Modeling Filter-Assisted Consumer Search with Inverse Reinforcement Learning*

Kohei Onzo† Asim Ansari‡ Raghuram Iyengar§

August 9, 2026

Abstract

Online stores typically offer product filters to help consumers narrow their search. In some contexts, these filters overlap so that the same product appears across multiple filters. Consumers can also vary in how they search: some apply several filters while others purchase directly from a listing page without search. We propose a model that accommodates overlapping filters and heterogeneous search behavior as a dynamic program. The resulting model generalizes the Weitzman framework and other approaches that yield index-based policies. As existing estimation methods are not suitable for handling our large state space, we develop a novel Bayesian MCMC method based on inverse reinforcement learning that uses neural network-based value function approximation for estimation. We apply our model to clickstream data from an online retailer of Japanese incense. Our results show that search costs increase substantially with row position and that the variability of the post-search shocks is greater than that of the pre-search shocks. In counterfactual simulations, consolidating price filters shifts the composition of filter selections toward non-price filters. We also find that prohibiting purchases directly from listing pages reduces the predicted inside-good purchase probability by roughly seven percentage points.

Keywords: Filter-assisted sequential search, Dynamic programming, Inverse reinforcement learning, Neural networks, Value function approximation, Bayesian estimation

*We gratefully acknowledge use of the research computing resources of the Empire AI Consortium, Inc, with support from Empire State Development of the State of New York, the Simons Foundation, and the Secunda Family Foundation (Bloom et al., 2025).

†Kohei Onzo is a doctoral student in Marketing, Columbia Business School, Columbia University, USA.

‡Asim Ansari is William T. Dillard Professor of Marketing, Columbia Business School, Columbia University, USA.

§Raghuram Iyengar is Miers-Busch, W'1885 Professor and Professor of Marketing, The Wharton School, University of Pennsylvania, USA.

1 Introduction

Online retailers routinely provide product filters to help consumers narrow the set of alternatives they consider purchasing. On websites such as Amazon and Best Buy, a customer looking for a gift can restrict the display to products within a price range or to those with a particular functional attribute and can then click on individual items to read their descriptions. A product filter imposes a constraint on the values of relevant product attributes and applying the filter returns an ordered subset of products on a list or grid page, with some basic information about the products. Narrowing down alternatives in this fashion can be beneficial to consumers when inspecting every alternative is costly and the number of alternatives is large (Hauser and Wernerfelt, 1990; Jacoby et al., 1974; Kuksov and Villas-Boas, 2010; Scheibehenne et al., 2010).

Two features of the filtering interface can influence how consumers navigate it. First, the page returned by a filter provides incomplete information about products. A consumer can see the price and infer that a displayed product satisfies the active filtering constraint. However, detailed specifications, images, and reviews that resolve the remaining uncertainty can only be obtained by clicking on the product to access a product description page. Second, filters overlap, as the same product can appear under multiple filters. For example, a product in a high price range may also have a premium attribute so that a customer who applies a high price filter learns basic information about products that are also returned by the filter associated with the premium attribute. Filtering and searching are therefore distinct activities with distinct costs, and consumers can interweave them in a search session.

Consumers differ in their use of filters. Some apply several filters in succession to scan candidate subsets before inspecting any single product, whereas others apply one filter and buy immediately from the list page without reading a description page at all. A consumer who does not need to resolve all uncertainty about a product, or whose search cost is high relative to the value of additional information to acquire, can rationally add an item to the shopping cart directly from a listing page. Such behavior is plausible in many e-commerce contexts and is consistent with customer actions in our empirical application.

The literature in marketing and economics has primarily studied the sequential search and purchasing behavior of consumers through the lens of the Weitzman framework (Weitzman, 1979). This literature has covered a wide range of empirical settings, including online demand (Kim et al., 2010, 2017), advertising and consumer awareness (Honka et al., 2017), search rankings (Ursu, 2018), preference heterogeneity under costly search (Morozov et al., 2021), consumer fatigue from search (Ursu et al., 2023), and consideration set formation (Morozov, 2023). Ursu et al. (2025) review this growing literature. The Weitzman framework is appealing because it yields a tractable index-based optimal policy for the Markov decision process (MDP) that governs structural models of sequential search. Each alternative is assigned a reservation value that can be computed in isolation and ex ante, and the optimal policy compares these indices to the maximum realized utility at any stage of the search session to decide which alternative to search next and when to stop searching. Two assumptions are particularly crucial for this optimal policy to be valid: (a) the utility distributions governing the alternatives must be independent of one another, and (b) the consumer can only purchase an alternative that has already been searched, or in other words, product uncertainty must be fully resolved before the product is purchased.

Using the Weitzman framework, several researchers have studied how consumers narrow their consideration sets on online platforms and how the tools a platform provides affect this behavior. Chen and Yao (2017) model sequential search with refinement using clickstream data. De los Santos and Koulayev (2017) study click-through under endogenous search refinement. Jiang and Zou (2020) have explored these behaviors theoretically. Koulayev (2014) and Gardete and Hunter (2024) model consumer search behavior but at the level of either an entire webpage or a specific product attribute, respectively, rather than at the product level. Also, researchers have modeled how consumers sequentially expand their consideration sets, a process sometimes referred to as product discovery (Greminger, 2022; Zhang et al., 2023; Fershtman and Pavan, 2024; Greminger, 2025).

As noted above, the literature has largely relied on index-based optimal policies to model consumer search behavior in a tractable manner. This reliance, however, requires the imposition of behavioral assumptions that may not necessarily be applicable in certain contexts. For example, index-based optimal policies including those in Weitzman (1979) and Greminger (2022) and more broadly that in Gittins et al. (2011) require that pulling one arm influence only that arm's state, in the language of multi-armed bandits. In other words, taking one action (e.g., applying a filter) in the MDP affects only the states associated with that action and does not alter the state associated with other actions (e.g., information about products that are also accessible through other filters). Consequently, models based on the Weitzman optimal policy, such as that developed in Chen and Yao (2017), would need to assume that applying a filter and searching a product in that filter does not provide information about products in any other filter. This means that filters cannot overlap and that each product is accessible only through a single filter. This assumption may not be applicable when the filters are based on semantically correlated attributes like price ranges and star ratings.

