Submitted to *Marketing Science*
manuscript MKSC-22-0238

Recommending for a Multi-sided Marketplace: A Multi-Objective Hierarchical Approach

(Authors' names blinded for peer review)

Many online systems today are multi-sided platforms. They create value by bringing buyers, sellers and other partners together, reducing search and transaction costs (Evans and Schmalensee 2016). A recommendation system for these platforms faces two main challenges. First, all sides of the marketplace have different and potentially conflicting utilities. Recommending in these contexts therefore entails jointly optimizing multiple conflicting objectives. Second, many platforms today present their recommendations hierarchically, where a recommendation item can be either a single product or a group of products. Off-the-shelf recommendation algorithms are not applicable in these settings. We propose MOHR, a multi-objective hierarchical recommender that tackles these challenges. MOHR combines machine learning with scalable multi-objective optimization for multi-sided recommendation, and incorporates a probabilistic model for hierarchical recommendation which accounts for consumers' browsing patterns. The hierarchical approach ensures consistent consumer experience across different levels of aggregations of the products, and provides transparency to the restaurant partners. We further develop an efficient optimization solution for serving MOHR in large-scale online platforms in real time. We implement MOHR at one of the largest industrial three-sided food delivery platforms in the world serving millions of consumers, and experiment with objectives including consumer happiness, marketplace fairness and partner earnings. Online experiments show a significant increase in consumer conversion, retention and gross bookings, which combined translate to $1.5 million weekly gain in revenue. Our work has been deployed globally by the industrial food delivery platform as the recommendation algorithm for its homepage.

Key words: recommender systems; multi-sided marketplace; multi-objective optimization; consumer browsing model

---

1. Introduction

1.1. Research Context

Over the past years, recommender systems are becoming increasingly ubiquitous in retail (Xiao and Benbasat 2007, Pathak et al. 2010, Zhang et al. 2011, Smith and Linden 2017), media (Miller et al. 2003, Bennett et al. 2007, Covington et al. 2016), travel (Ghose et al. 2012, Ursu 2018, Chen and Yao 2017, Noulas et al. 2012), news (Prawesh and Padmanabhan 2014, Dhillon and Aral 2021) and social platforms (Backstrom and Leskovec 2011, Li et al. 2017, Xie 2010). On one hand, they help the consumers by facilitating information acquisition and decision-making; On the other hand, they help the content providers and e-commerce sellers to efficiently target prospective consumers. In other words, recommender systems are a cost-efficient way for marketing. They have become the key drivers for consumer growth in many personalization platforms today. YouTube, the world's largest video sharing platform, reported that 70% of watch time is driven by recommendations (Solsman 2018). Netflix reported that 80% of what people watch on its platform is from personalized recommendations, and that this number was only 2% in 2001 (Gomez-Uribe and Hunt 2015, Govindarajan and Venkatraman 2022). As a result, Netflix has prevented more than $1 billion a year in canceled subscriptions thanks to its personalized recommendation engine (Govindarajan and Venkatraman 2022).

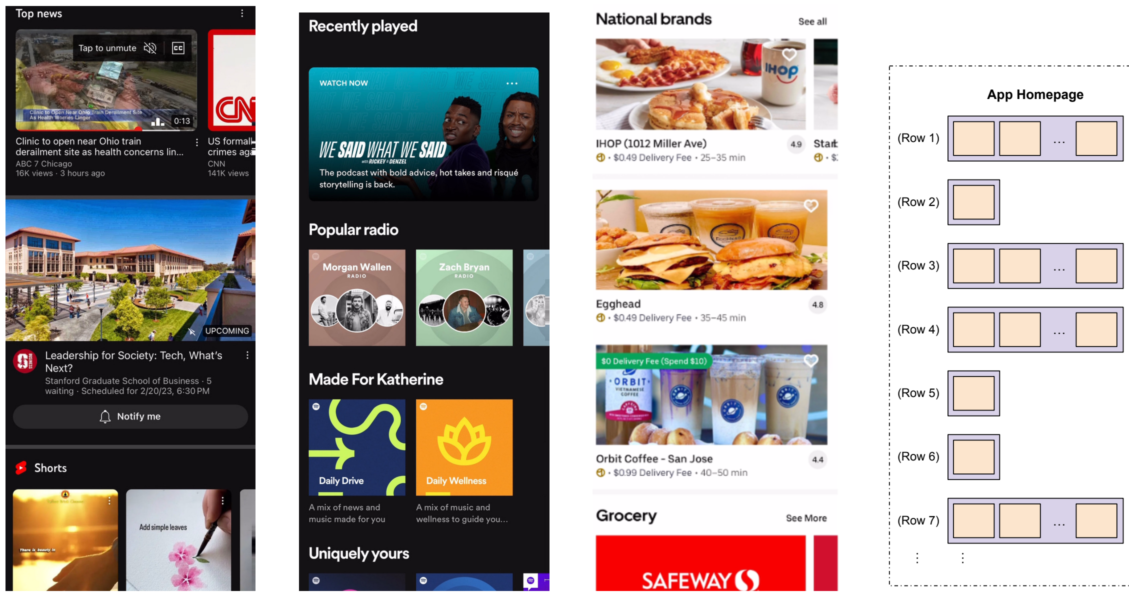
Many recommender systems today operate on multi-sided platforms which bring buyers, sellers and other partners together. These platforms create value by bringing more than one type of participants into the marketplace, reducing search and transaction costs (Evans and Schmalensee 2016). For example, Airbnb creates a two-sided marketplace that connects people who want to rent out their homes with people who are looking for accommodations in specific locations[1]. Content-sharing platforms such as YouTube create a three-sided marketplace by bringing users, content creators and advertisers together. Food and grocery delivery platforms such as Instacart, DoorDash, Uber Eats create a three-sided marketplace consisting of consumers, restaurant or retail partners, and delivery partners (Bahrami et al. 2021). With the recent breakthroughs in technology, multi-sided platforms have become much more prominent in the global economy than in the past decade (Evans and Schmalensee 2016).

1.2. Research Agenda and Challenges

A key element of the success of a multi-sided platform in the long-term is its ability to attract and retain participants from all sides of the business (Abdollahpouri et al. 2020), and balance short-term objectives with long-term ones (Wu et al. 2017). This usually entails a joint optimization of multiple objectives that are potentially conflicting with each other. For example, a news website could make more revenue from advertising by devoting larger spaces of the news page to ads; but then it would risk losing readers because of the overwhelming ads (Aribarg and Schwartz 2020). Such conflicting objectives exist even within the same side of the marketplace. For example, a content sharing platform that is optimized for short-term click-through rates could end up recommending click-baity contents or leading to pigeon-holing effect (Chaney et al. 2018, Fleder and Hosanagar 2009). As a result, it may cause consumer fatigue and boredom (Ursu et al. 2022, Liechty et al. 2005) and hurt consumer happiness in the long-term (Chen et al. 2021). Therefore, a delicate balance among the multiple objectives from different sides is necessary to maintain a healthy ecosystem and ensure the long-term success of the business. We emphasize that the multi-objective challenge is justified even without the multi-sided setup, as firms generally care about multiple conflicting objectives such as short-term profitability and long-term consumer satisfaction. The limits of user-centric or short-term focused recommendation has been recognized over the years (Abdollahpouri et al. 2020). However, there still lacks a mathematically principled and interpretable framework for modeling, understanding and optimizing the *multi-objective trade-off* with more than two objectives. The first part of our work aims at filling this gap.

Another challenge for building a recommender system for modern personalization platforms is the rising of hierarchically presented recommendations. For example, Fig. 1 shows the homepage for YouTube, Spotify and Uber Eats, as examples for a top video-sharing platform, a music-sharing platform and a food delivery platform respectively. In addition to single products, we see collections of products such as "Top News", "Trending Now" and "National Brands" appearing as recommendation items, which are presented as rows that can be scrolled through. Many platforms today adopt this strategy of hierarchical display, where a recommendation item can be either a *single product* or a *row of products*. The homepage is therefore a *two-dimensional grid*. A recommender system in this setting needs to decide the following, ideally in a personalized fashion: what and how many rows to show, and how to rank the single products together with rows of products. Off-the-shelf recommender systems are not directly applicable in these settings, as they focus on ranking products of the same type in a one-dimensional list.

It is, however, challenging to design a recommendation algorithm that ranks single products and rows of products together in a holistic framework, as it is tricky to calibrate across different levels of the aggregation of the products. For example, it is unclear how to compare the "appealingness" of a row of products against a single product in the homepage. As a result, many companies today rely on hand-crafted rules for a hierarchical display of the homepage. For example, certain vertical positions in the homepage are designated for



(a) YouTube. (b) Spotify. (c) Uber Eats. (d) 2-D grid.

Figure 1 **(a) - (c): Screenshots of the app homepages of three large-scale recommendation platforms; (d): Illustration of a hierarchical homepage with a 2-dimensional grid.**

displaying rows of products, and others for displaying single products. However, such rule-based solutions fail to provide transparency to the sellers or content providers as to why a certain product is ranked on top of others. Moreover, consumers have limited patience when browsing the app and may give up at any point. Existing recommendation algorithms fail to explicitly account for this fact when computing the ranking scores for each item. In the second part of our work, we look into modeling and understanding consumer behavior with the hierarchical display, and develop a recommendation solution accordingly.

1.3. Proposed Framework

In this work, we present a principled, generic, and scalable recommendation framework that simultaneously tackles the aforementioned challenges, i.e. multi-objective trade-off and hierarchical display. Specifically, we introduce a Multi-Objective Hierarchical Recommender (MOHR), a framework consisting of three modules: a machine learning module (MO), a hierarchical probabilistic aggregation module (H) and a scalable multi-objective optimization module (R). The MO-module consists of machine learning models for predicting the individual objectives for single products. The H-module generates predictions for rows of products based on a state-based consumer browsing model. The R-module decided within-row ranking and across-row ranking by solving a large-scale multi-objective optimization problem. Each module takes input from its preceding module(s). The final output is a personalized and hierarchically presented homepage that is optimized for the multiple objectives of interest.

We implement MOHR at one of the largest food delivery platforms in the world, which creates a three-sided marketplace consisting of consumers, restaurant partners and delivery partners (couriers). We design several consumer-level, firm-level and fairness objectives and show that they capture the long-term values of the multi-sided food-delivery platform; In particular, they are consumer conversion, consumer retention, basket value and marketplace fairness. Consumer conversion and retention objectives capture the utilities for the consumers and restaurant partners; Basket value objective captures the earnings for the restaurant partners and delivery partners; Marketplace fairness ensures that every restaurant partner on the platform has equal opportunity in this competitive marketplace[2].

We conduct online controlled experiments (A/B testing) on global consumers and compare MOHR with the company's latest machine learning based recommender system. Live experiments show that MOHR effectively pushes forward the Pareto frontier of the top line business metrics, leading to a significant improvement in consumer conversion, retention and revenue, which translate to \$1.5 million weekly gain in revenue. It is also able to achieve a significant performance gain for the new restaurants on the platform, without significantly impacting consumer-side metrics. Because of its significant business impact, MOHR has been deployed globally as the recommender system for the platform's app homepage and serves millions of users every day. It is also one of the largest launches on that platform during the past three years.

We further conduct additional experiments *within* the MOHR framework to understand the multi-objective trade-off on the multi-sided platform. For example, we show that a 2.3% increase in basket value comes at the cost of a 0.9% drop in consumer conversion[3]. MOHR effectively provides a tool to help managers make mathematically principled trade-offs among any number of objectives. More interestingly, we find that it is dangerous to do arbitrary extrapolation of the Pareto frontier, and too much emphasis on one objective may backfire and hurt that objective in practice.

1.4. Contributions

Our research contributes to the marketing community both methodologically and managerially. Methodologically, we propose a novel multi-objective hierarchical recommender framework (MOHR) that addresses two of the most prevalent challenges in recommender systems for online platforms: multiple conflicting objectives and hierarchical display. Specifically, we formulate the problem of "ranking for hierarchical display and with multiple objectives" as two sets of constrained optimization problems: one for within-row ranking and one for across-row ranking. We then solve the multi-objective optimization problem in a hierarchical setting through an innovative formulation of probabilistic consumer behavior modeling and constrained optimization. Empirically, we demonstrate that our approach works at scale and have deployed it on one of the world's largest food delivery platforms.

The managerial contributions of MOHR are three-fold. First, we showcase that long-term optimization via recommender systems can be achieved through a multi-objective approach, as opposed to existing single-score based approaches (Yang et al. 2022, Simester et al. 2020b). Managers of the online platforms should explicitly model and optimize the multiple aspects of the business to ensure the success of the business in the long run. Second, MOHR provides a convenient tool that acts as a knob for the managers to make mathematically principled and quantifiable trade-offs among conflicting objectives. We further provide theoretical insights on the Pareto frontiers of the multi-objective trade-off. Lastly, we empirically demonstrate that too much emphasis on a particular objective may hurt that objective and eventually backfire. A delicate balance among the multiple objectives is required in order to maintain a healthy ecosystem.

Although our findings may depend on the context of our research, our framework is general and readily applicable to other online platforms within and outside the food delivery industry. Amazon, eBay, and other online marketplaces are examples of well-known multi-sided platforms where MOHR can be applied. The three components of MOHR can also be applied in a modularized fashion: It can be applied to cases if the firm is only concerned with one of the two aforementioned challenges (multiple conflicting objectives *or* hierarchical display). In addition, our proposed framework is *not* subject to multi-sided platforms but rather any platforms with multiple conflicting objectives. In fact, most firms today care about multiple objectives such as short-term consumer engagement, consumer retention and satisfaction. Some examples include business-to-customer (B2C) platforms such as Meta and Netflix which need to balance short-term sales or consumer engagement with long-term profitability. Our proposed framework is readily applicable to these settings as well.

The rest of this paper is organized as follows. In Section 2, we discuss the related literature. In Section 3, we introduce the institutional background and data. We present the full MOHR framework in Section 4, and experiment results in Section 5. In Section 6, we conclude with a discussion on the contribution, implications and future research of our framework.

2. Related Work

2.1. Recommender Systems and Product Choice Modeling

Recommender systems are closely related to the product choice modeling literature in marketing in that they both aim to predict the consumer's future purchases or clicks. There is an abundance of literature in marketing in developing model-based approaches to predict individuals' choices (Guadagni and Little 1983, Wagner and Taudes 1986, Fader and Hardie 1996, Johnson et al. 2012, Farias and Li 2019). These approaches utilize consumer and product characteristics, as well as their interaction history, to enhance predictive accuracy and scalability (Jacobs et al. 2016, Yoganarasimhan 2020), or propose interpretable recommendation models with econometric terms (Bodapati 2008). Our work contributes to this strand of literature by modeling and understanding consumer behavior with *hierarchically* presented products in large-scale online recommendation platforms.

Methodology-wise, recommendation problems are usually formulated as a learning-to-rank task for information retrieval (IR) (Baeza-Yates et al. 1999). Learning-to-rank algorithms can be classified into three approaches: pointwise, pairwise, and listwise. These approaches differ in terms of whether they consider a single recommendation item (pointwise) (Adomavicius and Tuzhilin 2005), a pair of items (pairwise) (Yoganarasimhan 2020), or a list of items (listwise) at a time within the loss function (Liu et al. 2009). Model architecture wise, the evolution of recommender systems has progressed from simple matrix factorization (Adomavicius and Tuzhilin 2005) and SVM methods (Gong et al. 2014) to more sophisticated machine learning architectures such as tree-based models (He et al. 2014), deep neural networks (Chaudhuri et al. 2021, Covington et al. 2016) and reinforcement learning (Chen et al. 2019, Zheng et al. 2018, Liu 2022). Our framework makes a contribution to pointwise recommender systems using machine learning models, which are extensively utilized in large-scale industrial recommender systems (Covington et al. 2016, Smith and Linden 2017).

In terms of the information used to build the recommender systems, there are three types of recommender systems: content-based filtering, collaborative filtering, and hybrid methods (Adomavicius and Tuzhilin 2005, Ricci et al. 2015, Dhillon and Aral 2021). Content-based recommender systems are based on a description of the item and a profile of the consumer's preferences and recommend items that are similar to items that the consumer has enjoyed in the past (Aggarwal et al. 2016, Brusilovsky 2007, Mooney and Roy 2000). Collaborative filtering approaches are based on the assumption that consumers who liked similar items in the past will like similar kinds of items in the future (Breese et al. 1998, Billsus et al. 1998), and are further classified as memory-based (Adomavicius and Tuzhilin 2005, Delgado and Ishii 1999) and model-based (Billsus et al. 1998, Breese et al. 1998) approaches. Most recommender systems today adopt a hybrid approach combining collaborative filtering and content-based filtering (Balabanović and Shoham 1997, Adomavicius and Tuzhilin 2005, Tso-Sutter et al. 2008, Sahoo et al. 2012), which empirically performs better than pure approaches (Adomavicius and Tuzhilin 2005). In this work, we leverage both content features and model-based collaborative filtering features based on the activity history between the different players in the marketplace as inputs to machine learning based recommendation models, contributing to the literature of hybrid recommender systems.

2.2. The Effects of Recommendations

A large number of studies in marketing, information systems and computer science have developed understanding on the effects of recommender systems on consumer decision making. Xiao and Benbasat (2007) provided theoretical perspectives on the effects of recommender systems on consumer decision making processes and outcomes. Recommender systems affect consumers' consumption patterns from various aspects, including diversity (Fleder and Hosanagar 2009, Anderson et al. 2020), exploration (Datta et al. 2018, Chen et al. 2021), homogeneity (Chaney et al. 2018) and fragmentation (Hosanagar et al. 2014). In e-commerce, it has been demonstrated that recommendation and ranking positions have significant impact on the consumers, including search behavior (Narayanan and Kalyanam 2015, Ursu 2018), willingness to pay (Carare 2012, Adomavicius et al. 2018), trust (Wang et al. 2018) and even consumption preferences (Adomavicius et al. 2013). It has also been shown that recommender systems affect other parties of the e-commerce marketplace, through impacting demand levels (Ghose et al. 2014, Oestreicher-Singer and Sundararajan 2012, Bourreau and Gaudin 2022, Kumar and Hosanagar 2019), seller profits (Zhou and

Zou 2022, Chen et al. 2008, Das et al. 2009, Azaria et al. 2013) and social welfare (Zhang et al. 2021, Aridor and Gonçalves 2021, Donnelly et al. 2022). In our work, we show that recommender systems can have positive or negative impact on different sides in a multi-sided marketplace, and propose a recommendation framework that addresses the trade-off among the conflicting objectives in the marketplace in an interpretable and principled way.

2.3. Long-Term Optimization for Recommender Systems

Recommender systems that focus on optimizing consumer's immediate responses such as clicks and likes in the current session have gained tremendous success over the past years (Adomavicius and Tuzhilin 2005, Covington et al. 2016). However, it has become increasingly clear that over-indexing on short-term engagement can lead to undesirable recommendations, such as clickbait contents or pigeon-holing effects which hurt long-term user experience (Fleder and Hosanagar 2009, Chaney et al. 2018, Ursu et al. 2022, Liechty et al. 2005). Recognizing the drawbacks of short-term focused recommender systems, researchers and practitioners are shifting their focus towards optimizing long-term values via recommender systems (Besbes et al. 2016, Wu et al. 2017). This is also one of the motivations behind our work.

Long-term optimization for recommender systems is however challenging, as the desired outcome is sparse, noisy and naturally manifests over a long-horizon. Existing works fall into the following three categories: Advanced sequence modeling techniques such as Recurrent Neural Nets (RNNs) (Kang and McAuley 2018) and Transformer (Sun et al. 2019) that are able to account for longer consumer history; Reinforcement Learning (RL) methods (Afsar et al. 2022, Kokkodis and Ipeirotis 2021, Liu 2022) that allow the recommender system to plan several steps ahead; and surrogate objectives that are predictive of the long-term outcome but easier to optimize (Athey et al. 2019, Yang et al. 2022, Simester et al. 2020b, Wang et al. 2022). These methods attempt to capture the long-term values of a recommendation platform as a single objective. In this work, we argue that focusing exclusively on consumer-level objectives limits the platform's capability to optimize holistically. In particular, we demonstrate that it is important to adopt a multi-sided view in optimizing the long-term values of personalization platforms.

2.4. Multi-Objective Recommendation for Multi-sided Marketplace

With the increasing awareness of the limitations of single-objective systems, there has been an increasing number of literature on recommender systems with more than one objectives. For consumer-centric recommender systems, research has evolved from optimizing a single aspect of consumer feedback such as ratings or click-through rates (Adomavicius and Tuzhilin 2005, Hu et al. 2008), to utility-based recommender systems that capture multidimensional preferences of consumer utilities (Ghose et al. 2012, Li et al. 2017, Carbonell and Goldstein 1998, Chung and Rao 2012). With the rise of multi-sided platforms (Abdollahpouri et al. 2020), there has been emerging literature on recommender systems for multi-sided marketplace (Abdollahpouri et al. 2020). For example, researchers have considered seller earnings and platform profits by explicitly incorporating revenue or profit as objectives for the recommender systems (Chen et al. 2008, Das et al. 2009, Hosanagar et al. 2008, Azaria et al. 2013), and further maximize the total welfare of the system (Aridor and Gonçalves 2021, Zhang et al. 2021). In recent years, fairness has become an additional objective of interest to guarantee that different sellers or content creators are provided with equal opportunities on the platform (Beutel et al. 2019, Wang et al. 2021). Multi-sided recommender systems find its applications in various domains including e-commerce (Li et al. 2018), education (Zheng et al. 2019), loan (Lee et al. 2014), travel (Krasnodebski and Dines 2016), news (Tintarev et al. 2018), and content-sharing (Zhao et al. 2019).

