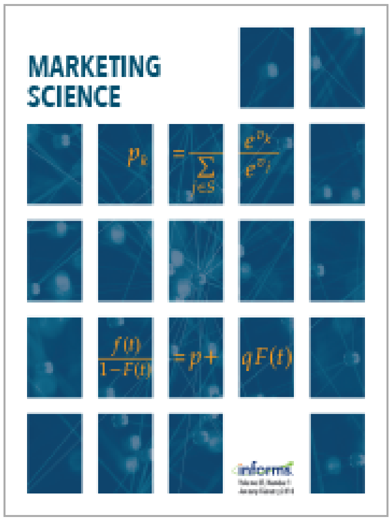


Marketing Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

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

Yuyan Wang; , Long Tao, Xian Xing Zhang

To cite this article:
Yuyan Wang; , Long Tao, Xian Xing Zhang (2025) Recommending for a Multi-Sided Marketplace: A Multi-Objective Hierarchical Approach. Marketing Science 44(1):1-29. https://doi.org/10.1287/mksc.2022.0238

informs
https://pubsonline.informs.org/journal/mksc

MARKETING SCIENCE
Vol. 44, No. 1, January–February 2025, pp. 1–29
ISSN 0732-2399 (print), ISSN 1526-548X (online)

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

Yuyan Wang,[a,*] Long Tao,[b] Xian Xing Zhang[b]

[a] Stanford Graduate School of Business, Stanford, California 94305; [b] Uber Technologies, Inc., San Francisco, California 94158
*Corresponding author
Contact: yuyanw@stanford.edu, https://orcid.org/0000-0001-5585-3789 (YW); long.tao@uber.com (LT); xx@pinterest.com (XXZ)

Received: June 30, 2022
Revised: October 3, 2023; April 5, 2024
Accepted: May 7, 2024
Published Online in Articles in Advance: August 14, 2024

https://doi.org/10.1287/mksc.2022.0238

Abstract. Recommender systems play a vital role in driving the long-term values for online platforms. However, developing recommender systems for multi-sided platforms faces two prominent challenges. First, recommending for multi-sided platforms typically involves a joint optimization of multiple, potentially conflicting objectives. Second, many platforms adopt hierarchical homepages, where items can either be individual products or groups of products. Off-the-shelf recommendation algorithms are not applicable in these settings. To address these challenges, we propose MOHR, a novel multi-objective hierarchical recommender. By combining machine learning, probabilistic hierarchical aggregation, and multi-objective optimization, MOHR efficiently solves the multi-objective ranking problem in a *hierarchical* setting through an innovative formulation of probabilistic consumer behavior modeling and constrained optimization. We implemented MOHR at Uber Eats, one of the world's largest food delivery platforms. Online experiments showed significant improvements in consumer conversion, retention, and gross bookings, resulting in a $1.5 million weekly increase in revenue. Moreover, MOHR offers managers a mathematically principled tool to make quantifiable and interpretable trade-offs across multiple objectives. As a result, it has been deployed globally as the recommender system for Uber Eats' app homepage.

History: Puneet Manchanda served as the senior editor for this article.

Supplemental Material: The online appendices and data files are available at https://doi.org/10.1287/mksc.2022.0238.

Keywords: recommender systems • multi-sided marketplace • food-delivery platforms

1. Introduction

1.1. Research Context

Recommender systems are becoming increasingly ubiquitous across retail (Xiao and Benbasat 2007, Zhang et al. 2011), media (Miller et al. 2003, Covington et al. 2016), travel (Ghose et al. 2012, Ursu 2018), news (Prawesh and Padmanabhan 2014, Dhillon and Aral 2021), and social platforms (Li et al. 2017, Xie 2010). They facilitate information acquisition and decision-making for the consumers while also helping sellers to efficiently target prospective consumers. Recommender systems have become the key drivers for consumer growth in many personalization platforms today. YouTube, the leading global video-sharing platform, has attributed 70% of watch time to recommendations (Solsman 2018). Netflix reported that personalized recommendations now contribute to 80% of the content consumption (Gomez-Uribe and Hunt 2015).

Today, many recommender systems operate on multi-sided platforms that bring buyers, sellers and other partners together. These platforms create value by reducing search and transaction costs (Evans and Schmalensee 2016). For example, Airbnb as a two-sided marketplace, connects people seeking accommodations with those offering their homes for rent. YouTube and other content-sharing platforms operate as three-sided marketplaces, engaging users, content creators, and advertisers. Food and grocery delivery platforms like Instacart and Uber Eats establish three-sided marketplaces involving consumers, sellers, and couriers (Bahrami et al. 2021). With the recent breakthroughs in technology, multi-sided platforms have become more prevalent than ever (Evans and Schmalensee 2016).

1.2. Research Agenda and Challenges

The success of a multi-sided platform relies on attracting and retaining participants from all sides (Schmalensee et al. 1989, Abdollahpouri et al. 2020) and balancing short-term goals with long-term ones (Wu et al. 2017). This entails a joint optimization of multiple objectives that are potentially conflicting. For example, a news website could make more short-term revenue by devoting larger spaces to ads; but then it would risk losing readers because of the overwhelming ads (Aribarg and Schwartz 2020). Conflicting objectives exist even within the same side of the marketplace. A content sharing platform that is optimized for click-through rates could end up recommending clickbait or pigeonholing content, hurting long-term consumer experience (Fleder and Hosanagar 2009, Chaney et al. 2018, Chen et al. 2021). Therefore, the platform needs to carefully trade off among the multiple objectives. 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 single-objective recommendation has been recognized in recent 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.

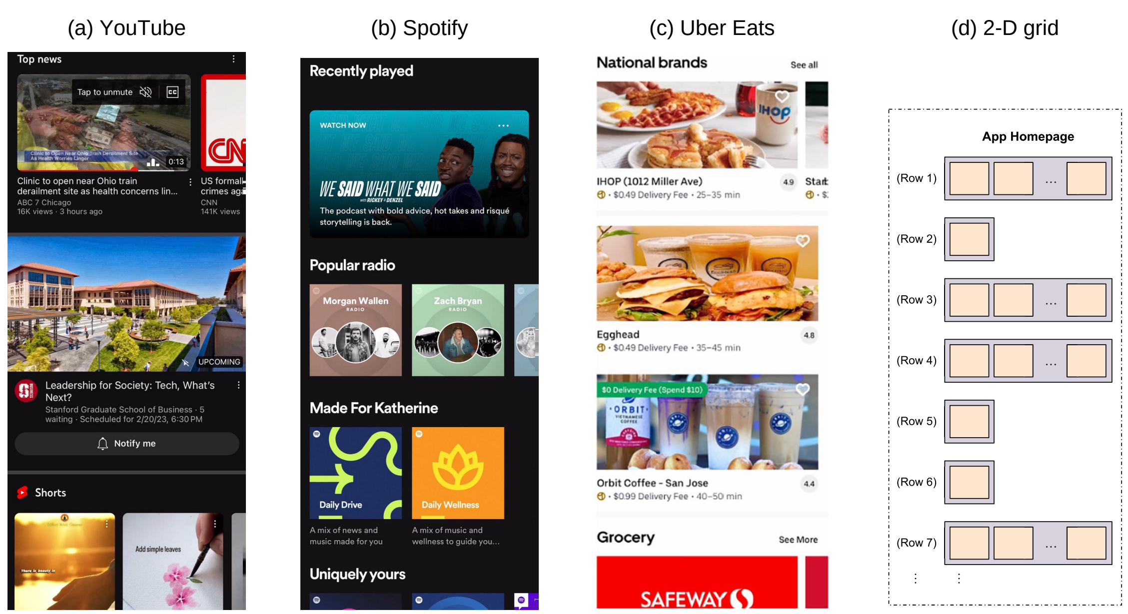
Another challenge for developing recommender systems for modern platforms is the prevalence of hierarchical recommendations. Figure 1 shows the homepages of YouTube, Spotify, and Uber Eats. In addition to individual products, the recommendations are also shown in the form of scrollable *rows* such as "Trending Now" and "National Brands." Off-the-shelf recommender systems are no longer applicable in this two-dimensional setting, as existing systems mostly focus on ranking items in a one-dimensional list. The multi-objective ranking problem becomes even more challenging in the presence of hierarchical recommendations. Specifically, it is unclear how to compare the "appealingness" of a row of products against a single product in the homepage, especially when multiple objectives are of interest. In other words, there lacks a mathematically principled framework capable of tackling the multi-objective trade-off with *hierarchical* recommendations.

1.3. Proposed Framework

In this work, we introduce MOHR, a principled, scalable recommendation framework for solving the multi-objective ranking problem in a *hierarchical* setting. Specifically, we formulate the problem of "hierarchical ranking with multiple objectives" as two sets of ranking 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.

We've implemented MOHR at Uber Eats, one of the world's largest food delivery platforms. By incorporating consumer-level, firm-level, and exploration objectives linked to long-term profitability, MOHR effectively

Figure 1. Examples of App Homepages and Illustration of a Two-Dimensional Hierarchical Homepage



Notes. (a)–(c) Examples of the app homepages. (d) Illustration of a two dimensional hierarchical homepage.

pushes forward the Pareto frontier of multiple business metrics, resulting in a weekly revenue increase of \$1.5 million. Because of its substantial impact, MOHR has been globally deployed as the recommender system for Uber Eats' app homepage, serving millions of consumers every day.

1.4. Contributions

Our research makes several contributions. Methodologically, we propose a novel machine learning (ML) framework that addresses two of the most prominent challenges in recommender systems for online platforms. Through an innovative formulation of consumer behavior modeling and constrained optimization, MOHR combines machine learning, probabilistic aggregation, and multi-objective optimization to deliver a personalized and hierarchical homepage optimized for multiple objectives. Empirically, we demonstrate that our approach works at scale and have deployed it at Uber Eats.

The managerial contributions of MOHR are threefold. First, we showcase that multi-objective optimization is effective at improving metrics tied to long-term profitability of the online platforms. Managers should explicitly model and optimize the multiple aspects of the business. Second, MOHR provides a convenient tool for the managers to make mathematically principled and quantifiable trade-offs among conflicting objectives. Last, we show that too much emphasis on one objective may hurt that objective and backfire.

Our framework is general and applicable to platforms within and outside the food delivery industry. MOHR can also be applied in a modularized fashion: It is applicable to cases if the firm is only concerned with one of the two aforementioned challenges (multiple conflicting objectives *or* hierarchical display). In addition, our framework is *not* limited to multi-sided platforms but rather any platforms with more than one objectives. In fact, most firms today care about multiple objectives such as short-term engagement and long-term retention. Our proposed framework is readily applicable to these settings as well.

The rest of this paper is organized as follows. Section 2 discusses the related literature. Section 3 introduces the institutional background and data. Section 4 presents the full ML framework. Section 5 and 6 provide offline and field experiment results. Section 7 discusses the contribution, implications and future research.

2. Related Work

2.1. Recommender Systems and Product Choice Modeling

Recommender systems are closely related to product choice modeling in the marketing literature, especially in predicting individuals' choices or preferences (Guadagni and Little 1983, Wagner and Taudes 1986, Fader and Hardie 1996, Ansari et al. 2000, Johnson et al. 2012, Farias and Li 2019). These works use consumer and product characteristics, as well as their interaction history, to enhance predictive accuracy and scalability (Bodapati 2008, Jacobs et al. 2016, Yoganarasimhan 2020). Our work contributes to this strand of literature by modeling consumer behavior with *hierarchically* presented products. More broadly, our work is related to personalized targeting (Blattberg et al. 2008) where ML models have been widely adopted. Simester et al. (2020b) looked into the robustness of different machine learning based targeting models to typical data challenges. Yoganarasimhan et al. (2023) evaluated different personalized ML targeting policies.

Methodology-wise, recommendation is formulated as a learning-to-rank task for information retrieval, which categorized into pointwise, pairwise, and listwise approaches (Liu 2009). Our framework contributes to pointwise recommender systems that are common in industrial recommender systems (Covington et al. 2016, Smith and Linden 2017). Regarding input information, there are three types of recommender systems: content-based filtering, collaborative filtering, and hybrid methods (Adomavicius and Tuzhilin 2005, Ricci et al. 2015). Content-based systems rely on item descriptions and consumer profiles (Mooney and Roy 2000, Brusilovsky 2007, Aggarwal 2016), whereas collaborative filtering assumes that consumers who liked similar items in the past will like similar items in the future (Billsus and Pazzani 1998, Breese et al. 1998). Most systems today use a hybrid approach, combining collaborative and content-based filtering (Adomavicius and Tuzhilin 2005, Sahoo et al. 2012). Our work contributes to the hybrid recommender system literature. To evaluate these systems offline, methods include inverse-probability weighting (IPW) (Horvitz and Thompson 1952), doubly robust estimation (Dudík et al. 2011, Simester et al. 2020a), and offline replay (Li et al. 2011, Aramayo et al. 2023) are widely adopted.

2.2. Effects of Recommendations

Recommender systems affect consumers' consumption patterns from various aspects (Xiao and Benbasat 2007), 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 impact 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). Recommender systems also affect other parties of the e-commerce marketplace through impacting demand levels (Oestreicher-Singer and Sundararajan 2012, Ghose et al. 2014, Kumar and Hosanagar 2019, Bourreau and Gaudin 2022), seller profits (Chen et al. 2008, Das et al. 2009, Azaria et al. 2013, Zhou and Zou 2023), and social welfare (Zhang et al. 2021, Aridor and Gonçalves 2022, Donnelly et al. 2023). In our work, we show that recommender systems can have a positive or negative impact on different sides in a multi-sided marketplace and propose a framework that addresses the conflicting objectives in an interpretable and principled way.

2.3. Multi-Objective Recommendation

Research has evolved from optimizing a single aspect of consumer feedback (Adomavicius and Tuzhilin 2005) to optimizing multidimensional preferences (Agarwal et al. 2012, Chung and Rao 2012, Ghose et al. 2012, Li et al. 2017). Building a recommender system for a multi-sided marketplace is essentially a multi-objective optimization problem (Sawaragi et al. 1985). Researchers have considered seller earnings and platform profits (Chen et al. 2008, Hosanagar et al. 2008, Das et al. 2009, Azaria et al. 2013) or total welfare (Zhang et al. 2021, Aridor and Gonçalves 2022) as objectives for the recommender systems.