In addition, most index-based optimal policies assume that a purchased product must have been searched in a given session to fully resolve uncertainty. Allowing purchase without search similarly does not result in a tractable optimal policy. Doval (2018) shows in a general setting that the index policy no longer exists once the decision maker can select an alternative without inspecting it. These common restrictions in the literature are usually made for tractability in estimation rather than for substantive reasons. The implications of such restrictions in a filtering context are unknown, as the issue has not yet been investigated. It is therefore important to develop a model that accommodates a more realistic filtering interface and to quantify what these standard assumptions imply for managerially relevant quantities.

In this paper, we develop a novel filter-assisted sequential search framework that models the filtering, search, and purchase decisions of consumers as a dynamic program. We also develop a Bayesian approach to estimate our model based on inverse reinforcement learning, which infers the preference and cost primitives that rationalize the observed sequences of actions (see Sutton and Barto (2018) for a textbook treatment of reinforcement learning). Specifically, we allow the product subsets returned by different filters to overlap, so that applying a filter can reveal information not only about the products it displays but also about those reachable through other overlapping filters. In addition, we allow consumers to purchase a product without searching it in a given session, by comparing the expected utility of an unsearched product with the realized utilities of the products they have already inspected. We let the cost of searching a product depend on the position in which the active filter displays it, and finally, we specify population distributions for the preference, search cost, and filter cost parameters to accommodate heterogeneity in search and purchase behavior. The resulting dynamic program generalizes the Weitzman framework and other index-based sequential search models.

On the methodological front, the flexibility of our specification comes at a computational cost. The state space of our model is large and partly continuous. Existing estimation approaches for dynamic discrete choice models (see Aguirregabiria and Mira (2010) for a survey) are not suitable for various reasons. The nested fixed point algorithm (Rust, 1987) requires computing the value function at every state. Methods based on conditional choice probabilities (Hotz and Miller, 1993; Hotz et al., 1994; Arcidiacono and Miller, 2011) require the state variables entering the choice probabilities to be observed, or when unobserved, to be low-dimensional and finite for computational reasons. In contrast, they are latent, multidimensional, and continuous in our case. The kernel-based approach developed by Imai et al. (2009) requires tracking approximate value function values for all discrete state variables and may not provide reliable approximation due to the curse of dimensionality unless the number of structural parameters and continuous-valued state variables is small. We therefore develop a Bayesian MCMC method that approximates the integrated value function of the MDP with a neural network trained at the beginning of each iteration during an initialization period.

Rather than supplying raw state variables to a generic feed-forward network or their ad hoc, simplified representation, we exploit our model structure and compute Deep Sets-based embeddings (Zaheer et al., 2017) of the product- and filter-level state variables. These embeddings are invariant to the ordering of the products and the filters. Imposing this invariance through the architecture, rather than letting the neural network learn it during training, can improve approximation quality for a given training budget. A generic network would have to discover this symmetry from potentially many permuted examples of the same state, spending network capacity and training samples that could otherwise be utilized to approximate the value function itself. Our MCMC sampler combines data augmentation, Gibbs steps, the independent Metropolis algorithm, and elliptical slice sampling (Murray et al., 2010) to draw the latent state variables and the model parameters. A few papers have used neural networks for value function approximation in dynamic discrete choice settings (Norets, 2012; Hodgson and Lewis, 2025; Kang et al., 2026). Hodgson and Lewis (2025) is closest in spirit to our approach, although it uses a plain feed-forward network trained once at the beginning of a maximum a posteriori estimation procedure, whereas we retrain a structured network within each MCMC iteration during an initialization period to adequately target the full posterior distribution. We empirically confirm via a Monte Carlo simulation that the structural parameters of the proposed model are recovered well using our estimation method.

We apply our model to clickstream data from an online store that sells Japanese incense, covering 7,706 sessions by 5,352 registered customers between September 2023 and August 2025. The store offers seven attribute-based filters alongside the unfiltered main page. Four of these filters define price ranges and three correspond to non-price attributes, and these filters overlap substantially. Our estimates show that the row position at which a product is displayed matters for search. The search cost increases by roughly 20% from the first to the second row, by 12% from the second to the third, and by 9% from the third to the fourth, which is consistent with the ranking effects documented by Ursu (2018). Applying a filter is estimated to be slightly less costly on average than searching a product. We also find that the variance of the post-search shocks is substantially larger than that of the pre-search shocks, which implies that the idiosyncratic information on a product description page is considerably more consequential for consumer decisions than what is visible on a listing page, net of what is observed by the researcher.

We use two counterfactual analyses to examine how the filtering mechanism shapes consumer behavior. When we consolidate the four price filters into two, our counterfactual analysis reveals that consumers would use the price filters only 44% of the time, down from more than half in the observed scenario, and the non-price filters absorb the difference as their usage rises from 35% to 42%. Consumers would rely more on non-price attributes to narrow their consideration sets under the coarser price structure. In the second counterfactual, we prohibit purchase without prior search by not allowing consumers to put a product directly in the cart, thereby imposing the assumption that the search literature almost invariably maintains. The probability of purchasing an inside good falls from 33% to 26% in this scenario. Maintaining the standard assumption would therefore understate demand by about seven percentage points in this context, which is a consequence a retailer evaluating its filtering interface would want to know about.

In summary, we make several contributions to the search literature. First, we develop a novel filter-assisted sequential search model that accommodates overlapping filters and purchase without prior search. Second, we develop a Bayesian estimation scheme for this model that uses a custom neural network architecture to approximate the value function within an MCMC sampler. This makes inverse reinforcement learning feasible for a search model whose state space is far larger than what existing methods can handle. Finally, we empirically show that the modeling restrictions the literature imposes for tractability can be consequential.

The rest of this paper is organized as follows. In the next section, we develop our model of filter-assisted search. We then describe our estimation approach and the neural value function approximation in Section 3. Section 4 presents the empirical application, the parameter estimates, and the counterfactual analyses. We conclude by discussing the limitations of our framework and by highlighting directions for further research in Section 5.

2 Model

2.1 Model Overview

We model the sequential search and purchase decisions of consumers where the search process involves the use of product filters. This setup corresponds to a shopping environment in which consumers can use filters to narrow the set of products that can be viewed and then inspect particular products from this set.

A product filter imposes a constraint on the values of one or more product attributes (such as products in a given price range, or products with a specific brand name and in a specific price range). We assume that viewing the products within this subset provides partial information to consumers. For example, from the list page returned by the filter, the consumer may observe basic attributes such as price and may infer that a product satisfies the constraint defining the active filter. However, the consumer must click on an individual product to obtain additional information such as the product's specifications, images, or reviews. At each step of this sequential process, the consumer may continue searching within the same subset, switch to a different filter, or terminate the session by either purchasing a single product or choosing the outside option.