Recommending with multiple objectives usually entails optimizing multiple objectives that are potentially conflicting with each other. For example, Hosanagar et al. (2008) looked into the trade-offs between the relevance to consumers and the margin for the firm, and between short-term and long-term profits, and showed that a profit-sensitive strategy led to an increased revenue without a significant loss in consumer satisfaction. On the other hand, Zhang et al. (2021) showed that maximizing profit can actually hurt consumer surplus. To the best of our knowledge, extant works mostly focus on understanding and optimizing the trade-offs between *two* objectives in a two-sided marketplace. In other words, it remains unclear how to model, understand and optimize the *multi-objective trade-off* in a marketplace with more than two sides or more than two objectives. In our work, we propose to adopt a mathematically principled framework that models the trade-off among any number of objectives, and validates its efficacy in a three-sided marketplace.

Building a recommender system for a multi-sided marketplace is essentially a multi-objective optimization problem (Sawaragi et al. 1985). Methods from the multi-objective optimization literature have been adapted for multi-sided recommender systems. Examples include constrained optimization (Rodriguez et al. 2012, Agarwal et al. 2015, 2011, 2012), learning-to-rerank (Nguyen et al. 2017) and multiple-gradient descent (Milojkovic et al. 2019). Extant works on multi-objective optimization all adopt a one-dimensional setting, where the output is a single ranked list of recommendation items. Therefore, they do not apply to hierarchical recommendation scenarios. Our proposed framework solves the multi-objective optimization problem in a hierarchical setting through an innovative formulation of probabilistic consumer behavior modeling and constrained optimization, contributing to the multi-objective ranking literature.

2.5. Consumer Behavior Modeling and Hierarchical Recommendation

Consumers' browsing behavior on personalization platforms is related to consumer search for decision making. Weitzman (1979) was among the first to model sequential search behavior. Built on this, Ursu (2018) proposed a sequential search model for understanding the effect of rankings on consumer online choices in the hotel industry. Shi and Trusov (2021) developed an empirical model for consumers' scroll behavior in search engine marketing (SEM) based on laboratory eye-tracking data. Jiang et al. (2021) developed a structural search model that characterizes the consumer search process. Dhillon and Aral (2021) proposed a neural matrix factorization approach to model consumers' dynamic interest over time. These works focused on building structural or temporal models for understanding consumer behavior, but did not leverage the model output or the understanding to improve the ranking or build a new recommender system. Closer to our work is Liebman et al. (2019), which leveraged consumers' in-session sequential behavior for online adaptation to listeners' music preferences. However, the proposed ranking model does not apply to the hierarchical display of recommendation items in our case.

Hierarchical recommendation is proposed to recommend products of different levels of aggregations. Methodologies based on hierarchical clustering (Zheng et al. 2013, King and Imbrasaitė 2015) and hierarchical reinforcement learning (Xie et al. 2021) are used to recommend aggregations of products. Oestreicher-Singer and Sundararajan (2012) studied the performance of a single-item recommendation in the context of a group of recommended products. Song et al. (2019) proposed a cascade model for consumers' sequential scrolling behavior and decision process, and a multicategory utility model for recommending items on category levels. To the best of our knowledge, existing works in this area either did not explicitly model and account for consumers' browsing behavior (Zheng et al. 2013, Xie et al. 2021), or the recommendation output is a homogeneous one-dimensional list although the consumer decision process is modeled in a hierarchical way (Song et al. 2019, Oestreicher-Singer and Sundararajan 2012, Agrawal et al. 2009). Our work bridges this gap by developing a state-based consumer browsing model with the hierarchical display of the homepage, which is incorporated into the final multi-objective ranking framework for across-row ranking. In the SEM context, it has been shown that consumers rarely looked at lower ranking results (Guan and Cutrell 2007) and their dominant browsing pattern looks like the letter F or a "golden triangle" (Nielsen 2006, Sherman 2005). Ursu (2018) shows that the click-through rate decreases with lower ranking positions on a hotel recommendation website. Our work contributes to this line of research. We demonstrate that in order to optimize consumer experience, it is critical to explicitly model how they browse a hierarchically arranged homepage and interact with products at different horizontal and vertical positions (Alvino and Basilico 2015).

3. Data and Institutional Background

3.1. Three-Sided Food Delivery Marketplace

There has been an emerging wave of food delivery platforms in the past decade. During the COVID-19 pandemic, the use of online food delivery services increased 67% globally (Muangmee et al. 2021). The total revenue of the food delivery industry is expected to increase to $388.74 billion by the year 2028, equaling a compound annual growth rate (CAGR) of 10.8% (GVR 2022). These services create a three-sided marketplace consisting of consumers, restaurant partners and delivery partners. Consumers place orders on food from the restaurants on the platform. Delivery partners pick up the food from the restaurants and deliver to the consumers, when consumers have the option to add a tip. The platform charges a fixed portion of the consumers' payment as the commission fee and pays the rest to the restaurant partners. The delivery partners earn income from consumers' tips and the platform's payment[4].

Recommender systems for these food delivery platforms play a critical role in the long-term success of the business. Each side in the food delivery marketplace has different and potentially conflicting utilities, and only optimizing for one may hurt others. For example, many personalization platforms focus on optimizing consumers' short-term conversion rate (i.e. placing an order) today. Such a recommendation strategy for a food delivery platform can lead to overly recommending popular and well-established restaurants. This causes several issues. On the restaurant side, new and low-volume restaurants will not get enough exposure on the platform, which discourages them from remaining on the platform. On the consumer side, the consumers face lack of selection and recommendation diversity as a result, leading to pigeon-holing effects (Fleder and Hosanagar 2009, Chaney et al. 2018) or consumer boredom (Ursu et al. 2022, Liechty et al. 2005) and hurting their long-term experience. On the courier and platform side, the highly skewed demand hurts supply efficiency and service reliability: Restaurants may not be able to fulfill a large quantity of orders in a short time, and there might not be enough delivery partners nearby. All of these issues hurt one or more sides of the marketplace and eventually the business. Therefore, a delicate balance among the multiple objectives from different sides is needed to maintain a healthy ecosystem and ensure the success of the business in the long term.

3.2. Hierarchically Presented Recommendations

As is shown in Figure 1, modern recommender systems often adopt a hierarchical display of the homepage. Specifically, a recommendation item can be either a single product or a row of products. Rows are also referred to as *"carousels"* in industry contexts, which emphasizes the fact that they can be scrolled horizontally. Carousels are widely adopted across different online platforms across different industries (Elahi and Chandrashekar 2020). They offer several advantages for recommendation purposes. First, carousels can be viewed as nudges tailored to the different modes of the consumers (e.g. in a hurry, looking for something healthy) and help them efficiently navigate through the contents. Second, the title of the carousels provide extra information about the products (e.g. cuisine type of the restaurant, genre of the movie) that may be critical to consumers' decision making, but are not clear to the consumers otherwise. Finally, they alleviate cold-start challenges for new products and new consumers. For example, "Popular Near You" or "National Favorites" are non-personalized recommendations based on only content information, which is a popular approach for solving cold-start problems in recommender systems (Schein et al. 2002).

While carousels are appealing in some contexts, at other times a single product is preferred as recommendation. For example, a consumer may repeatedly order the same product. An ideal recommendation setup is therefore a combination of carousels and single products. As a result, the homepage is a two-dimensional grid and a recommendation algorithm needs to decide both *within-row* ranking and *across-row* ranking.

3.3. Data

The company we work with is one of the largest food delivery platforms in the world serving millions of consumers every day. Consumers' interactions in the app are logged and processed through Apache Hive (Thusoo et al. 2009) for data extraction, transformation and loading (ETL). Specifically, an impression event is logged when the consumer scrolls through a product (a restaurant). An order event is logged when the consumer places an order. Contextual information is logged together with the event, including time of day, day of week and geolocation etc. A *session* is generated when a consumer opens the app and the backend will call the recommender system to generate real-time recommendations. A new session is generated when the consumer refreshes or comes back to the homepage (e.g. from the order page or search page), and the recommendations will be generated accordingly. Therefore, there will be at most one order from each consumer in every session, and multiple orders from the same consumer will be treated as multiple sessions. There are about 120 *predetermined carousel candidates* (curated by product managers at the company) and each carousel has a *fixed* set of qualifying candidates. In other words, the carousel candidates and restaurant candidates within the carousels are *external* to our framework[5].

The data we obtain from the company are randomly sampled from the global user logs, consisting of about 600 million impressions and 11 million orders between May 15, 2019 and May 28, 2019[6].

4. Machine Learning Framework

In this section, we describe our proposed Multi-Objective Hierarchical Recommender (MOHR) framework. Section 4.1 provides a high-level overview of the whole framework. Section 4.2, 4.3 and 4.4 describe the three modules in detail. Note again that although we describe this framework in the context of food delivery platforms, it is general and readily applicable to other platforms and other industries.

4.1. Overview of the framework

The proposed MOHR framework consists of three modules: a machine learning module (MO) for multiple outcomes, a hierarchical probabilistic module (H) for hierarchical aggregation, and a scalable multi-objective optimization module (R) for final ranking. In the

MO-module, we generate product-level predictions for the multiple outcomes using machine learning models. In the H-module, we generate row-level predictions by aggregating the product-level predictions from the MO-module with a consumer browsing model. In the R-module, we decide within-row ranking and across-row ranking by solving a large-scale multi-objective optimization problem. Each module takes as input the output from the previous modules, and the final output is a personalized and hierarchical homepage that is optimized for the multiple objectives. Figure 2 shows an overview of the full MOHR framework.

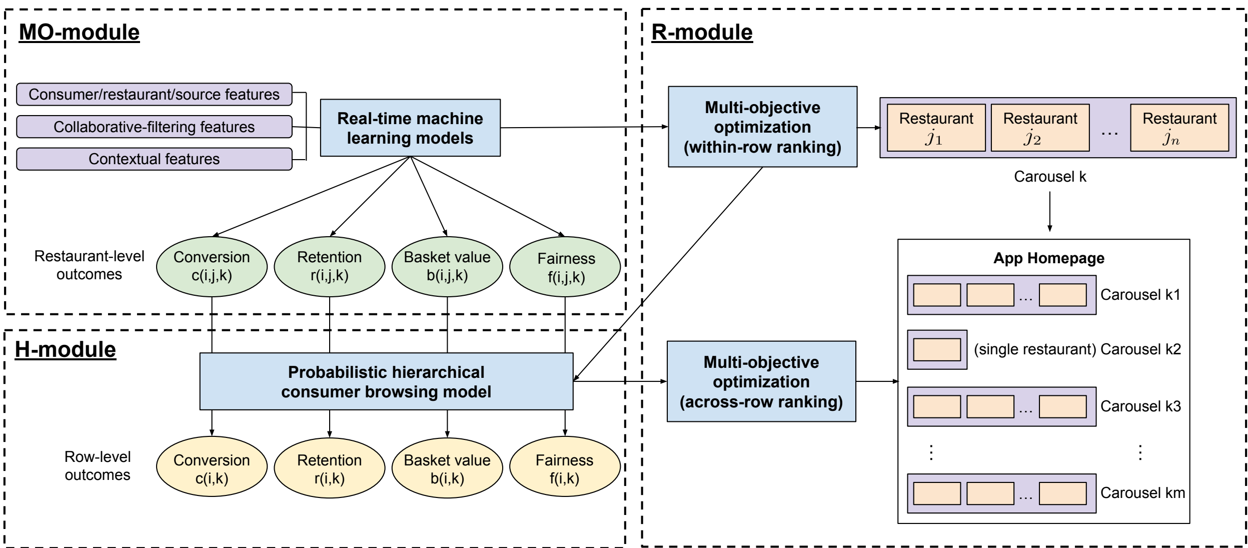


Figure 2 (Color online) An overview of MOHR.

4.2. MO-Module: Machine Learning Models for Product-Level Outcomes

4.2.1. Selection of Objectives. We select the objectives based on the following considerations. First, they should be directly capturing the utilities of one or multiple sides in the marketplace. Second, the objectives should have reasonable signal-to-noise (STN) ratio for the machine learning models to learn well. For example, we do *not* include consumer satisfaction as an objective, which is a relevant outcome for the consumer side. This is because in practice, there is usually no easy way to get clean and accurate signals for consumer satisfaction[7]. Lastly, the objective set should be small enough for model training and serving efficiency in large-scale online platforms. Based on these considerations, we define the following outcomes as the objectives for a food delivery platform:

- **Consumer conversion**: whether the consumer places an order. This captures the *short-term* outcome of the business and is a common objective used by many recommender systems (Zhang et al. 2019, Covington et al. 2016) today.
- **Consumer retention**: whether the consumer returns to the platform and orders again within the next time window[8], if the consumer orders in the current session[9]. This captures the *long-term* outcome of the business and serves as a proxy for consumer satisfaction.
- **Basket value**: dollar amount of the order, if the consumer orders in the current session. This captures the earnings for both *restaurant partners* and *delivery partners* in addition to the platform itself, as a fixed proportion of the total basket value (gross bookings) will be given to the partners as payments.
- **Marketplace fairness**: the exposure opportunities that new restaurants receive on the platform. The term "fairness" has been used in machine learning and ranking problems to refer to the notion of "equal opportunity" across different subgroups of population (Hardt et al. 2016, Beutel et al. 2019). Here the goal of the "marketplace fairness" objective is similar. Specifically, it track the amount of exposure provided to every restaurant partner, and ensure that the platform provides a fair marketplace by granting exposure to weak restaurants even if they perform poorly[10]. As a result, the restaurant partners would be more likely to stay with the platform, leading to a healthy ecosystem.

This is the minimal set that we come up with that best captures the objectives for the three-sided food delivery platform. Note that the delivery partners' objective is captured in the basket value objective, which determines their payment. Although all these objectives connect with the long-term success of the platform, they conflict with each other as they are capturing the different aspects of the multi-sided platform. As discussed in Section 3.1, over-focusing on consumer conversion would hurt consumer retention and marketplace fairness. And we will see in Section 5 that over-focusing on basket value (e.g. by recommending more expensive restaurants) would hurt consumer retention and even backfire. We will demonstrate that these four objectives are critical in capturing the long-term values of the business, and that a delicate balance across the multiple objectives is critical to ensure a healthy ecosystem and the success of the platform in the long run.

In the next, we describe in detail how each objective is modeled as a machine learning outcome[11] in the MO-module. Table 1 summarizes the notations used. Table 2 summarizes the outcomes. It's worth pointing out that although we only discussed the estimation of four outcomes in this section and they are developed in the context of food-delivery platforms, the MO-module is general and can incorporate *any* number of objectives that are of interest to *any* type of platforms.

Notation	Definition and comments
i	Index for consumers
j	Index for restaurants
k	Source of the restaurant, e.g. "Popular near you", or "Single" if appears as a single restaurant
q	Index for a recommendation item, which can be either a single restaurant within the carousel or a whole carousel
z	Context features such as time of day, day of week, meal period, country, geolocation
$O(i,j,k,z)$	(Product-level) Binary random variable taking value 1 if consumer i orders from restaurant j from source k under context z, 0 otherwise
$R(i,j,k,z)$	(Product-level) Binary random variable taking value 1 if consumer i returns to the platform and orders within 28 days of ordering from restaurant j from source k under context z, 0 otherwise
$B(i,j,k,z)$	(Product-level) Continuous random variable taking value as the dollar amount of the basket value if consumer i orders from restaurant j from source k under context z, 0 otherwise
$O(i,k,z)$	(Row-level) Binary random variable taking value 1 if consumer i orders from source k under context z, 0 otherwise
$R(i,k,z)$	(Row-level) Binary random variable taking value 1 if consumer i returns to the platform and orders within 28 days of ordering from source k under context z, 0 otherwise
$B(i,k,z)$	(Row-level) Continuous random variable taking value as the dollar amount of the basket value if consumer i orders from any restaurant from source k under context z, 0 otherwise
N_j	Number of impressions from product j
N_j^1	Number of orders from product j
I	Number of consumers
Q	Number of recommendation items (restaurants or carousels)
$\boldsymbol{x} = \{x_{iq}\}$	Ranking plan, where x_{iq} is the probability of serving item q to consumer i
$\boldsymbol{u} = \{u_{iq}\}$	Uniform ranking plan, where $u_{iq} \equiv \frac{1}{Q}$
$c_{iq}, r_{iq}, b_{iq}, f_{iq}$	Compact forms for the consumer conversion, consumer retention, basket value and fairness outcomes for consumer i and item q

Table 1 Summary of notations.

4.2.2. Prediction Using Machine Learning Models and Bayesian Statistics. Outcomes on individual products might vary given different sources of the product, where *source* is defined as the hierarchy information of the product (e.g. belongs to the "Italian food" carousel[12]). As an example, a restaurant appearing in the "Under 25 minutes" carousel is more appealing to a consumer who is in a hurry than the same restaurant

Outcome	Relevant sides	Level	Notation	Definition
Consumer conversion	Consumers, restaurant partners	Restaurant	$c(i,j,k)$	$\mathbf{P}[O(i,j,k,z)=1]$
		Carousel	$c(i,k)$	$\mathbf{P}[O(i,k,z)=1]$
Consumer retention	Consumers, restaurant partners	Restaurant	$r(i,j,k)$	$\mathbf{E}[R(i,j,k,z)\|O(i,j,k,z)=1]$
		Carousel	$r(i,k)$	$\mathbf{E}[R(i,k,z)\|O(i,k,z)=1]$
Basket value	Restaurant partners, delivery partners	Restaurant	$b(i,j,k)$	$\mathbf{E}[B(i,j,k,z)\|O(i,j,k,z)=1]$
		Carousel	$b(i,k)$	$\mathbf{E}[B(i,k,z)\|O(i,k,z)=1]$
Marketplace fairness	Restaurant partners	Restaurant	$f_r(j)$	$\sqrt{\mathrm{Var}(c_j\|\{O_{jm}\}_{m=1}^{N_j})}$
		Carousel	$f_k(k)$	$\sum_{l=1}^{n} f_r(j_l)\prod_{l'=1}^{l}(1-c(i,j_{l'-1},k))\cdot p_{l'-1,l'}$

Table 2 Summary of outcomes.

appearing in the "Popular near you" carousel, or as a single recommendation. Therefore, we explicitly account for this source effect by proposing to model each outcome for every $(consumer, restaurant, source)$ triplet. Different from usual recommender system settings where predictions are done on $(consumer, restaurant)$ pairs, our proposed triplet-level models are able to capture consumers' heterogeneous behavior with the hierarchical display.

Machine Learning Models for Consumer Conversion, Consumer Retention and Basket Value Outcomes. Using the notations in Table 1, we build machine learning models for consumer conversion, consumer retention and basket value as:

$$\begin{aligned} c(i,j,k) &= \mathbf{E}[O(i,j,k,z)=1] = f_c(a_i, a_j, a_k, a_{ij}, a_{ik}, a_{jk}, a_{ijk}, z), \\ r(i,j,k) &= \mathbf{E}[R(i,j,k,z)|O(i,j,k,z)=1] = f_r(a_i, a_j, a_k, a_{ij}, a_{ik}, a_{jk}, a_{ijk}, z), \\ b(i,j,k) &= \mathbf{E}[B(i,j,k,z)|O(i,j,k,z)=1] = f_b(a_i, a_j, a_k, a_{ij}, a_{ik}, a_{jk}, a_{ijk}, z), \end{aligned} \tag{1}$$

where we drop the dependency on context z for ease of notation. Here a_i represents the set of consumer-level features, a_{ij} represents the interaction history between consumer i and restaurant j, while a_{ijk} represents the interaction history between consumer i and restaurant j conditional on j appears in source k, etc.

For the consumer conversion and retention outcome which are binary classification problems, we adopt gradient boosting decision trees (Friedman 2001, 2002) as the nonlinear prediction function f_c and f_r, which are a family of ensemble methods that combines individual weak CART (Breiman et al. 2017) classifiers. For the basket value objective, we use gradient boosting regression tree as f_b with squared loss as the loss function (Hastie et al. 2009), and truncate the predictions to be non-negative[13]. They achieve a nice balance between predictive power and interpretability. Model training is done using the gradient boosting machine (GBM) on H2O (Click et al. 2017, Hastie et al. 2009), which is a popular distributed in-memory machine learning platform.

