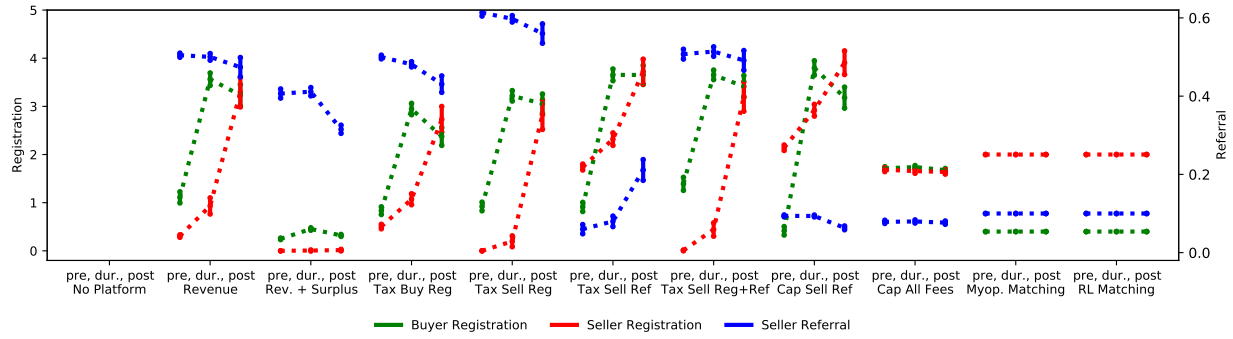(b) Platform fees set for each shock stage.
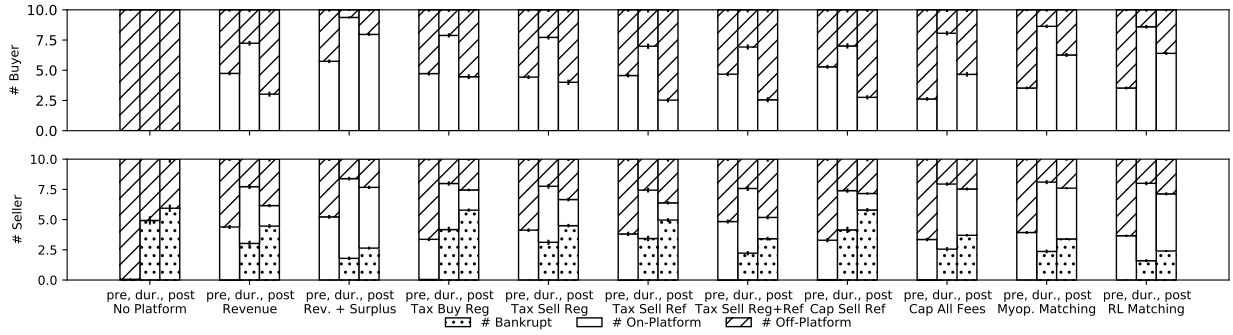
(c) The number of on- or off-platform agents and bankrupt sellers induced by different regulatory interventions.
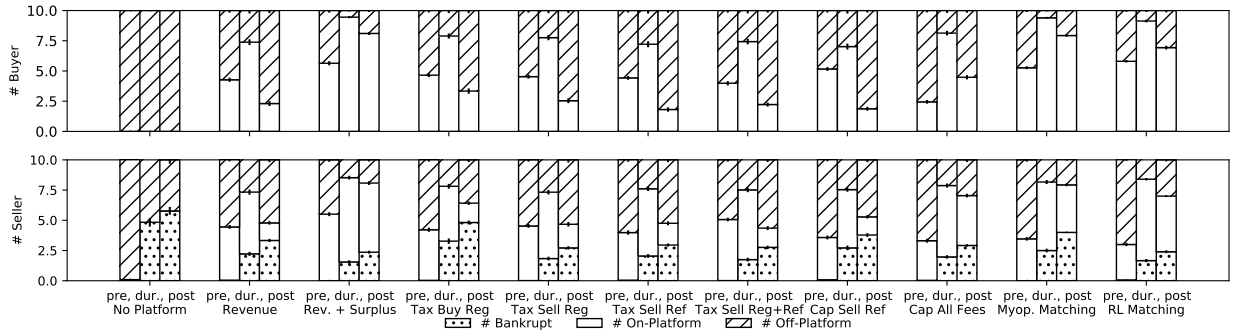
Figure 12: The welfare decomposition, platform fees, and agent states induced by different design objectives and considerations coming from regulation under the *uniform* market structure.

(a) Welfare decomposition achieved by different learned fee-setting policies in each shock stage.

(b) Platform fees set for each shock stage.

(c) The number of on- or off-platform agents and bankrupt sellers induced by different regulatory interventions.

Figure 13: The welfare decomposition, platform fees, and agent states induced by different design objectives and considerations coming from regulation under the *two-core* market structure.