Formally, we consider an e-commerce setting where the online store offers $M + 1$ product filters. Let $\mathcal{J}_m$ be the set of products displayed when filter $m \in \mathcal{M} = \{0, 1, \dots, M\}$ is used. Here, $m = 0$ corresponds to a "special" filter that returns the entire set of products $\mathcal{J} = \{1, \dots, J\}$, so that $\mathcal{J}_0 = \mathcal{J}$. This can be seen as a "no-filter" option. Our modeling setup allows for overlapping product filters, which distinguishes our setting from existing models of sequential search in the literature. This is consistent with how product filters are designed in most e-commerce settings. For the sequential search process, this means that when a consumer views products under one filter, the consumer may also know about products that appear under other filters. For example, a filter for expensive products may overlap with a filter based on a premium attribute that is present only among high-end products. Filter use therefore can reveal information not only about the currently displayed subset but also about products reachable through other filters. As we will discuss later, accommodating such overlapping filters in a model will violate a crucial assumption necessary for the use of tractable, index-based optimal policies, including the Weitzman policy.

Consumer $i \in \{1, \dots, n\}$ engages in one or more search sessions that are indexed by $t \in \{1, \dots, T_i\}$. A search session t can involve a number of sequential steps indexed by $h \in \{1, \dots, H_{it}\}$. We model the consumer's behavior within a session as a Markov decision process (MDP). Given the consumer's state at the beginning of step h, the consumer optimally chooses an action from a finite set of available actions to maximize discounted expected utility over the steps within the session. Consumer actions can be one of three types: applying a filter, searching a product, or purchasing an alternative. Applying a filter means selecting a filter and viewing the subset of products displayed under that filter. Searching a product involves clicking on a product that is listed by a filter to inspect it in greater detail. Searching and filtering are non-terminal actions in the MDP. Upon making a non-terminal action, the session continues to the next step $h + 1$, in which the consumer takes another action in a new state. The session terminates with the consumer choosing either a single alternative j from a subset of the J inside goods or an outside option indexed by $j = 0$.

Consistent with literature on sequential search, we assume that consumers are forward-looking over future steps within a session, but their activities are independent across sessions, given their preference and cost parameters. Thus, on a given shopping occasion, consumers take into account the future consequences of filtering, search, and purchase decisions that unfold only within that session. Table 1 illustrates different types of consumer sessions. We show actions for at most the first five steps for each session. Different letters denote different action types. The first row of the table shows that consumer $i = 1$ in session $t = 1$ applies a filter (denoted by F) in the first step. This can be the no filter option or one of the available attribute-based filters. Then, the consumer searches (denoted by S) sequentially two different products under that filter in steps 2 and 3. In Step 4, the consumer selects and applies another filter. Upon seeing a list of products under the new filter, the consumer makes a purchase (P) in step 5 and terminates the session. For this session, the consumer purchased the product that was searched in Step 3. We indicate this by using a bold **S** in Step 3. We see from the second row of the table that the same consumer begins a new session (e.g., on a different day) by selecting a filter in Step 1 but follows this by applying another filter in Step 2 without searching or purchasing any of the products returned in Step 1. The consumer searches a product in the second filter in Step 3 and then chooses an outside alternative (no purchase) in Step 4 to leave the session.

The other sessions in Table 1 show other interesting use cases. For example, the third session for the first consumer involves the selection of the outside alternative without any search and filtering actions. Session 1 of the second consumer is similar to the first session of the first consumer, except that this session ends with a purchase of an unsearched product from among the products returned by the two filters used by the consumer. This can be inferred from the fact that neither of the two S entries in the fourth row is bold.

We now describe the different components of the Markov decision process. These include the flow utility from purchasing an alternative, the flow cost of filtering and searching, the variables that describe the state space, the set of actions that are available at each step, the transition process between successive states, as well as the initial conditions for the decision problem. We then specify the value functions resulting from the MDP. We begin with the alternative-specific flow utilities.

Table 1: Timeline: an illustrative example

Consumer (i)	Session (t)	Step $h=1$	Step 2	Step 3	Step 4	Step 5	$\cdots$
1	1	F	S	**S**	F	P	
1	2	F	F	S	O		
1	3	O					
2	1	F	S	S	F	P	
3	1	F	P				
3	2	F	**S**	F	F	S	$\cdots$

F: apply a filter. S: search a product. P: purchase a product. O: choose an outside option.
Bold **S** corresponds to searching a product that the consumer eventually purchases in a given session.

2.2 Utility Specification

Consumer i receives the following (flow) utility from purchasing alternative j in session t:

$$u_{ijt} = \begin{cases} \underbrace{(\boldsymbol{x}'_{ijt}\boldsymbol{\beta}_i + \eta_{ijt})}_{\text{pre-search}} + \underbrace{(\boldsymbol{w}'_{ijt}\boldsymbol{\zeta}_i + \varepsilon_{ijt})}_{\text{post-search}} & \text{if } j \in \mathcal{J}, \\ \eta_{i0t} & \text{if } j = 0, \end{cases} \tag{1}$$

where the outside option is indexed by $j = 0$. The vector $\boldsymbol{x}_{ijt}$ includes the attributes for product j that are known to the consumer *prior* to search, whereas the vector $\boldsymbol{w}_{ijt}$ contains product attributes that are unknown to the consumer until they search the product (Choi and Mela, 2019). The researcher is assumed to observe both types of attributes in the data. Let $\eta_{ijt} \sim \mathrm{N}(0, \sigma^2_\eta)$ be a pre-search random shock for alternative $j \in \mathcal{J} \cup \{0\}$ and $\varepsilon_{ijt} \sim \mathrm{N}(0, \sigma^2_\varepsilon)$ be a post-search shock for product $j \in \mathcal{J}$. Consumers observe η_{ijt} prior to search while ε_{ijt} is observable only after searching the corresponding product. These random shocks are assumed to vary independently of each other. The researcher does not observe either of these random shocks. For notational clarity, we define a pre-search (flow) utility as $\delta_{ijt} = \boldsymbol{x}'_{ijt}\boldsymbol{\beta}_i + \eta_{ijt}$ if $j \in \mathcal{J}$ and $\delta_{ijt} = \eta_{i0t}$ if $j = 0$.