A full list of the features and the parameters of the machine learning models can be found in Appendix B.1.2. On a high level, there are three groups of features:

- **RFM features**: We adopt the well-known RFM (recency, frequency and monetary value) paradigm which is shown to have positive association with customer lifetime value (CLV) (Fader et al. 2005). In the context of food-delivery platforms, RFM maps to the time since last order (recency), number of past orders (frequency) and basket value of past orders (monetary value) respectively. We compute the RFM features for all possible combinations of the $(consumer, restaurant, source)$ triplet[14], in order to capture both personalized and non-personalized aspects of RFM features. The features are computed over multiple time horizons to capture the temporal effect of the RFM features. See Fig. 9 in Appendix B.1.2 for an illustration of the RFM features.
- **Contextual features**: We include features such as time of the day, day of the week, meal period, geolocation, language and device as the contextual features capturing the decision-making scenario of the customers. Vertical and horizontal position of the restaurant is also included as a feature to remove the position bias (Fig. 4). During serving time, the position feature is set to 0 (first position)[15]. This is a widely adopted method in recommender systems for removing selection bias, without the need for random ranking data or propensity score modeling which are costly to implement in practice (Zhao et al. 2019).
- **Collaborative filtering features**: We also include collaborative filtering (Breese et al. 1998) features using matrix factorization (Koren et al. 2009) to capture the similarities among consumers and items, based on the assumption that similar consumers like similar products. The output of the matrix factorization algorithm are embeddings for every customer, restaurant and source, which are used as input features for the machine learning models. Details are provided in Appendix B.1.1.

In total, there are about 200 features used in the MO-module. They include both content-based features and activity-based collaborative filtering features, making the model a hybrid recommender system (Burke 2002). The models are retrained daily based on most recent data. Table 9 in Appendix C.3 summarizes the most important features in each model using feature importance scores (Friedman 2001). For the RFM features, the monetary value features (e.g. historical basket values) are an important feature for the basket value outcome, and the recency and frequency features (e.g. historical order count and churn rates) are important features for the conversion and retention outcomes.

Contextual Multi-Armed Bandit for Marketplace Fairness Outcome. In this section, we show how the multi-armed bandit (MAB) framework can be leveraged to define and model the marketplace fairness outcome. For the platform, ensuring good consumer experience usually entails recommending popular and well-established restaurants to the consumers. However, this may lead to low exposure for the new and low-volume restaurants, or an unfair marketplace. The platform therefore faces a challenge of providing equal opportunities for all restaurant partners while ensuring good consumer experience. This problem can be viewed as a trade-off between *exploiting* well-established restaurants and *exploring* new restaurants.

This exploration-exploitation trade-off fits nicely into a contextual multi-armed bandit framework (Langford and Zhang 2007, Auer et al. 2002, Katehakis and Veinott Jr 1987) where every arm is a restaurant. A well-adopted approach for the contextual bandit problem is the upper-confidence bound (UCB) algorithm, where the optimal action chosen at each step is given by

$$j^* = \arg\max_j [Q(j) + \kappa\sigma(j)]. \tag{2}$$

Here $Q(j)$ is the estimated value of action j (i.e. recommending restaurant j), $\sigma(j)$ is the estimated standard deviation of the value of j, and $\kappa > 0$ controls the level of exploration. In our case, $Q(j)$ can be the estimated conversion rate for restaurant j, i.e. $c(i,j,k)$ from the previous section. $\sigma(j)$ measures the uncertainty of $c(i,j,k)$, which intuitively is high for new restaurants. Motivated by this, we propose to *define* the fairness outcome as the uncertainty value $\sigma(j)$, and we would like to maximize the exposure for the restaurants with high $\sigma(j)$.

There are two approaches to estimate the fairness outcome $\sigma(j)$. One approach is to leverage machine learning methods such as deep ensembles (Lakshminarayanan et al. 2017) that estimate a personalized real-time uncertainty. However, such methods are costly and unscalable as they require multiple copies of the same model to be trained and served in real time. They also provide noisy estimates due to the personalized nature. We therefore adopt an analytical approach which provides a fast and robust estimate of the uncertainty. We propose a restaurant-level Bayesian model with posterior inference, in which the fairness outcome is estimated as the posterior variance. Specifically, we assume the impressions on restaurant j follow a Bernoulli distribution with parameter equals the conversion rate. The posterior variance of the conversion rate can then be obtained by aggregating historical impression and order activities on the restaurant, yielding the *marketplace fairness outcome* $f_r(j)$ for restaurant j as

$$f_r(j) = \sigma(j) = \sqrt{\frac{(\alpha_j + N_j^1)(\beta_j + N_j - N_j^1)}{(\alpha_j + \beta_j + N_j)^2(\alpha_j + \beta_j + N_j + 1)}}, \tag{3}$$

where α_j and β_j are the parameters for the prior Beta distribution $\mathcal{B}(\alpha_j, \beta_j)$, and there are N_j impressions on restaurant j, out of which N_j^1 lead to orders. See Appendix B.1.3 for detailed notations and results.

As a sanity check, given α_j and β_j, lower values of N_j lead to *higher* value of the marketplace fairness outcome $f_r(j)$. In other words, the *fewer* impressions a restaurant receives, the *more* uncertain the system is about the estimation of the restaurant's conversion rate, hence the higher value for the marketplace fairness outcome. We discuss the choice for the prior parameters α_j and β_j in Appendix B.1.4. A trailing window of 120 days[16] is chosen for the impression counts N_j and order counts N_j^1.

The benefits of the marketplace fairness outcome $f_r(j)$ are threefold. First, it offers new restaurants more exposure on the platform. Second, it helps the learning of other objectives by introducing more training data on the new and low-volume restaurants. Third, by restricting to a trailing window for counting N_j, it provides a mechanism for adaptively boost new and low-volume restaurants over time: A restaurant will receive a high boost when first entering the platform, with Eq.(2) dominated by the second term ("**exploration**"); As it accumulates enough exposure, the boosting effects dies down and Eq.(2) is dominated by the point estimate $Q(j)$ ("**exploitation**"); Later on, when the restaurant is performing poorly in a certain time period by having a low $Q(j)$, it will lose exposure (i.e. having low N_j again). As a result $f_r(j)$ will go back up due to the decreasing of N_j and increasing of $\sigma(j)$, therefore offering a "second chance" to the restaurant ("**resurrection**").

Remark. We would like to call out that although the marketplace fairness outcome is derived from maximizing conversion under uncertainty and a UCB formulation, the effect of this outcome goes beyond a standard MAB framework. First, the goal of MAB maximizes the expected cumulative gain in the long term under uncertainty of the observed outcome, while the proposed marketplace fairness objective focuses more on the *immediate* impact on new and low-volume restaurants. This property is achieved only through the UCB-like formulation. Other MAB algorithms such as Thompson sampling (Thompson 1933) and epsilon-greedy (Sutton and Barto 2018) do not have the same effect on new and low-volume restaurants. See Appendix B.1.5 for additional analyses. Second, we do not choose a restaurant according to Eq. (2) as a standard UCB algorithm does, but rather just uses the uncertainty component in the UCB formulation to define the marketplace fairness outcome, which will be jointly optimized with other objectives in the R-module later. Third, the marketplace fairness outcome is non-personalized and on a per-restaurant level, while typical MAB frameworks applied to the recommendation setting would require a personalized uncertainty estimate for each $(consumer, restaurant)$ pair. This restaurant-centric property of the marketplace fairness outcome further distinguishes it from typical MAB frameworks. Lastly and most importantly, because of time window mechanism, the marketplace fairness outcome will *always* boost new and low-volume restaurants even if they are performing poorly, in which case they will be given a second chance through the "resurrection" stage as discussed above; while a typical MAB algorithm focuses on identifying the best arm and "bad" restaurants will eventually be ignored. This differentiates the marketplace fairness outcome from a typical MAB procedure, which focuses on lack of exploration when maximizing conversion under uncertainty[17].

4.3. H-Module: Hierarchical Probabilistic Aggregation for Row-Level Outcomes

In the H-module, the product-level outcomes from the MO-module are aggregated into row-level outcomes. This is achieved through a probabilistic hierarchical aggregation by leveraging Bayes' rule and the law of total expectation. An important component in this module is a state-based consumer browsing model, which is described below.

Consumers have limited patience when scrolling through the recommendations. Following the sequential search framework in Weitzman (1979), we assume that the consumer examines the products in the row (carousel) from left to right and one-by-one and may give up at a certain point. The consumer state is modeled as their viewing position (e.g. position X inside row Y). The state transitions are Markovian in that at every state, the consumer takes one of the following three actions: order from the current product, continue browsing the next product, or abandon the whole row[18][19].

We propose a *state-based consumer browsing model*, which outputs a set of scrolling probabilities that a consumer will scroll to the next position inside a row:

$$p_{l,l+1} = \mathbf{P}(\text{the consumer scrolls to position } l+1 \mid \text{currently at position } l). \tag{4}$$

Figure 3 illustrates the state-based consumer browsing model, where the position index starts at 1 and $p_{0,1} = 1$, meaning that consumers always browse the first product in each carousel[20]. Next, we show how to aggregate the product-level outcomes into row-level outcomes using the scrolling probabilities from the consumer browsing model.

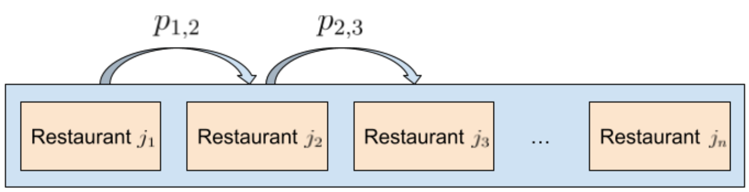


Figure 3 An illustration of the consumer browsing model and the scrolling factors for a carousel.

Row-Level Consumer Conversion. We first derive the conversion outcome $c(i,k)$[21], the probability that the consumer orders from any restaurant[22] in row k under context z. Assuming the restaurant at position l inside the row is indexed by j_l, by the law of total probability we have

$$\begin{aligned} c(i,k) &= \sum_{l=1}^{n} \Big[\mathbf{P}(\text{consumer } i \text{ orders from product } j_l \text{ at position } l \mid \text{scrolls to position } l) \\ &\qquad \times \mathbf{P}(\text{scrolls to position } l) \Big] \\ &= \sum_{l=1}^{n} \Big[c(i,j_l,k) \prod_{l'=1}^{l} \mathbf{P}(\text{consumer } i \text{ didn't order at position } l'-1, \text{ and scrolls to position } l') \Big] \\ &= \sum_{l=1}^{n} \Big[c(i,j_l,k) \prod_{l'=1}^{l} (1 - c(i,j_{l'-1},k)) \cdot p_{l'-1,l'} \Big], \end{aligned} \tag{5}$$

where we define $c(i,j_0,k) = 0$. Equation (5) is intuitive when viewing each term in the summation one by one: The first term, $c(i,j_1,k)$, is the probability that the consumer orders from the first product in the row; The second term, $(1 - c(i,j_1,k)) \cdot p_{1,2} \cdot c(i,j_2,k)$, is the probability that the consumer abandons the first product but scrolls to the second position and orders, etc[23].

Row-Level Basket Value. By law of total expectation, the expected basket value of a row can be decomposed as the sum of the basket value at each position inside the row:

$$\begin{aligned}\mathbf{E}[B(i,k,z)] &= \sum_{l=1}^{n} \mathbf{E}[B(i,j_l,k,z)|O(i,j_l,k,z)=1]P[O(i,j_l,k,z)=1] \\ &= \sum_{l=1}^{n} b(i,j_l,k)P[O(i,j_l,k,z)=1],\end{aligned} \tag{6}$$

which is a weighted combination of the basket value outcome of each individual product (restaurant) $b(i,j_l,k)$ inside the row, with the weights being the conversion probability at that position.

The basket value outcome of the row is therefore

$$\begin{aligned}b(i,k) &= \mathbf{E}[B(i,k,z)|O(i,k,z)=1] = \mathbf{E}[B(i,k,z)]/\mathbf{P}[O(i,k,z)=1] \\ &= \sum_{l=1}^{n} \frac{P[O(i,j_l,k,z)=1]}{\sum_{l=1}^{n} P[O(i,j_l,k,z)=1]} b(i,j_l,k),\end{aligned} \tag{7}$$

where $P[O(i,j_l,k,z)=1] = c(i,j_l,k)\prod_{l'=1}^{l}(1-c(i,j_{l'-1},k))\cdot p_{l'-1,l'}$ is the probability that the consumer scrolls to position l inside the row and orders from the product j_l, as computed in Eq.(5). Therefore, the row-level basket value is effectively a *weighted average* of the expected basket values of individual products inside the row, with the weights proportional to their predicted conversion at each position while accounting for the consumer's scrolling behavior.

Row-Level Consumer Retention. Following the same derivation above, the row-level consumer retention outcome can be computed as

$$r(i,k) = \sum_{l=1}^{n} \frac{P[O_l(i,k,z)=1]}{\sum_{l=1}^{n} P[O_l(i,k,z)=1]} r(i,j_l,k). \tag{8}$$

Row-Level Marketplace Fairness. The row-level marketplace fairness outcome $f_c(k)$ is slightly different as it is not conditioned on the consumer placing an order. By the same law of total expectation as in Eq.(5), we have

$$f_c(k) = \sum_{l=1}^{n} f_r(j_l)\mathbf{P}(\text{consumer } i \text{ scrolls to position } l) = \sum_{l=1}^{n} f_r(j_l)\prod_{l'=1}^{l}(1-c(i,j_{l'-1},k))\cdot p_{l'-1,l'}, \tag{9}$$

which is a weighted sum of product-level fairness outcomes $f_r(j_l)$ at each position inside the row, with the weights being the probability that the consumer scrolls to that position.

To sum up, the H-module provides an hierarchical probabilistic aggregation approach for estimating each outcome on the row level (carousels), as an aggregation of product-level outcomes within the row, which are obtained from the MO-module. A nice and important property for these row-level predictions is that they are calibrated against those for the single products. In other words, the predictions for a row (carousel) of restaurants is comparable to those for the single restaurants, so that one is able to rank them in a mixed and holistic fashion. This is critical for ensuring consistent consumer experience across different levels of aggregation of the products, and providing levels of transparency to the restaurant partners.

4.4. R-Module: Constrained Multi-Objective Optimization for Within-Row and Across-Row Ranking

4.4.1. Overview. The goal of the final module, R-module, is to determine a personalized layout of the hierarchical homepage while considering the multiple conflicting objectives from previous modules. We formulate the multi-objective hierarchical recommendation problem as two sets of ranking problems: one for *within-row ranking* which determines the ordering of restaurants within a carousel, and one for *across-row ranking* which determines the ranking of carousels[24].

The R-module is a universal approach that works for both ranking problems. Therefore, we introduce a general notation q as the index of a recommendation item. For within-row ranking, q indexes single products (restaurants within a carousel). For across-row ranking, q indexes rows of products (carousels). We also introduce subscripts for more compact notations. For example, c_{iq} denotes the conversion rate for consumer i on item q, and b_{iq}, r_{iq}, f_{iq} are defined similarly.

The problem of determining the ranking for a set of items is a combinatorial problem and usually infeasible to be solved in real-time for large-scale online platforms. As a result, the common practice in recommendation system literature is to solve the combinatorial ranking problem with a greedy solution, where every item is assigned a ranking score, and all the items are ranked according to the ordering of the scores (Liu et al. 2009). The greedy solution reduces the ranking problem from combinatorial complexity ($\mathcal{O}(n!)$) to log-linear complexity ($\mathcal{O}(n \log n)$) so that it is feasible for real-time large-scale systems. We adopt a probabilistic score as the ranking score. Specifically, the output of the R-module is a ranking plan $\boldsymbol{x} = \{x_{iq}\}$ which is a set of personalized scores, with x_{iq} being the probability of recommending item q to consumer i. Using probability to recommend as the ranking score is a popular choice in the recommender systems literature, as it enables one to compute the expected total outcome as shown in the next subsection. It is also used by the reinforcement learning (RL) based recommenders as the recommendation policy is always formulated as a probability distribution over all recommendation items (Chen et al. 2019)[25].

Given a ranking plan $\boldsymbol{x}$, the homepage layout is determined via a two-step procedure:

- **Step 1 (within-row ranking)**: Every product q within a row is assigned a ranking score x_{iq} from the R-module, which is a function of the product-level outcomes from the MO-module. They are then ranked according to the descending order of x_{iq}.
- **Step 2 (across-row ranking)**: Given within-row ranking, first compute the row-level outcomes from the H-module[26]. Then every row q is assigned a ranking score x_{iq} from the R-module, which is a function of the row-level outcomes from the H-module.

Next, we describe the R-module which takes input from the MO-module and H-module, and outputs a multi-objective ranking score $\boldsymbol{x} = \{x_{iq}\}$ for every product and every row. We first present the result here, and elaborate the optimization procedure in the next section and Appendix B.2.2.

Proposition 1 *Ranking according to x_{iq} is equivalent to ranking according to $\tilde{x}_{iq}$, where*

$$\tilde{x}_{iq} = c_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_f f_{iq}, \quad \forall i, q. \tag{10}$$

Here c_{iq}, r_{iq}, b_{iq} and f_{iq} are outputs from the MO-module (for within-row ranking) and H-module (for across-row ranking). λ_r, λ_b and λ_f are parameters related to the trade-off among multiple objectives which will be discussed in the next section.

4.4.2. Formulation and solution. With the R-module, we formulate the multi-objective hierarchical ranking problem as two sets of constrained optimization problems (one for within-row ranking and one for across-row ranking), with the constraints being the amount of sacrifice that the platform is willing to make in some objectives while optimizing for others.

For any ranking plan $\boldsymbol{x}$, its expected total numbers of orders, total gross bookings, total consumer retention and total fairness can be computed as

$$\begin{aligned} C(\boldsymbol{x}) &= \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}, \qquad B(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}b_{iq}, \\ R(\boldsymbol{x}) &= \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}r_{iq}, \qquad F(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}f_{iq}. \end{aligned} \tag{11}$$

These are the four objectives of interest, which are aggregations of the individual outcomes across all consumers and all items. Let

$$C^* = \max_{x\in\mathcal{E}} C(\boldsymbol{x}), \quad B^* = \max_{x\in\mathcal{E}} B(\boldsymbol{x}), R^* = \max_{x\in\mathcal{E}} R(\boldsymbol{x}), \quad F^* = \max_{x\in\mathcal{E}} F(\boldsymbol{x}), \tag{12}$$

be the optimal values for these objectives, where $\mathcal{E} = \{\boldsymbol{x} : x_{iq} \geq 0, \sum_q x_{iq} = 1, \forall i\}$ is the feasible region for $\boldsymbol{x}$. We formulate the multi-objective ranking problem as a *constrained optimization problem*:

$$\max_{\boldsymbol{x}\in\mathcal{E}} \; C(\boldsymbol{x}) \quad \text{s.t.} \quad B(\boldsymbol{x}) \geq \alpha_b B^*, \; R(\boldsymbol{x}) \geq \alpha_r R^*, \; F(\boldsymbol{x}) \geq \alpha_f F^*, \tag{13}$$

where $0 < \alpha_b, \alpha_r, \alpha_f < 1$ specifies the amount of *tolerable trade-off* for $B(\boldsymbol{x})$, $R(\boldsymbol{x})$ and $F(\boldsymbol{x})$ when optimizing for $C(\boldsymbol{x})$[27]. The linear programming problem in Eq.(13) can be viewed as a multi-objective optimization problem (Sawaragi et al. 1985). In Appendix B.2.1, we prove that the Pareto frontier[28] between any two objectives is concave, so that a small sacrifice in one objective can potentially lead to big improvement in the other.

Eq.(13) has $I * Q$ number of variables, which can be huge given millions of consumers (I) and thousands of items (Q). This causes scalability issues for solving and serving the solutions in real time for large-scale online platforms. To tackle this challenge, we adopt the trick in Agarwal et al. (2012) and add a quadratic penalty term to the objective function which leads to analytical solutions[29] for $\boldsymbol{x}$. The detailed optimization procedure and solution are provided in Appendix B.2.2. By leveraging KKT conditions, the final ranking function is reduced to what is shown in Prop. 1 above.

Remark. Taking a closer look at Eq.(10), the ranking function is essentially a *weighted combination* of the multiple objectives[30]. This analytical form enables us to efficiently serve a large-scale optimization problem online, without the need to solve a huge-scale linear programming problem in real time. This is especially important in our hierarchical setting where the constrained optimization problem needs to be solved twice and sequentially (one for within-row ranking and one for across-row ranking). We also don't need to compute B^*, R^* and F^* with the final ranking score, which further saves computation. For the objective weights λ_b, λ_r, and λ_f, while we can obtain them as functions of α_b, α_r and α_f by solving a linear system as shown in Appendix B.2.2, it can be prohibitively expensive due the large scale. In practice, we treat λ_b, λ_r and λ_f as tuning parameters directly to reduce computation, without the need to specify α_b, α_r and α_f. In Section 5 below, we show that λ_b, λ_r and λ_f can be used as knobs to make principled and quantifiable trade-offs across any number of objectives.

As another remark, compared with Agarwal et al. (2012) which proposes the quadratic penalty trick for multi-objective optimization, the R-module solves the multi-objective recommendation problem in a *hierarchical* setting with heterogeneous recommendation items. Specifically, we formulate the hierarchical multi-objective recommendation problem as two sets of constrained optimization problems: one for within-row ranking and one for across-row ranking. They are then solved through an innovative formulation of probabilistic consumer behavior modeling combined with constrained optimization. This is the main methodological contribution of the R-module.

4.5. Training and Optimization Considerations of MOHR

We now describe some additional training and optimization details of MOHR. The world is constantly evolving and so are consumer preferences. As such, machine learning based recommender systems are expected to adequately capture the dynamics of the evolving system in order to be successful in the long-term. In the next we discuss details of the feature updates and model updates in the MOHR framework to achieve this goal.

