ELECTRONIC COMPANION

EC.1. Comparison of Query Recommendation System (QRS), Auto-Completion, and Item Recommender System

Feature	Query Recommendation System (QRS)	Auto-Completion	Item Recommender System
Primary Objective	Encourage exploration by suggesting adjacent, diverse topics.	Optimize query entry efficiency by completing partial user inputs.	Recommend specific items/products to maximize relevance and sales.
Focus	Breadth: Expanding user consideration sets.	Depth: Narrowing down existing user intent.	Personalization: Matching user preferences with items.
Mechanism	Suggests related but diverse queries, often broadening search scope.	Predicts the user's intended query based on historical patterns and user-specific behavior.	Uses collaborative filtering or content-based algorithms to recommend items based on user behavior or attributes.
Output Type	Diverse query suggestions that may introduce new categories or merchants.	Personalized query completions tailored to refine user input.	Specific product or item directly relevant to user preferences.

Table EC.1 Comparison of Query Recommendation System (QRS), Auto-Completion, and Item Recommender System.

EC.2. Screenshot of the App Interface and Experimental Design

Figure EC.1 Screenshot of the App Landing Page

Figure EC.2 **Original Chinese Content of Query Recommender System Experiment**

EC.3. Query Recommendation Algorithm

The QRS system generates recommended queries via two main steps: recall and ranking. The system architecture is presented in FigureC1. During the first, recall, step, two methods are employed to generate a set of candidate queries for potential recommendation: a collaborative filter and query recall based on semantic similarity.

The collaborative filter identifies candidate queries as those that have historically been used by other users in proximity to the focal query (e.g., users who ran this query also ran that query). The collaborative filtering component of the algorithm draws on the item-based collaborative filter (ItemCF), first popularized by Amazon. The algorithm calculates the similarity between items (queries in this case), producing a query similarity matrix. The steps involved in the ItemCF can be summarized as follows:

1. Construct the user-query co-occurrence matrix: Based on user behavior, a co-occurrence matrix is constructed with users as rows and queries as columns.

2. Build the query similarity matrix: Using the co-occurrence matrix, calculate the similarity between each pair of queries (e.g., cosine distance).

3. Retrieve the top-n queries most similar to the focal query.

This initial set of candidate queries is then supplemented with additional queries, which are identified as semantically similar to the focal query. This is achieved by first obtaining vector embeddings of queries using BERT, a state-of-the-art, neural-network-based bidirectional encoder developed by Google, which has achieved remarkable success in various natural language processing (NLP) tasks. Using these vector embeddings, the system then draws on Facebook AI Similarity Search (FAISS), a tool that enables rapid online similarity search at scale, to identify candidate queries that are semantically similar to the focal query. Here, it is important to note that the system enforces a limiting threshold, such that queries that are *too* semantically similar are eliminated from consideration.

Once the complete set of candidate queries is constructed, the system then proceeds to rank them. The user's profile characteristics, recent behavior in the app, their context (e.g., time and place), recent transaction data, and the embedding of the focal query are passed as input to a ranking model, which scores each candidate query for presentation to the user. The QRS then ultimately displays the top-ranked candidates as query recommendations.