The multiple objectives of interest often conflict with each other. Hosanagar et al. (2008) examined trade-offs between consumer relevance and firm profit margins and short-term versus long-term profits. Zhang et al. (2021) showed that maximizing profit can hurt consumer surplus. To the best of our knowledge, extant works mostly focus on two objectives and adopt a one-dimensional setting, where the output is a single ranked list of recommendation items. Our proposed framework extends the multi-objective optimization problem to a hierarchical setting and provides a mathematically principled tool to model, understand, and optimize the multi-objective trade-off.

2.4. Consumer Behavior Modeling and Hierarchical Recommendation

Our work contributes to the literature on modeling consumer behavior in recommender systems. Weitzman (1979) was among the first to model sequential search behavior. Built on this, there has been several works on developing structural models for consumer search (Ursu 2018, Jiang et al. 2021, Shi and Trusov 2021). These works focused on building 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, their model does not apply to the hierarchical setting in our case.

For hierarchical recommendation, 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. 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 one-dimensional list, although consumer decision is modeled in a hierarchical way (Agrawal et al. 2009, Oestreicher-Singer and Sundararajan 2012, Song et al. 2019). Our work bridges this gap by directly modeling consumer behavior on hierarchically presented recommendations and using it in building a hierarchical recommender system.

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 by 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 platforms 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. 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.[1]

Single-objective recommender systems that focus on only one criterion may be suboptimal for multi-sided platforms in the long term. For example, many food delivery platforms focus on optimizing consumers' short-term conversion rate (i.e., placing an order). This can lead to overly recommending popular and well-established restaurants, which 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, they face lack of selection and recommendation diversity as a result, leading to pigeon-holing effects (Fleder and Hosanagar 2009, Chaney et al. 2018) and consumer boredom (Liechty et al. 2005, Ursu et al. 2023). On the courier and platform side, the highly skewed demand hurts efficiency: 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 these issues hurt one or more sides of the marketplace and eventually the business. Therefore, a careful trade-off among the multiple objectives from different sides is needed to maintain a healthy platform and ensure the success of the business in the long term.

3.2. Hierarchically Presented Recommendations

As is shown in Figure 1, a recommendation item can be either a single product or a row of products. Rows are also referred to as "carousels" in industry and are a widely adopted form of recommendation (Elahi and Chandrashekar 2020). They offer several advantages. First, these rows can be viewed as nudges tailored to the different modes of the consumers (e.g., in a hurry, looking for something healthy). Second, the title of the rows provide extra information about the products (e.g., cuisine type) that may be critical to consumers' decision making but may not be readily visible otherwise. Last, they alleviate the cold-start challenges for new consumers and new products, by showing nonpersonalized recommendations such as "Popular Near You" or "National Favorites."

Although rows 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 rows and single products. As a result, the homepage is a two-dimensional grid (Figure 1(d)), and off-the-shelf recommender systems are no longer applicable.

3.3. Data

We work with Uber Eats, one of the largest food delivery platforms in the world. Consumers' interactions in the app are logged and processed through Apache Hive (Thusoo et al. 2009) for data extraction, transformation, and loading (ETL). 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, geolocation, and so on. 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 backend will call the recommender system to generate real-time recommendations.[2] There are about 120 *predetermined* row candidates (curated by product managers at the company), and each row has a *fixed* set of qualifying candidates.[3]

4. ML Framework

In this section, we describe the Multi-Objective Hierarchical Recommender (MOHR) framework. Section 4.1 provides a high-level overview of the whole framework. Sections 4.2, 4.3, and 4.4 describe the three modules in detail.

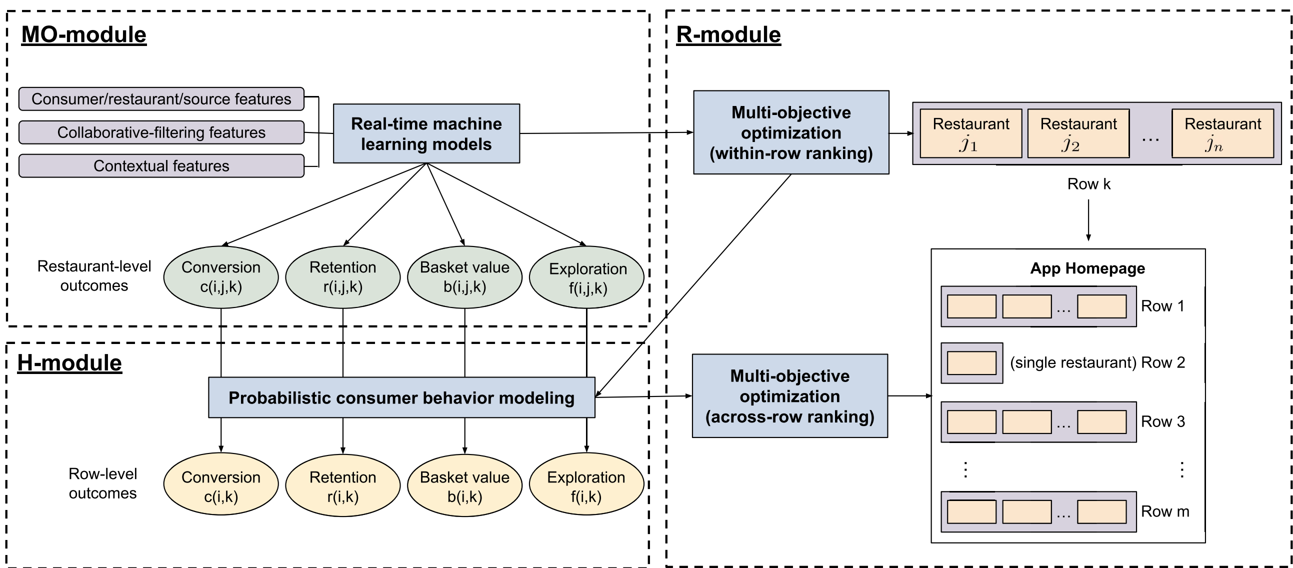
4.1. Overview and Selection of Objectives

4.1.1. Overview of the Framework. MOHR consists of three modules: a machine learning module (MO) for multiple outcomes, a hierarchical module (H) for hierarchical aggregation, and a scalable multi-objective optimization module (R) for ranking. The MO-module generates ML predictions for the multiple outcomes at the product (restaurant) level. The H-module aggregate these product-level predictions into row-level predictions. The R-module solves a large-scale multi-objective optimization problem for within-row ranking and across-row ranking. Each module takes input from the preceding 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.

4.1.2. Selection of Objectives. The objectives are selected based on several considerations. First, they should align with the platform's long-term profitability. Second, these objectives should exhibit a reasonable signal-to-noise (STN) ratio. For example, we do *not* include consumer satisfaction (measured by surveys), 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 it.[4] Last, the set of objectives should be sufficiently compact for efficient model training and deployment on large-scale platforms. Based on these considerations, we incorporate the following outcomes:

- **Consumer conversion**: Whether the consumer places an order. This captures the *short-term* order volume and is a common objective used by many recommender systems today (Covington et al. 2016). This is also the single objective used by the latest production recommender system at the company and captures the short-term revenue of the platform.
- **Consumer retention**: Whether the consumer returns to the platform and orders again within the time window,[5] if the consumer orders in the current session.[6] This outcome captures *long-term* consumer experience and long-term revenue and serves as a proxy for consumer satisfaction (Wu et al. 2017, Wang et al. 2022).
- **Basket value**: Dollar amount of the order, if the consumer orders.[7] This captures the earnings for both restaurant partners and delivery partners and directly ties to the platform's revenue.
- **Restaurant exploration**: Captures uncertainty on potentially successful restaurants. For new restaurants on the platform which the system has limited data on, this objective helps the ML system to more efficiently estimate their true values for better personalization. As

Figure 2. Overview of MOHR



a result, it offers new restaurants more incentives to join and stay with the platform, contributing to the platform variety and potentially increased profits in the long term.

This is the minimal set that we come up with that align closely with the long-term goals for the three-sided platform. The delivery partners' utility is captured by the basket value outcome, which determines their payment. Although all these outcomes contribute to the platform's long-term success, they potentially conflict with each other. As discussed in Section 3.1, overfocusing on consumer conversion would hurt consumer retention and new restaurant exposure. We will see in Section 6 that overfocusing on basket value would hurt consumer retention and even backfire. We will demonstrate that long-term optimization for the platform can be achieved through a multi-objective approach with a delicate balance across these objectives.

In the next section, we describe in detail how each objective is modeled as a machine learning outcome[8]

Table 1. Summary of Notations

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 (either a single restaurant within the row or a whole row
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, N_j^1	Number of impressions (N_j) and orders (N_j^1) on product j
I, Q	Number of consumers (I) and recommendation items (restaurants or rows) (Q)
$\boldsymbol{x} = \{x_{iq}\}$	Ranking plan, where x_{iq} is the probability of serving item q to consumer i
$\boldsymbol{u} = \{u_{iq}\}$	Uniform ranking plan, where $u_{iq} \equiv \frac{1}{Q}$
$c_{iq}, r_{iq}, b_{iq}, e_{iq}$	Compact forms for the consumer conversion, consumer retention, basket value and restaurant exploration

Table 2. Summary of Outcomes

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

in the MO-module. Table 1 summarizes the notations used. Table 2 summarizes the outcomes. We emphasize that, although we only discuss four outcomes 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* platforms.

4.2. MO-Module: ML Models for Product-Level Outcomes

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" row[9]). For example, a restaurant appearing in the "Under 25 minutes" row may be more appealing to a consumer who is in a hurry than the same restaurant appearing in the "Popular near you" row. 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.

4.2.1. ML 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

$$\begin{aligned} c(i,j,k) &= \mathbf{E}[O(i,j,k,z)=1] \\ &= f_c(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k, \boldsymbol{a}_{ij}, \boldsymbol{a}_{ik}, \boldsymbol{a}_{jk}, \boldsymbol{a}_{ijk}, \boldsymbol{z}), \\ r(i,j,k) &= \mathbf{E}[R(i,j,k,z)\mid O(i,j,k,z)=1] \\ &= f_r(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k, \boldsymbol{a}_{ij}, \boldsymbol{a}_{ik}, \boldsymbol{a}_{jk}, \boldsymbol{a}_{ijk}, \boldsymbol{z}), \\ b(i,j,k) &= \mathbf{E}[B(i,j,k,z)\mid O(i,j,k,z)=1] \\ &= f_b(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k, \boldsymbol{a}_{ij}, \boldsymbol{a}_{ik}, \boldsymbol{a}_{jk}, \boldsymbol{a}_{ijk}, \boldsymbol{z}), \end{aligned} \tag{1}$$

where we drop the dependency on context z for ease of notation. Here $\boldsymbol{a}_i$ represents the consumer-level features, $\boldsymbol{a}_{ijk}$ captures the interaction history between consumer i and restaurant j given that restaurant j appears in source k, and so on.

For consumer conversion and retention, which are binary outcomes, we adopt gradient boosting decision trees (Friedman 2001, 2002) as the nonlinear prediction function f_c and f_r. For the basket value outcome, we use gradient boosting regression tree as f_b with squared loss as the loss function (Hastie et al. 2009). They achieve a nice balance between predictive power and interpretability. Model training is done using the gradient boosting machine (GBM) on H2O (Hastie et al. 2009, Click et al. 2017), 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 Online Appendix A.1. On a high level, there are three groups of features:

- **RFM features**: We adopt the RFM (recency, frequency, and monetary value) paradigm in 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 and over multiple time horizons to capture the personalized, nonpersonalized, and temporal effects. See Figure A.1 in Online Appendix A.1 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. Vertical and horizontal position of the restaurant is also included as a feature to remove the position bias. During serving time, the position feature is set to zero (first position). This is a widely adopted method in recommender systems for removing off-policy training 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**: Collaborative filtering (Breese et al. 1998) features based on matrix

factorization are included to capture the similarities among consumers and products, based on the assumption that similar consumers like similar products. The output consists of embeddings for each customer, restaurant, and source, which are subsequently used as input features. Details are provided in Appendix A.1.

In total, there are about 200 features. See Appendix B.1 for model performance and important features for each ML model. Among the RFM features, not surprisingly, monetary value features (e.g., historical basket values) are important for the basket value outcome, and recency and frequency features (e.g., historical order count and churn rates) are important for the conversion and retention outcomes.

4.2.2. Bayesian Uncertainty for Restaurant Exploration Outcome. The restaurant exploration outcome can be viewed as the recommender system navigating an exploration-exploitation trade-off: It exploits established and popular restaurants while simultaneously exploring new restaurants that have the potential to become successful. This exploration-exploitation trade-off (March 1991) fits nicely into a contextual multi-armed bandit (MAB) framework (Katehakis and Veinott 1987, Auer et al. 2002). 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

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

where $Q(j)$ is the estimated value of action j, $\sigma(j)$ is the estimated uncertainty of the value of j, and $\kappa > 0$ controls the level of exploration.

We adopt a restaurant-level Bayesian model to estimate $\sigma(j)$ as the posterior standard deviation for the conversion rate for restaurant 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 parameters for the prior Beta distribution $\mathcal{B}(\alpha_j, \beta_j)$. There are N_j impressions on restaurant j, out of which N_j^1 lead to orders. See Appendix A.2 for a detailed derivation of Equation (3).

We define the restaurant exploration outcome $e_r(j)$ to be exactly the *uncertainty estimate* $\sigma(j)$ in Equation (3). As a sanity check, given α_j and β_j, lower N_j lead to *higher* $e_r(j)$. In other words, the *fewer* impressions a restaurant receives, the *more* uncertain the system is about the restaurant and hence the higher value for $e_r(j)$. The choices for the prior parameters α_j and β_j are discussed in Appendix A.2.1. Instead of using lifetime history, a trailing window of 120 days[10] is chosen for the impression counts N_j and order counts N_j^1.

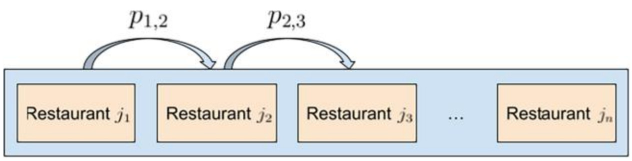
Remark 1. The benefits of the restaurant exploration outcome are threefold. First, it offers new restaurants more exposure on the platform. Second, it helps the learning of other objectives by introducing more training data on the new restaurants. Third, by restricting to a trailing window for counting N_j, it provides a mechanism for *adaptively* help new and low-volume restaurants over time: A restaurant will receive a high boost when first entering the platform, with Equation (2) dominated by the second term ("exploration"); As it gradually accumulates enough exposure, the boosting effect dies down, and Equation (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, $e_r(j)$ will go back up due to decreasing N_j and increasing $\sigma(j)$, effectively offering a "second chance" to the restaurant ("resurrection").