Feature Updates. The consumers are expected to see a different homepage layout with different recommendations every time they open the app or refresh the homepage within the same visit. This is realized by feature updates from the MO-module. With every homepage refresh, a new session is generated together with the updated features[31]. Specifically, the output of the MO-module, i.e. the ML predictions for the product-level outcome, would be different for different sessions as the input feature values are different. Therefore, the predictions for the row-level outcomes would also be different from the H-module, which means the input to the R-module would be different. As a result, a new ranking (output of the R-module) would be generated with every homepage refresh to capture any real-time changes in the three-sided marketplace. The analytical solution from the R-module is extremely helpful to enable this property, as one just needs to plug in the ML predictions into Eq. (10) without the need to solve huge-scale linear programming problems with every page refresh.

Model Updates. Due to their black-box nature, a big weakness of ML models is their inability to reason about the data generation process as in a structural model (Farrell et al. 2020). Indeed, they rely on exploiting correlations in the data to make predictions, therefore there is no guarantee that the model primitives are invariant. To tackle this challenge, we implement the following practices in updating the ML models in the MOHR framework. First, all ML models are retrained daily with past 30 days observation to capture any temporal shifts in the data. Second, although all ML models are trained using observational data, we correct the off-policy bias by including both vertical and horizontal position as input feature to the ML models during training, which are set to 0 or the true horizontal position during serving (as described in Section 4.2 and 4.4). Third, all ML models are validated with the latest out-of-sample data: With 30 days training data, the first 29 days are used for training, and the last 1 day is used for validation. The temporal split of training and validation dataset mimics the production environment, where models trained using data up to yesterday are served on today's traffic.

These practices help make sure that off-policy bias is removed to the maximal extent, and the ML models are always accurately predicting the outcome of interest in real time. In other words, our **real-time feature updating** and **continuous model retraining with off-policy correction** strategy helps the MOHR framework to capture the evolving dynamics of the multi-sided marketplace. The fact that this sequential retraining strategy performs well in live experiments (Section 5) validates that the MOHR framework is able to capture at least the local equilibrium well.

In addition, the state-based consumer browsing model in the H-module is estimated exogenously. We think this is a reasonable assumption to make in our case, as the recommendation algorithm changes are typically *imperceptible* to the consumers (they always see different recommendations with every homepage refresh, even under the same recommendation algorithm). This is also validated in the data from the food-delivery platform we work with.

5. Results

5.1. Experiment Setup and Performance Measures

5.1.1. Experiment Setup. To the best of our knowledge, we are the first to propose a hierarchical recommender that ranks contents of various levels of aggregation using a single holistic framework. The closest baseline is the latest production recommender system at the company. It decomposes the hierarchical recommendation problem into two parts, where carousels and individual restaurants are ranked separately using disjoint state-of-art hybrid machine learning recommendation algorithms, with consumer conversion as the single objective. All carousels are ranked above all single restaurants with another machine learning model determining how many carousels to display. See Appendix C.1 for a detailed description.

Our experiments were conducted over 28 days[32] in June 2019 on 2% of the company's global consumers[33]. Every consumer in the experiment traffic is assigned with a unique consumer identifier[34] which is randomly hashed into the treatment group or control group. The treatment group information is logged together with consumers' activities on the platform during the experiment period.

5.1.2. Performance Measures and Statistical Hypothesis Testing. Table 3 summarize a list of metrics for the online experiments, which correspond to the multiple objectives for the multiple sides in the three-sided marketplace that are critical to the business.

A consumer may visit the app multiple times during the experiment period. Let S_i be the number of sessions that consumer i generates during the experiment period, and O_{is}, B_{is} be the binary indicator for whether consumer i orders from session s, and the basket value for the session (0 if there is no order) respectively. To measure consumer retention, we let R_{is} be the binary indicator of whether consumer i returns to the app and places another order in the next 14 days following the current session s. Note that basket value and retention are measured only on ordered sessions. For marketplace fairness, we measure the performance of the new restaurants on the platform, which are those joining the platform within 21 days of the experiment start date. Specifically, we measure the percentage of the overall impressions and orders from the platform that are on those newly onboarded restaurants.

Note that the sessions generated by the same consumer are correlated with each other as they reflect the behavior of the same consumer. Therefore, ratio metrics such as conversion

Measure	Definition / Explanation	Relevant sides
Conversion rate	$\frac{\sum_i \sum_s O_{is}}{\sum_i S_i}$	Consumers, restaurant partners
Basket value per order	$\frac{\sum_i \sum_s O_{is} B_{is}}{\sum_i \sum_s O_{is}}$	Restaurant/delivery partners
Retention rate	$\frac{\sum_i \sum_s O_{is} R_{is}}{\sum_i \sum_s O_{is}}$	Consumers, restaurant partners
Orders per consumer	$\frac{1}{I} \sum_i \sum_s O_{is}$	Consumers, restaurant/delivery partners
New restaurant impression ratio	% of impressions on new restaurants	Restaurant partners
New restaurant order ratio	% of orders on new restaurants	Restaurant partners

Table 3 List of Measurements.

rate, basket value per order and retention in Table 3 are *not* from i.i.d. samples. We explicitly account for this intra-consumer correlation when computing the variance for those test statistics in hypothesis testing for the online experiments. The resulting p-values are larger than when the examples are treated as i.i.d., therefore our tests are more rigorous and conservative and less likely to claim the treatment as effective. See Appendix C.2 for details.

5.2. Offline Analysis

5.2.1. Hyperparameter Selection with Offline Replay. The ranking function in Eq.(10) contains three hyperparameters λ_b, λ_r and λ_f controlling the relative importance of the objectives. It is costly to run online experiments to select the optimal values for these hyperparameters. It is also risky to serve a new framework in production with arbitrary hyperparameters before we have an understanding of their effects on the platform. Therefore, it is necessary to develop an offline evaluation procedure to pick hyperparameter values for MOHR to be experimented online.

The data for offline evaluation is critical for the quality of the evaluation, as we are faced with the typical challenge of position bias (Ursu 2018) and off-policy evaluation (Strehl et al. 2010, Schnabel et al. 2016). To understand position bias, the company has set aside a small percentage of random sessions for random ranking, where the vertical list of restaurants are ranked completely at random. Figure 4 confirms position bias on the number of impressions, number of orders and conversion rate on the random ranking data, showing that the same restaurant at different positions may appeal very differently to the consumers. Position bias causes challenges for performing off-policy evaluation. For example, if MOHR framework predicts to rank restaurant j_0 at a certain position for a consumer, but the existing production system has never presented restaurant j_0 at that position to her, then it is hard to predict whether the consumer would have ordered from that restaurant.

We adopt the *offline replay* method proposed by Li et al. (2011), which utilizes random data for off-policy evaluation, but in the context of bandit algorithms. Offline replay method is a popular alternative to inverse-propensity weighting (IPW) based methods (Yang et al. 2022, Yoganarasimhan et al. 2022, Hitsch and Misra 2018, Simester et al. 2020a) for off-policy evaluation, when the underlying logging policy is not stochastic. We adapt their method to the ranking scenario. Specifically, if it happens that the new algorithm chooses the same restaurant to be ranked on the top position as in the random ranking data[35], then that event is retained and will be used for estimating the performance of the new algorithm. In other words, the replay method is essentially looking for events in the random ranking data that can serve as "replaying" the ranking under the new algorithm to be evaluated. The replay method is proven to provide unbiased offline evaluation (Li et al. 2011) without running online experiments.

Figure 5 shows the offline Pareto frontiers from the offline replay analysis[36]. As expected, larger values of λ_b and λ_f result in better basket value and marketplace fairness at the cost of the conversion objective[37]. As consumer conversion is the top-tier business metric for the company, for online experiments we pick the values for λ_b and λ_f such that the drop in conversion is minimal.

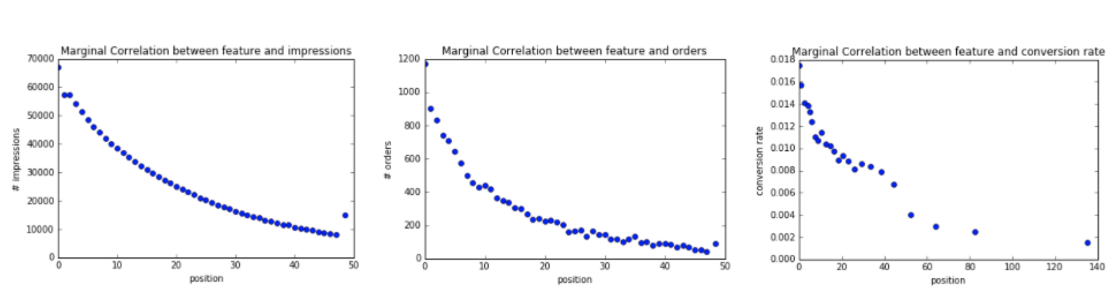


Figure 4 Impressions (left) , orders (middle) and conversion (right) vs. position on random ranking data.

5.2.2. Feature Importance and Model Performance. See Appendix C.3 for the model performance and feature importance of the machine learning-based objectives in the MO-module.

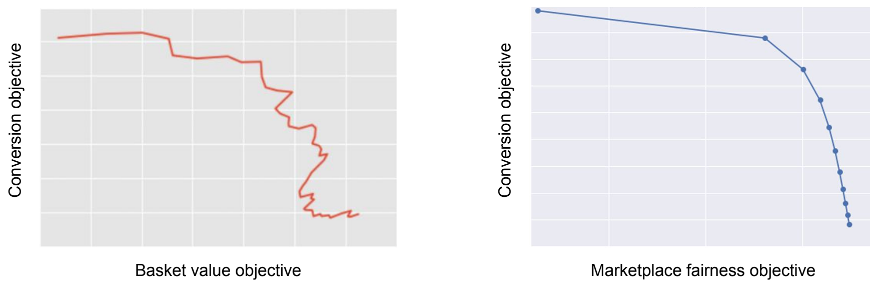


Figure 5 **Offline Pareto frontiers generated by the offline replay analysis. Left: conversion objective (y-axis) vs. basket value objective (x-axis); Right: conversion objective (y-axis) vs. marketplace fairness objective objective (x-axis); We omit the tick values for business compliance reasons. The upper and right directions correspond to better objective values.**

5.3. Online Experiment Results

We present the results from the large-scale online controlled experiment (i.e. A/B testing) at the company. Compared with the latest production system at the company, there are two major changes from the MOHR framework: recommending for the three-sided marketplace using multi-objective optimization (MO-module and R-module), and recommending hierarchical items using hierarchical modeling (H-module). To understand the contributions of the two changes separately, we conduct three sets of experiments: (1) **multi-objective recommendation ("MOR")**: the MOHR framework *without* the hierarchical hierarchical modeling component (H-module), where the restaurants are ranked using the multi-objective ranking score in Eq.(10); (2) **hierarchical single-objective recommendation ("H")**: the MOHR framework *without* the multi-objective optimization component (MO-module and R-module), where the contents are ranked together holistically by the H-module, with conversion as the single objective; (3) **multi-objective hierarchical recommendation ("MOHR")**: the full MOHR framework combining (1) and (2).

We also conduct additional robustness checks for the experiment, including a randomization check and elimination of possible novelty effects. See Appendix C.5 for more details.

5.3.1. Results on Multi-Objective Recommendation ("MOR").

We experiment with adding basket value, consumer retention and marketplace fairness objectives to the production system which uses conversion as the single objective. Without the H-module, the MOR framework is not applicable to rank the carousels (vertically) together with the restaurants. Therefore, we keep the production system's ranking for the carousels, while use the MOR framework for restaurant-level rankings, namely within-carousel (horizontal) ranking and vertical single restaurant ranking.

Constrained by the number of online experiments we can run on live consumer traffic, we adopt a greedy approach in understanding the effect of incorporating each new objective into the system. Specifically, we sequentially add more additive terms in the ranking function in Eq.(10).

Table 4 reports the metrics defined in Table 3 with statistically significant[38] differences between each treatment and control group. We see that by carefully picking the weights for each objective, we are able to achieve Pareto improvements for the three-sided marketplace without hurting consumer conversion. With the basket value objective, we observe a 0.5% relative increase in average basket value per order. In particular, the average basket value of the *top* recommended restaurant has increased by 4.5%, confirming the position effect of the treatments. With the retention objective, we observe a 0.7% relative increase in consumer 14-day retention, indicating that the consumers are coming back to the platform and ordering more often, which also leads to a 0.8% increase in orders per consumer. Lastly with the marketplace fairness objective, the number of impressions and orders on the new restaurants are more than doubled, increasing by 150% and 108% respectively, without a significant drop in the performance of the well-established restaurants on the platform. The fact that introducing the marketplace fairness objective by boosting new restaurants does not significantly hurt consumer conversion is an interesting result to us. This is explained by two observations. First, it has been shown in Appendix B.2.1 that the Pareto frontier for the constrained optimization problem is concave, in that a small sacrifice in one objective can lead to large improvements in others. In this case, the Pareto frontier is concave enough, so that we are able to achieve a significant improvement in marketplace fairness without hurting other objectives significantly. Second, this can be explained as the benefit of consumer exploration (Chen et al. 2021), where boosting new contents helps the consumers discover new interests, and arguably does not hurt consumer experience – sometimes even improving it.

The combined impact for MOR by including all the objectives is summarized as the last column in Table 4. Note that only the relative changes of the metrics are reported, as we are not allowed to reveal the actual values of the key business metrics for compliance reasons. Although the relative changes in the key metrics are small (less than 1%), they

	Basket value	Consumer retention	Marketplace fairness	Combined
Conversion rate	-	-	-	-
Basket value per order	+0.5%	-	-	+0.5%
Retention rate	-	+0.7%	-	+0.7%
Orders per consumer	-	+0.8%	-	+0.8%
New restaurants impression ratio	-	-	+150%	+150%
New restaurants order ratio	-	-	+108%	+108%

Table 4 Results on multi-objective recommendation ("MOR"). Metric differences that are statistically significant at 95% confidence interval are reported, in the form of relative changes over the control group.

translate to considerable business impact given the large scale and consumer base of the company's global platform. Specifically, the MOR framework has led to \$1.3 million weekly gain in revenue.

5.3.2. Results on Hierarchical Single-Objective Recommendation ("H"). A key input to the H-module is the consumer browsing model, which outputs the scrolling factors $p_{l,l+1}$ at each position l as defined in Eq.(4). We adopt a global estimation procedure that estimates a set of non-personalized scrolling factors for the consumer browsing model. Specifically, at each position l inside the carousel, we compute the ratio of the impressions happening at position l that are followed by an impression event at position $l+1$ as the estimate for $p_{l,l+1}$:

$$\hat{p}_{l,l+1} = \frac{\text{number of impressions happened at position } l+1}{\text{number of impressions happened at position } l}. \tag{14}$$

Appendix C.4 reports the estimated consumer scrolling factors. In practice, we find the global estimation works well. In addition to consumer conversion, we monitor two other metrics that are related to consumers' conversion behavior and the quality of the recommendations: *average vertical order position* and *search rate*. The former measures the average vertical position of an order in the homepage, and the latter measures the percentage of the sessions where the consumers go to the search tab, which is a signal that the recommendations on the homepage are not relevant or interesting to them.

Metric	Conversion rate	Average vertical order position	Search rate
Relative change over control	+1.5%**	-5.7%***	-0.9%***

Table 5 Results on hierarchical single-objective recommendation ("H"). Metrics are reported as relative changes over the control group. $^{***}p < 0.01$, $^{**}p < 0.05$.

From Table 5, the hierarchical single-objective recommender improves conversion rate by 1.5%, which translates to $1.1 million weekly gain in revenue. There is a significant 5.7% reduction in average vertical order position and 0.9% reduction in search rate, indicating that the recommendations on the homepage are of higher relevance so the consumers don't need to scroll as much[39] or go to the search page to find what they want.

Ablation Study on Consumer Browsing Model. We also experiment with ablating the consumer browsing model in the H-module. Specifically, we let $p_{0,1} = 1$ and $p_{l,l+1} = 0$ for $l > 0$, instead of using $\hat{p}_{l,l+1}$ in Eq.(14) as the consumer scrolling factors. Using the production recommender system as the baseline, the ablated MOHR framework decreases the average order position by 5.5%, which confirms the benefits of recommending carousels and single restaurants intelligently together, but does not change any other business metrics significantly. Compared with the results in Table 5 where the H-module increases conversion rate, retention of new consumers and decreases search rate, the result suggests that the hierarchical consumer browsing model is critical for the improvements from the H-module.

5.3.3. Results on the Full MOHR. Table 6 summarizes the results on the full MOHR as illustrated in Fig.2. We observe Pareto improvements in all key metrics, which together translates to $1.5 million weekly gain in revenue. Note that the improvements in conversion rate (+0.5%), average vertical order position (-3.2%) and search rate (-0.8%) are smaller compared with H-module only (Table 5). This is an expected result of the trade-off between the additional objectives (basket value, retention and marketplace fairness) and the original conversion objective, which also explains the fact that the revenue gain from the MOHR framework ($1.5 million weekly) is less than the sum of that from MOR ($1.3 million weekly) and H ($1.1 million weekly) treatment groups. Nevertheless, we would like to emphasize that compared with the latest production recommender system, our MOHR framework is able to deliver *Pareto improvements* on all key business metrics at no cost to any of the objectives or any sides in the marketplace.

While MOHR effectively pushes forward the Pareto frontier for the three-sided marketplace, trade-offs still exist as they are the nature of multi-objective optimization. To better understand how the objective weights in the MOHR framework moderate the trade-offs among the online metrics, we conduct additional experiments on the basket value objective with varying λ_b. There are three findings as shown in Fig.6. First, with larger λ_b, the average *predicted* basket values of the recommended restaurants are also larger (Fig.6a)

Metric	Conversion rate	Basket value per order	Retention rate
Relative change	+0.5%**	+0.5%***	+0.7%***
Metric	Orders per consumer	New restaurants impression ratio	New restaurants order ratio
Relative change	0.9%***	+150%***	+108%***
Metric	Average vertical order position	Search rate	
Relative change	-3.2%***	-0.8%**	

Table 6 Results on the full multi-objective hierarchical recommender ("MOHR"). Metrics are reported as relative changes over the control group. $^{***}p < 0.01$, $^{**}p < 0.05$.

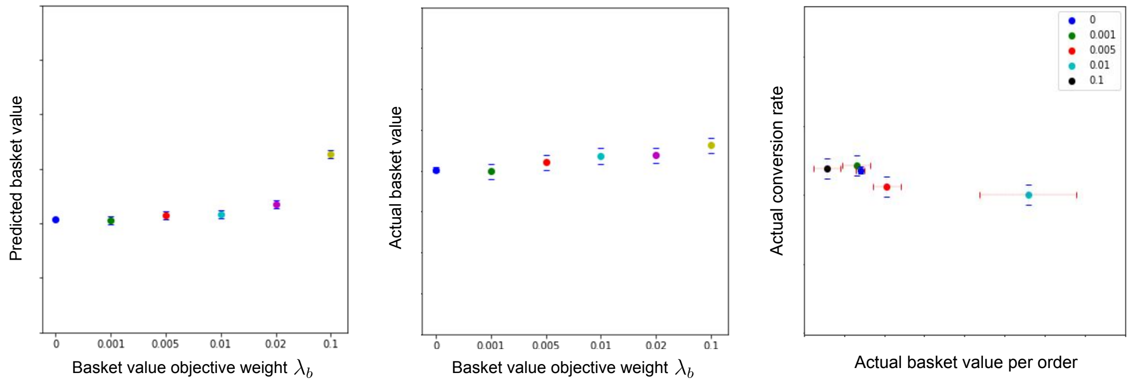
as expected. As a result, the *actual* basket size per order is also increased (Fig.6b). This confirms the effectiveness of the basket value objective and its weight λ_b as the tuning parameter. Note that the increasing trend in the actual basket value (Fig.6b) is less that in the predicted basket value (Fig.6a). This is expected as the recommendation algorithm has full control on what can be shown (predicted basket value), but only partial influence on what the consumers will order (actual basket value). Second, within a reasonable range of λ_b values, we see a trade-off between conversion rate and basket value (Fig.6c) in the online metrics, corroborating the findings from the offline analysis. Finally and most interestingly, when the weight λ_b is huge so that the ranking function in Eq.(10) is dominated by the basket value objective[40], we observe a significant +2.1% increase in search rate, and a 2.7% drop in retention of new consumers. This suggests two consequences when more expensive restaurants and carousels appear in the homepage recommendation: Consumers are more likely to abandon the homepage recommendation and search to order instead; New consumers (who are not yet familiar with the platform) are left with an impression that the selections on the platform are beyond their affordability, hurting their willingness to come back in the future. In other words, aggressive boosting of the objectives may backfire and hurt both the long-term consumer experience and the business.

Because of its significant business impact, the MOHR framework has been deployed globally by the company and is currently serving as its recommender system for the homepage. It was one of the company's most successful launches over the past few years.

6. Discussion

6.1. Research Contributions

This paper proposes a general recommendation framework that addresses two of the most prominent challenges in multi-sided personalization platforms, namely multi-sided trade-off and hierarchical homepage. We propose MOHR, which is a model-based recommendation



(a) Average **predicted** basket value of the recommendation.

(b) Average **actual** basket value of the recommendation.

(c) Actual trade-off between consumer conversion and basket value.