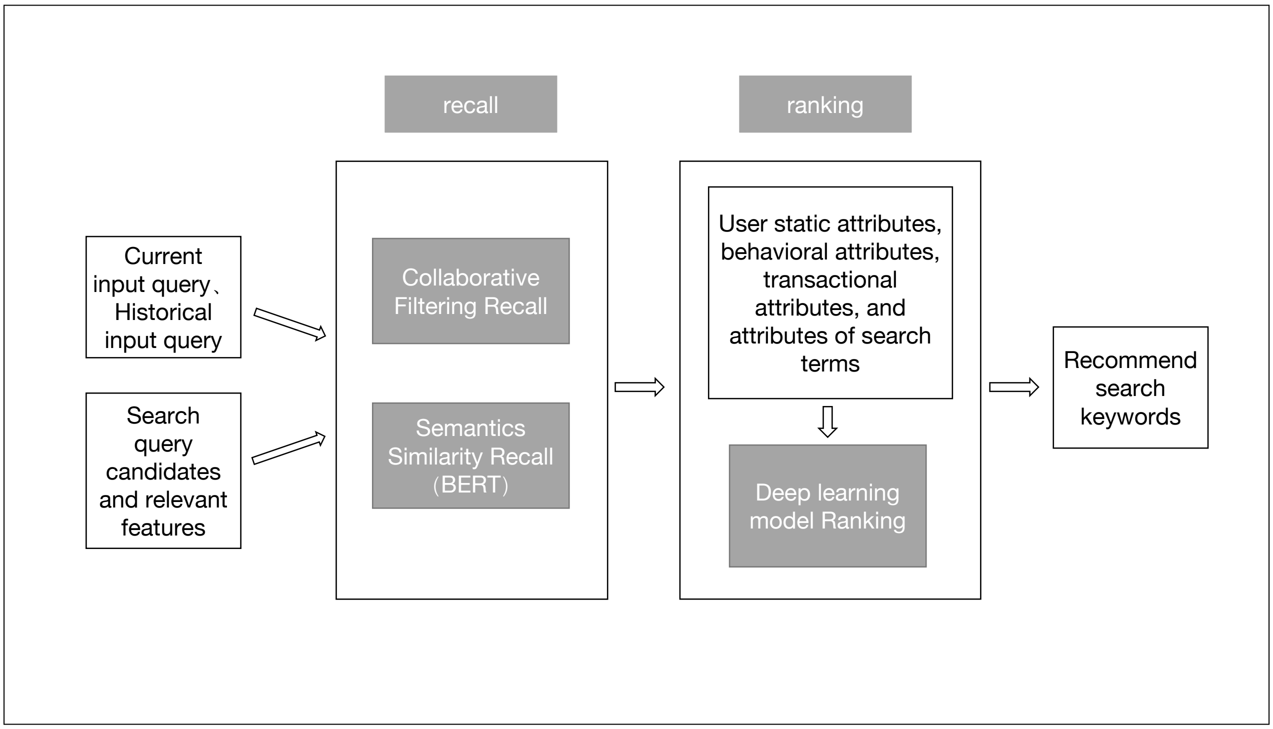


Figure C1 **Query Recommender Algorithm Architecture**

EC.4. The Search Process Using the Query Recommender System Feature

Step 1: Tab the Search Bar **Step 2: Enter Search Query** **Step 3: Query Recommendation and Auto-complete** **Step 4: Click the Recommended Query to the Result Page**

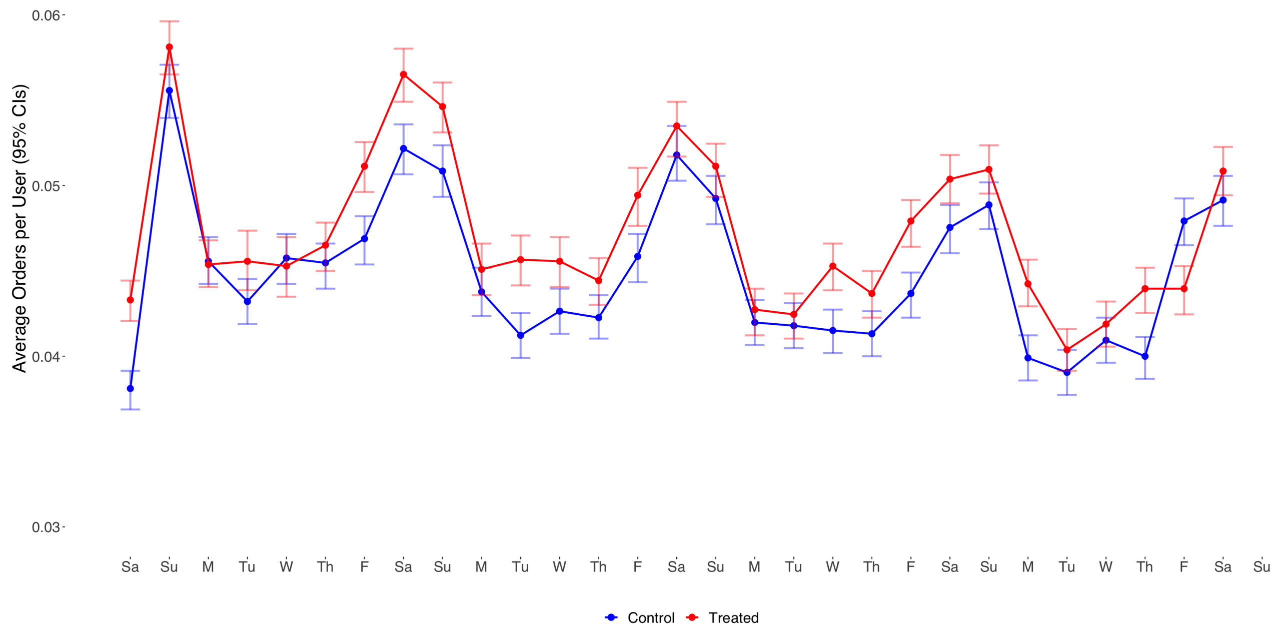
Figure D1 The Search Process with Query Recommender System Feature

EC.5. Variable Summary Statistics

Table EC.2 Variable Summary Statistics

Variable Name	Mean	Std. Dev.	Min	Max
Panel A: Background Variables				
Gender	0.414	0.747	0	2
Age	2.546	1.426	0	5
Membership Tier	3.937	1.457	1	7
Education	0.506	0.499	0	1
Spending Level	1.920	0.799	0	3
Marriage Status	0.002	0.051	0	1
Panel B: Dependent Variables				
Query Volume	23.209	27.479	0	518
Tap Volume	20.063	25.172	0	766
Order Volume	4.391	5.566	0	43

EC.6. Daily Effect Estimates



Note: We plot average order volumes in the treatment and control conditions, by experiment day, with 95% confidence intervals. The x-axis indicates daily progression through the experiment, noting the days of the week. As noted earlier, most users will complete no more than one order on any given day. Accordingly, the difference in average order counts between the two groups translates roughly to differences between the two groups in the fraction of users completing an order on each day. We see clear evidence that the treatment group exhibits systematically higher order volumes.