Remark 2. Although the restaurant exploration outcome is derived from a MAB framework for maximizing conversion under uncertainty, the effect of this outcome goes beyond it. First, the goal of MAB is to maximize the cumulative reward (in this case, cumulative conversion), whereas the proposed restaurant exploration outcome focuses on the *immediate* impact on new and low-volume restaurants. This property is achieved only through the UCB formulation, whereas other MAB algorithms such as Thompson sampling (Thompson 1933) and epsilon-greedy (Sutton and Barto 2018) do not have the same effect. See Appendix B.3 for details. Second, we do *not* select restaurants based on Equation (2) as a standard UCB procedure would. Instead, we only use the uncertainty component to define the restaurant exploration outcome, which will be jointly optimized with other objectives later in the R-module. Third, the restaurant exploration outcome is *nonpersonalized* and on a per-restaurant level, whereas a typical MAB framework would require a *personalized* uncertainty estimate. The restaurant-centric property further distinguishes it from typical MAB frameworks that are user-centric. Last, because of the time window mechanism, the restaurant exploration outcome consistently benefits new and low-volume restaurants, offering a "second chance" even if they experience poor performance during a certain time period. In contrast, a typical MAB algorithm aims to identify the best arm, leading to under-performing restaurants being ignored over time.

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 through a probabilistic hierarchical aggregation. An

Figure 3. Illustration of Consumer Scrolling Probabilities Within a Row



important component in this module is modeling consumers' scrolling behaviors within the hierarchical homepage, which we elaborate below.

We assume that the consumer examines the products in a row from left to right and one-by-one (Weitzman 1979) and may give up at a certain point as they have limited patience. The consumer state is defined as their viewing position (e.g., position X inside row Y). The state transitions are Markovian, and at every state, the consumer takes one of the following three actions[11]: order from the current product, continue browsing the next product, or abandon the whole row. To estimate these transition probabilities, we define a set of scrolling probabilities that a consumer will scroll to the next position inside a row,[12] as illustrated in Figure 3:

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

Next, we derive the row-level outcomes by aggregating the product-level outcomes using these scrolling probabilities.

We first derive the row-level conversion outcome $c(i, k)$, the probability that the consumer orders from any restaurant in row k. Assuming the restaurant at position l inside the row is indexed by j_l, and that there are n positions in total inside each row, by the law of total probability, we have

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

where we define $c(i, j_0, k) = 0$. Equation (5) is intuitive: The first term in the summation, $c(i, j_1, k)$, 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, and so on.

Similarly, by law of total expectation and law of total probability, we can obtain the row-level aggregation for the other three outcomes:

$$b(i,k) = \sum_{l=1}^{n} \frac{\mathbf{P}[O(i,j_l,k,z)=1]}{\sum_{l=1}^{n} \mathbf{P}[O(i,j_l,k,z)=1]} b(i,j_l,k),$$

$$r(i,k) = \sum_{l=1}^{n} \frac{\mathbf{P}[O(i,j_l,k,z)=1]}{\sum_{l=1}^{n} \mathbf{P}[O(i,j_l,k,z)=1]} r(i,j_l,k),$$

$$\begin{aligned} e_c(k) &= \sum_{l=1}^{n} e_r(j_l) \mathbf{P}(\text{consumer } i \text{ scrolls to position } l) \\ &= \sum_{l=1}^{n} e_r(j_l) \prod_{l'=1}^{l} (1 - c(i,j_{l'-1},k)) \cdot p_{l'-1,l'}, \end{aligned} \tag{6}$$

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 and orders at position l, as computed in Equation (5). See Appendix A.3 for detailed derivations. It is not hard to see that every row-level outcome is effectively a *weighted average* of the corresponding product-level outcomes within the row, with the weights determined by the consumer's ordering and scrolling behavior at each position.

Remark 3. 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 of restaurants is comparable to those for the single restaurants, so that one is able to rank them together in a mixed and holistic fashion. This is critical for ensuring consistent consumer experience across different levels of aggregations and providing transparency to the restaurant partners. Moreover, the H-module is easily generalizable to higher levels of aggregation as well. For example, if the company designs nested rows (e.g., "rows of rows"), one just need to extend the consumer browsing probabilities to include across-row browsing behavior and apply one more level of aggregation.

4.4. R-Module: Constrained Multi-Objective Optimization for Ranking

4.4.1. Overview. We formulate the multi-objective hierarchical recommendation problem as two sets of ranking problems: one for *within-row ranking* that determines the ordering of restaurants within a row and one for *across-row ranking* that determines the ordering of rows. The R-module is a universal approach that works for both. We introduce a general notation q as the index of a recommendation item. For within-row ranking, q indexes single products (restaurants) within the row. For across-row ranking, q indexes rows. We also leverage subscripts for compact notations: c_{iq} denotes the conversion rate for consumer i on item q, and b_{iq}, r_{iq}, e_{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, a common practice is to resort to a greedy solution, where every item is assigned a ranking score and ranked accordingly (Liu 2009). The greedy solution reduces the complexity from combinatorial ($\mathcal{O}(n!)$) to log-linear ($\mathcal{O}(n\log n)$). We adopt a probabilistic formulation as the ranking score: The output of the R-module is a ranking plan $\boldsymbol{x} = \{x_{iq}\}$, with x_{iq} being the probability of recommending item q to consumer i.

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

- Step 1 **(within-row ranking)**: Each product q within a row is assigned a ranking score x_{iq}. Their within-row positions are determined by the descending order of x_{iq}.
- Step 2 **(across-row ranking)**: Row-level outcomes are computed from the H-module.[13] Then, each row q is assigned a ranking score x_{iq}. The rows and single restaurants (i.e., rows with one restaurant) are then ranked vertically by the descending order of x_{iq}.

We present the final output of the R-module in Proposition 1 and defer the formulation and solution to the next section and Appendix A.4.

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

$$\tilde{x}_{iq} = c_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_e e_{iq}, \quad \forall i, q. \tag{7}$$

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

4.4.2. Formulation and Solution. For any ranking plan $\boldsymbol{x}$, its expected total number of orders, total gross bookings, total consumer retention, and total restaurant exploration can be computed as

$$C(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq} c_{iq}, \qquad B(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq} c_{iq} b_{iq},$$
$$R(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq} c_{iq} r_{iq}, \qquad E(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq} e_{iq}. \tag{8}$$

These are the four objectives of interest, which are aggregations of the individual outcomes across all consumers and all items. Let $C^* = \max_{x\in\mathcal{E}} C(\boldsymbol{x}), B^* = \max_{x\in\mathcal{E}} B(\boldsymbol{x}), R^* = \max_{x\in\mathcal{E}} R(\boldsymbol{x}), E^* = \max_{x\in\mathcal{E}} E(\boldsymbol{x})$ 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*:

$$\max_{x\in\mathcal{E}} \quad C(\boldsymbol{x}) \quad \text{s.t.} \quad B(\boldsymbol{x}) \geq \alpha_b B^*, \ R(\boldsymbol{x}) \geq \alpha_r R^*, \ E(\boldsymbol{x}) \geq \alpha_e E^*, \tag{9}$$

where $0 < \alpha_b, \alpha_r, \alpha_e < 1$ specifies the amount of *tolerable trade-off* for $B(\boldsymbol{x})$, $R(\boldsymbol{x})$, and $E(\boldsymbol{x})$ when optimizing for $C(\boldsymbol{x})$.[14] In Online Appendix A.2, we prove that the Pareto frontier between any two objectives is concave, so that a small sacrifice in one objective can potentially lead to big improvement in others.

Equation (9) 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 serving 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 that leads to analytical solutions[15] for $\boldsymbol{x}$. In Appendix A.4, we provide full mathematical details. By leveraging KKT conditions, the final ranking function is reduced to what is shown in Proposition 1.

Remark 4. Taking a closer look at Equation (7), the ranking function is essentially a *weighted combination* of the multiple objectives.[16] 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. We also do not need to compute B^*, R^*, and E^* with the final ranking score, which further saves computation. For the objective weights λ_b, λ_r, and λ_e, although we can obtain them as functions of α_b, α_r, and α_e by solving a linear system as shown in Appendix A.4, it can be prohibitively expensive due the large scale. In practice, we treat λ_b, λ_r, and λ_e as tuning parameters directly to reduce computation, without the need to specify α_b, α_r, and α_e. In Section 6, we show that λ_b, λ_r, and λ_e can be used as knobs to make principled and quantifiable trade-offs across different objectives.

Remark 5. Compared with Agarwal et al. (2012), who propose the quadratic penalty trick for multi-objective optimization, the R-module solves the multi-objective recommendation problem in a *hierarchical* setting. 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.

5. Model Training and Offline Results

5.1. Training and Optimization Considerations

The world is constantly evolving and so are consumer preferences. As a result, recommender systems are expected to adequately capture the evolving dynamics to be successful in the long term. However, because of 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) as they rely on exploiting correlations in the data to make predictions. In other words, there is no guarantee that the model primitives are invariant.

To tackle this challenge, we implement the following practices in the MOHR framework. First, all ML models are retrained daily using the past 30 days of observation to capture any temporal shifts in the data,[17] and the feature values are updated in real time (Appendix B.2). 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 features to remove position bias (as described in Sections 4.2 and 4.4). This approach can be understood as introducing a "fixed effect" for position to correct for the bias it introduced. It is a widely adopted method in recommender systems to mitigate off-policy training bias without requiring random ranking data or propensity score modeling (Zhao et al. 2019). Third, all ML models are validated with the latest out-of-sample data: With 30-day training data, the initial 29 days are used for training, whereas the last day is reserved for validation. This temporal split of the training and validation datasets mimics the online setting, where models trained using data up until the previous day are deployed to handle the current day's traffic. In Figure B.3 in Online Appendix B.1, we show that the next-day performance for the ML models in the MO-module are stable over time, demonstrating the effectiveness of these ML model updates in capturing the evolving dynamics.

These practices ensures that off-policy bias is removed to the maximal extent and that the ML models are accurately predicting the outcomes of interest in real time. In other words, they help the MOHR framework to efficiently capture the evolving dynamics of the multi-sided marketplace.[18] In addition, the consumer browsing probabilities in the H-module are 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.[19]

5.2. Offline Analysis

5.2.1. Offline Evaluation with Random Ranking Data.

The ranking function in Equation (7) contains three hyperparameters λ_b, λ_r, and λ_e. 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 understanding their effects. Therefore, it is necessary to develop an offline evaluation procedure to pick hyperparameter values to be experimented online.

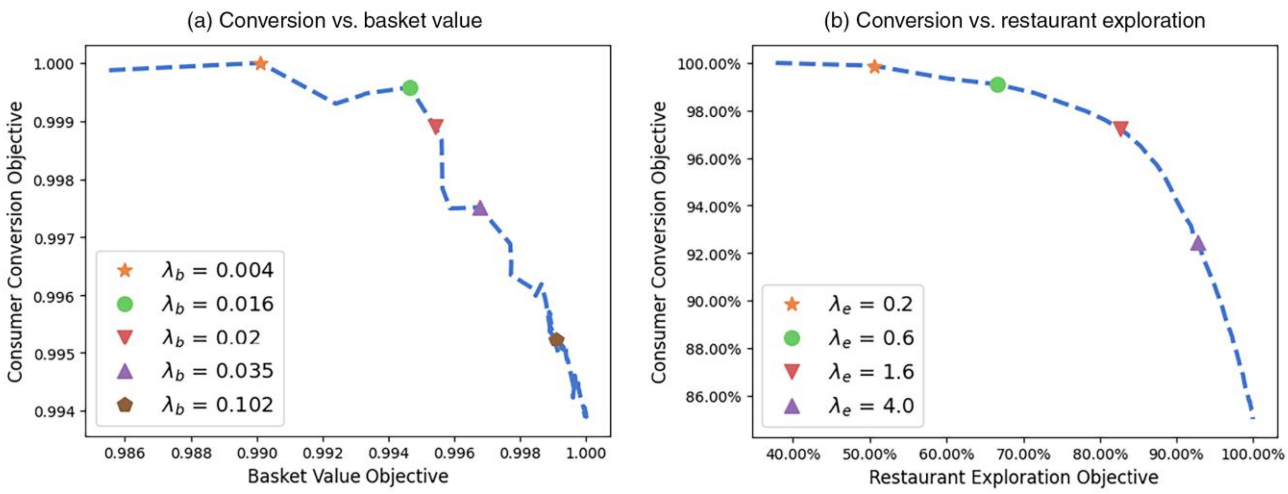
High-quality data are essential for reliable offline evaluation in recommender systems, considering the common challenges of position bias (Ursu 2018) and off-policy evaluation (Strehl et al. 2010, Schnabel et al. 2016). To investigate position bias, the company has allocated a small percentage of random sessions for random ranking, where the restaurants are ranked entirely at random. Figure B.1 in Appendix B.4.1 confirms position bias, showing that top-ranked restaurants get more orders even if the ranking is completely random. Position bias poses a significant challenge for offline evaluation. For example, if the new policy predicts that restaurant j_0 will be ranked at the top for a consumer, but the current system has never placed j_0 in that position for her, it becomes difficult to predict if the consumer would have actually ordered from that restaurant if placed at the top position.

To correct for such off-policy bias in evaluation, we adopt the popular *offline replay* method proposed by Li et al. (2011). It uses random data, has theoretical guarantees for unbiasedness, and has been adopted for offline evaluation in marketing contexts (Aramayo et al. 2023), mostly in multi-armed bandit settings. In Appendix B.4.2, we describe the procedure to adapt their method to the ranking context.

5.2.2. Offline Pareto Frontiers.

Figure 4 shows the Pareto frontiers from the offline replay analysis.[20] Aligned with the theoretical concavity proved in Online Appendix A.2, the offline Pareto frontiers exhibit a concave shape.[21] Notably, there is an almost "flat" region for consumer conversion (top-left region) when the weights assigned to other objectives are small (e.g., $\lambda_b \in [0, 0.004]$, $\lambda_e \in [0, 0.2]$). However, when the weights are relatively large (e.g., $\lambda_b \in [0.016, 0.1]$, $\lambda_e \in [1.6, 4.0]$), the improvements in the basket value objective and restaurant exploration objective come at a much greater expense to the conversion objective. In Online Appendix B.2.1, we present an offline analysis of short-term session-level bookings, calculated by multiplying conversion rates by basket values, as a proxy metric for short-term profits. These offline analyses demonstrate how MOHR serves as a mathematically principled tool for characterizing Pareto frontiers, allowing managers to

Figure 4. Offline Pareto Frontiers



Note. Tick values are presented as percentages relative to the maximum values of the objectives, whereas actual values are omitted due to compliance reasons.

make informed decisions about reasonable trade-offs among the different objectives.