Figure 6 Additional experiments on the basket value objective. Tick values are omitted for business compliance.

framework combining machine learning, hierarchical probabilistic aggregation and multi-objective optimization. In the first step (MO-module), we develop machine learning models for real-time personalized predictions of the multiple objectives at individual product level, with content-based, collaborative-filtering based and real-time contextual features. In the second step (H-module), we generate predictions for rows of products. Specifically, row-level outcomes are modeled as an aggregation of product-level outcomes, using a state-based consumer browsing model which captures consumers' browsing patterns on the homepage. In the final step (R-module), we formulate the multi-objective recommendation problem as a constrained multi-objective optimization problem, taking as input the predictions from the previous modules. The variables are the probabilities of serving each hierarchical item to each consumer, and the constraints specify the amount of tolerable trade-off among the multiple objectives. To deal with scalability challenges, We adopt a quadratic penalty term for the objective function, the solution becomes a combination of the multiple objectives with an analytical form. Each objective is associated with a weight, which we treat as tuning parameters controlling the trade-off across multiple objectives. The output of the framework is a hierarchical ranking function that accounts for consumers' browsing patterns and combines multiple objectives, providing recommendations on heterogeneous and hierarchical products that are optimized for the multi-sided marketplace.

Methodologically, we formulate the problem of "ranking for hierarchical display and with multiple objectives" as two sets of constrained optimization problems: one for within-row ranking and one for across-row ranking. We then solve the multi-objective optimization problem in a hierarchical setting through an innovative formulation of probabilistic consumer behavior modeling and constrained optimization. The benefits of the proposed MOHR framework are three-fold. First, it provides a general and mathematically principled way to model and optimize for the multiple sides in the marketplace in order to optimize the long-term values of the platform. The weights associated with each objective can be treated as tuning parameters, which offers practitioners full control over the trade-off across multiple objectives. The offline Pareto frontiers generated by the replay analysis further facilitates the understanding and decision-making under the multi-sided trade-off when online experiments are expensive. Second, the hierarchical probabilistic aggregation approach guarantees the interpretability of the ranking function, and that the predictions for rows of products are calibrated against those for the single products. This ensures consistent consumer experience across different levels of aggregations, and provides transparency to the restaurant partners. Lastly, the analytical solution from the R-module provides a fast and efficient way to do hierarchical recommendation without the need to solve huge linear programming problems online, making it possible to serve the MOHR framework in any large-scale online systems in real time.

6.2. Managerial Implications

Managerially, we show that long-term optimization via recommender systems can be achieved through a multi-objective approach, as opposed to existing single-score based targeting approaches. In particular, it is beneficial to explicitly model and optimize multiple conflicting aspects of the platform in order to maintain a healthy ecosystem and be successful in the long term.

Our proposed MOHR framework is general, flexible and can be readily applied to other recommendation applications within and outside the food delivery industry. Doordash and Grubhub as examples of other food delivery platforms, YouTube as a video streaming platform, and Airbnb as a peer home-sharing platform, are all operating in multi-sided marketplaces and the products for recommendation can be heterogeneous and hierarchical. The MOHR framework is readily applicable to these platforms, with any number of objectives. We would like to emphasize that our proposed framework is *not* subject to the multi-sided settings. In fact, most firms today care about multiple objectives such as short-term consumer engagement, long-term consumer retention and satisfaction. MOHR is readily applicable to these settings as well. The holistic framework also reduces the burden of maintaining separate machine learning systems for recommending products of different levels of aggregations.

Components of the MOHR framework can be applied in a modularized fashion. Section 5.3 demonstrates that a subset of its components, namely MOR and H, can act as a complete framework to address a particular challenge. Therefore, if a platform is only concerned with multi-objective recommendation in a multi-sided marketplace but the recommendation contents are not hierarchical, it can adopt the MOHR framework without the H-module (i.e "MOR" in Section 5.3.1); if a platform is only concerned with hierarchical recommendation but do not need to optimize for more than one objective, it can adopt only the H-module (i.e "H" in Section 5.3.2).

Results and analysis under the MOHR framework provide insights on the trade-offs among multiple objectives in a multi-sided marketplace. On one of the world's largest food delivery platforms, we experimented with objectives including conversion and retention for the consumers, marketplace fairness for the restaurant partners, and earnings for the delivery partners. Compared with the latest production system, the MOHR framework is able to achieve Pareto improvements on all outcomes. In particular, it improves long-term consumer experience (retention), marketplace fairness and partner earnings without significantly impacting consumers' short-term engagement (conversion). Within the MOHR framework, trade-offs exist as a natural outcome for optimizing multiple conflicting objectives. As it is expensive to generate the full Pareto frontiers in online experiments, we propose to adopt offline replay analysis to generate Pareto frontiers using offline data, to help understand the trade-off between multiple objectives. We also observe that if the weight for a particular objective is too large, it will hurt overall consumer experience and backfire. For example, an aggressive boost of the basket value objective leads the new consumers on the platform to believe that the selections on the platform are expensive, hurting the long-term experience of those with low price elasticity. Insights like these help inform better managerial decision-making on multi-sided platforms.

Lastly, we would like to call out the connection between the proposed marketplace fairness objective and the cold-start problem as a well-known challenge for recommender systems. For new products or products with low exposure on the platform (i.e. cold-started items), the marketplace fairness objective assigns a high value to them, leading to an increased exposure. This also helps the machine learning models generate more accurate predictions for the new products. Over time, as the new products accumulate more exposure, the marketplace fairness objective assigns a lower value to them, leading to a "graduation" from the cold-start phase. As a result, the items will be recommended mainly based on the values of the other objectives. Therefore, the proposed marketplace fairness objective addresses the cold-start challenge in a dynamic, adaptive and data-dependent fashion. In addition, the marketplace fairness objective does not necessarily hurt consumer experience as observed in our experiments, which can be explained as the benefit of consumer exploration (Chen et al. 2021), where boosting new contents helps the consumers discover new interests and potentially improves long-term consumer experience.

6.3. Challenges, Limitations and Future Research

A challenge and limitation of the MOHR framework is its scalability with large numbers of objectives. With an increasing number of objectives, it could become unscalable to tune the weights for each objective in an A/B testing framework with a combinatorial number of candidates for the weight vector. Multi-armed bandit experiments (Burtini et al. 2015) are more efficient experiment designs than A/B testing, where the experiment traffic is dynamically allocated to different treatment groups based on their short-term performance metrics. However, they are not feasible for long-term objectives such as consumer retention in our application, which requires the consumer to consistently receive the same treatment for an extended period of time. Another alternative is to learn the optimal weight combination offline using more sophisticated methods such as Bayesian optimization. However, we found those methods suffer from training-serving skew due to its off-policy nature, which introduces additional challenges for off-policy learning in addition to the reward design. In practice, we adopt a greedy approach for adding new objectives, where each objective is added and tuned sequentially. This reduces the tuning complexity from exponential to linear in the number of objectives.

The state-based consumer browsing model has two limitations. First, in our application the model is a global static estimate based on a snapshot of consumer behavior logs. It is not personalized and could become outdated after the model is launched to global traffic. In addition, different consumers have different browsing patterns, and even the same consumer could have different browsing patterns under different contexts. A future research direction is to build a personalized and real-time consumer browsing model, which takes as input the consumer's history, current in-session behavior and real-time contextual features, and generates a real-time prediction of consumer state transition probabilities. The whole MOHR framework still holds in this case, but with $p_{l,l+1}$ in Eq.(4) plugged in as the output from a personalized real-time machine learning model instead. Second, the consumer browsing model assumes a linear browsing pattern (i.e. consumers inspect one item at a time without going back and forth), following the sequential search framework proposed by Weitzman (1979). This assumption can be relaxed by assuming that the consumers first inspect a set of items and then choose one from the set, which calls for a choice modeling component with position bias taken into account.

Lastly, the objectives in the MOHR framework are estimated by separate machine learning models. However, different outcomes may be related to each other in addition to being conflicting, and one may leverage the relatedness among these outcomes for better predictive power. For example, consumers' short-term engagement might be indicative of their long-term happiness. Multi-task deep learning models (Ruder 2017) are well-suited in this case to jointly and efficiently learn multiple related and conflicting objectives. The multiple machine learning models in the MO-module can be replaced with a single large multi-task deep learning model, with the other components of MOHR unchanged.

Endnotes

[1]The Airbnb platform itself can be considered as a third player in the marketplace, leading to a three-sided marketplace. Our proposed framework in this paper applies to both specifications.

[2]The marketplace fairness objective does not (and should not) ensure equal exposure or equal outcome (i.e. every restaurant receives equal amount of exposure or equal number of orders), but instead that the machine learning-based recommender system has learned a fair targeting rule so that every restaurant has equal opportunity to be surfaced to the right consumers. We will elaborate this point more later in Section 4.

[3]Although the percentage numbers seem small, these trade-offs are actually huge in practice (in the scale of million dollars changes in weekly revenue) given the scale of the business.

[4]Therefore, earnings for the restaurant partners, delivery partners and the platform are all positively correlated with consumers' payments, or gross bookings.

[5]This is a common setup (i.e. fixed candidate set) for industrial recommender systems. Also note that the same restaurant can be qualified for multiple carousels (e.g. a restaurant can belong to both "national favorite" and "order again" carousel). The platform has a personalized diversification and deduplication algorithm to reduce the chances of the same restaurant appearing at multiple places in the homepage. But such instances can still happen if the restaurant has an extremely high ranking score.

[6]We cannot disclose the number of consumers and number of restaurants involved in the dataset due to business compliance reasons. But we would like to point out that these are huge numbers which bring scalability challenges, as will be discussed in Section 4.4.

[7]Consumer satisfaction signals are usually collected from online surveys, which have extremely low response rate, and the respondents are usually not a representative sample of the whole population.

[8]The time window is chosen as 14 days in our experiment, which is aligned with the key business metric for the company.

[9]The condition is counterfactual, meaning that the machine learning model will have a prediction for this objective regardless of whether the consumer orders in the current session.

[10]Note that the goal of marketplace fairness is to ensure "equal opportunities" but not necessarily "equal outcomes" for the restaurant partners.

[11]We used the term *"outcome"* in the MO-module as the dependent variable for the machine learning models. Later in the R-module, these outcomes are aggregated into *"objectives"* for optimization purposes.

[12]If the restaurant appears as a single recommendation time, we say it belongs to a single-restaurant carousel.

[13]Theoretically it is possible for gradient boosting regression trees to generate negative predictions even if all training labels are positive, although we didn't observe this from our models.

[14]That is, seven combinations in total: $(consumer, restaurant, source)$, $(consumer, restaurant)$, $(consumer, source)$, $(restaurant, source)$, $(consumer)$, $(restaurant)$, $(source)$.

[15]For across-row ranking where the restaurant ordering within a carousel is determined, the horizontal position feature is set to the true position of the restaurant within the carousel. For more details see the R-module.

[16]We also experimented with 90 days and 180 days as the time window. The results were not statistically different.

[17]As a secondary effect of boosting exposure for weak restaurants, the marketplace fairness objective also enjoys two more benefits: better training data coverage to reduce uncertainty in the estimate of other outcomes (as in a typical MAB procedure); and better consumer experience with novel recommendations (Chen et al. 2021). We focus on its most salient effect, which is exposure for weak restaurants.

[18]When the consumer abandons the current row, she can either go to the next row that's immediately below the current one, or abandon the session completely. For the vertical scrolling behavior, we assume that the consumer's vertical browsing is continuous (they will not skip a row to browse the next row). This is also validated from the data we have. We do not explicitly model the vertical (across-row) browsing probabilities in the MOHR framework as they do not affect the final ranking. The reason is that all ranking algorithms will place higher-scored rows on top of lower-scored rows regardless of how deep the consumer scrolls vertically. So we only model with horizontal (within-row) browsing probabilities as they determine the row-level scores.

[19]The termination state is placing an order. The consumer can place at most one order in each session. This is because when they place an order, a session ends and when they return to the home screen, a new session starts. Therefore, the H-module also applies to other platforms where multiple consumptions can happen sequentially (e.g. people watch multiple YouTube videos sequentially), in which case a separate session is defined each consumption and the ranking scores are recomputed after each consumption. This is in fact the exact practice that major platforms such as YouTube are adopting today (Covington et al. 2016, Wang et al. 2022).

[20]This is empirically guaranteed to be true by the design of the homepage of the app.

[21]We drop the dependency on z in $c(i,j,k), c(i,k), b(i,j,k), b(i,k), r(i,j,k), r(i,k)$ for ease of notation, but we would like to emphasize that these estimates all take contextual features z as input.

[22]The consumer can place at most one order in each session. This is because when they place an order, a session ends and when they return to the home screen, a new session starts. The H-module also applies to other platforms where multiple engagements can happen sequentially (e.g. people can watch multiple YouTube videos sequentially), in which case a separate session is defined for each engagement and the ranking scores are regenerated after each

engagement. This is in fact the exact practice that major platforms such as YouTube are adopting today (Covington et al. 2016, Wang et al. 2022).

[23]When $n=1$, Eq.(5) computes the conversion rate of a single-restaurant row, which equals the conversion rate of the only restaurant inside it.

[24]As described in Section 3.3, the candidates for within-row and across-row ranking are given. The R-module only determines the ordering of them. This is a typical setup (i.e. fixed candidate set) for a recommender system.

[25]These probability estimates also allow us to assess the uncertainty of our estimate, which can be useful for downstream applications that require such uncertainty estimation.

[26]Note that the computation of row-level outcomes depends on the within-row ranking as derived in the H-module. In addition, the restaurant position within the carousel, which is determined by the within-row ranking, is used as an input feature in generating the ML predictions within a carousel. That's why there is an arrow from the R-module to the H-module in Fig. 2.

[27]The formulation is equivalent to having $B(\boldsymbol{x})$, $R(\boldsymbol{x})$ or $F(\boldsymbol{x})$ as the objective while constraining on others. This is because the primal problem in Eq.(13) is feasible and bounded, so strong duality holds.

[28]Solution $\boldsymbol{x}_1$ is said to dominate solution $\boldsymbol{x}_2$ if $(C(\boldsymbol{x}_1), B(\boldsymbol{x}_1), R(\boldsymbol{x}_1), F(\boldsymbol{x}_1)) \geq (C(\boldsymbol{x}_2), B(\boldsymbol{x}_2), R(\boldsymbol{x}_2), F(\boldsymbol{x}_2))$ element-wisely, and at least one of the inequalities is strict. A solution $\boldsymbol{x}$ is called *Pareto optimal* if there is no solution $\boldsymbol{x}' \neq \boldsymbol{x}$ such that $\boldsymbol{x}'$ dominates $\boldsymbol{x}$. *Pareto frontier* is the set of all Pareto optimal solutions.

[29]The benefit of having analytical solutions is that we don't need to solve the large-scale linear programming problem online, and only need to plug in the values for the analytical form instead.

[30]Note that the basket value b_{iq} and retention r_{iq} is multiplied by the conversion c_{iq} while the marketplace fairness f_{iq} is not. This is because the basket value and retention outcome is a *counterfactual* estimation conditioning on the consumer placing an order in the current session, while the marketplace fairness objective is not (Table 2).

[31]For example, their interaction history in the last session will be incorporated in features such as number of impressions / clicks / orders in the past X days. Also contextual features such as time of day would change across different sessions.

[32]Note that a 28-day experiment is considered to be a long-term experiment at the company, and the long-term consumer retention metric the company monitors is also defined using a 28-day window.

[33]We are not allowed to disclose the number of consumers, sessions and orders from the experiments due to business compliance reasons. But we would like to point out that given the large scale of the business, the data gives us more than enough statistical power to conduct hypothesis testing on the performance metrics defined in the next section.

[34]We only look at signed-in consumers as sign-in is required to place an order on the app.

[35]The random ranking data provided by the company is restaurant-level random ranking and we unfortunately don't have carousel-level random ranking data from the company. Nevertheless, the selected parameters from the replay analysis using the restaurant-level random ranking data perform reasonably well in the online experiments.

[36]We unfortunately could not generate the Pareto frontier for the consumer retention objective using the replay method. The reason is that it requires at least 28 days of random ranking data to observe consumer retention, but the random ranking data we have from the company is only one week.

[37]Note that the Pareto frontier for the basket value objective is noisy, while the Pareto frontier for the marketplace fairness objective is much smoother. This is expected as the marketplace fairness objective is measured by the number of *impressions* a restaurant receives, which the recommender system has (almost) full control by determining which restaurants to put on top. On the other hand, the basket value objective depends on the consumer *placing an order* on the restaurant, which is a stochastic event that the recommender system only has partial influence on. In other words, the basket value objective incorporates one extra layer of randomness, leading to a noisier Pareto frontier.

[38]The statistical significance is measured under 0.95 confidence level.

[39]Note that the control group ranks all carousels on top of all single restaurants. So the MOHR framework actually presents *fewer* contents in the top positions, yet it's still able to reduce the average vertical order position by 5.7% compared with control. This further confirms the increased quality of the homepage.

[40]The basket value objective, which is measured in dollar amounts, is about 2 orders of magnitude larger than the other three objectives. Therefore $\lambda_b = 0.1$ means that the term for the basket value objective, $\lambda_b c_{iq} b_{iq}$, is roughly 10 times the value of the other terms, making the ranking function in Eq.(10) dominated by the basket value objective.

7. Declarations

7.1. Funding and Competing Interests.

All authors were employed and supported by Uber Technologies, Inc. when the work was done, and have held stocks of the company that may constitute a material financial position. Uber Technologies, Inc. recognizes a potential need to disclose certain confidential information, and to protect such information from unauthorized use and disclosure. Uber Technologies, Inc. had the right to remove its intellectual property or trade secrets from the paper subject to the following stipulations:

1. Removing Uber confidential information from the paper, including summary statistics such as conversion rate, number of impressions / clicks / orders per carousel etc.

2. Compliance with Uber's obligations as it relates to applicable laws, including the Data Protection Law, security laws, confidentiality requirements, and contractual commitments.

References

Abdollahpouri H, Adomavicius G, Burke R, Guy I, Jannach D, Kamishima T, Krasnodebski J, Pizzato L (2020) Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* 30(1):127–158.

Adomavicius G, Bockstedt JC, Curley SP, Zhang J (2013) Do recommender systems manipulate consumer preferences? a study of anchoring effects. *Information Systems Research* 24(4):956–975.

Adomavicius G, Bockstedt JC, Curley SP, Zhang J (2018) Effects of online recommendations on consumers' willingness to pay. *Information Systems Research* 29(1):84–102.

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

Afsar MM, Crump T, Far B (2022) Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys* 55(7):1–38.

Agarwal D, Chatterjee S, Yang Y, Zhang L (2015) Constrained optimization for homepage relevance. *Proceedings of the 24th International Conference on World Wide Web*, 375–384.

Agarwal D, Chen BC, Elango P, Wang X (2011) Click shaping to optimize multiple objectives. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 132–140.

Agarwal D, Chen BC, Elango P, Wang X (2012) Personalized click shaping through lagrangian duality for online recommendation. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 485–494.

Aggarwal CC, et al. (2016) *Recommender systems*, volume 1 (Springer).

Agrawal R, Gollapudi S, Halverson A, Ieong S (2009) Diversifying search results. *Proceedings of the second ACM international conference on web search and data mining*, 5–14.

Agrawal S, Goyal N (2017) Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)* 64(5):1–24.

Alvino C, Basilico J (2015) Learning a personalized homepage. *The Netflix Tech Blog* 9.

Anderson A, Maystre L, Anderson I, Mehrotra R, Lalmas M (2020) Algorithmic effects on the diversity of consumption on spotify. *Proceedings of The Web Conference 2020*, 2155–2165.

Aribarg A, Schwartz EM (2020) Native advertising in online news: Trade-offs among clicks, brand recognition, and website trustworthiness. *Journal of Marketing Research* 57(1):20–34.

Aridor G, Gonçalves D (2021) Recommenders' originals: The welfare effects of the dual role of platforms as producers and recommender systems. *Available at SSRN 3928005* .

Athey S, Chetty R, Imbens GW, Kang H (2019) The surrogate index: Combining short-term proxies to estimate long-term treatment effects more rapidly and precisely. Technical report, National Bureau of Economic Research.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2):235–256.

Azaria A, Hassidim A, Kraus S, Eshkol A, Weintraub O, Netanely I (2013) Movie recommender system for profit maximization. *Proceedings of the 7th ACM conference on Recommender systems*, 121–128.

Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. *Proceedings of the fourth ACM international conference on Web search and data mining*, 635–644.

Baeza-Yates R, Ribeiro-Neto B, et al. (1999) *Modern information retrieval*, volume 463 (ACM press New York).

Bahrami S, Nourinejad M, Yin Y, Wang H (2021) The three-sided market of on-demand delivery. *Available at SSRN 3944559* .

Balabanović M, Shoham Y (1997) Fab: content-based, collaborative recommendation. *Communications of the ACM* 40(3):66–72.

Bennett J, Lanning S, et al. (2007) The netflix prize. *Proceedings of KDD cup and workshop*, volume 2007, 35 (New York, NY, USA.).

Besbes O, Gur Y, Zeevi A (2016) Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing & Service Operations Management* 18(1):15–33.

Beutel A, Chen J, Doshi T, Qian H, Wei L, Wu Y, Heldt L, Zhao Z, Hong L, Chi EH, et al. (2019) Fairness in recommendation ranking through pairwise comparisons. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2212–2220.

Billsus D, Pazzani MJ, et al. (1998) Learning collaborative information filters. *Icml*, volume 98, 46–54.