EC.7. Exploration of the Underlying Mechanism

EC.7.1. Query Categories Recommended by QRS

The app maintains a set of more than 700 food category labels, and queries are automatically labeled with a food category at the time of their entry. Note that some queries may be too generic to receive a meaningful label. Accordingly, the food category labels also include an 'other' category. Using these food category labels, we assessed the overlap in food categories associated with search queries that users manually type into the search box and those associated with QRS-recommended queries. We observe relatively little overlap, just 32.6%. This result indicates that the QRS recommends queries that relate to food categories that a user would not enter on their own.

EC.7.2. Effects on Query Composition of User Input

We can also examine the treatment's effect on more indirect measures of users' exposure to novel merchants and products. Conceptually, customers may enter two types of initial queries before observing an auto-completion suggestion or query recommendation: specific queries and generic queries. In the case of specific search queries, users type a highly specific keyword that aims to locate a particular merchant or food menu item. For example, "McDonald's" or "McFlurry." In contrast, in the case of generic queries, users enter keywords that are coarse or vague, i.e., which do not mention a specific product or merchant. For example, a user might query for "fast food." Queries are automatically labeled by the system as "generic" or "specific." This labeling is achieved using BERT, a deep-learning architecture applied to obtain the associated vector embedding. The vectorized query is finally passed to a neural network-based classifier, which classifies the query into either a generic or specific type. The app management team has indicated that this classifier is highly accurate, correctly identifying the query type with an accuracy of approximately 94% in a hand-labeled, class-balanced sample. Based on the generic-specific labels, we conduct two more analyses to understand the type of queries that the QRS produces and whether access to the QRS also induces users to initially enter more generic (versus specific) queries into the search bar when beginning a search.

Examining the proportion of generic (versus specific) query recommendations, we find that the majority are indeed labeled as generic. In particular, 56.81% of recommended queries are labeled as generic. Next, estimating the effect of treatment on the volume of generic versus specific queries that users enter into the search box. Note that we do not include the QRS-generated queries in these counts; these measures examine only what users manually enter into the search bar. On average, we observe the results reported in Table EC.3. The positive and statistically significant coefficient in Column (1) indicates that the QRS drives users to enter systematically more generic queries into the search bar, on average. Interestingly, we find a statistically insignificant result in Column (2), indicating that we have no evidence whether the QRS drives a decline in the prevalence of specific search queries. This result indicates that the QRS drives an overall increase in query entry, and that most of those additional queries are generic in nature.

Beyond the generic-versus-specific nature of the queries that users enter, we also considered whether the QRS causally affects query length, as generic queries are presumably also shorter, given their lack of detail. In Column (3), we estimate that treated users enter queries that are systematically shorter, on average.

EC.7.3. Effects on New Search Query Categories

Finally, we examined the treatment's effect on the total count of unique food categories that a user's queries are labeled with by the system, as well as the volume of queries a user enters that come to be labeled with product categories that a user has not purchased or searched for in the two months prior to the experiment. These results are presented

Table EC.3 Additional Analyses: Impact of Query Recommender System on Search Query Content

	(1) *ln Generic Query*	(2) *ln Specific Query*	(3) *ln Query Length*	(4) *ln Product Categories*	(5) *ln New Product Categories*
Query Recommender System	0.024***	−0.002	−0.006**	0.011*	0.013*
	(0.008)	(0.008)	(0.002)	(0.006)	(0.006)
Constant	2.214***	1.226***	1.417***	2.379***	1.719***
	(0.017)	(0.017)	(0.005)	(0.013)	(0.014)
Controls	Y	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519	58,519
R^2	0.084	0.080	0.011	0.167	0.068

Note: This table reports the effect of the Query Recommender System on the characteristics of users' search queries. Generic search queries refer to broad search terms that are not specific to any SKU. Specific search queries indicate detailed and narrowly defined search terms for a particular merchant or food item. Query length is the average number of characters in a search query. Product categories represent the total number of distinct product categories associated with user queries. New product categories denote the total number of previously unexplored product categories introduced in user queries. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$. Robust standard errors are reported in parentheses.

in Columns (4) and (5). We find that the QRS causes users to enter queries associated with a systematically larger set of food categories and, further, that the queries entered are systematically more likely to be labeled with food categories that are new to the user.

These findings collectively suggest that the QRS induces users to engage in more expansive search by reducing the query generation efforts. First, and most directly, we find that the QRS tends to recommend generic (rather than specific) search queries to a user, and it also tends to recommend queries that pertain to food categories that are new to the user. Second, the QRS's influence appears to be indirect, as it also induces the user to begin manually entering more generic queries and queries that are tied to new food categories, when the user initiates a search.

EC.8. Individual-level and Market-level Sales Diversity

EC.8.1. Individual-level Sales Diversity

The table below reports the impact of the Query Recommender System on individual-level sales diversity, specifically on the tap and order volumes for new and repeat merchants. Columns (1)–(4) show results for new merchants, indicating that QRS significantly increases tap and order volumes, as well as their logarithmic transformations, suggesting heightened exploration and engagement with new merchants. Columns (5)–(8) present results for repeat merchants, revealing no significant effect on tap volume but a negative impact on order volume, indicating reduced purchase concentration among familiar merchants.