The flat region observed in the offline Pareto frontier indicates the possibility of achieving Pareto improvements. Given that conversion is the most important business metric for the company, for online experiments we choose candidate values for λ_b and λ_e so that the trade-offs lie in the flat region of the Pareto frontiers.[22]

6. Field Experiments

6.1. Experiment Setup and Performance Measures

6.1.1. Experiment Setup. The experiments were conducted over 28 days[23] in June 2019 on 2% of Uber Eats' global consumers. To the best of our knowledge, our proposed framework represents the first multi-objective recommender system that is able to rank hierarchical products within a unified framework. The closest baseline is Uber Eats' own latest production recommender system, which uses disjoint machine learning models to rank rows and individual restaurants separately, with consumer conversion as the single objective. See Appendix C.2 for a detailed description of the baseline.

Compared with the baseline, MOHR introduces two changes: multi-objective optimization (MO-module and R-module) and hierarchical recommendation (H-module). To evaluate them separately, we designed three treatment groups: (1) multi-objective recommender (MOR): only the MO-module and R-module *without* the hierarchical component; (2) hierarchical single-objective recommender (H): only the H-module with conversion as the *single* objective; and (3) multi-objective hierarchical recommender (MOHR): the complete MOHR framework combining (1) and (2).

6.1.2. Performance Measures and Statistical Hypothesis Testing. Table 3 summarize a list of metrics for the online experiments to track the performance of the multiple sides in the marketplace. Let S_i be the number of sessions that consumer i has during the experiment period, and O_{is} and B_{is} be the binary indicator for

Table 3. List of Measurements

Measure	Definition/explanation	Relevant sides
Conversion rate	$\sum_i \sum_s O_{is} / \sum_i S_i$	Consumers, restaurant partners
Basket value per ordered restaurant	$\sum_i \sum_s O_{is} B_{is} / \sum_i \sum_s O_{is}$	Restaurant/delivery partners
Retention rate	$\sum_i \sum_s O_{is} R_{is} / \sum_i \sum_s O_{is}$	Consumers, restaurant partners
Orders per consumer	$\sum_i \sum_s O_{is} / I$	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 4. Live Experiment Results

	MOR				H	MOHR
Metric	Basket value	Retention	Exploration	Combined	Single objective	All four objectives
Conversion rate	—	—	—	—	+1.5%**	+0.5%**
Basket value per ordered restaurant	+0.5%**	—	—	+0.5%**	—	+0.5%**
Retention rate	—	+0.7%***	—	+0.7%***	—	+0.7%***
Orders per consumer	—	+0.8%***	—	+0.8%***	—	+0.9%***
New restaurants impression ratio	—	—	+150%***	+150%***	—	+150%***
New restaurants order ratio	—	—	+100%***	+100%***	—	+108%***
Average vertical order position	—	—	—	—	−5.7%***	−3.2%***
Search rate	—	—	—	—	−0.9%***	−0.8%***

Note. Metrics are reported as relative changes over control.
***$p < 0.01$; **$p < 0.05$.

whether consumer i orders from session s and the basket value, respectively. To measure retention, we let R_{is} be the binary indicator of whether consumer i places another order within the next 14 days following the current session s. For the restaurant exploration outcome, we measure the percentage of overall impressions and orders that are generated from the new restaurants, which are those that joined Uber Eats' platform within 21 days of the experiment's start date.[24]

Sessions from the same consumer are *correlated* as they reflect the behavior of the same person. Therefore, metrics in Table 3 are *not* from independent and identically distributed (i.i.d.) samples. We correct for this intraconsumer correlation when computing the variance for the test statistics through the *delta method* (Oehlert 1992). The resulting p values are larger than when the samples are treated as i.i.d. Therefore, our tests are more rigorous and conservative and less likely to claim the treatment (i.e., MOHR) as effective. See Appendix C.1 for details.

6.2. Field Experiment Results

Now, we present the field experiment (A/B testing) results obtained with the optimal hyperparameter values for λ_b, λ_r, and λ_e. In the next section, we discuss the implications of these objective weights and the actual trade-offs observed.

6.2.1. Multi-Objective Recommender (MOR) Results.

Without the H-module, the MOR framework is not applicable for ranking the rows (vertically) together with single restaurants. Therefore, we keep the production system's ranking for the rows while using the MOR framework for restaurant-level rankings, namely within-row ranking and vertical single restaurant ranking.

Constrained by the number of online experiments we can run on live traffic, we adopt a greedy approach in understanding the effect of incorporating each new objective into the system. This involves sequentially introducing each additive term to the ranking function in Equation (7). The first four columns of Table 4 report the metric changes. With the basket value objective, we observe a 0.5% relative increase in average basket value per ordered restaurant. 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. With the restaurant exploration objective, the number of impressions and orders on the new restaurants more than doubled, increasing by about 150% and 100%, respectively, without hurting the performance of the popular restaurants on the platform.

Understanding the Pareto Improvements. We see that with the optimal weights for each objective, MOR is able to achieve Pareto improvements in other metrics without significantly hurting consumer conversion. We dig deeper into the data and experiments to understand why such Pareto improvements are possible. Initially, one might consider that the sample size is insufficient to detect significant decreases in conversion. However, it is worth noting that the number of consumers in each experiment group is in the order of millions and determined through a sample size calculation to ensure that a relative conversion rate change as small as 0.15% is detectable. In other words, the company is willing to accept a conversion sacrifice that is less than 0.15% and remains undetectable, provided that there are substantial improvements in other desired metrics (as observed in our experiments). Second, the offline Pareto frontiers in Figure 4 indicate that the Pareto frontiers are concave, with an almost flat region when the objective weights are small. As reported in Section 6.3, the optimal objective weights for the basket value, consumer retention, and restaurant exploration are indeed small. Therefore, the realized trade-off is likely situated within the flat region of the Pareto frontier, resulting in Pareto improvements. Finally, in terms of the new restaurant metric improvements, the new restaurants account for only less than 0.3% of the total orders on the platform. Therefore, it is understandable that the mid-level and popular restaurants were not impacted significantly. Moreover, the fact that recommending more new restaurants does not hurt consumer conversion can be attributed to the benefits 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.[25]

The combined impact for MOR, incorporating all objectives, is summarized as "combined" under "MOR" in Table 4. Only the relative changes in metrics are reported due to compliance reasons. Despite the small relative changes in these key metrics (less than 1%), they have resulted in significant business impact given the scale of the platform. Specifically, the MOR framework has led to a weekly revenue increase of $1.3 million.

6.2.2. Hierarchical Single-Objective Recommender (H) Results. We adopt a global estimation procedure for estimating $p_{l,l+1}$ in the H-module. Specifically, for each position l within the row, we compute the ratio of impressions occurring at position l that are subsequently followed by an impression event at position $l+1$. This ratio serves as an estimate for $p_{l,l+1}$:

$$\hat{p}_{l,l+1} = \frac{\text{number of impressions happened at position } l+1}{\text{number of impressions happened at position } l}. \tag{10}$$

Online Appendix C.1 reports the estimated consumer scrolling probabilities. In practice, we find the global estimation works well. In addition to consumer conversion, we monitor two other metrics that provide insights into consumer behavior and the quality of recommendations: *average vertical order position* and *search rate*. The average vertical order position serves as an indicator of the consumer's search effort, whereas the search rate measures the percentage of sessions in which consumers navigate to the search tab. A high search rate indicates that the recommendations on the homepage are not relevant or interesting to the consumers.

From Table 4, the hierarchical single-objective recommender (H) improves conversion rate by 1.5%, which translates to a weekly revenue gain of $1.1 million. Moreover, there is a significant 5.7% reduction in the average vertical order position and a 0.9% decrease in the search rate. These findings indicate that the H-module not only improves the outcome of interest but also reduces search effort for the consumers.[26] In Appendix C.3, we provide two ablation studies to further validate the efficacy of the consumer scrolling probabilities in the H-module.

6.2.3. Full Multi-Objective Hierarchical Recommender (MOHR) Results. With the full MOHR framework, we observe Pareto improvements across all key metrics (the last column in Table 4), which together translates to a weekly revenue gain of $1.5 million. The improvements in conversion rate (+0.5%), average vertical order position (−3.2%), and search rate (−0.8%) are relatively smaller compared with those from the H-module only. This is an expected result of the trade-offs between the additional objectives and the original conversion objective.[27] Consequently, the revenue gain from the full MOHR ($1.5 million weekly) is lower than the combined gains from the MOR ($1.3 million weekly) and H ($1.1 million weekly) treatment groups. However, we emphasize that, compared with the latest production recommender system, MOHR achieves *Pareto improvements* across all key business metrics at no cost to any objectives or any sides in the marketplace.

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

6.3. Insights on the Objective Weights and Trade-Offs

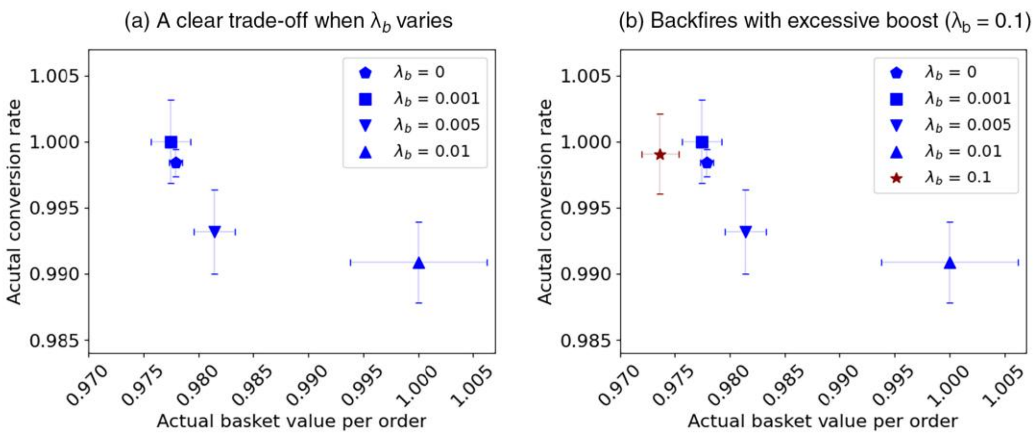
6.3.1. Final Objective Weights and Relative Objective Importance. The first row of Table 5 reports the optimal objective weights used in the final ranking function. The scale of different outcomes are very different, so the absolute magnitude of the weights does *not* reflect the actual relative importance of each of these objectives.[28] To make the weights comparable, we rewrite

Table 5. Final Weights, Standard Deviation, and Mean Calibrated Weights for Each Objective

Statistics	Consumer conversion	Consumer retention	Basket value	Restaurant exploration
Final weights	—	$\lambda_r = 0.6$	$\lambda_b = 0.002$	$\lambda_e = 1.3$
Standard deviation	$\nu_c = 0.0211$	$\nu_r = 0.3721$	$\nu_b = 6.8619$	$\nu_e = 0.0109$
Mean calibrated weights	$avg(\tilde{\lambda}_c) = 0.0211$	$avg(\tilde{\lambda}_r) = 0.0028$	$avg(\tilde{\lambda}_b) = 0.0002$	$avg(\tilde{\lambda}_b) = 0.0142$

Note. The mean values for each objective are also omitted for business compliance reasons.

Figure 5. Trade-offs Between Actual Consumer Conversion and Actual Basket Value as Weight λ_b Varies



Note. To preserve confidentiality, the tick values represent the percentage over the maximum values for each objective.

the ranking function in Equation (7) by normalizing each objective's value by its standard deviation[29]:

$$\tilde{x}_{iq} = \tilde{\lambda}_c \cdot \frac{c_{iq}}{\nu_c} + \tilde{\lambda}_r \cdot \frac{r_{iq}}{\nu_r} + \tilde{\lambda}_b \cdot \frac{b_{iq}}{\nu_b} + \tilde{\lambda}_e \cdot \frac{e_{iq}}{\nu_e}, \quad \forall i, q, \qquad (11)$$

where $\nu_c, \nu_r, \nu_b, \nu_e$ are standard deviations of conversion, retention, basket value, and restaurant exploration outcome, respectively, and $\tilde{\lambda}_c = \nu_c, \tilde{\lambda}_r = \lambda_r c_{iq} \nu_r, \tilde{\lambda}_b = \lambda_b c_{iq} \nu_b, \tilde{\lambda}_e = \lambda_e \nu_e$, are the corresponding *calibrated objective weights*. The last row of Table 5 reports the mean values ($avg(\cdot)$) of these objective weights. The calibrated objective weights $(\tilde{\lambda}_c, \tilde{\lambda}_r, \tilde{\lambda}_b, \tilde{\lambda}_e)$ can be understood as a proxy of the relative importance of each objective in contributing to the long-term outcomes of the platform. The conversion objective dominates the other three objectives (with $avg(\tilde{\lambda}_c)$ being the largest), which is expected as it is the most important business metric for the company. The calibrated weights for the retention and basket value are one to two orders of magnitude smaller than the conversion objective. The restaurant exploration objective is relatively more important than the retention and basket value objective, with the calibrated weight about 67% that of the conversion objective.

6.3.2. Actual Trade-Offs with Different Objective Weights. Although 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[30] with varying λ_b.

There are several observations. First, both the average *predicted* basket value and the *actual* basket value of top recommended restaurant increase with higher values of λ_b (Figure C.2 in Appendix C.4) as expected. This confirms the role of the weight λ_b that controls the influence of the basket value objective on the final ranking results and consumer behaviors. Second, within a reasonable range of λ_b values ($\lambda_b \in [0.001, 0.01]$), there is a clear trade-off between the actual conversion rate and actual basket value[31] (Figure 5(a)), corroborating the findings from the offline analysis. In particular, compared with zero weight on basket value objective (i.e., $\lambda_b = 0$ in Figure 5(a)), a 2.2% relative increase in basket value (corresponding to $\lambda_b = 0.01$) would entail a 0.8% drop in conversion.