Bodapati AV (2008) Recommendation systems with purchase data. *Journal of marketing research* 45(1):77–93.

Bourreau M, Gaudin G (2022) Streaming platform and strategic recommendation bias. *Journal of Economics & Management Strategy* 31(1):25–47.

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 43–52.

Breiman L, Friedman JH, Olshen RA, Stone CJ (2017) *Classification and regression trees* (Routledge).

Brusilovsky P (2007) Adaptive navigation support. *The adaptive web*, 263–290 (Springer).

Burke R (2002) Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12(4):331–370.

Burtini G, Loeppky J, Lawrence R (2015) A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757* .

Carare O (2012) The impact of bestseller rank on demand: Evidence from the app market. *International Economic Review* 53(3):717–742.

Carbonell J, Goldstein J (1998) The use of mmr, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336.

Chaney AJ, Stewart BM, Engelhardt BE (2018) How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. *Proceedings of the 12th ACM Conference on Recommender Systems*, 224–232.

Chaudhuri N, Gupta G, Vamsi V, Bose I (2021) On the platform but will they buy? predicting customers' purchase behavior using deep learning. *Decision Support Systems* 149:113622.

Chen LS, Hsu FH, Chen MC, Hsu YC (2008) Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences* 178(4):1032–1048.

Chen M, Beutel A, Covington P, Jain S, Belletti F, Chi EH (2019) Top-k off-policy correction for a reinforce recommender system. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 456–464.

Chen M, Wang Y, Xu C, Le Y, Sharma M, Richardson L, Wu SL, Chi E (2021) Values of user exploration in recommender systems. *Fifteenth ACM Conference on Recommender Systems*, 85–95.

Chen Y, Yao S (2017) Sequential search with refinement: Model and application with click-stream data. *Management Science* 63(12):4345–4365.

Chung J, Rao VR (2012) A general consumer preference model for experience products: application to internet recommendation services. *Journal of marketing research* 49(3):289–305.

Click C, Malohlava M, Candel A, Roark H, Parmar V (2017) Gradient boosting machine with h2o. *H2O.ai* .

Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. *Recsys' 16*, 191–198.

Das A, Mathieu C, Ricketts D (2009) Maximizing profit using recommender systems. *arXiv:0908.3633* .

Datta H, Knox G, Bronnenberg BJ (2018) Changing their tune: How consumers' adoption of online streaming affects music consumption and discovery. *Marketing Science* 37(1):5–21.

Delgado J, Ishii N (1999) Memory-based weighted majority prediction. *SIGIR Workshop Recomm. Syst. Citeseer*, 85 (Citeseer).

Dhillon PS, Aral S (2021) Modeling dynamic user interests: A neural matrix factorization approach. *Marketing Science* 40(6):1059–1080.

Donnelly R, Kanodia A, Morozov I (2022) Welfare effects of personalized rankings. *Available at SSRN 3649342* .

Elahi E, Chandrashekar A (2020) Learning representations of hierarchical slates in collaborative filtering. *Proceedings of the 14th ACM Conference on Recommender Systems*, 703–707.

Evans DS, Schmalensee R (2016) *Matchmakers: The new economics of multisided platforms* (Harvard Business Review Press).

Fader PS, Hardie BG (1996) Modeling consumer choice among skus. *Journal of marketing Research* 33(4):442–452.

Fader PS, Hardie BG, Lee KL (2005) Rfm and clv: Using iso-value curves for customer base analysis. *Journal of marketing research* 42(4):415–430.

Farias VF, Li AA (2019) Learning preferences with side information. *Management Science* 65(7):3131–3149.

Farrell MH, Liang T, Misra S (2020) Deep learning for individual heterogeneity: An automatic inference framework. *arXiv preprint arXiv:2010.14694* .

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

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

Friedman JH (2002) Stochastic gradient boosting. *Computational statistics & data analysis* 38(4):367–378.

Ghose A, Ipeirotis PG, Li B (2012) Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. *Marketing Science* 31(3):493–520.

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

Gomez-Uribe CA, Hunt N (2015) The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6(4):1–19.

Gong NZ, Talwalkar A, Mackey L, Huang L, Shin ECR, Stefanov E, Shi E, Song D (2014) Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5(2):1–20.

Govindarajan V, Venkatraman NV (2022) The next great digital advantage. *Harvard Business Review* .

Guadagni PM, Little JD (1983) A logit model of brand choice calibrated on scanner data. *Marketing science* 2(3):203–238.

Guan Z, Cutrell E (2007) An eye tracking study of the effect of target rank on web search. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 417–420.

GVR (2022) Online food delivery market size; share report, 2028. URL `https://www.grandviewresearch.com/industry-analysis/online-food-delivery-market-report`.

Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29.

Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) *The elements of statistical learning: data mining, inference, and prediction*, volume 2 (Springer).

He X, Pan J, Jin O, Xu T, Liu B, Xu T, Shi Y, Atallah A, Herbrich R, Bowers S, et al. (2014) Practical lessons from predicting clicks on ads at facebook. *Proceedings of the eighth international workshop on data mining for online advertising*, 1–9.

Hitsch GJ, Misra S (2018) Heterogeneous treatment effects and optimal targeting policy evaluation. *Available at SSRN 3111957* .

Hosanagar K, Fleder D, Lee D, Buja A (2014) Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation. *Management Science* 60(4):805–823.

Hosanagar K, Krishnan R, Ma L (2008) Recomended for you: The impact of profit incentives on the relevance of online recommendations. *ICIS 2008 Proceedings* 31.

Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. *2008 Eighth IEEE international conference on data mining*, 263–272 (Ieee).

Jacobs BJ, Donkers B, Fok D (2016) Model-based purchase predictions for large assortments. *Marketing Science* 35(3):389–404.

Jiang Z, Chan T, Che H, Wang Y (2021) Consumer search and purchase: An empirical investigation of retargeting based on consumer online behaviors. *Marketing Science* 40(2):219–240.

Johnson EJ, Shu SB, Dellaert BG, Fox C, Goldstein DG, Häubl G, Larrick RP, Payne JW, Peters E, Schkade D, et al. (2012) Beyond nudges: Tools of a choice architecture. *Marketing letters* 23:487–504.

Kang WC, McAuley J (2018) Self-attentive sequential recommendation. *2018 IEEE international conference on data mining (ICDM)*, 197–206 (IEEE).

Katehakis MN, Veinott Jr AF (1987) The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12(2):262–268.

King J, Imbrasaitė V (2015) Generating music playlists with hierarchical clustering and q-learning. *European Conference on Information Retrieval*, 315–326 (Springer).

Koch M, von Luck K, Schwarzer J, Draheim S (2018) The novelty effect in large display deployments–experiences and lessons-learned for evaluating prototypes. *Proceedings of 16th European conference on computer-supported cooperative work-exploratory papers* (European Society for Socially Embedded Technologies (EUSSET)).

Kokkodis M, Ipeirotis PG (2021) Demand-aware career path recommendations: A reinforcement learning approach. *Management Science* 67(7):4362–4383.

Koren Y (2009) Collaborative filtering with temporal dynamics. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 447–456.

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Krasnodebski J, Dines J (2016) Considering supplier relations and monetization in designing recommendation systems. *Proceedings of the 10th ACM Conference on Recommender Systems*, 381–382.

Kumar A, Hosanagar K (2019) Measuring the value of recommendation links on product demand. *Information Systems Research* 30(3):819–838.

Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30.

Langford J, Zhang T (2007) The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems* 20(1):96–1.

Lee EL, Lou JK, Chen WM, Chen YC, Lin SD, Chiang YS, Chen KT (2014) Fairness-aware loan recommendation for microfinance services. *Proceedings of the 2014 international conference on social computing*, 1–4.

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

Li L, Chu W, Langford J, Wang X (2011) Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM international conference on Web search and data mining*, 297–306.

Li Z, Fang X, Bai X, Sheng ORL (2017) Utility-based link recommendation for online social networks. *Management Science* 63(6):1938–1952.

Liebman E, Saar-Tsechansky M, Stone P (2019) The right music at the right time: Adaptive personalized playlists based on sequence modeling. *MIS Quarterly* 43(3).

Liechty JC, Fong DK, DeSarbo WS (2005) Dynamic models incorporating individual heterogeneity: Utility evolution in conjoint analysis. *Marketing Science* 24(2):285–293.

Liu TY, et al. (2009) Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3(3):225–331.

Liu X (2022) Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Marketing Science* .

Miller BN, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: experiences with an occasionally connected recommender system. *Proceedings of the 8th international conference on Intelligent user interfaces*, 263–266.

Milojkovic N, Antognini D, Bergamin G, Faltings B, Musat C (2019) Multi-gradient descent for multi-objective recommender systems. *arXiv preprint arXiv:2001.00846* .

Mooney RJ, Roy L (2000) Content-based book recommending using learning for text categorization. *Proceedings of the fifth ACM conference on Digital libraries*, 195–204.

Muangmee C, Kot S, Meekaewkunchorn N, Kassakorn N, Khalid B (2021) Factors determining the behavioral intention of using food delivery apps during covid-19 pandemics. *Journal of theoretical and applied electronic commerce research* 16(5):1297–1310.

Murphy KP (2006) Binomial and multinomial distributions. *University of British Columbia, Tech. Rep* .

Narayanan S, Kalyanam K (2015) Position effects in search advertising and their moderators: A regression discontinuity approach. *Marketing Science* 34(3):388–407.

Nguyen P, Dines J, Krasnodebski J (2017) A multi-objective learning to re-rank approach to optimize online marketplaces for multiple stakeholders. *arXiv preprint arXiv:1708.00651* .

Nielsen J (2006) " f-shaped pattern for reading web content," jakob nielsen's alertbox. *http://www. useit. com/alertbox/reading_pattern. html* .

Noulas A, Scellato S, Lathia N, Mascolo C (2012) A random walk around the city: New venue recommendation in location-based social networks. *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT)*, 144–153 (IEEE Computer Society).

Oestreicher-Singer G, Sundararajan A (2012) The visible hand? demand effects of recommendation networks in electronic markets. *Management science* 58(11):1963–1981.

Pathak B, Garfinkel R, Gopal RD, Venkatesan R, Yin F (2010) Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems* 27(2):159–188.

Prawesh S, Padmanabhan B (2014) The "most popular news" recommender: Count amplification and manipulation resistance. *Information Systems Research* 25(3):569–589.

Ricci F, Rokach L, Shapira B (2015) Recommender systems: introduction and challenges. *Recommender systems handbook*, 1–34 (Springer).

Rodriguez M, Posse C, Zhang E (2012) Multiple objective optimization in recommender systems. *Proceedings of the sixth ACM conference on Recommender systems*, 11–18.

Ruder S (2017) An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* .

Sahoo N, Singh PV, Mukhopadhyay T (2012) A hidden markov model for collaborative filtering. *MIS quarterly* 1329–1356.

Saito Y, Joachims T (2021) Counterfactual learning and evaluation for recommender systems: Foundations, implementations, and recent advances. *Proceedings of the 15th ACM Conference on Recommender Systems*, 828–830.

Sawaragi Y, Nakayama H, Tanino T (1985) *Theory of multiobjective optimization* (Elsevier).

Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 253–260.

Schnabel T, Swaminathan A, Singh A, Chandak N, Joachims T (2016) Recommendations as treatments: Debiasing learning and evaluation. *international conference on machine learning*, 1670–1679 (PMLR).

Sherman C (2005) A new f-word for google search results. *Search Engine Watch* (March 7).

Shi SW, Trusov M (2021) The path to click: Are you on it? *Marketing Science* 40(2):344–365.

Simester D, Timoshenko A, Zoumpoulis SI (2020a) Efficiently evaluating targeting policies: Improving on champion vs. challenger experiments. *Management Science* 66(8):3412–3424.

Simester D, Timoshenko A, Zoumpoulis SI (2020b) Targeting prospective customers: Robustness of machine-learning methods to typical data challenges. *Management Science* 66(6):2495–2522.

Smith B, Linden G (2017) Two decades of recommender systems at amazon.com. *Ieee internet computing* 21(3):12–18.

Solsman JE (2018) Youtube's ai is the puppet master over most of what you watch. *CNET, January* 10.

Song Y, Sahoo N, Ofek E (2019) When and how to diversify—a multicategory utility model for personalized content recommendation. *Management Science* 65(8):3737–3757.

Strehl A, Langford J, Li L, Kakade SM (2010) Learning from logged implicit exploration data. *Advances in neural information processing systems* 23.

Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P (2019) Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Sutton RS, Barto AG (2018) *Reinforcement learning: An introduction* (MIT press).

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3-4):285–294.

Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Anthony S, Liu H, Wyckoff P, Murthy R (2009) Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment* 2(2):1626–1629.

Tintarev N, Sullivan E, Guldin D, Qiu S, Odjik D (2018) Same, same, but different: algorithmic diversification of viewpoints in news. *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, 7–13.

Tso-Sutter KH, Marinho LB, Schmidt-Thieme L (2008) Tag-aware recommender systems by fusion of collaborative filtering algorithms. *Proceedings of the 2008 ACM symposium on Applied computing*, 1995–1999.

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

Ursu RM, Zhang Q, Honka E (2022) Search gaps and consumer fatigue. *Marketing Science* .

Wagner U, Taudes A (1986) A multivariate polya model of brand choice and purchase incidence. *Marketing Science* 5(3):219–244.

Wang W, Xu J, Wang M (2018) Effects of recommendation neutrality and sponsorship disclosure on trust vs. distrust in online recommendation agents: Moderating role of explanations for organic recommendations. *Management science* 64(11):5198–5219.

Wang Y, Sharma M, Xu C, Badam S, Sun Q, Richardson L, Chung L, Chi EH, Chen M (2022) Surrogate for long-term user experience in recommender systems. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4100–4109.

Wang Y, Wang X, Beutel A, Prost F, Chen J, Chi EH (2021) Understanding and improving fairness-accuracy trade-offs in multi-task learning. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1748–1757.

Weitzman ML (1979) Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society* 641–654.

Wu Q, Wang H, Hong L, Shi Y (2017) Returning is believing: Optimizing long-term user engagement in recommender systems. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1927–1936.

Xiao B, Benbasat I (2007) E-commerce product recommendation agents: Use, characteristics, and impact. *MIS quarterly* 137–209.

Xie R, Zhang S, Wang R, Xia F, Lin L (2021) Hierarchical reinforcement learning for integrated recommendation. *Proceedings of AAAI*.

Xie X (2010) Potential friend recommendation in online social network. *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, 831–835 (IEEE).

Yang J, Eckles D, Dhillon P, Aral S (2022) Targeting for long-term outcomes. *Management Science, forthcoming* .

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

Yoganarasimhan H, Barzegary E, Pani A (2022) Design and evaluation of optimal free trials. *Management Science* .

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

Zhang TC, Agarwal R, Lucas Jr HC (2011) The value of it-enabled retailer learning: personalized product recommendations and customer store loyalty in electronic markets. *Mis Quarterly* 859–881.

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

Zhao Z, Hong L, Wei L, Chen J, Nath A, Andrews S, Kumthekar A, Sathiamoorthy M, Yi X, Chi E (2019) Recommending what video to watch next: a multitask ranking system. *Proceedings of the 13th ACM Conference on Recommender Systems*, 43–51.

Zheng G, Zhang F, Zheng Z, Xiang Y, Yuan NJ, Xie X, Li Z (2018) Drn: A deep reinforcement learning framework for news recommendation. *Proceedings of the 2018 world wide web conference*, 167–176.

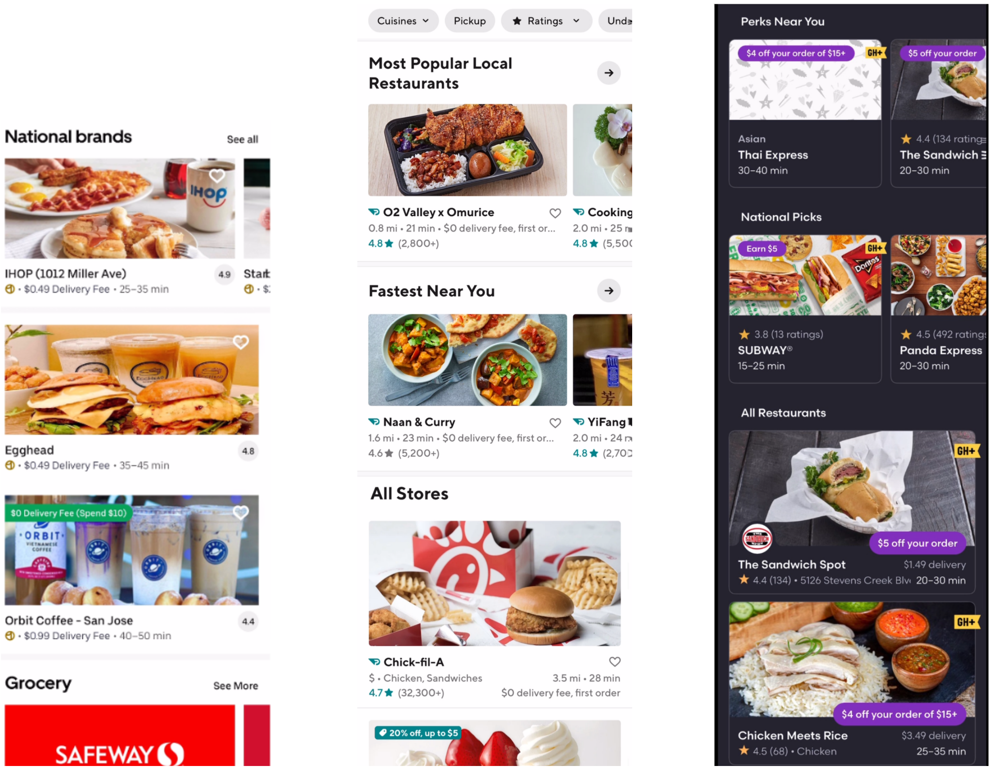
Zheng L, Li L, Hong W, Li T (2013) Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications* 40(6):2127–2136.

Zheng Y, Ghane N, Sabouri M (2019) Personalized educational learning with multi-stakeholder optimizations. *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 283–289.

Zhou B, Zou T (2022) Competing for recommendations: The strategic impact of personalized product recommendations in online marketplaces. *Marketing Science* .

Appendix. Recommending for a Three-Sided Food Delivery Marketplace: A Multi-Objective Hierarchical Approach

A. Illustration of hierarchical recommendation items



(a) Uber Eats. (b) DoorDash. (c) Grubhub.

Figure 7 Screenshots of the homepage of three major food delivery apps.

B. Additional Technical Details for MOHR

B.1. MO-Module

B.1.1. Collaborative filtering features based on matrix factorization. To leverage the idea of collaborative filtering that similar consumers enjoy similar contents, we build matrix factorization models to learn a latent vector representation for every consumer, restaurant and source. The idea for matrix factorization for collaborative filtering (Koren et al. 2009) is to factor the huge consumer-item interaction matrix as the product of two lower dimensional matrices, the first one has a row for each consumer, while the second has a column for each item. The row or column associated to a specific user or item is referred to as latent factors. Here an item refers to either a restaurant or a product. Figure 8 illustrates the idea of matrix factorization for collaborative filtering.

Suppose there are I consumers and J restaurants in total, we learn their latent representations by

$$\{u_i\}_{i=1}^I, \{v_j\}_{j=1}^J = \underset{\{u_i\}_{i=1}^I, \{v_j\}_{j=1}^J}{\arg\min} \sum_{i,j \in S} (u_i^T v_j - r_{ij})^2 + \lambda_u \sum_{i=1}^I \|u_i\|_2 + \lambda_v \sum_{j=1}^J \|v_j\|_2, \tag{15}$$

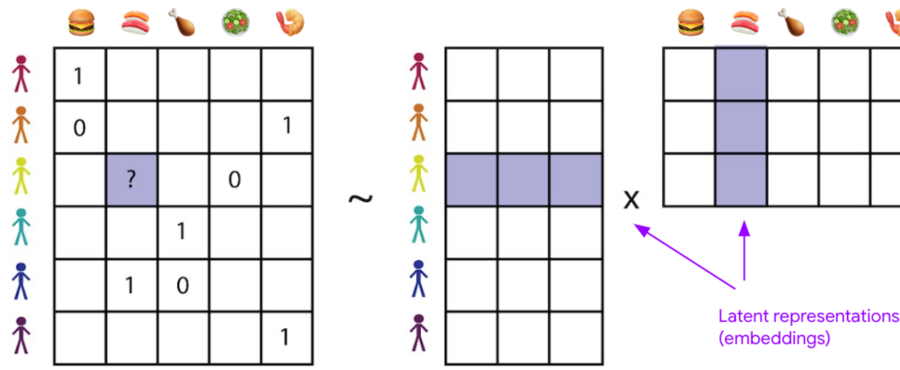


Figure 8 Matrix factorization for collaborative filtering.

where S is the set of observations (where consumer i was recommended restaurant j), r_{ij} is the number of orders between consumer i and restaurant j (0 if the consumer never ordered from the restaurant), λ_u and λ_v are positive penalization coefficients preventing the optimization from learning wild values. This optimization problem also has a Bayesian interpretation with Gaussian prior on the representations, in which case λ_u and λ_v are determined by the variance parameter of the priors. See Section 4.2 in Dhillon and Aral (2021) as an example.