Table EC.4 Treatment Effects on Individual-level Sales Diversity

	New Merchants				Repeat Merchants			
	Tap Volume	ln *Tap Volume*	*Order Volume*	ln *Order Volume*	*Tap Volume*	ln *Tap Volume*	*Order Volume*	ln *Order Volume*
Query Recommender System	0.331**	0.014*	0.078**	0.013**	0.066	0.005	−0.040**	−0.010**
	(0.143)	(0.008)	(0.034)	(0.006)	(0.090)	(0.006)	(0.017)	(0.004)
Constant	11.589***	1.800***	4.038***	1.130***	2.941***	0.539***	0.834***	0.303***
	(0.300)	(0.018)	(0.077)	(0.014)	(0.237)	(0.013)	(0.036)	(0.008)
Controls	Y	Y	Y	Y	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519	58,519	58,519	58,519	58,519
R^2	0.038	0.093	0.074	0.092	0.003	0.025	0.014	0.031

Note: This table reports the effects of the Query Recommender System on tap and order volumes for new and repeat merchants. "New Merchants" refer to previously unvisited merchants, while "Repeat Merchants" refer to merchants with prior interactions. Both the new and repeated merchants are defined based on the 6-month pre-treatment period data. Log-transformed variables are indicated with "ln." *** $p < 0.01$; ** $p < 0.05$; * $p < 0.10$. Robust standard errors are reported in parentheses.

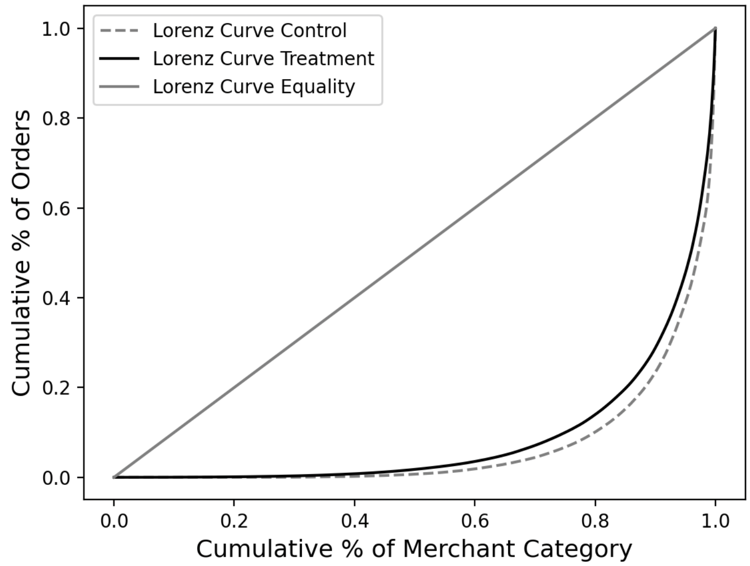
EC.8.2. Market-level Sales Diversity

The Lorenz curve, traditionally used in economics to represent income or wealth distribution within a population, has also frequently been employed to visualize market concentration. The curve depicts the distribution of market share among firms in a marketplace. On the horizontal axis, firms are ordered from the smallest to the largest in terms of market share, with movement to the right indicating the cumulative percentage of firms in the market. The vertical axis indicates the cumulative percentage of market share associated with these firms.

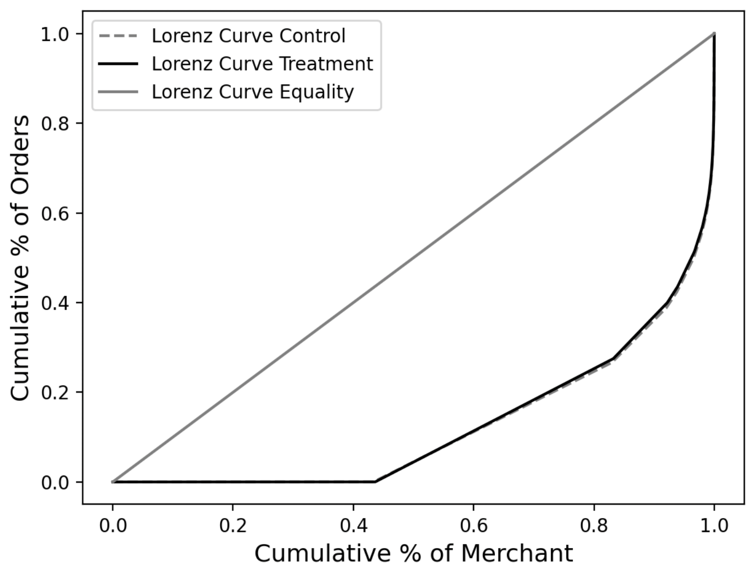
In a perfectly competitive market, where firms all hold equal market share, the Lorenz curve would be a 45-degree line. In markets where only a few firms dominate, the curve bows significantly below the line of equality. The greater the deviation, the greater the market concentration.