Last and most interestingly, when the weight λ_b is huge ($\lambda_b = 0.1$) so that the ranking function in Equation (7) is dominated by the basket value objective,[32] we saw a significant 0.4% drop in actual basket value (compared with $\lambda_b = 0$) (Figure 5(b)) as opposed to a huge lift as we expected from the offline Pareto frontier (Figure 4). The drop in basket value was accompanied by a significant 2.1% increase in search rate, a significant 3.8% increase in vertical order position, and a 2.7% drop in retention of new consumers, whose first interaction with the app is within the experiment period. In addition, the conversion rate remains roughly the same (statistically indistinguishable from when $\lambda_b = 0$ and $\lambda_b = 0.001$). In Online Appendix B.2.2, we also present the results online short-term session-level bookings with varying λ_b's, which also shows a significant decline at $\lambda_b = 0.1$.

Based on these metrics, we think a probable mechanism driving the drop in basket value per ordered restaurant at the extreme weight $\lambda_b = 0.1$ is *consumer trust*. When the basket value objective dominates the final ranking function, the recommended restaurants are mostly expensive ones as they generate higher basket values and may be over the budget for some consumers as a result. This leads to two consequences: Experienced consumers are more likely to scroll deeper to find cheaper restaurants or go to the search tab to order, which explains the increase in search rate and the fact that conversion remains relatively unchanged; 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 (which explains the drop in new consumer retention). In other words, maximizing the basket value per *recommended* restaurant does not necessarily lead to the maximal value of basket value per *ordered* restaurant. Such "backfire" effects are *not* observed in the offline Pareto frontier in Figure 4(a). This can be attributed to a fundamental limitation of offline evaluation when dealing with out-of-distribution data: this backfire effect was not observed offline, because the offline random ranking data used by the offline evaluation is unlikely to cover extreme ranking outcomes such as exclusively recommending expensive restaurants. In Online Appendix C.4, we provide a detailed discussion with additional simulations that demonstrate this point.

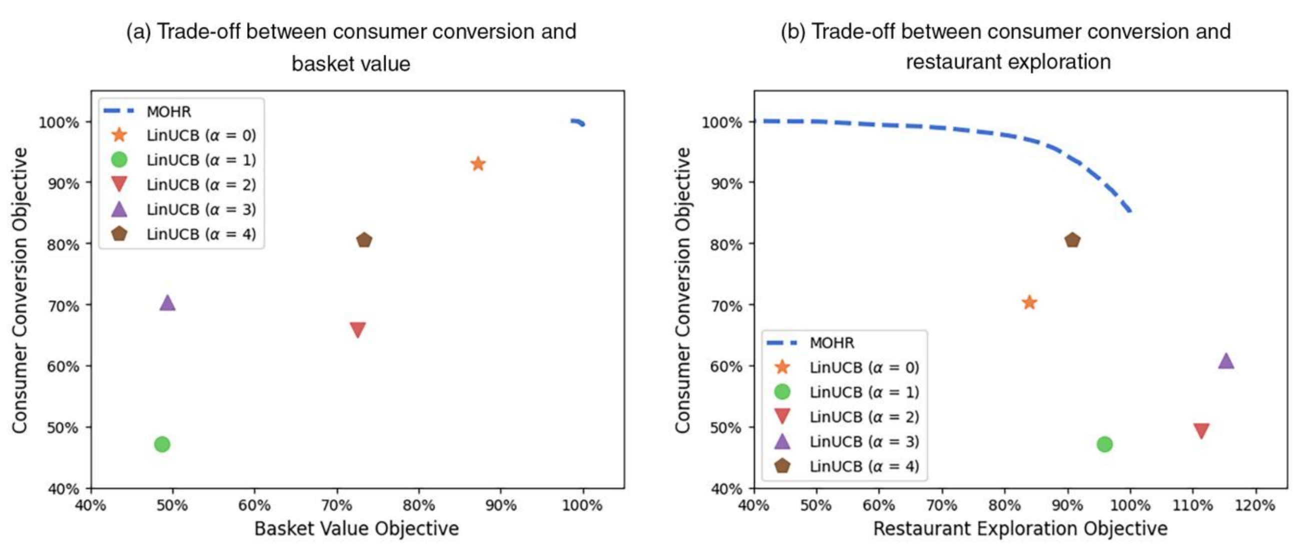
To summarize, there are mainly two takeaways from these observations. First, MOHR provides a mathematically principled tool that acts as a "knob" to help managers make quantifiable and interpretable trade-offs. Second, one should not do arbitrary extrapolation on the Pareto frontiers. Specifically, too much emphasis on one of the objectives may backfire and hurt that objective, in addition to hurting others.

6.4. Additional Baselines and Robustness Checks

6.4.1. LinUCB as an Additional Baseline. We also evaluated MOHR's performance against the most popular algorithm for MAB, LinUCB proposed by Li et al. (2010). LinUCB's design aims to balance exploration and exploitation moderated by a hyperparameter α, by updating a linear reward model and choosing actions with the highest upper confidence bound. Because LinUCB is not able to handle the hierarchical ranking setup, we compare its performance with MOHR under a fixed within-row ranking.

Figure 6 shows that MOHR significantly outperforms LinUCB,[33] where LinUCB uses consumer conversion as the reward. Two key insights emerge from these experiments. First, the underperformance of LinUCB can primarily be attributed to its dependence on a linear reward model, which is restrictive and prone to misrepresentation in complex real-world applications like conversion rate prediction. In contrast, MOHR leverages ML predictions within a mathematically principled framework, offering superior predictive accuracy and thus better Pareto efficiency. Second, unlike MOHR where the objective weights control the trade-offs in a monotone way (Figure 4), there is no monotone relationship between the α values and the trade-off curve. Last, Figure 6(b) sheds lights on the distinction between MOHR's restaurant exploration objective and the uncertainty component of a conventional MAB algorithm. LinUCB can provide aggressive exploration (e.g., at $\alpha = 2$ and $\alpha = 3$) but at a cost of a 40% drop in conversion, which is a trade-off not acceptable to most platforms. In

Figure 6. Comparison of MOHR Against LinUCB with Various α Values, Where LinUCB Uses Consumer Conversion as the Single Objective



Note. Tick values are presented as percentages relative to the maximum values of the objectives under MOHR, whereas actual values are omitted due to business compliance reasons.

the regions where the trade-offs are practical to the company (top left region of the Pareto frontiers), MOHR strictly dominates the LinUCB approaches.

In Online Appendix C.2, we present a detailed discussion on the comparison between MOHR and LinUCB, and additional results on alternative reward functions for LinUCB.

6.4.2. Additional Baselines for Hierarchical Ranking. Because of the challenge of calibrating between rows of products and individual items, existing solutions for hierarchical product ranking on online platforms predominantly rely on heuristic or proxy-based methods. In particular, there are three main approaches: heuristic-based positioning that fix the vertical position of the rows in the homepage ("heuristic-based"), disjoint systems that consist of separate ML models for ranking rows and single products, with an overarching model for determining how many rows to display ("disjoint systems"), and dimension reduction to simplify ranking by considering only the first item in each row ("dimension-reduction").

Disjoint systems were previously shown to outperform heuristic-based approaches by the company, leading the company to adopt them as their production recommender system, as described in Appendix C.2. In the online experiments, we showed that MOHR outperformed the disjoint systems, leading us to believe that it also outperforms the heuristic-based approaches. The ablation study for the H-module in Section 6.2.2 showed that MOHR also outperforms dimension-reduction approaches. In summary, our work establishes that MOHR achieves superior performances to all three recognized approaches for hierarchical ranking. See Online Appendix C.3 for a complete discussion.

6.4.3. Robustness Checks. In Online Appendix C.6, we provide robustness checks including randomization checks and the elimination of novelty effects.

7. Discussion

7.1. Methodological Contributions

This paper proposes MOHR, a novel recommendation framework that addresses two of the most prominent challenges in multi-sided platforms: multi-sided trade-off and hierarchical homepage. Compared with the existing literature on multi-objective recommendation (Agarwal et al. 2012), the methodological contribution of MOHR is solving the multi-objective ranking problem in a *hierarchical* setting, through an innovative formulation of probabilistic consumer behavior modeling and constrained optimization. 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, and propose a probabilistic consumer behavior model to connect the two problems which are then solved through constrained optimization. To the best of our knowledge, we are the first to provide a holistic recommendation solution for hierarchical homepages and with multiple objectives, which are a common setup in many online platforms across different industries today. Empirically, we demonstrate that our proposed MOHR framework works at scale. It has been deployed at Uber Eats, one of the world's largest food delivery platforms serving millions of consumers everyday.

7.2. Managerial Implications

We demonstrate that multi-objective optimization is effective at improving metrics tied to long-term profitability of the platforms. Our proposed MOHR framework provides a convenient tool that acts as a knob for the managers to make mathematically principled trade-offs among conflicting objectives. We also show that if the weight for a particular objective is too large, it will hurt consumer experience and backfire, suggesting the need for a delicate balance across the multiple objectives to achieve long-term profitability.

In the context of food delivery platforms, we show that consumer conversion, consumer retention, basket value and restaurant exploration are four objectives that are tied to the long-term outcomes of the three-sided platform. We obtain quantifiable trade-offs within the MOHR framework. We also discuss the relative importance of each objective through their final weights, where the normalized weights can be understood as our proxy of the relative importance of each objective toward improving the long-term profitability of the platform.

7.3. Generalizability

MOHR is general, flexible, and readily applicable to other platforms within and outside the food delivery industry. YouTube and Netflix as a video streaming platform, and Airbnb as a peer home-sharing platform, are all operating in multi-sided marketplaces with a hierarchically displayed homepage. In addition, these platforms are usually concerned with multiple objectives. For example, content recommendation platforms such as YouTube and Netflix would care about the short-term engagement and long-term satisfaction of the consumers, content creators and advertisers; C2C platforms such as Ebay, Craigslist, and Airbnb would care about the retention of both the consumers and the sellers or hosts. MOHR is readily applicable to these platforms. MOHR is also applicable to cases where multiple consecutive consumptions are expected, which is a common scenario for content sharing platforms such as YouTube. In these cases, every consumption is viewed as a separate session, and MOHR naturally applies.[34]

Components of MOHR can be applied in a modularized fashion as demonstrated in Section 6.2. For example, if a platform is concerned with multi objectives but the recommendation contents are not hierarchical, it can adopt MOHR without the H-module (i.e., MOR); if a platform is only concerned with hierarchical recommendation but does not need to optimize for more than one objective, it can adopt only the H-module (i.e., H). The holistic framework also reduces the burden of maintaining separate machine learning systems for recommending products of different levels of aggregations.

Last, we emphasize that our proposed framework is *not* limited to the multi-sided settings. Firms generally care about multiple objectives such as short-term consumer engagement, long-term consumer retention, and satisfaction. Therefore, the multi-objective criterion is justified even *without* the multi-sided setup of the platform. MOHR is readily applicable to these general settings as well.

7.4. Challenges, Limitations, and Future Research

A limitation of the MOHR framework is its scalability with large numbers of objectives. With an increasing number of objectives, it becomes infeasible to tune the weights in an A/B testing framework with a combinatorial number of candidate weights. 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.

The probabilistic consumer behavior model has two limitations. First, the model is a global static estimate based on a snapshot of behavior logs. It is not personalized and could become outdated after the model is deployed globally. A future research direction is to build a personalized and real-time consumer behavior model, which takes as input the consumer's history, in-session behavior and contextual features. The whole MOHR framework still holds in this case, but with $p_{l,l+1}$ in Equation (4) plugged in as the output from a personalized real-time ML model instead. Second, the consumer behavior model assumes a linear browsing pattern (i.e., 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.

Last, the objectives in the MOHR framework are estimated by separate ML models. However, different outcomes may be related to each other, 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. Multitask deep learning models (Ruder 2017) are well suited in this case to jointly and efficiently predict multiple outcomes. The multiple ML models in the MO-module can be replaced with a single multitask deep learning model, with the other components of MOHR unchanged.

Acknowledgments

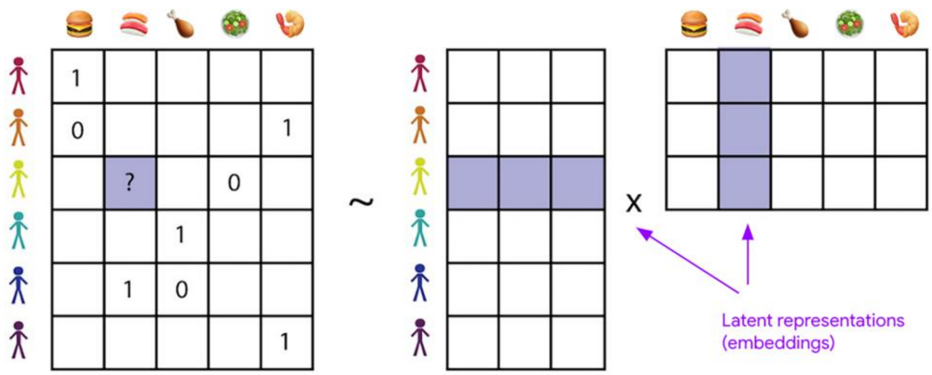
The authors thank the senior editor, associate editor, and two anonymous reviewers for thoughtful suggestions throughout the review process; Alex Burnap, Xinyu Cao, Wesley R. Hartmann, Tongil TI Kim, Beibei Li, Pan Li, Song Lin, Nikhil Malik, Emaad Manzoor, Sridhar Narayanan, Ram Rao, Navdeep S. Sahni, Robbie Sanders, Shunyuan Zhang, and Yingjie Zhang, as well as seminar participants at Cornell University, UT Dallas, Hong Kong University of Science and Technology, the Chinese University of Hong Kong, Northwestern-Kellogg, Stanford GSB, Wharton, NYU Stern, UC Riverside, Temple University, USC Marshall, UChicago Booth, Peking University, University of Florida, University of Arizona, Berkeley Haas, Columbia Business School, UCLA Anderson, and the 2022 Marketing Science Conference, the 2022 INFORMS Annual Meeting, the Virtual Quant Marketing Seminar Series, the 2022 Conference on Information Systems and Technology (CIST), the 16th ACM Conference on Recommender Systems (Recsys 2022), the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2023), and the 2023 Summer Institute in Competitive Strategy (SICS) Conference for helpful suggestions on improving the paper; and Xiaoting Yin, Issac Liu, Ambika Krishnamachar, Yuanchi Ning, and Chen Peng at Uber Technologies for their engineering, product, and leadership support. All authors were employed and financially supported by Uber Technologies, Inc. (Uber) when the work was done. Uber recognizes a potential need to disclose certain confidential information and to protect such information from unauthorized use and disclosure. Uber had the right to remove its intellectual property or trade secrets subject to the following stipulations: (1) removing Uber confidential information from the paper, including data, code, and statistics such as conversion rate, number of impressions, clicks, orders per row, and so on and (2) compliance with Uber's obligations as it relates to applicable laws, including the Data Protection Law, security laws, confidentiality requirements, and contractual commitments.