Consumers have rational expectations and we assume that, prior to applying any filter, they know $(\{(\mathcal{J}_m, F^{\boldsymbol{x}}_m)\}_{m\in\mathcal{M}}, F^{\boldsymbol{w}}, u_{i0t})$. Here, $F^{\boldsymbol{x}}_m$ is the consumer's belief distribution of pre-search product attributes $\boldsymbol{x}_{ijt}$ for the subset of products returned by filter m. The belief distribution $F^{\boldsymbol{w}}$, which is common across products, captures consumer uncertainty about the post-search product attributes. We assume that the post-search attributes, $\boldsymbol{w}_{ijt}$, are independent of the pre-search attributes $\boldsymbol{x}_{ijt}$ and of the pre- and post-search shocks. This is similar to assuming that the distribution of the post-search shocks ε_{ijt} is independent of other variables. Upon applying a filter m in step h, consumer i knows the values of $(\boldsymbol{x}_{ijt}, \eta_{ijt})$ for all $j \in \mathcal{J}_m$ for all subsequent steps $h' > h$, in session t.

2.3 Cost Specification

We now describe the cost specifications in our model. Let $\chi_i > 0$ denote the cost that consumer i incurs each time a filter is used. In addition, consumer i incurs a search cost c_{ir} when searching a product. We specify c_{ir} to be a position-dependent cost and allow it to vary with the *row-position* r in which the searched product is displayed under a given filter. To be specific, consumer i incurs search cost $c_{i,r(j,m)}$ when the consumer searches product j under filter m, where $r(j,m)$ maps a pair of product j and filter m to the row-position r in which the product is displayed under filter m. For the function $r(\cdot,\cdot)$ to be well-defined, we endow the set $\mathcal{J}_m$ with an order to indicate in which row $r \in \{1,\dots,R\}$ on a webpage each product is displayed under filter m, for each $m \in \mathcal{M}$. Thus, filters not only return specific subsets of products but also sort the products in a particular order. In our empirical application, products are displayed in a grid format. We focus only on the row-position and do not take into account the column positions of the products.[1]

2.4 State Variables

The state Ω_{ith} at the beginning of step h within session t for consumer i is defined as

$$\Omega_{ith} = \left(S_{ith}, \tilde{u}_{ith}, \tilde{j}_{ith}, m_{ith}, \{\delta_{ijt}\}_{j \in \mathcal{J}_{ith}}, \mathcal{J}_{ith}\right). \tag{2}$$

The different components of the state can be described as follows:

- S_{ith} is the set of products consumer i has already searched until the beginning of step h in the session,
- $\tilde{u}_{ith}$ is the maximum utility (as in Equation (1)) among the searched products and the outside option,
- $\tilde{j}_{ith}$ is an index that corresponds to the alternative with the highest utility, i.e., the alternative associated with $\tilde{u}_{ith}$,
- $m_{ith} \in \mathcal{M}$ is the currently active filter at the beginning of step h. This means that consumer i is viewing products displayed under that filter. If $m_{ith} = m$, then, in step $h-1$, the consumer either

[1] It is straightforward to extend our modeling framework to accommodate both the row and column positions as well as a list format, in which only a single product is displayed in each row.

selected filter m or searched a product under $m_{i,t,h-1} = m$,

- $\mathcal{J}_{ith}$ is the set of all the products that have been shown to the consumer until step h in session t. Thus $\mathcal{J}_{ith} = \bigcup_{h'=2}^{h} \mathcal{J}_{m_{ith'}}$, if $h \geq 2$ and is $\emptyset$, if $h = 1$, and
- the set $\{\delta_{ijt}\}_{j \in \mathcal{J}_{ith}}$ contains the pre-search utilities of the products in $\mathcal{J}_{ith}$. Each δ_{ijt} is constructed based on the realizations of $\{(\boldsymbol{x}_{ijt}, \eta_{ijt})\}_{j \in \mathcal{J}_{ith}}$ and consumer i's preference parameter $\boldsymbol{\beta}_i$ for the pre-search attributes.

We now define a few variables that are constructed from the state variables to facilitate exposition. For $j \in \mathcal{J}_{ith}$, let $\overline{u}_{ijt}$ be the mean utility conditional on the pre-search utility δ_{ijt} such that

$$\overline{u}_{ijt} = \mathbb{E}[u_{ijt} \mid \delta_{ijt}] = \delta_{ijt} + \mathbb{E}[\boldsymbol{w}'_{ijt}\boldsymbol{\zeta}_i + \varepsilon_{ijt}] = \delta_{ijt} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i,$$

where $\overline{\boldsymbol{w}} = \mathbb{E}[\boldsymbol{w}_{ijt}]$. This expectation is with respect to $F^{\boldsymbol{w}}$. We also define

$$u^*_{ith} = \max\left\{\tilde{u}_{ith}, \max_{j \in \mathcal{J}_{ith} \setminus S_{ith}} \{\overline{u}_{ijt}\}\right\}, \tag{3}$$

such that it is the maximum of $\tilde{u}_{ith}$, (i.e., the maximum utility among the searched products and the outside option) and the maximum mean utility among the unsearched products. Finally, we define j^*_{ith} as the index pertaining to the alternative associated with u^*_{ith}, i.e.,

$$j^*_{ith} = \begin{cases} \tilde{j}_{ith} & \text{if } \tilde{u}_{ith} > \max_{j \in \mathcal{J}_{ith} \setminus S_{ith}} \{\overline{u}_{ijt}\} \\ \arg\max_{j \in \mathcal{J}_{ith} \setminus S_{ith}} \{\overline{u}_{ijt}\} & \text{otherwise} \end{cases}.$$

2.5 Consumer Actions

Recall that at any step of the sequential search process a consumer can apply a filter, search a product, or purchase an alternative. More formally, the consumer takes a single action $a_{ith} \in \mathcal{A}_{ith}$, where $\mathcal{A}_{ith}$ is a set of feasible actions defined as:

$$\mathcal{A}_{ith} = \left(\bigcup_{j \in \{j^*_{ith}\}} \{\text{purchase alternative } j\}\right) \cup \left(\bigcup_{j \in \mathcal{J}_{m_{ith}} \setminus S_{ith}} \{\text{search product } j\}\right) \cup \left(\bigcup_{m \in \mathcal{M} \setminus \{m_{ith}\}} \{\text{apply filter } m\}\right). \tag{4}$$