We generate Lorenz curves reflecting market-level concentration and diversity in sales at both the *merchant-category level* (food menu items are assigned to mutually exclusive categories, and restaurants are then categorized based on the most common category of food item that they sell. The system maintains more than 700 food categories. Examples include hotpot, BBQ, seafood, poultry, vegetable, meat, soup, noodle, dumpling and fried food.) and at the *merchant level* (for computational reasons, we limit this analysis to only those restaurants that obtained at least one transaction from at least one of the sampled customers during the 3-months leading up to the experiment). Panel A in Figure H3 depicts our curves at the merchant category level and Panel B at the merchant level. Each figure depicts two lines, reflecting the degree of market concentration in the control group and in the treatment group, respectively, in addition to the 45-degree line conveying the ideal of perfect equality.

As can be seen, the treatment group exhibits systematically greater consumption diversity (equality in the distribution of sales) than the control group. That is, the line associated with the distribution of demand in the treatment group is closer to the ideal than that of the control group. To quantify these effects numerically, we next calculate a Gini coefficient associated with each experimental group, for each diversity measure (category-level and merchant-level). To calculate a Gini coefficient from a Lorenz curve, we measure the area between the line of perfect equality and the observed Lorenz curve, and express that area as a fraction of the total area below the line of equality. Gini coefficients thus range from 0



(a) Lorenz Curve for Merchant Category Diversity



(b) Lorenz Curve for Merchant Diversity

Figure H3 Lorenz Curves of Consumption Diversity

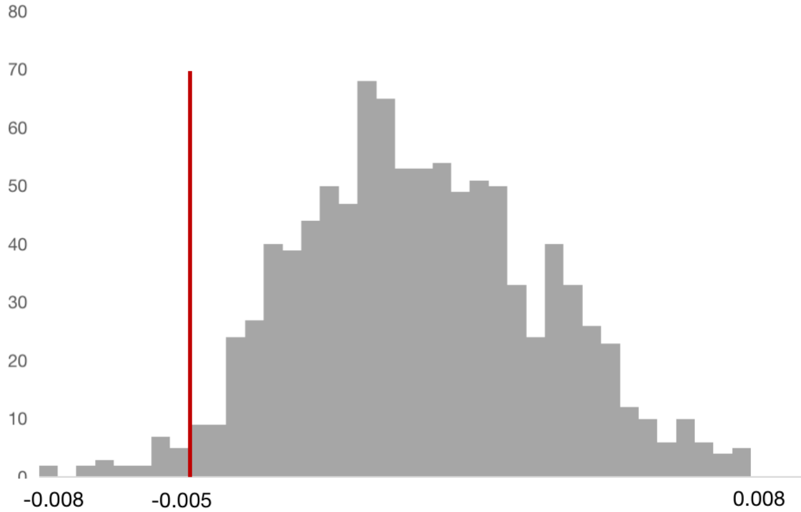
Note: The Lorenz curves depict sales concentration and diversity across merchant categories (Panel A) and individual merchants (Panel B). The 45-degree line indicates perfect equality, where all firms share equal market proportions. Deviations from this line reflect market concentration. In the treatment group, the Query Recommender System reduces concentration by fostering diversity at both levels. Greater deviations from the equality line signify higher concentration.

to 1, where 0 represents perfect equality (every entity has an equal share) and 1 represents perfect inequality (one category or seller dominates the entire market). Greater market level diversity will, therefore, manifest as a smaller Gini coefficient. To operationalize the effect of our treatment on market-level diversity, we calculate the difference in the Gini values between the treatment and control group ($Gini_{\mathrm{t}} - Gini_{\mathrm{c}}$).

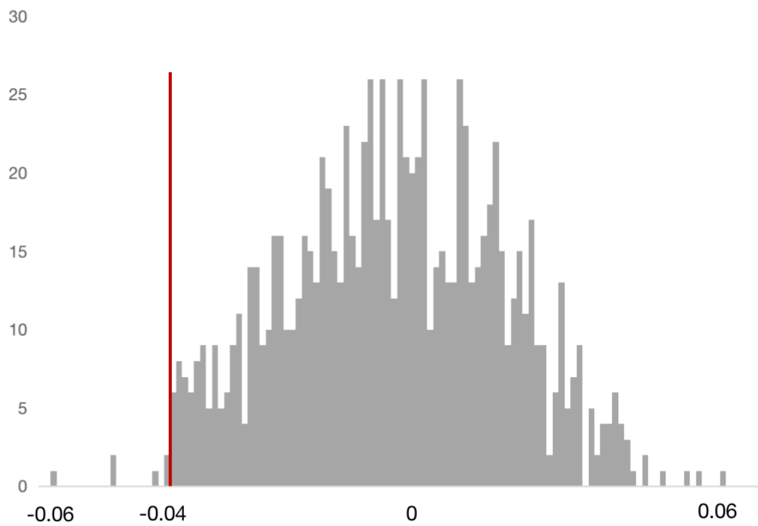
Of course, on its own, the difference between two Gini coefficients lacks any notion of statistical significance. To gauge that, to understand whether our treatment's effects are meaningful, we follow the approach of Lee and Hosanagar (2019), implementing a permutation test (Dufour et al. 2019), i.e., randomization inference. We thereby recover an empirical distribution of differences in Gini coefficients between treatment and control under the null hypothesis of no difference by repeatedly shuffling the true treatment assignment indicator at random, across users, such that we break its association with changes in the Gini coefficient, by design. With each shuffling of the treatment vector, we recalculate group-specific Gini coefficients and their difference, storing the value. We repeat this process 1,000 times. Finally, we compare the true observed difference in Gini values between experimental conditions, based on the true vector of treatment assignments, and examine how extreme the true value is relative to the distribution of values obtained under the null. An empirical p-value is then calculated as the fraction of all Gini differences under the null that are more extreme than the observed value.