Appendix A. Additional Technical Details for MOHR

A.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

Figure A.1. Matrix Factorization for Collaborative Filtering



two lower-dimensional matrices: The first one has a row for each consumer, whereas the second has a column for each item. The row or column associated to a specific consumer or item is referred to as latent factors. Here an item refers to either a restaurant or a row. Figure A.1 illustrates the idea of matrix factorization for collaborative filtering.

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

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

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 (zero if the consumer never ordered from the restaurant), and 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.

Equation (A.1) 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 (A.1) but changed $\boldsymbol{v}_j$ to source representation $\tilde{\boldsymbol{w}}_k$ and r_{ij} to r_{ik}, the order counts between consumer i and source k:

$$\{\tilde{\boldsymbol{u}}_i\}_{i=1}^{I}, \{\tilde{\boldsymbol{w}}_k\}_{k=1}^{K} = \underset{\{\tilde{u}_i\}_{i=1}^{I}, \{\tilde{w}_k\}_{k=1}^{K}}{\arg\min} \sum_{i,k \in \tilde{S}} (\tilde{\boldsymbol{u}}_i^T \tilde{\boldsymbol{w}}_k - r_{ik})^2 + \lambda_u \sum_{i=1}^{I} \|\tilde{\boldsymbol{u}}_i\|_2 + \lambda_w \sum_{k=1}^{K} \|\tilde{\boldsymbol{w}}_k\|_2, \tag{A.2}$$

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

A.2. Bayesian Modeling for the Restaurant Exploration Outcome

We now describe the Bayesian modeling procedure to estimate $\sigma(j)$, the posterior variance for $c(i,j,k)$ as the value for the restaurant exploration outcome. 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 A.1 states the posterior for $c(i,j,k)$.

Proposition A.1. *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 $\sigma(j)^2 = (\alpha_j+N_j^1)(\beta_j+N_j-N_j^1)/(\alpha_j+\beta_j+N_j)^2(\alpha_j+\beta_j+N_j+1)$.*

Proof of Proposition A.1. 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, $O_{j1},\ldots,O_{jN_j}$ are random variables represents the corresponding conversion events where $O_{jm}=1$ means the mth impression on restaurant j leads to an order and zero otherwise. In other words, $O_{j1},\ldots,O_{jN_j} \overset{i.i.d.}{\sim} \text{Bernoulli}(c_j)$. The conjugate prior for Bernoulli distribution is the Beta distribution $Beta(\alpha_j,\beta_j)$.

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

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

which concludes the proof. □

A.2.1. Choice of Prior Parameters for the Restaurant Exploration Outcome. 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 reduce discrimination among 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 Equation (A.3) 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 $\mathcal{B}(\alpha, \beta)$, which equals $\alpha/(\alpha+\beta)$, should be close to 2%. The second condition implies that α and β should be large enough to guard the posterior against noisy data, whereas the third condition implies that α and β should be small enough so that the numerator and denominator in Equation (A.3) are not dominated by them. Given these considerations, we set $\alpha = 2$ and $\beta = 98$ and find them to work well empirically.

A.3. Derivation of Row-Level Outcomes in the H-Module

A.3.1. Row-Level Basket Value Outcome. By law of total expectation, the expected basket value of a row can be decomposed as the sum of basket values at each position l inside it:

$$\begin{aligned}
&\mathbf{E}[B(i,k,z)] \\
&= \sum_{l=1}^{n} \mathbf{E}[B(i,j_l,k,z) | O(i,j_l,k,z) = 1] P[O(i,j_l,k,z) = 1] \\
&= \sum_{l=1}^{n} b(i,j_l,k) P[O(i,j_l,k,z) = 1],
\end{aligned} \tag{A.4}$$

which is a weighted combination of the basket value outcome of each individual 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

$$\begin{aligned}
&b(i,k) \\
&= \mathbf{E}[B(i,k,z) | O(i,k,z) = 1] = \mathbf{E}[B(i,k,z)] / \mathbf{P}[O(i,k,z) = 1] \\
&= \sum_{l=1}^{n} \frac{P[O(i,j_l,k,z) = 1]}{\sum_{l=1}^{n} P[O(i,j_l,k,z) = 1]} b(i,j_l,k),
\end{aligned} \tag{A.5}$$

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 j_l, as computed in Equation (5). Thus, the row-level basket value is effectively a *weighted average* of the expected basket values of individual restaurants within the row, with the weights proportional to their predicted conversion at each position while accounting for the consumer's scrolling behavior.

A.3.2. Row-Level Consumer Retention Outcome. Following the same derivation above, the row-level consumer retention outcome can be computed as

$$r(i,k) = \sum_{l=1}^{n} \frac{P[O(i,j_l,k,z) = 1]}{\sum_{l=1}^{n} P[O(i,j_l,k,z) = 1]} r(i,j_l,k). \tag{A.6}$$

A.3.3. Row-Level Restaurant Exploration Outcome. The row-level restaurant exploration outcome $e_c(k)$ is slightly different as it is not conditioned on the consumer placing an order. By the same law of total expectation as in Equation (5), we have

$$\begin{aligned}
e_c(k) &= \sum_{l=1}^{n} e_r(j_l) \mathbf{P}(\text{consumer } i \text{ scrolls to position } l) \\
&= \sum_{l=1}^{n} e_r(j_l) \prod_{l'=1}^{l} (1 - c(i,j_{l'-1},k)) \cdot p_{l'-1,l'},
\end{aligned} \tag{A.7}$$

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

A.4. Formulation and Solution for the Constrained Optimization Problem in the R-Module

We adopt the trick in Agarwal et al. (2012) and add a quadratic penalty term to the objective function in Equation (9) 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 $\boldsymbol{u} = \left\{u_{iq} = \frac{1}{Q}, \forall i, q\right\}$ that assigns equal probability to all items for all consumers[35]:

$$\begin{aligned}
&\max_{x \in \mathcal{E}} C(\boldsymbol{x}) - \frac{\gamma}{2} \|\boldsymbol{x} - \boldsymbol{u}\|_F^2 \\
&\text{s.t. } B(\boldsymbol{x}) \geq \alpha_b B^*, R(\boldsymbol{x}) \geq \alpha_r R^*, E(\boldsymbol{x}) \geq \alpha_e E^*,
\end{aligned} \tag{A.8}$$

Proposition A.2 provides the solutions to Equation (A.8). Proposition A.3 provides guidance on serving the solution for online systems.

Proposition A.2. *The solution to Equation* (A.8) *is*

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_e e_{iq} - \mu_i) + \frac{1}{Q}, \tag{A.9}$$

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

Proof of Proposition A.2. First, we write out the element-wise form of Equation (A.8):

$$\begin{aligned}
&\max_{\{x_{iq}\} \in \mathcal{E}} \sum_{i,q} \left(x_{iq} c_{iq} - \frac{\gamma}{2} \left(x_{iq} - \frac{1}{Q} \right)^2 \right) \\
&\text{s.t. } \sum_{i,q} x_{iq} c_{iq} b_{iq} \geq \alpha_b B^*, \\
&\qquad \sum_{i,q} x_{iq} c_{iq} r_{iq} \geq \alpha_r R^*, \\
&\qquad \sum_{i,q} x_{iq} e_{iq} \geq \alpha_e E^*, \\
&\qquad x_{iq} \geq 0, \ i = 1, \dots, I, q = 1, \dots, Q, \\
&\qquad \sum_{q} x_{iq} = 1, \ i = 1, \dots, I,
\end{aligned} \tag{A.10}$$

where c_{iq}, r_{iq}, b_{iq}, and e_{iq} are the values for the consumer conversion outcome, consumer retention outcome, basket value outcome and restaurant exploration outcome between consumer i

and item q, respectively. The objective for the maximization problem in Equation (A.10) is concave, and the inequality is all affine functions. Therefore, the KKT conditions are *necessary and sufficient* conditions for optimality. We use them to solve Equation (A.10).

Let $\lambda_b, \lambda_r, \lambda_e, \delta_{iq}$, and μ_i be the nonnegative slack variables for the five sets of constraints in Equation (A.10), which are used to define the Lagrangian:

$$\begin{aligned} &L(\{x_{iq}\}, \lambda_b, \lambda_r, \lambda_e, \{\delta_{iq}\}, \{\mu_i\}) \\ &= \sum_{i,q}\left(x_{iq}c_{iq} - \frac{\gamma}{2}\left(x_{iq} - \frac{1}{Q}\right)^2\right) - \lambda_b\left(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*\right) \\ &\quad - \lambda_r\left(\sum_{i,q} x_{iq}c_{iq}r_{iq} - \alpha_r R^*\right) - \lambda_e\left(\sum_{i,q} x_{iq}e_{iq} - \alpha_e E^*\right) \\ &\quad - \delta_{iq}x_{iq} + \mu_i\left(\sum_q x_{iq} - 1\right). \end{aligned} \tag{A.11}$$

By *stationarity* from the KKT conditions, we have

$$-c_{iq} + \gamma\left(x_{iq} - \frac{1}{Q}\right) - \lambda_b c_{iq}b_{iq} - \lambda_r c_{iq}r_{iq} - \lambda_e e_{iq} - \delta_{iq} + \mu_i = 0, \tag{A.12}$$

which yields

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_e e_{iq} + \delta_{iq} - \mu_i) + \frac{1}{Q}. \tag{A.13}$$

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

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_e e_{iq} - \mu_i) + \frac{1}{Q} \tag{A.14}$$

for any $x_{iq} > 0$. □

Proposition A.3. *Ranking according to x_{iq} in Equation (A.9) is equivalent to ranking according to*

$$\tilde{x}_{iq} = c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_e e_{iq}. \tag{A.15}$$

Proof of Proposition A.3. When serving the ranking plan 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 affect the final ranking results. □

We now show that λ_b, λ_r, and λ_e can be solved as functions of α_b, α_r and α_e in addition to the other inputs. By *primal feasibility* from the KKT conditions, we have $\sum_q x_{iq} = 1$, $\forall i$. Plugging in Equation (A.13) and solve for μ_i, we have

$$\mu_i = \frac{1}{Q}\sum_q(c_i q + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_e e_{iq} + \delta_{iq}),\ i = 1, \ldots, I, \tag{A.16}$$

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

By complementary slackness from the KKT conditions, we have

$$\begin{aligned} \lambda_b\left(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*\right) &= 0, \\ \lambda_r\left(\sum_{i,q} x_{iq}c_{iq}r_{iq} - \alpha_r R^*\right) &= 0, \\ \lambda_e\left(\sum_{i,q} x_{iq}e_{iq} - \alpha_e e^*\right) &= 0. \end{aligned} \tag{A.17}$$

The first equation in Equation (A.17) 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 $\mu_i (i = 1, \ldots, I)$, λ_b, λ_r, and λ_e after plugging in Equation (A.13). Similar observations hold for the other two equations in Equation (A.17). Therefore, combining Equations (A.16) and (A.17), 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 that, instead of solving λ_b, λ_r, and λ_e as a function of α_b, α_r, and α_e, which are treated as tuning parameters, we treat λ_b, λ_r and λ_e as tuning parameters directly to reduce computation. In addition, λ_b, λ_r, and λ_e can also be viewed as the weights controlling the relative importance of the different objectives.

Appendix B. Additional Details on Offline Evaluation

B.1. Model Performance and Important Features in the MO-Module

Table B.1 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).

B.2. Feature Updates Within the MOHR Framework

The consumers are expected to see a different homepage layout and different recommendations with every homepage refresh. This is realized by feature updates from the MO-module. With every homepage refresh, a new session is generated together with the updated features. Specifically, the output of the MO-module, that is, the ML predictions for the product-level outcomes, 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 Equation (7) without the need to solve huge-scale linear programming problems with every page refresh.

B.3. Comparison Against MAB Algorithms for the Restaurant Exploration Outcome

In this section, we provide theoretical analysis to show that the benefit of the restaurant exploration outcome is beyond

Table B.1. Top 10 Important Features for the ML Models in the MO-Module, Measured by the Feature Importance Score of the Gradient Boosted Trees

Model name	Model performance	Top 10 important features
Consumer conversion	Test AUC = 0.8562	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

what a standard multiarmed bandit (MAB) procedure can provide. The restaurant exploration outcome is defined as the uncertainty estimate of the conversion rate motivated by an upper confidence bound (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 (popular) 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 the point estimate $\mu_o > \mu_n$ and the standard deviation $\sigma_o < \sigma_n$. In other words, restaurant 1 has a higher estimated conversion rate and lower uncertainty estimate than restaurant 2. Restaurant 2 has a higher value of restaurant exploration 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) restaurant exploration objective (UCB formulation), (2) Thompson sampling, and (3) epsilon-greedy strategy.

1. Restaurant exploration objective: With a UCB formulation as presented in Section 4.2.2, the ranking score r_o, r_n for the two restaurants are

$$r_o = \mu_o + \kappa\sigma_o,$$
$$r_n = \mu_n + \kappa\sigma_n. \tag{B.1}$$

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. In other words, *every* new restaurant on the platform is guaranteed to receive more exposure thanks to the restaurant exploration objective.

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:

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

where the last inequality is because $(\mu_o - \mu_n)/\sqrt{\sigma_o^2 + \sigma_n^2} > 0$ and $\int_0^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = 0.5$. Therefore under Thompson sampling, the new restaurant has less than half of the chance of being recommended.

Table B.2. Chance of Recommending a New Restaurant Under UCB, Thompson Sampling, and Epsilon-Greedy Strategy (with $\epsilon = 0.1$)

Strategy	Control	Thompson sampling	Epsilon-greedy	Restaurant exploration (ours)
$\mathbb{P}$(Recommend new restaurant)	0%	less than 50%	5%	100% given large enough κ

Note. "Control" uses only the estimated conversion for ranking with no treatment on boosting new restaurants.

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." 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 B.2 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 due to the fact that Thompson sampling and epsilon-greedy strategies involve stochasticity (e.g., sampling from distribution, or random draws), and there is no guarantee that new restaurants will always receive a boost under these stochastic sampling procedures. Instead, the UCB formulation (as adopted by our restaurant exploration outcome) consistently assigns higher scores to all new restaurants.