The expression inside the first pair of parentheses indicates that consumers can purchase only the alternative with the highest expected utility u^*_{ith}, defined in Equation (3), if they decide to purchase. The second pair of parentheses implies that when they search, consumers can only do so for the products that have not been searched yet within a session and are displayed under the currently selected filter. These search actions do not alter the filter at the beginning of the subsequent step, i.e., $m_{i,t,h+1} = m_{ith}$. The third pair of parentheses specifies that consumers can only apply a filter that differs from the currently active one. Finally, we define $\mathcal{A}$ to be the set of all actions (some of which may be infeasible in a given step), i.e.,

$$\mathcal{A} = \left(\bigcup_{j \in \mathcal{J} \cup \{0\}} \{\text{purchase alternative } j\} \right) \cup \left(\bigcup_{j \in \mathcal{J}} \{\text{search product } j\} \right) \cup \left(\bigcup_{m \in \mathcal{M}} \{\text{apply filter } m\} \right).$$

The set of all feasible actions $\mathcal{A}_{ith}$ at any given state is always a subset of $\mathcal{A}$ and as $\mathcal{A}$ is of finite cardinality, only a finite number of actions are available at any given state.

2.6 Transition Mechanism

For a consumer in state Ω_{ith}, a purchase in step h (i.e., $a_{ith} = \text{purchase } j^*_{ith}$) results in the end of the session. For non-terminating actions $a_{ith} \in \mathcal{A}_{ith} \setminus \{\text{purchase } j^*_{ith}\}$, the state evolves as follows:

$$\Omega_{i,t,h+1} \mid [\Omega_{ith}, a_{ith}] = \left(S_{i,t,h+1}, \tilde{u}_{i,t,h+1}, \tilde{j}_{i,t,h+1}, m_{i,t,h+1}, \{\delta_{ijt}\}_{j \in \mathcal{J}_{i,t,h+1}}, \mathcal{J}_{i,t,h+1} \right), \tag{5}$$

where

$$S_{i,t,h+1} = \begin{cases} S_{ith} \cup \{j\} & \text{if } a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ S_{ith} & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases}$$

$$\tilde{u}_{i,t,h+1} = \begin{cases} \max\{\tilde{u}_{ith}, u_{ijt}\} & \text{if } a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ \tilde{u}_{ith} & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases}$$

$$\tilde{j}_{i,t,h+1} = \begin{cases} j & \text{if } [a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith}] \wedge [u_{ijt} \geq \tilde{u}_{ith}] \\ \tilde{j}_{ith} & \text{if } [a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith}] \wedge [u_{ijt} < \tilde{u}_{ith}] \\ \tilde{j}_{ith} & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases}$$

$$m_{i,t,h+1} = \begin{cases} m_{ith} & \text{if } a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ m & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases}$$

$$\{\delta_{ij't}\}_{j'\in\mathcal{J}_{i,t,h+1}} = \begin{cases} \{\delta_{ij't}\}_{j'\in\mathcal{J}_{ith}} & \text{if } a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ \{\delta_{ij't}\}_{j'\in\mathcal{J}_{ith}} \cup \{\delta_{ij't}\}_{j'\in\mathcal{J}_m\setminus\mathcal{J}_{ith}} & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases}$$

$$\mathcal{J}_{i,t,h+1} = \begin{cases} \mathcal{J}_{ith} & \text{if } a_{ith} = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ \mathcal{J}_{ith} \cup (\mathcal{J}_m \setminus \mathcal{J}_{ith}) = \mathcal{J}_{ith} \cup \mathcal{J}_m & \text{if } a_{ith} = \text{apply } m \in \mathcal{M} \setminus \{m_{ith}\}. \end{cases}$$

The transition for each variable depends upon the current state of the consumer and the action taken in that state. When the consumer selects filter m in step h, pre-search product attributes $\boldsymbol{x}_{ijt} \sim F_m^{\boldsymbol{x}}$ and pre-search shocks $\eta_{ijt} \sim \mathrm{N}(0, \sigma_\eta^2)$ are revealed for the products $j \in \mathcal{J}_m \setminus \mathcal{J}_{ith}$. These are the products displayed by filter m that the consumer has not seen previously in the current session t. The pre-search utility for those products is then constructed based on these product attributes and pre-search shocks so that $\delta_{ijt} = \boldsymbol{x}'_{ijt}\boldsymbol{\beta}_i + \eta_{ijt}$. The post-search attributes $\boldsymbol{w}_{ijt} \sim F^{\boldsymbol{w}}$ and the post-search shock $\varepsilon_{ijt} \sim \mathrm{N}(0, \sigma_\varepsilon^2)$ are revealed to the consumer upon searching product j, and the utility $u_{ijt} = \delta_{ijt} + \boldsymbol{w}'_{ijt}\boldsymbol{\zeta}_i + \varepsilon_{ijt}$ becomes known.

The transitions of S_{ith}, m_{ith}, and $\mathcal{J}_{ith}$ are deterministic given the current state and action. Those of $\tilde{u}_{ith}$, $\tilde{j}_{ith}$, and $\{\delta_{ij't}\}_{j'\in\mathcal{J}_{ith}}$ are stochastic, although the transition distributions can be degenerate, depending on the current state and action. For example, when $\mathcal{J}_m \setminus \mathcal{J}_{ith} = \emptyset$, the consumer does not see any new products when filter m is applied and therefore, the transitions of all state variables are deterministic. The state variables $\{\delta_{ij't}\}_{j'\in\mathcal{J}_{ith}}$ do not change when the consumer searches a product within the current filter. Similarly, $\tilde{u}_{ith}$ and $\tilde{j}_{ith}$ do not change when the consumer's action corresponds to the selection of a filter.

2.7 Initial Conditions within Each Session

In the initial step, $h = 1$, consumers cannot search a product directly as no filter is active in the beginning of a session. Therefore, consumers can either apply a filter or choose the outside option. The set of actions available in the initial step is then given by

$$\mathcal{A}_{it1} = \{\text{choose an outside option } j = 0\} \cup \left(\bigcup_{m\in\mathcal{M}} \{\text{apply filter } m\} \right).$$

This is consistent with our definition of the generic action set in Equation (4). The initial state is defined as

$$\Omega_{it1} = \left(S_{it1}, \tilde{u}_{it1}, \tilde{j}_{it1}, m_{it1}, \{\delta_{ijt}\}_{j\in\mathcal{J}_{it1}}, \mathcal{J}_{it1}\right),$$

where $S_{it1} = \emptyset$, $\tilde{u}_{it1} = u_{i0t} = \eta_{i0t}$, and $\tilde{j}_{it1} = 0$. We set $m_{it1} = \texttt{null}$ to indicate that there is no currently active filter in the first step and the consumer can apply any filter $m \in \mathcal{M} \setminus \{\texttt{null}\} = \mathcal{M}$. Also, $\{\delta_{ijt}\}_{j \in \mathcal{J}_{it1}} = \emptyset$, because $\mathcal{J}_{it1} = \emptyset$. Given the above, we can continue to use the definition of the state in Equation (2) and the transition mechanisms in Equation (5) for all steps h, including $h = 1$.