In Figure H4, we depict the results of the permutation test for our merchant category-level and merchant-level measures. In each figure, the gray histograms reflect the distribution of differences in Gini values recovered from our 1,000 shuffling–permutation steps. The red vertical line reflects the difference in Gini values that we would obtain using the true treatment assignment vector. Panel A depicts the results for our merchant level measure, and Panel B the results for our merchant-category level measure. In both cases, the treatment effect drives a statistically significant decline in the Gini coefficient ($p < 0.05$).

Notably, these results differ slightly from those observed in the context of item recommenders, as reported by Lee and Hosanagar (2019). Those authors report that RSs induce



(a) Difference in Gini Coefficients at Merchant Level



(b) Difference in Gini Coefficients at Merchant Category Level

Figure H4 Permutation Tests of Treatment Effects on Gini Coefficients

Note: This figure visualizes the results of permutation tests assessing the treatment effect on market-level diversity, measured using the Gini coefficient, at the merchant level (Panel A) and merchant category level (Panel B). The Gini coefficient quantifies inequality, where 0 represents perfect equality (all entities have an equal share) and 1 represents perfect inequality (one entity dominates entirely). The gray histograms in each panel represent the empirical distribution of differences in Gini coefficients derived from 1,000 random permutations of the treatment assignment vector under the null hypothesis of no treatment effect. The red vertical line marks the observed difference in Gini coefficients between the treatment and control groups based on the actual experimental data. In both panels, the observed difference lies at the extreme end of the null distribution, indicating a statistically significant reduction in market concentration driven by the treatment ($p < 0.05$). This reflects increased diversity at both the merchant and merchant category levels due to the Query Recommender System intervention.

greater market-level concentration in customer purchases. Here, we have found that a QRS increases the diversity of customer purchases not only at the individual level, but also at the market level.

EC.9. Heterogeneous Effects across Users with Different Levels of Reliance on Auto-completion

Results of this moderated regression are presented in EC.5. Considering estimates in Columns (1) and (3), we see that greater pre-treatment use of auto-completion associates with a larger, more positive effect of the QRS on tap volumes and order volumes. In particular, we observe a positive and statistically significant coefficient associated with the *Query Recommender System* × *Auto-completion Reliance* term in each column. Interestingly, the main effect of the treatment (the *Query Recommender System* term) on tap volumes is significant and negative, implying that the treatment systematically reduces the number of merchants viewed among users who did not rely on auto-completion in the pre-treatment period. Paired with the finding that the treatment does not systematically affect the volume of orders these users make (the main effect of *Query Recommender System* in Columns (3) and (4) is insignificant), this result indicates that, although these treated users do not begin to make more orders, their search seems, at least, to be more efficient. This latter result also highlights the importance of the auto-completion feature as a complement to the QRS: absent auto-completion use, it appears that the QRS triggers reduced search, perhaps because the QRS is recommending systematically more specific queries to these users, or perhaps because the greater complexity of the search interface serves to overload these users, increasing their perceived search costs.

Taken together, these results indicate comprehensively that the QRS enables more expansive exploration of the product catalog, and that such exploration is made possible by users' simultaneous access to and use of the auto-completion feature.

Table EC.5 Moderating Effect of Pre-treatment Auto-completion Reliance

	(1) *Tap Volume*	(2) *ln Tap Volume*	(3) *Order Volume*	(4) *ln Order Volume*
Query Recommender System	−0.542**	−0.019	−0.054	−0.007
	(0.256)	(0.016)	(0.035)	(0.010)
Query Recommender System × *Auto-completion Reliance*	1.234***	0.044**	0.120***	0.026**
	(0.314)	(0.018)	(0.041)	(0.012)
Auto-completion Reliance	4.223***	0.382***	0.223***	0.112***
	(0.230)	(0.013)	(0.030)	(0.008)
Constant	9.592***	1.762***	1.272***	0.480***
	(0.347)	(0.018)	(0.044)	(0.012)
Controls	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519
R^2	0.076	0.142	0.031	0.048

Note: This table shows the moderating effects of customers' pre-treatment reliance on the auto-completion feature. Column (1) and Column (2) examine the impact on tap volumes (absolute and log-transformed), while Column (3) and Column (4) focus on order volumes (absolute and log-transformed).. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$. Robust standard errors are reported in parentheses.

EC.10. Robustness Checks

EC.10.1. Ruling Out Novelty Effect

We begin by addressing potential novelty effects. On average, users complete slightly fewer than four orders during the 30-day experimental period. Given that users typically place no more than one order per day, this suggests that users in our sample tend to make purchases approximately twice a month. Consequently, while our findings indicate that the positive treatment effects persist throughout the experimental period, it is possible that these effects are partially driven by the novelty of the feature, particularly for users encountering it for the first time toward the latter part of the experiment.