Eq.(15) is a biconvex problem and can be solved efficiently using alternating least squares (ALS) (Koren 2009). The output of the optimization problem, u_i's and v_j's, are used as latent representations for the consumers and restaurants. We build another matrix factorization model on $(consumer, source)$ level similar to Equation (15) but changed v_j to source representation $\tilde{w_k}$ and r_{ij} to r_{ik}, the order counts between consumer i and source k:

$$\{\tilde{u_i}\}_{i=1}^{I}, \{\tilde{w_k}\}_{k=1}^{K} = \underset{\{\tilde{u_i}\}_{i=1}^{I}, \{\tilde{w_k}\}_{k=1}^{K}}{\arg\min} \sum_{i,k\in\tilde{S}} (\tilde{u_i}^T \tilde{w_k} - r_{ik})^2 + \lambda_u \sum_{i=1}^{I} \|\tilde{u_i}\|_2 + \lambda_w \sum_{k=1}^{K} \|\tilde{w_k}\|_2, \tag{16}$$

and obtain another set of representations, which are $\tilde{u}_i$ for consumers and $\tilde{w}_k$ for sources. For the individual ML models defined in Eq.(1), the learned embeddings u_i and $\tilde{u}_i$ are included as part of consumer-level features a_i, v_j is part of restaurant-level features a_j, and $\tilde{w}_k$ is part of source-level features a_k.

B.1.2. Details for the ML models. Table 7 summarizes the features used for predicting consumer conversion, consumer retention and basket value. Note that for the count-based features such as the number of impressions/views/orders, we include both the raw count and the *normalized count* as features, where the normalized count are divided by the average impression/view/order count at the position of the event, in order to correct for the position bias as illustrated in Fig.4. The name/id of the source (e.g. i.e. the name of the carousel or "single" if the training instance is a single restaurant recommendation) is explicitly used as a feature.

For the gradient boosted trees as the predictive ML model, we use learning rate of 0.1, and maximum depth of the tree as 8, which is the same model architecture and capacity as the latest production system at the company. For the conversion rate model, the training data is unbalanced with a positive sample ratio as low as 1.8%, which could potentially cause challenges to the binary classification models. We therefore experimented with down-sampling the negative examples. However, we didn't find a performance boost on the test data, which could be explained as fact that the training data is big enough (around 600 million). In the final version of MOHR, the individual ML models are all trained without data reweighting or resampling.

The data used to train the ML models are *observational* data from randomly sampled global user logs, consisting of about 600 million impressions and 11 million orders. As an industry-wide challenge, observational data potentially brings bias in the training data due to their off-policy nature (Saito and Joachims 2021). Alternatively, one could rely on experimental data, such as the random ranking data as described in Section 5.2, which is free of off-policy bias. However, experimental data are usually costly to obtain (random ranking significantly hurt consumer experience). Therefore the platform is usually only able to set aside a very small percentage of sessions as experimental data. In our case, the random ranking data we obtained from the company is only on 5% of overall sessions and only on vertical ranking (i.e. restaurants within a carousel are not randomized). As a result, they provide poor coverage of all the consumers and all the restaurants to be served as training data. Because of these challenges, we do now rely on random ranking data, but rather all available observational data as the training data for the ML models. We adopt off-policy correction techniques in training (described as part of the contextual features in Section 4.2) to alleviate the off-policy bias.

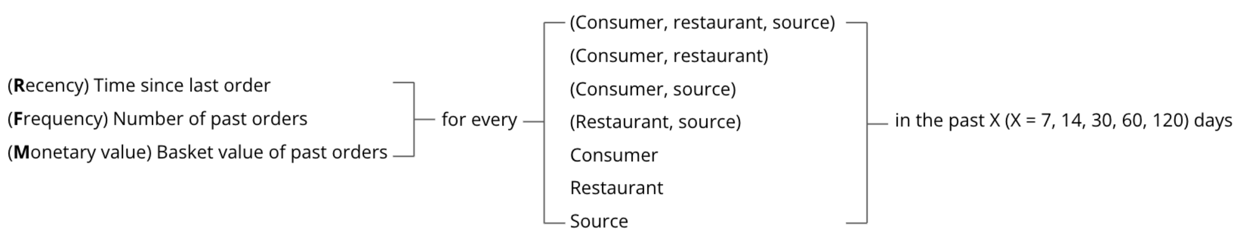


Figure 9 RFM features in the MO-module.

B.1.3. Bayesian modeling for the marketplace fairness objective. We now describe the Bayesian modeling procedure to estimate $\sigma(j)$, the posterior variance for $c(i,j,k)$ as the value for the fairness objective. The order event $O(i,j,k,z)$ is a Bernoulli random variable with parameter $c(i,j,k)$. Therefore, we choose Beta distribution as the prior for $c(i,j,k)$. Proposition 2 below states the posterior for $c(i,j,k)$.

Proposition 2 *Suppose the prior distribution for $c(i,j,k)$ is $\mathcal{B}(\alpha_j,\beta_j)$, and that there are N_j impressions on restaurant j, out of which N_j^1 lead to orders. Then the posterior distribution for $c(i,j,k)$ is $\mathcal{B}(\alpha_j+N_j^1, \beta_j+N_j-N_j^1)$, and its posterior variance is $\sigma(j)^2 = \frac{(\alpha_j+N_j^1)(\beta_j+N_j-N_j^1)}{(\alpha_j+\beta_j+N_j)^2(\alpha_j+\beta_j+N_j+1)}$.*

Proof of Proposition 2. For ease of notation we drop the dependency on i, k for now and denote $c(i,j,k)$ as c_j for restaurant j. Suppose there are N_j impressions on restaurant j, $O_{j1},...,O_{jN_j}$ are random variables represents the corresponding conversion events where $O_{jm}=1$ means the m-th impression on restaurant j leads to an order, and 0 otherwise. In other words, $O_{j1},...,O_{jN_j} \overset{i.i.d.}{\sim}$ Bernoulli(c_j). The conjugate prior for Bernoulli distribution is the Beta distribution $Beta(\alpha_j,\beta_j)$.

Feature	Definition
a_i	(Normalized) Impression/view/order count/ratio from consumer i in the past T days Average basket values from consumer i in the past T days Consumer embedding u_i and $\tilde{u_i}$ from matrix factorization
a_j	(Normalized) Impression/view/order count/ratio on restaurant j in the past T days Average basket values from restaurant j in the past T days Percentage of consumers churned after ordering from restaurant j in the past T days Restaurant embedding v_j from matrix factorization Delivery radius of restaurant j
a_k	Source name (name of the carousel or "single" if appearing as single restaurant recommendation) (Normalized) Impression/view/order count/ratio from source k in the past T days Average basket values from source k in the past T days Source embedding $\tilde{w_k}$ from matrix factorization
a_{ij}	(Normalized) Impression/view/order count/ratio between consumer i and restaurant j in past T days Haversine distance between restaurant j and consumer's delivery location Estimated delivery time range Delivery fee, busy area fee, service fee $u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding from matrix factorization $cos(u_i, v_j)$, i.e. cosine similarity between consumer embedding and restaurant embedding
a_{ik}	(Normalized) Impression/view/order count/ratio from consumer i in source k in the past T days Average basket values from consumer i in source k in the past T days $\tilde{u_i}^T \tilde{w_k}$, i.e. dot product of consumer embedding and source embedding from matrix factorization $cos(\tilde{u_i}, \tilde{w_k})$, i.e. cosine similarity between consumer embedding and source embedding
a_{jk}	(Normalized) Impression/view/order count/ratio from restaurant j in source k in the past T days Average basket values from restaurant j in source k in the past T days
a_{ijk}	(Normalized) Impression/view/order count/ratio from consumer i, restaurant j, source k in the past T days Average basket values from consumer i, restaurant j, source k in past T days
z	Source k (name of the carousel or "single restaurant") Vertical & horizontal position of the recommendation item City, geolocation, language, device Temporal features including day of week, local hour of day, meal period

Table 7 List of features for the ML models in the MO-module. T=7,14,30,60,120.

Following the known result on the conjugate Beta posterior distribution with Beta prior and Binomial likelihood (Murphy 2006), we get the posterior distribution for c_j as $Beta(\alpha_j + N_j^1, \beta_j + N_j - N_j^1)$.

Plugging in the formula for the variance of Beta distribution, we get the posterior variance for c_j as

$$\sigma(j)^2 = \frac{(\alpha_j + N_j^1)(\beta_j + N_j - N_j^1)}{(\alpha_j + \beta_j + N_j)^2(\alpha_j + \beta_j + N_j + 1)} \tag{17}$$

which concludes the proof. □

B.1.4. Choice of prior parameters for the marketplace fairness objective. To reduce the number of parameters, we let $\alpha_j = \alpha, \beta_j = \beta, \forall j$, that is, all restaurants follow the same prior distribution for its conversion rate. This is a reasonable assumption to ensure fairness across all restaurants (i.e. no prior bias for any of the restaurants). There are three considerations for picking the values for α and β for the prior distribution. First, the prior mean should not be too far from the actual point estimate for the conversion rate, which is around 2% in our training data. Second, it is preferable to have the posterior relatively stable and robust to bot attacks such as a huge amount of fake view and orders from a new restaurant. Third, the posterior variance in Eq.(17) should be able to differentiate new restaurants with few impressions and orders from the well-established restaurants. The first condition implies the mean of a Beta distribution $\mathcal{B}(\alpha,\beta)$, $\frac{\alpha}{\alpha+\beta}$ should be close to 2%. The second condition implies that α and β should be large enough to guard the posterior against noisy data, while the third condition implies that α and β should be small enough so that the numerator and denominator in Eq.(17) is not dominated by them. Given these considerations, we set $\alpha=2$ and $\beta=98$ and find them to work well empirically.

B.1.5. Comparison against alternative multi-armed bandit algorithms for the marketplace fairness outcome. In this section, we provide theoretical analysis to show that the benefit of the marketplace fairness outcome is beyond what a standard multi-armed bandit (MAB) procedure can provide. The marketplace fairness outcome is defined as the uncertainty estimate of the conversion rate motivated by a UCB formulation of MAB. We show that other MAB algorithms such as Thompson sampling (Thompson 1933) and epsilon-greedy (Sutton and Barto 2018) do not have the same boosting effect on new and low-volume restaurants.

For simplicity and without loss of generality, assume that there are two restaurants in total: Restaurant 1 (old) is a popular and well-established restaurant with conversion rate $C_o \sim N(\mu_o,\sigma_o^2)$; Restaurant 2 (new) is a new or low-volume restaurant with conversion rate $C_n \sim N(\mu_n,\sigma_n^2)$. Because restaurant 1 is popular and well established and has more training data than restaurant 2, we assume that $\mu_o > \mu_n$ and $\sigma_o < \sigma_n$. Therefore restaurant 2 has a marketplace fairness outcome as it has a higher uncertainty.

The goal of the recommender system is to choose one from the two restaurants to recommend. Next we discuss how often will the new restaurant (restaurant 2) be recommended under (1) UCB (our formulation), (2) Thompson sampling and (3) epsilon-greedy strategy.

(1) UCB: With UCB, the ranking score r_o, r_n for the two restaurants are:

$$\begin{aligned} r_o &= \mu_o + \kappa\sigma_o, \\ r_n &= \mu_n + \kappa\sigma_n. \end{aligned} \tag{18}$$

Therefore, one has $r_o < r_n$ as long as $\kappa > \frac{\mu_o-\mu_n}{\sigma_n-\sigma_o}$. In other words, restaurant 2 will *always* be recommended given large enough κ. Therefore UCB offers a strategy to *deterministically* boost new restaurants.

(2) Thompson sampling: With Thompson sampling, the ranking score r_o, r_n for the two restaurants are sampled from their posterior distribution $C_o \sim N(\mu_o,\sigma_o^2)$ and $C_n \sim N(\mu_n,\sigma_n^2)$ respectively. Therefore, the probability that restaurant 2 is recommended is the probability that random variable C_n is larger than C_o:

$$\mathbb{P}(C_n > C_o) = \mathbb{P}(C_n - C_o > 0) = \int_{(\mu_o-\mu_n)/\sqrt{\sigma_o^2+\sigma_n^2}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx < 0.5, \tag{19}$$

where the last inequality is because $(\mu_o - \mu_n)/\sqrt{\sigma_o^2+\sigma_n^2} > 0$. Therefore under Thompson sampling, the new restaurant has less than half of the chance of being recommended.

(3) Epsilon-greedy. Epsilon-greedy strategy selects the best arm with probability $1-\epsilon$ (exploitation), and select a random arm (with uniform probability) with probability ϵ (exploration). In our example, as $\mu_o > \mu_n$, restaurant 1 is the "best arm". And restaurant 2 will be selected during the exploration stage, with probability $\epsilon/2$ as there are two restaurants in total. ϵ controls the degree of exploration and typically takes a small value such as $\epsilon = 0.1$ (Sutton and Barto 2018).

Table 8 summarizes the probability of recommending the new restaurant for the three methods. We see that the probability is less than 50% for both Thompson sampling and epsilon-greedy strategy, whereas UCB guarantees 100% of recommending the new restaurant given large enough weight. In other words, although all three MAB strategies enjoy the theoretical guarantees on the regret bound (Auer et al. 2002, Agrawal and Goyal 2017), only the UCB formulation deterministically boosts new restaurants. This is the reason why we adopt a UCB-like formulation for the marketplace fairness outcome.

Strategy	**UCB**	**Thompson Sampling**	**Epsilon-Greedy**
$\mathbb{P}$(Recommending new restaurant)	100 % given large enough κ	less than 50 %	5 %

Table 8 The chance of recommending a new restaurant under Upper Confidence Bound (UCB), Thompson sampling and epsilon-greedy strategy (with $\epsilon = 0.1$).

B.2. R-Module

B.2.1. Proof of the concavity of the Pareto frontier.

Proof of the concavity of the Pareto frontier. We prove the case for the trade-off between the conversion objective $C(\boldsymbol{x})$ and the basket value objective $B(\boldsymbol{x})$. The cases for other objectives readily follow.

Given B^* is a fixed constant independent of $\boldsymbol{x}$, we let $\lambda := \alpha_b B^*$ and rewrite the optimization problem as

$$\begin{aligned} &\max_{\boldsymbol{x}\in\mathcal{F}} C(\boldsymbol{x}) \\ &\text{s.t. } B(\boldsymbol{x}) \geq \lambda \end{aligned} \tag{20}$$

where $\mathcal{F} = \{\boldsymbol{x} \in \mathcal{E} : R(\boldsymbol{x}) \geq \alpha_r R^*,\ F(\boldsymbol{x}) \geq \alpha_f F^*\}$ is the feasible region for $\boldsymbol{x}$. Let $\boldsymbol{x}^*_\lambda$ be the solution to Eq.(20) which is a function of λ. We would like to show that $C(\boldsymbol{x}^*_\lambda)$ is concave in $B(\boldsymbol{x}^*_\lambda)$ as a function of λ. We decompose the proof into two steps, which are the two claims below.

Claim 1: $z(\lambda) := C(\boldsymbol{x}^*_\lambda)$ is a concave function of λ.

Proof. Define the Lagrangian function as

$$L_\lambda(\boldsymbol{x},\mu) = C(\boldsymbol{x}) + \mu(B(\boldsymbol{x}) - \lambda), \quad \mu \geq 0. \tag{21}$$

Therefore the dual problem for Eq.(20) can be written as

$$\begin{aligned} D_\lambda(\mu) &= \max_{\boldsymbol{x}\in\mathcal{F}} L_\lambda(\boldsymbol{x},\mu) = -\mu\lambda + \max_{\boldsymbol{x}\in\mathcal{F}} (C(\boldsymbol{x}) + \mu B(\boldsymbol{x})) \\ &:= -\mu\lambda + \kappa(\mu), \end{aligned} \tag{22}$$

where $\kappa(\mu) := \max_{\boldsymbol{x}\in\mathcal{F}} (C(\boldsymbol{x}) + \mu B(\boldsymbol{x}))$. Because Eq.(20) is a feasible linear optimization problem, strong duality holds, i.e.

$$z(\lambda) = \max_{\boldsymbol{x}\in\mathcal{G}_\lambda} C(\boldsymbol{x}) = \min_{\mu\geq 0} D_\lambda(\mu), \tag{23}$$

where $\mathcal{G}_\lambda = \{\boldsymbol{x} \in \mathcal{F} : B(\boldsymbol{x}) \geq \lambda\}$ is the feasible region for Eq.(20). For any positive λ_1, λ_2 and $t \in [0,1]$, we have

$$\begin{aligned} z(t\lambda_1 + (1-t)\lambda_2) &= \min_{\mu \geq 0} (-\mu(t\lambda_1 + (1-t)\lambda_2) + \kappa(\mu)) \\ &\geq t \min_{\mu \geq 0}(-\mu\lambda_1 + \kappa(\mu)) + (1-t)\min_{\mu \geq 0}(-\mu\lambda_2 + \kappa(\mu)) \\ &= tD_{\lambda_1}(\mu) + (1-t)D_{\lambda_2}(\mu) \\ &= tz(\lambda_1) + (1-t)z(\lambda_2), \ \forall t \in [0,1]. \end{aligned} \tag{24}$$

Therefore by definition of concavity, $z(\lambda)$ is concave in λ.

Claim 2: $B(\boldsymbol{x}_\lambda^*)$ is a piecewise linear function of λ. Specifically, $B(\boldsymbol{x}_\lambda^*) = \lambda_0$ for $\lambda \leq \lambda_0$, $B(\boldsymbol{x}_\lambda^*) = \lambda$ for $\lambda > \lambda_0$.

Proof. Let

$$\boldsymbol{x}_0 = \arg\max_{\boldsymbol{x} \in \mathcal{F}} C(\boldsymbol{x}) \tag{25}$$

be the solution to a modified version of Eq.(20) that relaxes the feasible region from $\mathcal{G}$ to $\mathcal{F}$. If there is more than one solution to Eq.(25), pick x_0 to be the one such that $B(\boldsymbol{x})$ is *maximized*. Let $\lambda_0 = B(\boldsymbol{x}_0)$. λ_0 can be bigger or smaller than λ. We discuss the two cases separately below.

If $\lambda \leq \lambda_0$, then x_0 is also the solution to the original optimization problem in Eq.(20). Therefore $B(\boldsymbol{x}_\lambda^*) = \lambda_0$.

Otherwise, if $\lambda > \lambda_0$, next we show that $B(\boldsymbol{x}_\lambda^*) = \lambda$. Because $\mathcal{G} \subseteq \mathcal{F}$ and $\boldsymbol{x}_\lambda^* = \arg\max_{\boldsymbol{x} \in \mathcal{G}} C(\boldsymbol{x})$, we have

$$C(\boldsymbol{x}_\lambda^*) \leq C(\boldsymbol{x}_0). \tag{26}$$

We know that

$$B(\boldsymbol{x}_0) = \lambda_0 < \lambda \leq B(\boldsymbol{x}_\lambda^*). \tag{27}$$

If $C(\boldsymbol{x}_\lambda^*) = C(\boldsymbol{x}_0)$, by Eq.(27) it contradicts with the assumption that $\boldsymbol{x}_0$ is picked among the optimal solutions such that $B(\boldsymbol{x})$ is maximized. Therefore the inequality in Eq.(26) is strict, i.e.

$$C(\boldsymbol{x}_\lambda^*) < C(\boldsymbol{x}_0). \tag{28}$$

Note that if $B(\boldsymbol{x}_\lambda^*) > \lambda$, we have $B(\boldsymbol{x}_\lambda^*) > \lambda > B(\boldsymbol{x}_0)$. By linearity of $B(\cdot)$, we have that there exists a $\boldsymbol{x}' = t'\boldsymbol{x}_0 + (1-t')\boldsymbol{x}_\lambda^*$ such that $t' \in (0,1)$ and $B(\boldsymbol{x}') = \lambda$. Then by linearity of $C(\cdot)$ and Eq.(28), we have

$$\begin{aligned} C(\boldsymbol{x}') &= t'C(\boldsymbol{x}_0) + (1-t')C(\boldsymbol{x}_\lambda^*) \\ &> t'C(\boldsymbol{x}_\lambda^*) + (1-t')C(\boldsymbol{x}_\lambda^*) \\ &= C(\boldsymbol{x}_\lambda^*). \end{aligned} \tag{29}$$

Because the feasible region $\mathcal{G}$ is convex and $\boldsymbol{x}'$ is a linear combination of two points within $\mathcal{G}$, we have $\boldsymbol{x}' \in \mathcal{G}$ but $C(\boldsymbol{x}') > C(\boldsymbol{x}_\lambda^*)$. This contradicts the fact that $\boldsymbol{x}_\lambda^*$ is the optimal solution for Eq.(20). So we must have $B(\boldsymbol{x}_\lambda^*) = \lambda$ for $\lambda > \lambda_0$. So $B(\boldsymbol{x}_\lambda^*)$ is a piecewise linear function in λ.