Why is a consistent boost in new restaurant visibility preferable to stochastic boosts? The preference arises from the sensitivity of new restaurant partner retention to their performance during the initial days on the platform. When adopting stochastic boosting methods such as Thompson sampling or epsilon-greedy, there exists a nonnegligible probability that a new restaurant may not receive sufficient exposure in their initial days, potentially increasing their likelihood of churning. In contrast, our UCB-based approach provides a deterministic boost, guaranteeing every restaurant higher exposure levels. This guarantee plays a crucial role in optimizing their performance during the critical early period after onboarding. This is the reason for our decision to adopt a UCB-like formulation for the restaurant exploration outcome.

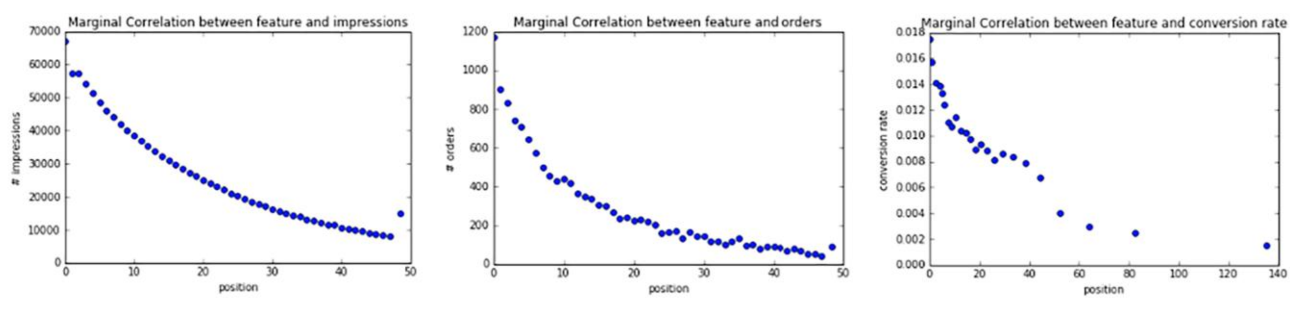
B.4. Position Bias and Offline Replay Method for Off-Policy Evaluation of MOHR

B.4.1. Position Bias. Figure B.1 shows the average number of impressions, average number of orders and conversion rate at each position from the random ranking data. We can see a clear decreasing trend in all three metrics as position increases (i.e. further down in the feed). This suggests that the same restaurant at different positions may appeal very differently to the consumers.

B.4.2. Offline Replay Method for Off-Policy Evaluation The offline replay method, proposed by Li et al. (2011), is a popular alternative to inverse-propensity weighting based methods (Simester et al. 2020a, Yoganarasimhan et al. 2023, Hitsch et al. 2024, Yang et al. 2024) for off-policy evaluation. To evaluate the performance of MOHR using offline data, we adapt their method for the ranking scenario. Specifically, if it happens that the new policy (i.e., MOHR) chooses the same restaurant to be ranked on the top position as in the random ranking data, then that event is retained and will be used for estimating the performance of the new policy. In other words, the replay method is essentially looking for events in the random ranking data that can serve as "re-playing" the ranking under the new policy to be evaluated. The replay method is proven to provide unbiased offline evaluation (Li et al. 2011) without the need to run online experiments.

As a side note, the random ranking data provided by the company is restaurant-level random ranking, and we unfortunately do not have row-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.

Figure B.1. Impressions (Left), Orders (Middle), and Conversion (Right) vs. Position on Random Ranking Data



Appendix C. Additional Details on Online Experiments

C.1. Variance Correction Using Delta Method

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

$$\overline{O} = \frac{1}{I}\sum_i \sum_s O_{is}, \ \ \overline{S} = \frac{1}{I}\sum_i S_i \tag{C.1}$$

be the average number of orders O and average number of sessions S *per consumer*. Therefore, the conversion rate metric $C = \overline{O}/\overline{S}$ 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

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

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, respectively. By multivariate delta method, we have the conversion rate

$$C = \overline{O}/\overline{S} \sim N(\mu_O/\mu_S, \sigma_C^2), \tag{C.3}$$

where

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

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. Therefore, the variance correction in Equation (C.4) yields a p value that is larger than if treating all sessions as i.i.d., making the hypothesis testing more rigorous and conservative, that is, less likely to claim the treatment (MOHR) as effective.

C.2. Latest Production Recommender System at the Company

The latest production recommender system at Uber Eats is a framework using three disjoint machine learning (ML) models to rank rows 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, that is, 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 row (ML Model A); (2) a (*consumer*, *row*)-level model predicting the conversion objective on row level, i.e. the probability that the consumer will order from *any* restaurant inside the row in the current session, which is used to determine the ranking among the rows (ML Model B); (3) a (*consumer*, *number of rows*)-level model predicting the conversion rate under different number of rows recommended, which is used to determine how many rows to display in the current session (ML Model C). Figure C.1 shows an overview of the production recommender system.

All of the models are real-time personalized ML models, using the state-of-art hybrid recommender systems based on gradient boosting decision trees with the features and hyperparameters same as those in Online Appendix A.1. 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 rows and single restaurants, all of the rows are ranked above all of the single restaurants in the production recommender system.

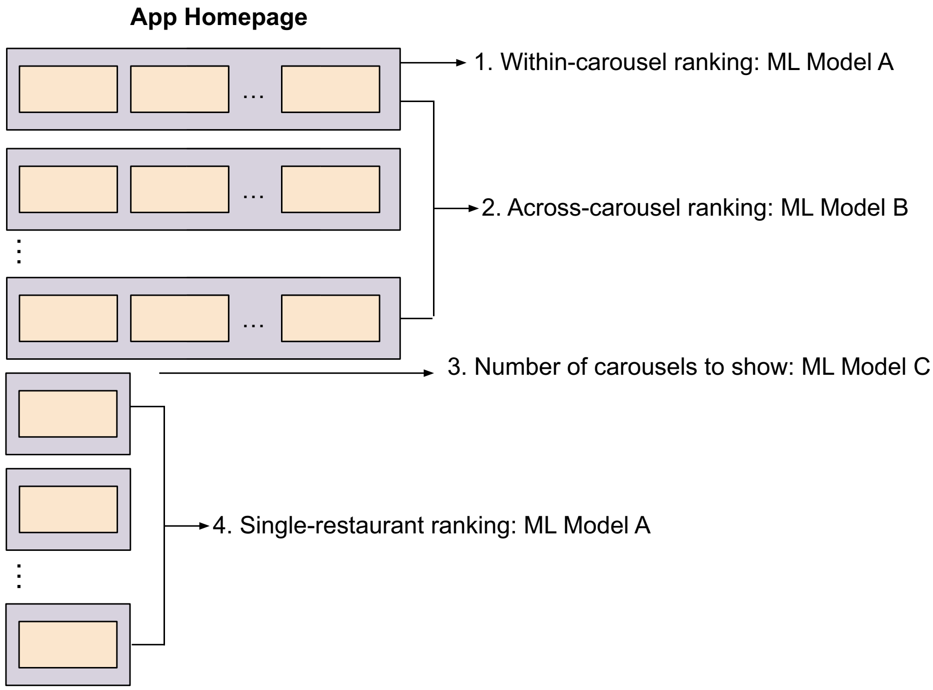
C.3. Ablation Study on the Probabilistic Consumer Behavior Model

To further investigate the impact of the consumer scrolling probabilities from the consumer behavior model in the H-module, we conducted two ablation experiments within the H-module.

C.3.1. Using Only the First Restaurant in Each Row in Computing Row-Level Outcomes. We set $p_{0,1} = 1$ and $p_{l,l+1} = 0$ for $l > 0$, instead of using $\hat{p}_{l,l+1}$ in Equation (10) as the consumer scrolling factors. Comparing this ablated version of MOHR with the production recommender system as the baseline, we observed a 5.5% decrease in the average order position. This finding reaffirms the benefits of intelligently recommending rows and single restaurants together. However, no significant changes were observed in other business metrics. In comparison with the results presented in Table 4, where the H-module increased the conversion rate and decreased the search rate, this outcome suggests that the proposed consumer scrolling behavior modeling is crucial for the improvements attained through the H-module.

C.3.2. A Weighted Sum Learned with a Linear Regression Model. In the H-module, the row-level outcome is essentially a weighted sum of individual outcomes within

Figure C.1. Overview of the Latest Production Recommender System at the Company



that row, where the weights for each restaurant are functions of the scrolling probabilities $p_{l,l+1}$'s. Alternatively, the weights can be learned from a linear regression model by predicting the row-level conversion rate from the conversation of individual restaurants inside the row:

$$c = w_0 + \sum_{l=1}^{7} w_l c_l + \epsilon, \tag{C.5}$$

where c is the conversion label of a row, c_l is the predicted conversion rate for the restaurant at the l-th position inside the row, w_l is the weight for the l-th position, and w_0 is the intercept term. The H-step is only applied to the top seven positions within each row. This is because at the time of the development of MOHR, there were at most seven restaurants presented in every row. To see more restaurants within the row, there is a "see all" button at the top right corner of every row.

We experimented with this setting as an additional baseline for the H-module. Table C.1 reports the fitted values for the weights. When these are compared with the values of $p_{l,l+1}$, as reported in Table 2 in Online Appendix C.1, two significant concerns regarding the learned weights emerge. First, unlike the expected monotonic relationship observed for $p_{l,l+1}$, the learned weights do not exhibit a decreasing pattern across positions. This implies that restaurants displayed later in a row disproportionately influence the "appeal" of a carousel, contradicting intuition which would suggest those placed earlier should have more impact. Second, certain weights, such as w_5 and w_6, are negative. This is problematic as it may inadvertently prompt the recommendation algorithm to favor restaurants with *lower* conversion rates in order to maximize the row-level conversion rate, potentially leading to low-quality restaurants being recommended.

In the live A/B experiment, we replace Equation (5) (i.e., computing row-level conversion) with the following equation:

$$c(i,k) = w_0 + \sum_{l=1}^{7} w_l c(i, j_l, k) \tag{C.6}$$

while keeping everything else unchanged. Compared with this baseline, MOHR improved homepage conversion rate by 0.2% and reduced average vertical order position by 0.5%, all statistically significant. This validates the superiority of the aggregation formula based on consumer scrolling behaviors (Equation (5)) in computing row-level outcomes.

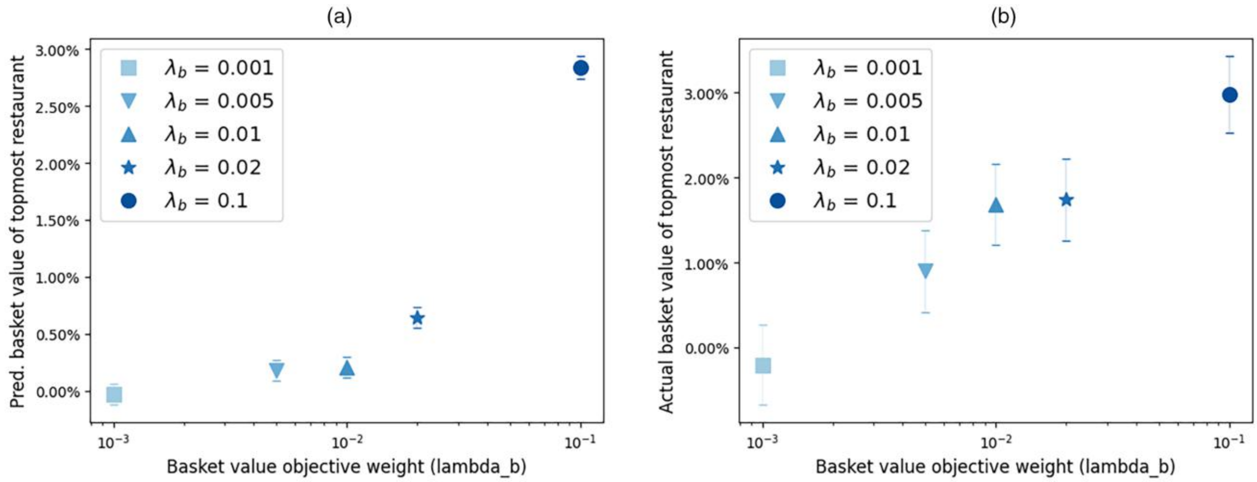
C.4. Additional Experiment Results on the Basket Value Objective

Figure C.2 displays the average predicted and actual basket values for the highest-ranked single restaurant, with different weights assigned to the basket value objective. It is worth noting that the highest-ranked single restaurant may or may not be positioned at the top in the feed, as the MOHR framework

Table C.1. Estimated Weights for Each Position from the Linear Model

Coefficient	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7
Value	0.0015	0.2364	0.3213	0.1959	0.1317	−0.0044	−0.047	0.3123

Figure C.2. Additional Experiment Results on the Basket Value Objective



Notes. (a) Average predicted basket value of the highest-ranked single restaurant in the home feed. (b) Average actual basket value of the highest-ranked single restaurant in the home feed. The y axis represents the percentage change with respect to $\lambda_b = 0$.

combines rows and single restaurants in a mixed ranking. However, we focus on this single position so that the basket value objective is straightforward to measure.

We notice that larger values of λ_b lead to an increase in the average *predicted* basket value of the topmost single restaurant, as expected (Figure C.2(a)). Consequently, the *actual* basket value from the topmost restaurant also rises with increasing objective weight λ_b (Figure C.2(b)). This confirms the efficacy of the basket value objective and its weight λ_b as the tuning parameter, which controls the relative importance of the objective.

Endnotes

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

[2] By definition, 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.

[3] In other words, the row candidates and restaurant candidates within each row are *external* to our framework. This is a common setup (i.e., fixed candidate set) for industrial recommender systems.

[4] 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.

[5] The time window is 14 days in our experiment, which aligns with the retention metric by the company.

[6] 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.

[7] This is also a counterfactual prediction similar to the consumer retention outcome.

[8] 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.

[9] If the restaurant appears as a single recommendation item, we say it belongs to a single-restaurant row.

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

[11] We do not explicitly model the vertical (across-row) browsing probabilities in MOHR as they do not affect the final ranking. The reason is that consumer browsing behaviors on the low level (e.g., within row) are only needed for ranking on a high level (e.g., across row).

[12] The position index starts at one and $p_{0,1} = 1$, meaning that consumers always browse the first product in each row. This is empirically guaranteed to be true by the design of the app homepage.

[13] The computation of row-level outcomes depends on the within-row ranking from Step 1, as derived in the H-module. In addition, the restaurant position within the row, which is determined by the within-row ranking, is used as an input feature in the ML models in the MO-module. That's why there is an arrow from the R-module to the H-module in Figure 2.