2.8 Value Functions

Consistent with the literature in dynamic discrete choice models (Rust, 1987; Aguirregabiria and Mira, 2010), we use idiosyncratic shocks $\boldsymbol{e}_{ith}$ to smooth out the likelihood. The ex-post value function after these shocks are realized, $\tilde{V}(\Omega_{ith}, \boldsymbol{e}_{ith})$, can be written as

$$
\begin{aligned}
\tilde{V}(\Omega_{ith}, \boldsymbol{e}_{ith}) &= \max\{\text{purchase, search, apply filter}\} \\
&= \max\Bigg\{ u^*_{ith} + e^{(1)}_{ith}, \\
&\qquad \max_{j \in \mathcal{J}_{m_{ith}} \setminus S_{ith}} \left\{ -c_{i,r(j,m_{ith})} + e^{(2)}_{ithj} + \kappa \cdot \mathbb{E}[V(\Omega_{i,t,h+1}) \mid \Omega_{ith}, a_{ith} = \text{search product } j] \right\}, \\
&\qquad \max_{m \in \mathcal{M} \setminus \{m_{ith}\}} \left\{ -\chi_i + e^{(3)}_{ithm} + \kappa \cdot \mathbb{E}[V(\Omega_{i,t,h+1}) \mid \Omega_{ith}, a_{ith} = \text{apply filter } m] \right\} \Bigg\},
\end{aligned} \tag{6}
$$

where $\kappa \in (0, 1)$ is a discount factor. We set $\kappa = 0.99$ in our application. We assume that each element of the idiosyncratic shocks,

$$
\boldsymbol{e}_{ith} = \left(e^{(1)}_{ith}, \left\{ e^{(2)}_{ithj} \right\}_{j \in \mathcal{J}_{m_{ith}} \setminus S_{ith}}, \left\{ e^{(3)}_{ithm} \right\}_{m \in \mathcal{M} \setminus \{m_{ith}\}} \right),
$$

is independently and identically distributed according to the type-1 extreme value distribution. Then, the integrated (ex-ante) value function $V(\Omega_{ith})$ is given by

$$
V(\Omega_{ith}) = \mathbb{E}_{\boldsymbol{e}_{ith}} \left[\tilde{V}(\Omega_{ith}, \boldsymbol{e}_{ith}) \right] = \gamma_{\mathtt{E}} + \log \left(\sum_{a \in \mathcal{A}_{ith}} \exp(V_a(\Omega_{ith})) \right), \tag{7}
$$

where $\gamma_{\mathtt{E}}$ is Euler's constant. $V_a(\Omega_{ith})$ is an action-specific value function for action a that is defined as

$$V_a(\Omega_{ith}) = \begin{cases} u^*_{ith} & \text{if } a = \text{purchase } j^*_{ith} \\ -c_{i,r(j,m_{ith})} + \kappa \cdot \mathbb{E}[V(\Omega_{i,t,h+1}) \mid \Omega_{ith}, a = \text{search product } j] & \text{if } a = \text{search } j \in \mathcal{J}_{m_{ith}} \setminus S_{ith} \\ -\chi_i + \kappa \cdot \mathbb{E}[V(\Omega_{i,t,h+1}) \mid \Omega_{ith}, a = \text{apply filter } m] & \text{if } a = \text{apply filter } m \in \mathcal{M} \setminus \{m_{ith}\} \end{cases} \tag{8}$$

for each action $a \in \mathcal{A}_{ith}$.

Our model specification differs from the majority of sequential search models in the marketing literature as it allows consumers to purchase a product even without searching it.[2] Consumers in our model may do so when they do not need to resolve all uncertainty about the product's characteristics or when their search costs do not justify additional search. This can be seen by looking at how u^*_{ith} (as defined in Equation (3)) appears in the Bellman equation in Equation (6). Consumers can purchase an unsearched product by evaluating its expected utility if this is higher than the maximum of the realized utilities among the searched products, where the expectation is with respect to the post-search attributes and post-search shocks. Thus our modeling framework deviates from Weitzman-based sequential search models.

Our model does not yield a simple index policy as in Weitzman (1979) or Greminger (2022) because we allow for overlapping filters and for product purchase without search. Crucially, overlapping filters imply that a single filtering action can alter the states for multiple filters. The same restriction on how actions influence states (the independence assumption) is also crucial in Greminger (2022), and more broadly, for obtaining the Gittins index policy (Gittins et al., 2011).

Finally, the conditional choice probability, i.e., the probability of taking a particular action given the current state, calculated before the idiosyncratic shocks are realized, can be written in terms of the alternative-specific value functions as

$$\Pr(a_{ith} = a \mid \Omega_{ith}) = \frac{\exp(V_a(\Omega_{ith}))}{\sum_{a' \in \mathcal{A}_{ith}} \exp(V_{a'}(\Omega_{ith}))}. \tag{9}$$

2.9 Customer Heterogeneity

Thus far, we have focused on the within-session dynamics, conditional on the individual-level preference and cost parameters that are common across all the sessions belonging to a customer. We now specify how

[2] For example, in e-commerce settings, consumers can purchase a product by adding it directly to the cart, without opening its product description page.

these parameters vary across customers via population distributions that characterize heterogeneity.

Let $\boldsymbol{\xi}_i$ denote a transformed vector of the preference parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\zeta}_i$ in Equation (1) and assume that it follows a normal distribution,

$$\boldsymbol{\xi}_i \sim \mathrm{N}(\boldsymbol{\mu}, \Sigma), \tag{10}$$

where $\boldsymbol{\mu}$ is the population mean vector and Σ is the population variance matrix. We assume that the transformation from vector $[\boldsymbol{\beta}_i', \boldsymbol{\zeta}_i']'$ to vector $\boldsymbol{\xi}_i$ is a known, deterministic, and element-wise bijection. This assumption accommodates a wide range of empirical contexts. For example, a sign constraint on a few elements of $\boldsymbol{\beta}_i$ and $\boldsymbol{\zeta}_i$ can be easily imposed to accommodate a negative price coefficient. The bijection can be set to identity when the researcher does not need to impose constraints on the support of $\boldsymbol{\beta}_i$ or $\boldsymbol{\zeta}_i$.