Finally, combining Claim 1 and 2, we arrive at the conclusion that $C(\boldsymbol{x}_\lambda^*)$ is concave in $B(\boldsymbol{x}_\lambda^*)$. In other words, the trade-off curve between $C(\boldsymbol{x}_\lambda^*)$ and $B(\boldsymbol{x}_\lambda^*)$ with varying λ is a concave curve. The benefit of a concave trade-off curve is illustrated in Fig.10. Comparing with point A on the trade-off curve, point B achieves a big boost in $B(\boldsymbol{x}_\lambda^*)$ with only a small sacrifice in $C(\boldsymbol{x}_\lambda^*)$. $\square$

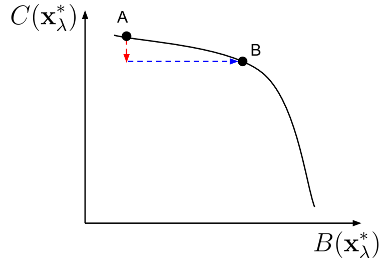


Figure 10 An illustration of a concave trade-off curve. A small sacrifice in one of the objectives can lead to a big improvement in the other.

B.2.2. Formulation and solution for the constrained optimization problem in R-module. We adopt the trick in Agarwal et al. (2012) and add a quadratic penalty term to the objective function in Eq.(13) for an efficient and scalable solution that can be readily served in large-scale online systems. Specifically, we penalize the squared Frobeniums norm between $\boldsymbol{x}$ and a uniform ranking plan $\boldsymbol{u} = \{u_{iq} = \frac{1}{Q}, \forall i, q\}$ that assigns equal probability to all items for all consumers[41]:

$$\begin{aligned} \max_{\boldsymbol{x}\in\mathcal{E}} \; & C(\boldsymbol{x}) - \frac{\gamma}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_F^2 \\ \text{s.t. } & B(\boldsymbol{x}) \geq \alpha_b B^*,\; R(\boldsymbol{x}) \geq \alpha_r R^*,\; F(\boldsymbol{x}) \geq \alpha_f F^*, \end{aligned} \tag{30}$$

Proposition 3 below provides the solutions to Eq.(30). Propsition 4 provides guidance on serving the solution for online systems.

Proposition 3 *The solution to Eq.(30) is*

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_f f_{iq} - \mu_i) + \frac{1}{Q}, \tag{31}$$

for any $x_{iq} > 0$. Here $\lambda_b, \lambda_r, \lambda_f$ are the slack variables for the constraints on $B(\boldsymbol{x}), R(\boldsymbol{x})$ and $F(\boldsymbol{x})$ respectively, and are functions of α_b, α_r and α_f. μ_i is the slack variable for the constraint $\sum_q x_{iq} = 1$.

Proof of Proposition 3. First, we write out the element-wise form of Eq.(30):

$$\begin{aligned} \max_{\{x_{iq}\}\in\mathcal{E}} \; & \sum_{i,q}(x_{iq}c_{iq} - \frac{\gamma}{2}(x_{iq} - \frac{1}{Q})^2) \\ \text{s.t. } & \sum_{i,q} x_{iq} c_{iq} b_{iq} \geq \alpha_b B^*, \\ & \sum_{i,q} x_{iq} c_{iq} r_{iq} \geq \alpha_r R^*, \\ & \sum_{i,q} x_{iq} f_{iq} \geq \alpha_f F^*, \\ & x_{iq} \geq 0,\; i = 1, ..., I, q = 1, ..., Q, \\ & \sum_{q} x_{iq} = 1,\; i = 1, ..., I, \end{aligned} \tag{32}$$

where $c_i q$, $r_i q$, $b_i q$, and $f_i q$ are the values for the consumer conversion objective, consumer retention objective, basket value objective and fairness objective between consumer i and item q, respectively. The objective for the maximization problem in Eq.(32) is concave, the inequality are all affine functions. Therefore, the KKT conditions are *necessary and sufficient* conditions for optimality. We use them to solve Eq.(32).

Let $\lambda_b, \lambda_r, \lambda_f, \delta_{iq}$ and μ_i be the non-negative slack variables for the five sets of constraints in Eq.(32), which are used to define the Lagrangian:

$$\begin{aligned} L(\{x_{iq}\}, \lambda_b, \lambda_r, \lambda_f, \{\delta_{iq}\}, \{\mu_i\}) = & \sum_{i,q}(x_{iq}c_{iq} - \frac{\gamma}{2}(x_{iq} - \frac{1}{Q})^2) - \lambda_b(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*) \\ & - \lambda_r(\sum_{i,q} x_{iq}c_{iq}r_{iq} - \alpha_r R^*) - \lambda_f(\sum_{i,q} x_{iq}f_{iq} - \alpha_f F^*) \\ & - \delta_{iq}x_{iq} + \mu_i(\sum_q x_{iq} - 1). \end{aligned} \tag{33}$$

By *stationarity* from the KKT conditions, we have

$$-c_{iq} + \gamma(x_{iq} - \frac{1}{Q}) - \lambda_b c_{iq}b_{iq} - \lambda_r c_{iq}r_{iq} - \lambda_f f_{iq} - \delta_{iq} + \mu_i = 0, \tag{34}$$

which yields

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq} + \delta_{iq} - \mu_i) + \frac{1}{Q}. \tag{35}$$

By *complementary slackness* from the KKT conditions, $x_{iq} > 0$ only when $\delta_{iq} = 0$. Therefore

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq} - \mu_i) + \frac{1}{Q} \tag{36}$$

for any $x_{iq} > 0$. □

Proposition 4 *Ranking according to x_{iq} in Eq.(31) is equivalent to ranking according to*

$$\tilde{x}_{iq} = c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq}. \tag{37}$$

Proof of Proposition 4. When serving the ranking plan $\boldsymbol{x}$ for consumer i, only the relative ordering of x_{iq} matters. Therefore the intercept $\frac{1}{Q}$, the multiplier $\frac{1}{\gamma}$ and μ_i do not have affect the final ranking results. □

We now show that λ_b, λ_r, and λ_f can be solved as functions of α_b, α_r and α_f in addition to the other inputs. By *primal feasibility* from the KKT conditions, we have $\sum_q x_{iq} = 1, \forall i$. Plugging in Eq.(35) and solve for μ_i we have

$$\mu_i = \frac{1}{Q}\sum_q(c_iq + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq} + \delta_{iq}),\ \ i = 1, ..., I, \tag{38}$$

Which are I linear equations involving the unknown variables $\mu_i (i = 1, ..., I)$, λ_b, λ_r and λ_f.

By complementary slackness from the KKT conditions, we have

$$\begin{aligned} \lambda_b(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*) = 0, \\ \lambda_r(\sum_{i,q} x_{iq}c_{iq}r_{iq} - \alpha_r R^*) = 0, \\ \lambda_f(\sum_{i,q} x_{iq}f_{iq} - \alpha_f F^*) = 0. \end{aligned} \tag{39}$$

The first equation in Eq.(39) implies either $\lambda_b = 0$, or $(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*) = 0$, which is another linear equation for $\mu_i (i = 1, ..., I)$, λ_b, λ_r and λ_f after plugging in Eq.(35). Similar observations hold for the other two equations in Eq.(39). Therefore, combining Eq.(38) and Eq.(39), we have a linear system with $I+3$ unknowns and $I+3$ equations, which can be solved using any linear equation solver.

In practice, I is the number of consumers, so solving the linear system directly can be expensive. We propose instead of solving λ_b, λ_r and λ_f as a function of α_b, α_r and α_f which are treated as tuning parameters, we propose to treat λ_b, λ_r and λ_f as tuning parameters directly to reduce computation. In addition, λ_b, λ_r and λ_f can also be viewed as the weights controlling the relative importance of the different objectives.

C. Additional Experiment Details

C.1. Latest Production Recommender System at the Company

The latest production recommender system at the company is a framework using three disjoint machine learning (ML) models to rank carousels and single restaurants in the homepage, based on conversion rate as the single objective: (1) A $(consumer, restaurant)$-level model predicting the conversion objective on restaurant level, i.e. the probability that the consumer will order from the restaurant in the current session, which is used to determine the ranking among the single restaurants and within each carousel (**ML Model A**); (2) A $(consumer, carousel)$-level model predicting the conversion objective on carousel level, i.e. the probability that the consumer will order from *any* restaurant inside the carousel in the current session, which is used to determine the ranking among the carousels (**ML Model B**); (3) A $(consumer, number\ of\ carousels)$-level model predicting the conversion rate under different number of carousels recommended, which is used to determine how many carousels to display in the current session (**ML Model C**). Figure 11 shows an overview of the production recommender system.

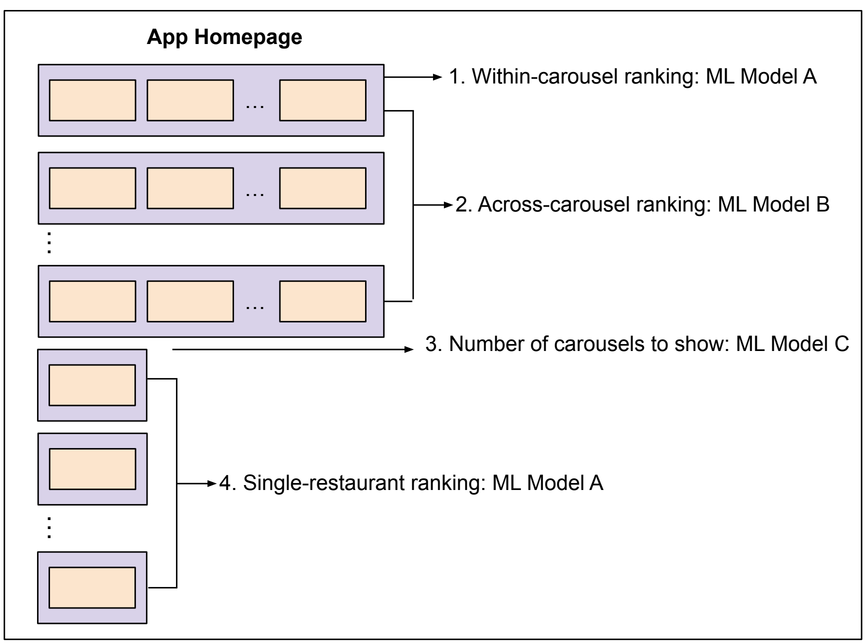


Figure 11 An overview of the latest production recommender system at the company.

All of the models are real-time personalized ML models, using the state-of-art hybrid recommender systems (Burke 2002) based on gradient boosting decision trees with the features and hyperparameters same as those in Appendix B.1.2. For fair comparison, we adopt the same model architecture and model size for estimating the individual objectives in the MO-module of the MOHR framework for the experiments at the company.

Because the framework is unable to generate calibrated ranking scores across carousels and single restaurants, all of the carousels are ranked above all of the single restaurants in the production recommender system.

C.2. Variance Correction For Ratio Metrics with Intra-consumer Correlation

Different sessions from the same consumer during the experiment period could be correlated with each other. To explicitly account for this intra-consumer correlation, we derive the corrected variance calculation for the three ratio metrics in Table 3 in the hypothesis testing procedure. Without loss of generality, we present the derivation for the conversion rate metric below. The derivation for the basket value per order and retention rate readily follows.

Following the notation in Table 3, let

$$\bar{O} = \frac{1}{I}\sum_i \sum_s O_{is}, \;\; \bar{S} = \frac{1}{I}\sum_i S_i \tag{40}$$

be the average number of orders O and average number of sessions S *per consumer*. Therefore, the conversion rate metric $C = \bar{O}/\bar{S}$ is the ratio of the two. We assume that the observations within each consumer could be correlated, but the observations across different consumers are independent. By multivariate central limit theorem, we have

$$\begin{pmatrix}\bar{O}\\ \bar{S}\end{pmatrix} \overset{I\to\infty}{\sim} N\left[\begin{pmatrix}\mu_O\\ \mu_S\end{pmatrix}, \begin{pmatrix}\sigma_O^2/I & Cov(O,S)/I\\ Cov(O,S)/I & \sigma_S^2/I\end{pmatrix}\right] \tag{41}$$

where μ_O and σ_O^2 are the mean and variance of the random variable O (number of orders from each consumer), μ_S and σ_S^2 are the variance of the random variable S (number of sessions from each consumer), and $Cov(O,S)$ is the covariance between O and S. By multivariate delta method, we have the conversion rate

$$C = \bar{O}/\bar{S} \sim N(\mu_O/\mu_S, \sigma_C^2), \tag{42}$$

where

$$\begin{aligned}
\sigma_C^2 &= Var(\frac{\bar{O}}{\bar{S}})\\
&= \begin{pmatrix}\frac{\partial}{\partial\bar{O}}(\frac{\bar{O}}{\bar{S}}) & \frac{\partial}{\partial\bar{S}}(\frac{\bar{O}}{\bar{S}})\end{pmatrix} \begin{pmatrix}\sigma_O^2/I & Cov(O,S)/I\\ Cov(O,S)/I & \sigma_S^2/I\end{pmatrix} \begin{pmatrix}\frac{\partial}{\partial\bar{O}}(\frac{\bar{O}}{\bar{S}})\\ \frac{\partial}{\partial\bar{S}}(\frac{\bar{O}}{\bar{S}})\end{pmatrix}\\
&= \begin{pmatrix}\frac{1}{\bar{S}} & -\frac{\bar{O}}{\bar{S}^2}\end{pmatrix} \begin{pmatrix}\sigma_O^2/I & Cov(O,S)/I\\ Cov(O,S)/I & \sigma_S^2/I\end{pmatrix} \begin{pmatrix}\frac{1}{\bar{S}}\\ -\frac{\bar{O}}{\bar{S}^2}\end{pmatrix}\\
&= \frac{1}{I}\left[\frac{\sigma_O^2}{\bar{S}} + \frac{\bar{O}^2}{\bar{S}^4}\sigma_S^2 - \frac{2\bar{O}}{\bar{S}^3}Cov(O,S)\right].
\end{aligned} \tag{43}$$

When computing the p-values for C, σ_O^2, σ_S^2 and $Cov(O,S)$ can be plugged in as the sample variance and covariance estimated from the data. Generally speaking, the estimated variance is *larger* when considering the intra-consumer correlation compared with treating all sessions to be i.i.d.. So the variance correction in Eq.(43) yields a p-value that's larger than if treating all sessions as i.i.d., making the hypothesis testing more rigorous and conservative.

C.3. Results on the MO-module

Table 9 summarizes the model performance and top important features for the ML-based objectives, namely consumer conversion, consumer retention and basket value. The feature importance score for gradient boosted trees is defined as in (Friedman 2001).

Model name	Model performance	Top 10 important features
Consumer conversion	Test AUC = 0.8797	Normalized (consumer, restaurant) order count
		Local hour of day
		Consumer view count
		Normalized (consumer, restaurant) impression count
		Normalized (consumer, restaurant) click count
		$u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding
		Consumer order-to-impression ratio
		Restaurant delivery time
		Meal period
		(restaurant, source) order-to-impression ratio
Consumer retention	Test AUC = 0.7847	Consumer order counts in the past 120 days
		Restaurant average basket value
		Consumer order counts in the past 14 days
		Consumer order counts in the past 7 days
		Delivery radius
		Consumer ride count
		City
		% of consumers churned after ordering from restaurant j in past 60 days
		% of consumers churned after ordering from restaurant j in past 30 days
		% of consumers churned after ordering from restaurant j in past 120 days
Basket value	Test rMSE = 0.1135	(consumer, restaurant) average basket value in the past 120 days
		Consumer average basket value in the past 120 days
		$u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding
		Local hour of day
		(restaurant, source) average basket value in the past 120 days
		Source name
		Restaurant average basket value in the past 120 days
		$cos(u_i, v_j)$, i.e. cosine similarity between consumer and restaurant
		(consumer, restaurant) average basket value in the past 60 days
		Consumer average basket value in the past 60 days

Table 9 Top 10 important features for the ML models in the MO-module, measured by the feature importance score of the gradient boosted trees.

C.4. Results on the H-Module

Table 10 and Fig.12 presents the estimated scrolling factors in the experiment. Note that there are at most 6 restaurants presented in every carousel. To see more restaurants within the carousel, there is a "see all" button at the top right corner of every carousel. The H-module is only applied to the top 6 positions within each carousel.

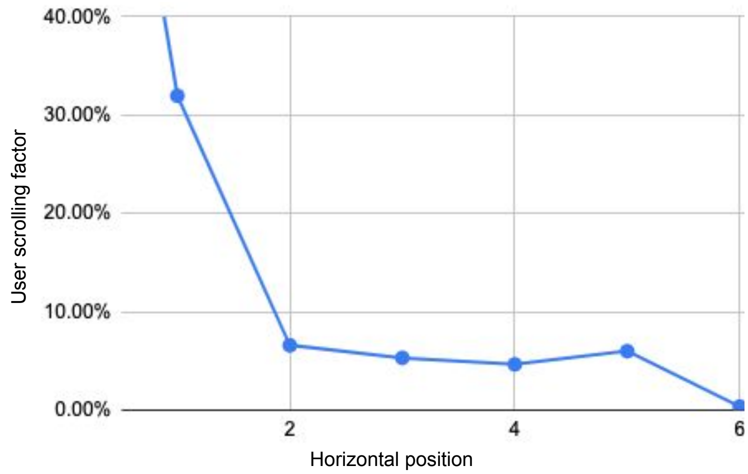


Figure 12 **Scrolling factors from the consumer browsing model.**

Horizontal position	Consumer scrolling factor $\hat{p}_{l,l+1}$
0	100.00%
1	31.96%
2	6.58%
3	5.32%
4	4.65%
5	5.99%
6	0.36%

Table 10 **Estimated values for the scrolling factors from the consumer browsing model.**

C.5. Robustness Checks

C.5.1. Randomization Check. To check if the random assignment for the online experiments truly holds, we inspect the treatment and control consumers *before* the experiment start date, when they were all receiving recommendations from the same production recommender system. This is called A/A test in the industry. Specifically, we collect 472 metrics related to different aspects of consumer behaviors[42] including the key business metrics in the results above, and conduct hypothesis testing on whether the differences between treatment and control groups are statistically significant before the experiment start date. Specifically, we compute the p-values for the metric differences between treatment and control group 28 days *before* the experiment start date, when both treatment and control consumers are expected to receive recommendations generated by the same algorithm. Table 11 shows that the A/A testing p-values are all greater than 0.05 (or 0.10 depending on the significance level of choice), suggesting that there is no significant difference in the treatment and control group in terms of the key business metrics, before the experiment start date.

Metric	Conversion rate	Basket value per order	Retention rate	Orders per consumer	Search rate
A/A testing p-value	0.326	0.452	0.947	0.853	0.286

Table 11 p-values for the A/A testing on key business metrics.

We further collected a comprehensive set of 472 metrics capturing various aspects of consumer behavior on the platform and across different surfaces, and computed the p-values for the 472 metric differences. Under the null hypothesis that treatment and control group consumers are not statistically different, the p-values should follow a uniform distribution. We conduct the Kolmogorov-Smirnov (KS) test on the empirical distribution of those 472 p-values, and could not reject the null that they follow a uniform distribution on [0,1] (Fig.13), suggesting that our randomization holds true.

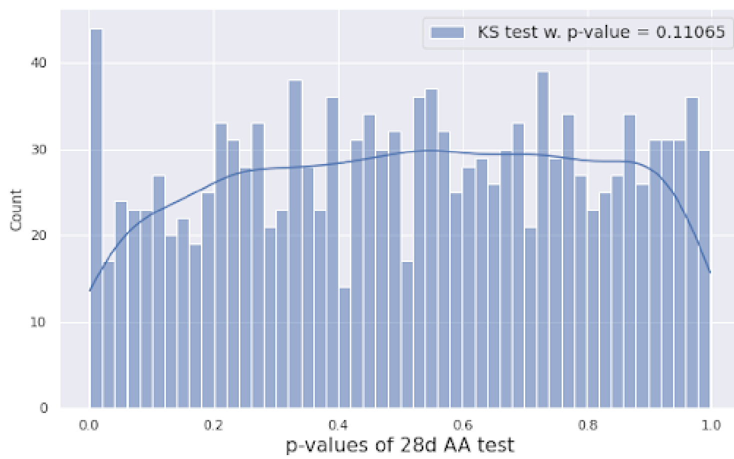


Figure 13 Histogram of the p-values for the 472 metric differences for A/A test. Kolmogorov-Smirnov (KS) test which compares the empirical distribution of the p-values against the uniform distribution on [0,1] has p-value of 0.11, which fails to reject the null hypotheses that these metrics are not statistically significantly different during the A/A testing period.

C.5.2. Eliminating the Novelty Effects. With the MOHR framework, carousels and single restaurants are mixed together and the homepage appears as a heterogeneous arrangement of items. One might argue that the new display may introduce a novelty effect (Koch et al. 2018) and consumers' engagement levels with the platform might be higher in the beginning than when they become familiar with the new design. We identified three pieces of evidence to counter this argument. First, the first three days of experiment data is discarded in computing the metrics reported in the previous sections, which eliminates part of the novelty effect. Second, there is an improvement in long-term consumer retention (+0.7%) which by definition measures consumers' future engagement with the platform after finishing the current session. This means that the treatment effect of MOHR persists for at least 28 days. Lastly, we measured the metrics for the new consumers during the experiment, whose *first* interaction with the platform is either always under the current production system (control consumers) or always under the new display under the MOHR framework (treatment consumers). Therefore, there is no novelty effect at play for those consumers. We observe a significant 5.5% increase in the retention of new consumers which is even larger than the overall retention increase. This suggests that the novelty effect, if it exists, actually impacts the effects of the MOHR framework negatively as it introduces a "shock" to the existing consumers with a new display, so that the positive effects on them are actually smaller than the new consumers who have no prior experience with the platform.