[14] The formulation is equivalent to having $B(\boldsymbol{x})$, $R(\boldsymbol{x})$ or $E(\boldsymbol{x})$ as the objective while constraining on others. This is because the primal problem in Equation (9) is feasible and bounded, so strong duality holds.

[15] The benefit of having analytical solutions is that we do not need to solve the large-scale linear programming problem online and only need to plug in the values for the analytical form instead.

[16] Note that b_{iq} and r_{iq} is multiplied by c_{iq}, whereas e_{iq} is not. This is because the basket value and retention outcome is a *counterfactual* estimation conditioning on the consumer placing an order.

[17] In the ideal case, the models would be retrained with every new data point that comes in, that is, in an online learning setting. However, online learning is not practical in real-word large-scale platforms, so most industry applications adopt an "off-policy batch training" practice, where models are retrained on a regular basis (e.g., every couple hours or daily).

[18] The fact that this sequential retraining strategy performs well in live experiments (Section 6) validates that the MOHR framework is able to capture at least the local equilibrium well.

[19] They always see different recommendations with every refresh, even under the same recommendation algorithm.

[20] We unfortunately could not generate the Pareto frontier for retention as it requires at least 28 days to be observed, but the random ranking data we had at the time of our project spans only one week.

[21] The Pareto frontier of basket value is noisier than that of the restaurant exploration. This is expected as the restaurant exploration objective is measured by the number of *impressions* a restaurant receives, which the recommender system has almost full control; The basket value objective relies on the consumer actually *placing an order* at a restaurant, which is a stochastic event influenced only partially by the recommender system. In other words, the basket value objective introduces an additional layer of randomness, resulting in a noisier Pareto frontier.

[22] Therefore, the ranges for the lambdas are relatively small, and only small lambda values are of practical importance to the platform as they offer minimal drop in conversion.

[23] A 28-day experiment is considered a long-term experiment by the company.

[24] Another relevant metric for new restaurants is their retention. However, in practice, restaurants rarely offboard themselves from the platform, as there is no cost associated with remaining on it. Therefore, restaurant retention is too stable to be used as a metric for online experiments.

[25] Offline analysis alone was unable to capture the benefits of exploration probably due to the low percentage of new restaurants, leading to the prediction in Figure 4 that an increase in the restaurant exploration objective would always result in a decrease in conversion.

[26] The control group ranks all rows above single restaurants. Therefore, the experiment group actually presents *fewer* restaurants in the top positions of the feed. Despite this, it still reduces the average vertical order position.

[27] We did not observe this trade-off for MOR, probably because MOR is only effective on part of the homepage.

[28] The final weight for the basket value is extremely small (λ_b = 0.002) compared with the weight for the other two objectives (λ_r = 0.6, λ_e = 1.3). This is because the basket value objective, which is measured in dollar amounts, is about two to three orders of magnitude larger than the other three objectives (consumer conversion, consumer retention, restaurant exploration).

[29] Only the standard deviation (or variance) of each objective matters in the final ranking results. The mean value of each objective only shifts the ranking score by a constant and does not affect the final ranking.

[30] As live experiments are costly to run, we were only able to conduct these online trade-off experiments for one objective. However, we think the results would be generalizable to other objectives.

[31] The actual basket value (defined in Table 3) captures the dollar amount of an order *conditional* on the consumer placing an order. In other words, it does not capture the consumer's conversion behavior.

[32] The basket value objective, which is measured in dollar amounts, is about two to three 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 Equation (7) dominated by the basket value objective.

[33] The MOHR trade-off curves (in blue dashed lines) in Figure 6, (a) and (b), are exactly those in Figure 4 but plotted on a much wider x axis and y axis.

[34] For platforms like YouTube, each video watched is treated as an individual recommendation session. For example, when a user finishes watching a video and returns to the YouTube homepage, a new recommendation session is generated, resulting in the presentation of new recommendations. MOHR is readily applicable to each of the recommendation sessions. This practice aligns with the current approach adopted by major video recommendation platforms (Covington et al. 2016, Wang et al. 2022).

[35] The penalty term can also be viewed as a "fairness" regularization by encouraging every item having equal opportunity to be shown to the consumers.

References

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

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge Data Engrg.* 17(6):734–749.

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

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

Agarwal D, Chen BC, Elango P, Wang X (2012) Personalized click shaping through Lagrangian duality for online recommendation. *Proc. 35th Internat. ACM SIGIR Conf. Res. Development Inform. Retrieval* 12:485–494.

Aggarwal CC (2016) *Recommender Systems*, vol. 1 (Springer, Berlin).

Agrawal R, Gollapudi S, Halverson A, Ieong S (2009) Diversifying search results. *Proc. Second ACM Internat. Conf. Web Search Data Mining* 09:5–14.

Anderson A, Maystre L, Anderson I, Mehrotra R, Lalmas M (2020) Algorithmic effects on the diversity of consumption on Spotify. *Proce. Web Conf.* 20:2155–2165.

Ansari A, Essegaier S, Kohli R (2000) Internet recommendation systems. *J. Marketing Res*. 37(3):363–375.

Aramayo N, Schiappacasse M, Goic M (2023) A multiarmed bandit approach for house ads recommendations. *Marketing Sci.* 42(2): 271–292.

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

Aridor G, Gonçalves D (2022) Recommenders' originals: The welfare effects of the dual role of platforms as producers and recommender systems. *Internat. J. Indust. Organ.* 83:102845.

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

Azaria A, Hassidim A, Kraus S, Eshkol A, Weintraub O, Netanely I (2013) Movie recommender system for profit maximization. *Recsys* 13:121–128.

Bahrami S, Nourinejad M, Yin Y, Wang H (2021) The three-sided market of on-demand delivery. Preprint, submitted October 18, https://dx.doi.org/10.2139/ssrn.3944559.

Billsus D, Pazzani MJ (1998) Learning collaborative information filters. *Proc. Fifteenth Internat. Conf. Machine Learning* 98:46–54.

Blattberg RC, Kim BD, Neslin SA, Blattberg RC, Kim BD, Neslin SA (2008) *Why Database Marketing?* (Springer, Berlin).

Bodapati AV (2008) Recommendation systems with purchase data. *J. Marketing Res.* 45(1):77–93.

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

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. *Proc. Fourteenth Conf. Uncertainty Artificial Intelligence* 98:43–52.

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

Burtini G, Loeppky J, Lawrence R (2015) A survey of online experiment design with the stochastic multi-armed bandit. Preprint, submitted October 2, https://arxiv.org/abs/1510.00757.

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

Chaney AJ, Stewart BM, Engelhardt BE (2018) How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. *Proc. 12th ACM Conf. Recommender Systems* 18:224–232.

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

Chen M, Wang Y, Xu C, Le Y, Sharma M, Richardson L, Wu SL, et al. (2021) Values of user exploration in recommender systems. *Proc. 15th ACM Conf. Recommender Systems*, 85–95.

Chung J, Rao VR (2012) A general consumer preference model for experience products: Application to Internet recommendation services. *J. Marketing Res.* 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. *Proc. 10th ACM Conf. Recommender Systems*, 191–198.

Das A, Mathieu C, Ricketts D (2009) Maximizing profit using recommender systems. Preprint, submitted August 25, https://arxiv.org/abs/0908.3633.

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

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

Donnelly R, Kanodia A, Morozov I (2023) Welfare effects of personalized rankings. *Marketing Sci.* 43(1):92–113.

Dudík M, Langford J, Li L (2011) Doubly robust policy evaluation and learning. *Proc. 28th Internat. Conf. Machine Learn.*, 1097–1104.

Elahi E, Chandrashekar A (2020) Learning representations of hierarchical slates in collaborative filtering. *Recsys* 21:703–707.

Evans DS, Schmalensee R (2016) *Matchmakers: The New Economics of Multisided Platforms* (Harvard Business Review Press, Cambridge, MA).

Fader PS, Hardie BG (1996) Modeling consumer choice among SKUs. *J. Marketing Res.* 33(4):442–452.

Fader PS, Hardie BG, Lee KL (2005) RFM and CLV: Using iso-value curves for customer base analysis. *J. Marketing Res.* 42(4):415–430.

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

Farrell MH, Liang T, Misra S (2020) Deep learning for individual heterogeneity: An automatic inference framework. Preprint, submitted October 28, https://arxiv.org/abs/2010.14694.

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

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Ann. Statist.* 29(5):1189–1232.

Friedman JH (2002) Stochastic gradient boosting. *Comput. Statist. Data Anal.* 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 Sci.* 31(3):493–520.

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

Gomez-Uribe CA, Hunt N (2015) The Netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Management Inform. Systems* 6(4):1–19.

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

GVR (2022) Online food delivery market size; share report, 2028. Accessed August 1, 2023, https://www.grandviewresearch.com/industry-analysis/online-food-delivery-market-report.

Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2 (Springer, Berlin).

Hitsch GJ, Misra S, Zhang WW (2024) Heterogeneous treatment effects and optimal targeting policy evaluation. *Quant. Marketing Econom.* 22(2):115–168.

Horvitz DG, Thompson DJ (1952) A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.* 47(260):663–685.

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

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 Sci.* 60(4):805–823.

Jacobs BJ, Donkers B, Fok D (2016) Model-based purchase predictions for large assortments. *Marketing Sci.* 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 Sci.* 40(2):219–240.

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

Katehakis MN, Veinott AF Jr (1987) The multi-armed bandit problem: Decomposition and computation. *Math. Oper. Res.* 12(2):262–268.

King J, Imbrasaitė V (2015) Generating music playlists with hierarchical clustering and q-learning. *37th European Conf. IR Res. ECIR 2015*, vol. 15 (Springer International Publishing, New York), 315–326.

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

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. *Proc. 19th Internat. Conf. World Wide Web*, 661–670.

Li L, Chu W, Langford J, Wang X (2011) Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proc. Fourth ACM Internat. Conf. Web Search Data Mining* 11:297–306.

Li Z, Fang X, Bai X, Sheng ORL (2017) Utility-based link recommendation for online social networks. *Management Sci.* 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. *Management Inform. Systems Quart.* 43(3):765–786.

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

Liu TY (2009) Learning to rank for information retrieval. *Foundations Trends Inform. Retrieval* 3(3):225–331.

March JG (1991) Exploration and exploitation in organizational learning. *Organ. Sci.* 2(1):71–87.

Miller BN, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: Experiences with an occasionally connected recommender system. *Proc. 8th Internat. Conf. Intelligent User Interfaces,* 263–266.

Mooney RJ, Roy L (2000) Content-based book recommending using learning for text categorization. *Proc. Fifth ACM Conf. 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. *J. Theoretical Appl. Electronic Commerce Res.* 16(5):1297–1310.

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

Oehlert GW (1992) A note on the delta method. *Amer. Statist.* 46(1):27–29.

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

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

Ricci F, Rokach L, Shapira B (2015) Recommender systems: Introduction and challenges. *Recommender Systems Handbook* (Springer, Berlin), 1–34.

Ruder S (2017) An overview of multi-task learning in deep neural networks. Preprint, submitted June 15, https://arxiv.org/abs/1706.05098.

Sahoo N, Singh PV, Mukhopadhyay T (2012) A hidden Markov model for collaborative filtering. *Management Inform. Systems Quart.* 36(4):1329–1356.

Sawaragi Y, Nakayama H, Tanino T (1985) *Theory of Multiobjective Optimization* (Elsevier, New York).

Schmalensee R, Armstrong M, Willig RD, Porter RH (1989) *Handbook of Industrial Organization* (Elsevier, New York).

Schnabel T, Swaminathan A, Singh A, Chandak N, Joachims T (2016) Recommendations as treatments: Debiasing learning and evaluation. *Internat. Conf. Machine Learn.* 16:1670–1679.

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

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

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

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

Solsman JE (2018) YouTube's AI is the puppet master over most of what you watch. *CNET* (January10), https://www.cnet.com/tech/services-and-software/youtube-ces-2018-neal-mohan/.

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

Strehl A, Langford J, Li L, Kakade SM (2010) Learning from logged implicit exploration data. *Adv. Neural Inform. Processing Systems* 23.

Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).

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, et al. (2009) Hive: A warehousing solution over a map-reduce framework. *Proc. VLDB Endowment* 2(2):1626–1629.

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

Ursu RM, Zhang Q, Honka E (2023) Search gaps and consumer fatigue. *Marketing Sci.* 42(1):110–136.

Wagner U, Taudes A (1986) A multivariate polya model of brand choice and purchase incidence. *Marketing Sci.* 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 Sci.* 64(11):5198–5219.

Wang Y, Sharma M, Xu C, Badam S, Sun Q, Richardson L, Chung L, et al. (2022) Surrogate for long-term user experience in recommender systems. *Conf. Knowledge Discovery Data Mining* 22:4100–4109.

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

Wu Q, Wang H, Hong L, Shi Y (2017) Returning is believing: Optimizing long-term user engagement in recommender systems. *Conf. Inform. Knowledge Management* 17:1927–1936.

Xiao B, Benbasat I (2007) E-commerce product recommendation agents: Use, characteristics, and impact. *Management Inform. Systems Quart.* 31(1):137–209.

Xie X (2010) Potential friend recommendation in online social network. *Proc. IEEE/ACM Internat. Conf. Green Comput. Comm. Internat. Conf. Cyber Physical Social Comput.* (IEEE, New York), 831–835.

Xie R, Zhang S, Wang R, Xia F, Lin L (2021) *Hierarchical Reinforcement Learning for Integrated Recommendation* (AAAI Press, Palo Alto, CA).

Yang J, Eckles D, Dhillon P, Aral S (2024) Targeting for long-term outcomes. *Management Sci.* 70(6):3841–3855.

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

Yoganarasimhan H, Barzegary E, Pani A (2023) Design and evaluation of optimal free trials. *Management Sci.* 69(6):3220–3240.

Zhang TC, Agarwal R, Lucas HC Jr (2011) The value of it-enabled retailer learning: Personalized product recommendations and customer store loyalty in electronic markets. *Management Inform. Systems Quart.* 35(4):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. *Management Inform. Systems Quart.* 45(1):1–43.

Zhao Z, Hong L, Wei L, Chen J, Nath A, Andrews S, Kumthekar A, et al. (2019) Recommending what video to watch next: A multitask ranking system. *Recsys* 19:43–51.

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

Zhou B, Zou T (2023) Competing for recommendations: The strategic impact of personalized product recommendations in online marketplaces. *Marketing Sci.* 42(2):360–376.