The search cost heterogeneity for each row position $r \in \{1, \ldots, R\}$ is modeled as

$$\log c_{ir} \sim \mathrm{N}\left(\boldsymbol{z}_{ir}'\boldsymbol{\gamma}, \sigma_c^2\right), \tag{11}$$

where the vector $\boldsymbol{z}_{ir}$ contains any applicable covariates such as the row position r itself. In our application, we set it to $\boldsymbol{z}_{ir} = [1, \log r]'$, and $\boldsymbol{\gamma} = [\gamma_0, \gamma_1]'$ is the associated parameter. Thus, γ_0 captures the baseline search cost that consumers incur when they search a product, and γ_1 measures the row position effect on the search cost. The filtering cost heterogeneity is specified as follows:

$$\log \chi_i \sim \mathrm{N}(\lambda, \sigma_\chi^2). \tag{12}$$

Figure 1 shows the directed acyclic graph for our proposed model, which clarifies the conditional independence assumptions that characterize our specification.

2.10 Identification

We rely on the sequential search literature (Ursu et al., 2025), and especially studies that use panel data (Morozov et al., 2021; Onzo and Ansari, 2025) to describe qualitatively how our model parameters are identified from different sources of variation in the data.

The preference parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\zeta}_i$ are identified from variations in the pre- and post-search product attributes $(\boldsymbol{x}_{ijt}, \boldsymbol{w}_{ijt})$ and the panel data structure. Note that the preference parameters influence all three actions—which filters to use, which products to search, and which product to purchase. This interlink-

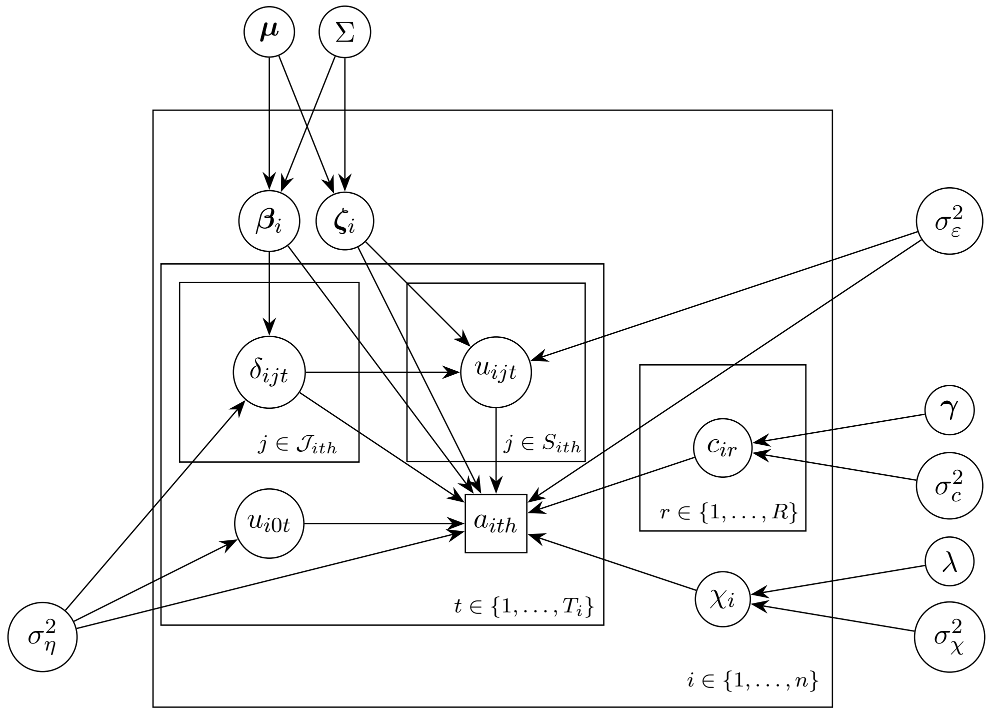

Figure 1: Directed acyclic graph for the proposed model

Note: Unobserved quantities are shown in circles, and observed ones using squares. Observed, exogenous variables (e.g., $\boldsymbol{x}_{ijt}$) are not shown to reduce visual clutter.

age among decisions is a key aspect of our model—when making purchase or search decisions, consumers consider the realized pre-search attributes, and when selecting a filter, they utilize the distribution of the pre-search attributes for the products they have not yet encountered in a given session. In addition, they take into account the post-search attributes when they are about to purchase a searched product and consider their distribution for the products they have not searched in a given session. As these attributes are weighted by $\boldsymbol{\beta}_i$ and $\boldsymbol{\zeta}_i$, they inform all three decisions. The population mean $\boldsymbol{\mu}$ and covariance Σ for the preference parameters that characterize preference heterogeneity are estimated based on the identification of the consumer-level preference parameters.

The number of times consumers apply filters is a key input for estimating the filtering cost parameter χ_i as well as the parameters λ and σ_χ^2 in its heterogeneity distribution. Similarly, differences in the extent to which consumers search products that appear in different row positions identify the search cost related parameters c_{ir}, as well as $\boldsymbol{\gamma}$ and σ_c^2. In particular, how much the row position of a product influences the frequency with which the product is searched helps determine the relevant parameter in $\boldsymbol{\gamma}$.

Having identified the preference parameters, we can utilize purchase data from consumers with a sufficiently low search cost to pin down the variance of the post-search shocks, as argued in Onzo and Ansari (2025). The variance of the pre-search shocks is informed by the assumption that when making a filter selection decision consumers rely on the expectation of the pre-search utility δ_{ijt} for products they have not seen. This is possible because the (empirical) distribution of the product attributes is observed in the data.

3 Estimation

We develop a novel Bayesian approach for model estimation. We outline our estimation methodology here and provide additional details in Appendix A. The full conditionals that are used in our Markov chain Monte Carlo (MCMC) scheme are described in Appendix W.A. We provide an overview of the sampling scheme, describe our neural value function approximation, outline the data inputs for estimation, and report a small simulation study that verifies the validity of our method.