Although our ability to fully rule out this possibility is limited, we take a step toward addressing it by re-estimating the effects after excluding taps and orders from users' initial sessions during the experimental period. Specifically, we drop data associated with users' first to third sessions and reanalyze the results. The outcomes of this analysis are reported in Table EC.6 through Table EC.8.

Table EC.6 Ruling Out Novelty Effect: Impact of Query Recommender System on Merchant Visits and Orders (Excluding First Exposure)

	(1) *Tap Volume*	(2) ln *Tap Volume*	(3) *Order Volume*	(4) ln *Order Volume*
Query Recommender System	0.399**	0.015*	0.051*	0.012*
	(0.154)	(0.008)	(0.021)	(0.005)
Constant	11.378***	1.854***	1.340***	0.491***
	(0.334)	(0.018)	(0.042)	(0.011)
Controls	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519
R^2	0.060	0.105	0.024	0.045

Note: This table reports the effects of the QRS on total taps (*Tap Volume*) and orders (*Order Volume*) after excluding data from users' first login (session) during the experimental period. This approach aims to address potential novelty effects. Robust standard errors are reported in parentheses. $^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.10$.

Table EC.7 Ruling Out Novelty Effect: Impact on Merchant Visits and Order Volumes (Excluding First Two Exposures)

	(1) *Tap Volume*	(2) ln *Tap Volume*	(3) *Order Volume*	(4) ln *Order Volume*
Query Recommender System	0.410**	0.013*	0.029*	0.010*
	(0.164)	(0.008)	(0.017)	(0.005)
Constant	11.686***	1.903***	1.256***	0.506***
	(0.357)	(0.017)	(0.037)	(0.011)
Controls	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519
R^2	0.054	0.109	0.027	0.040

Note: This table reports the effects of the QRS on total taps (*Tap Volume*) and orders (*Order Volume*) after excluding data from users' first two logins (sessions) during the experimental period. This approach aims to address potential novelty effects. Robust standard errors are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

Table EC.8 Ruling Out Novelty Effect: Impact on Merchant Visits and Order Volumes (Excluding First Three Exposures)

	(1) *Tap Volume*	(2) ln *Tap Volume*	(3) *Order Volume*	(4) ln *Order Volume*
Query Recommender System	0.367**	0.014*	0.032*	0.010**
	(0.159)	(0.008)	(0.017)	(0.005)
Constant	11.307***	1.870***	1.204***	0.491***
	(0.346)	(0.017)	(0.036)	(0.011)
Controls	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519
R^2	0.054	0.109	0.026	0.040

Note: This table reports the effects of the QRS on total taps (*Tap Volume*) and orders (*Order Volume*) after excluding data from users' first three logins (sessions) during the experimental period. This approach aims to address potential novelty effects. Robust standard errors are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

EC.10.2. Session-level Regressions

Although we have reported clear evidence that the QRS drives a systematic rise in order volumes, the query volumes and merchant tap volumes also increase significantly. This raises questions about the efficiency of the users' search experience. Indeed, referring back to our descriptive evidence of the treatment effect, that positive effects occur only on weekends may reflect that the benefits emerge specifically when users have the time and flexibility to engage in a less efficient, more expansive search.

To gain a better sense of the implications of the QRS for search efficiency, we repeat our primary estimations considering data aggregated to the level of a user-session, rather than to the individual user. A search session is defined as when a customer conducts search queries with relevant keyword entries for a particular shopping intent within an app log-in activity. Results of these regressions are reported in Table EC.9. The estimates are largely consistent with our main findings. Most importantly, the findings indicate that the increase in order volumes does not just arise at the user level; increases arise within sessions as well, indicating that the QRS does not drive users to conduct multiple sessions before arriving at a purchase.

Table EC.9 Impact on Taps and Purchases at Session Level

	(1) *Tap Volume*	(2) ln *Tap Volume*	(3) *Order Volume*	(4) ln *Order Volume*
Query Recommender System	0.013**	0.003**	0.002*	0.001**
	(0.006)	(0.001)	(0.001)	(0.000)
Constant	1.431***	0.782***	0.117***	0.080***
	(0.011)	(0.003)	(0.002)	(0.001)
Controls	Y	Y	Y	Y
Observations	707,544	707,544	707,544	707,544
R^2	0.313	0.409	0.489	0.380

Note: This table reports the impact of the Query Recommender System on the total number of taps (*Tap Volume*) and orders (*Order Volume*) at the session level. Clustered standard errors at the individual level are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

EC.10.3. Robustness Checks Omitting Control Variables

Table J1 Impact of the Query Recommender System on Users' Total Taps and Orders (Omitting Control Variables)

	(1) *Tap Volume*	(2) ln *Tap Volume*	(3) *Order Volume*	(4) ln *Order Volume*
Query Recommender System	0.352**	0.011*	0.034*	0.011**
	(0.157)	(0.008)	(0.018)	(0.005)
Constant	16.598***	2.420***	1.333***	0.586***
	(0.111)	(0.005)	(0.012)	(0.003)
Controls	N	N	N	N
Observations	58,519	58,519	58,519	58,519
R^2	0.000	0.000	0.062	0.035

Note: This table reports the impact of the Query Recommender System on users' total taps (*Tap Volume*) and orders (*Order Volume*), omitting control variables. Robust standard errors are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

Table J2 Impact of the Query Recommender System on Manual vs. Auto-completed Search Queries (Omitting Control Variables)

	(1) *Auto-completed Queries*	(2) ln *Auto-completed Queries*	(3) *Manual Queries*	(4) ln *Manual Queries*
Query Recommender System	0.760***	0.024***	−0.261***	−0.017**
	(0.140)	(0.008)	(0.094)	(0.008)
Constant	13.295***	2.205***	7.651***	1.609***
	(0.099)	(0.005)	(0.067)	(0.006)
Controls	N	N	N	N
Observations	58,519	58,519	58,519	58,519
R^2	0.000	0.002	0.005	0.045

Note: This table reports the impact of the Query Recommender System on the number of manual vs. auto-completed search queries. *Auto-completed Queries* refers to searches generated using auto-completion, and *Manual Queries* refers to searches typed manually. Robust standard errors are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

Table J3 Impact of the Query Recommender System on Total Taps and Orders via Manual vs. Auto-complete Queries (Omitting Control Variables)

Section A: Auto-completion Query	(1) Tap Volume	(2) ln Tap Volume	(3) Order Volume	(4) ln Order Volume
Query Recommender System	0.508***	0.020**	0.053***	0.015***
	(0.126)	(0.008)	(0.009)	(0.004)
Constant	11.735***	2.070***	0.886***	0.435***
	(0.089)	(0.005)	(0.009)	(0.003)
Controls	N	N	N	N
Observations	58,519	58,519	58,519	58,519
R^2	0.000	0.016	0.003	0.002

Section B: Manual Query	(1) Tap Volume	(2) ln Tap Volume	(3) Order Volume	(4) ln Order Volume
Query Recommender System	−0.213***	−0.019***	−0.028***	−0.006*
	(0.063)	(0.005)	(0.009)	(0.003)
Constant	5.026***	1.283***	0.465***	0.236***
	(0.045)	(0.005)	(0.006)	(0.002)
Controls	N	N	N	N
Observations	58,519	58,519	58,519	58,519
R^2	0.002	0.015	0.003	0.071

Note: This table reports the impact of the Query Recommender System on users' total taps (*Tap Volume*) and orders (*Order Volume*) initiated via manual and auto-complete search queries. Robust standard errors are reported in parentheses. ***$p < 0.01$; **$p < 0.05$; *$p < 0.10$.

EC.11. Alternative Estimators

Table K3 Robustness to Alternative Estimators

Section A: Auto-completion vs. Manual Query Taps	**(1) Poisson (Auto-completion Taps)**	**(2) Negative Binomial (Auto-completion Taps)**	**(3) Poisson (Manual Taps)**	**(4) Negative Binomial (Manual Taps)**	**(5) Poisson (Total Taps)**	**(6) Negative Binomial (Total Taps)**
Query Recommender System	0.044***	0.040***	−0.040***	−0.032***	0.023**	0.025***
	(0.010)	(0.008)	(0.012)	(0.010)	(0.009)	(0.007)
Constant	1.903***	1.929***	1.685***	1.685***	2.426***	2.441***
	(0.010)	(0.016)	(0.028)	(0.021)	(0.021)	(0.308)
Controls	Y	Y	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519	58,519	58,519
Section B: Auto-completion vs. Manual Query Orders	**(1) Poisson (Auto-completion Orders)**	**(2) Negative Binomial (Auto-completion Orders)**	**(3) Poisson (Manual Orders)**	**(4) Negative Binomial (Manual Orders)**	**(5) Poisson (Total Orders)**	**(6) Negative Binomial (Total Orders)**
Query Recommender System	0.060***	0.058***	−0.059***	−0.051***	0.027**	0.028**
	(0.014)	(0.013)	(0.021)	(0.018)	(0.013)	(0.012)
Constant	−0.432***	−0.359***	−0.192***	−0.133***	0.249***	0.323***
	(0.032)	(0.027)	(0.046)	(0.037)	(0.029)	(0.025)
Controls	Y	Y	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519	58,519	58,519
Section C: Total Search Queries by Query Type	**(1) Poisson (Auto-completion Search Queries)**	**(2) Negative Binomial (Auto-completion Search Queries)**	**(3) Poisson (Manual Search Queries)**	**(4) Negative Binomial (Manual Search Queries)**	**(5) Poisson (Total Search Queries)**	**(6) Negative Binomial (Total Search Queries)**
Query Recommender System	0.057***	0.052***	−0.032***	−0.027**	0.026**	0.027***
	(0.009)	(0.009)	(0.012)	(0.012)	(0.008)	(0.008)
Constant	2.060***	2.087***	2.073***	2.075***	2.702***	2.716***
	(0.023)	(0.022)	(0.026)	(0.026)	(0.020)	(0.019)
Controls	Y	Y	Y	Y	Y	Y
Observations	58,519	58,519	58,519	58,519	58,519	58,519

Note: This table reports robustness checks using alternative regression estimators. Poisson and Negative Binomial regression models assess the impact of the Query Recommender System on users' query behavior and outcomes. Columns refer to Auto-complete and Manual Taps, Orders, and Total Search Queries. $^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.10$. Robust standard errors are reported in parentheses.