3.1 Posterior Sampling

We build upon the Bayesian estimation frameworks for full-information static discrete choice models (Albert and Chib, 1993; McCulloch and Rossi, 1994; McCulloch et al., 2000) and Weitzman-based sequential search models (Morozov, 2023; Onzo and Ansari, 2025) to develop an MCMC scheme for our model framework. Our approach combines data augmentation, the independent Metropolis algorithm, elliptical slice sampling (Murray et al., 2010), and Gibbs sampling steps to sample the posterior draws of the unknown quantities and latent variables collected in

$$\Upsilon = \left(\{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \boldsymbol{\mu}, \Sigma^{-1}, \boldsymbol{\gamma}, \sigma_c^2, \lambda, \sigma_\chi^2, \sigma_\varepsilon^2, \sigma_\eta^2\right). \tag{13}$$

The entities in Equation (13) consist of global- and consumer-level parameters we wish to interpret as well as auxiliary variables such as the flow utilities. Draws for the individual-level preference and cost related parameters allow us to infer the preferences of specific consumers, and the draws of the population-level parameters provide insights about the individual differences that exist in the population. As our model captures consumers' forward-looking behavior across the different steps within a session, many of the full conditional distributions require the evaluation of the probability of taking an action given a state as in

Equation (9). There are two major challenges in evaluating these probabilities.

First, some of our state variables are latent and therefore are not observed in the data. Moreover, these are multidimensional and continuous variables. We therefore use data augmentation to infer these unobserved state variables from augmented flow utilities and pre-search flow utilities as part of our Bayesian estimation scheme. Augmenting these variables also allows us to maintain conditional conjugacy of the full conditionals for most of the model parameters. We do not have closed-form full conditional distributions for some of the individual-level preference parameters because of constraints on their support, as is the case for negative price coefficients in $\boldsymbol{\beta}_i$ or ζ_i. We handle such non-conjugacy by using elliptical slice sampling (Murray et al., 2010), which is a tuning-free sampling algorithm unlike other common algorithms including the random walk Metropolis-Hastings. Such a tuning-free method is very useful in complex hierarchical settings such as ours.

Evaluating the probability expression in Equation (9) is also challenging because it requires the computation of the action-specific value functions defined in Equation (8) for arbitrary actions and states, or more specifically, the (integrated) value function in Equation (7). We address this challenge by utilizing a neural network to approximate the integrated value function. We develop a custom neural network architecture that is specifically designed for our modeling task, instead of relying on a generic feed-forward neural network architecture as such customization can improve the quality of the value function approximation.

We train a neural network using a random set of state variables and model parameters at the beginning of each MCMC iteration. This batch is drawn from a distribution that is centered at the current MCMC draws of the relevant augmented latent variables and parameters. This choice reflects the fact that we typically need value function estimates only in a neighborhood of the current MCMC draws in each iteration. Once the neural network training is complete in a given MCMC iteration, we sample from the full conditionals of the latent variables and parameters. We now describe our specific approach for approximating the value function and discuss its suitability for our modeling framework.

3.2 Neural Value Function Approximation

A variety of approaches have been used to estimate dynamic discrete choice models. Rust (1987) developed the nested fixed point (NFXP) algorithm for maximum likelihood estimation when the state space is not too large. In this approach, the value function for each state is computed in an inner loop and the structural parameters are updated in an outer loop of the optimization algorithm. Imai et al. (2009) developed a

Bayesian estimation approach that relies on a kernel-based approximation of the value function within each MCMC iteration. Another stream of literature uses estimation methods that utilize the conditional choice probabilities (CCP) (Hotz and Miller, 1993; Hotz et al., 1994; Arcidiacono and Miller, 2011). Such an approach does not require the computation of the value function.

We cannot use these estimation methods in our context for multiple reasons. It is infeasible to use the NFXP algorithm as the state space of our model is large (note that some state variables are continuous-valued and thus the cardinality of the state space is infinite). CCP-based estimation is also impractical as some of our state variables, e.g., the flow utilities, are not observed in the data. While Arcidiacono and Miller (2011) proposed a version of the EM algorithm to handle such unobserved variables that enter the CCP, the method is not ideal when the latent variables are continuous and multidimensional, as in our case. The Bayesian approach in Imai et al. (2009) is also not suitable as it requires keeping track of approximate value function values for all discrete state variables. Such accounting is computationally infeasible as the discrete state variables in our model take a very large number of values. For instance, S_{ith} (the set of products consumer i has searched till step h) can take 2^J distinct values. This number grows exponentially with the number of products. The method can also struggle unless the number of structural parameters and continuous-valued state variables is small. This occurs because it approximates the value function using a kernel method whose approximation quality can deteriorate in high dimensional cases (Stone, 1982).

Given the above, we use a neural network to approximate the value function. As mentioned previously, we use a custom neural network architecture that leverages our model structure to approximate the integrated value function. A few existing papers have used generic neural network architectures for value function approximation (Norets, 2012; Hodgson and Lewis, 2025; Kang et al., 2026). Of these, Hodgson and Lewis (2025) is the closest in spirit to our approach. However, it uses a plain feed-forward neural network and trains it once at the beginning of its maximum a posteriori parameter estimation procedure for approximate Bayesian inference.

Specifically, we approximate the ex-ante value function $V(\Omega_{ith})$ in Equation (7) at an arbitrary state Ω_{ith} by a neural network $\hat{V}_{\psi}(\Omega_{ith})$ that is parameterized by trainable neural network parameters in ψ. It is helpful to denote the ex-ante value function as $V(\Omega_{ith};\boldsymbol{\theta}_i)$ and its neural approximation as $\hat{V}_{\psi}(\Omega_{ith};\boldsymbol{\theta}_i)$ to emphasize their dependence on the structural model parameters, where $\boldsymbol{\theta}_i = (\boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}_{r=1}^{R}, \chi_i, \sigma_{\varepsilon}^2, \sigma_{\eta}^2)$ is the tuple of structural parameters that enter the value function. Note that the state Ω_{ith} itself is a function of some components of $\boldsymbol{\theta}_i$, but the entire $\boldsymbol{\theta}_i$ can influence the value function because it characterizes the

Markov decision process of the model.

In general, it is computationally challenging to train a neural network to reliably approximate the value function $V(\Omega_{ith}; \boldsymbol{\theta}_i)$. We therefore exploit our model structure to design the neural network architecture. Rather than supplying raw state variables to a plain feed-forward network, we compute Deep Sets-based embeddings (Zaheer et al., 2017) of the product- and filter-level state variables. These embeddings are invariant to the indexing of the products and filters. Incorporating such invariance results in more effective neural value function approximation as this obviates the need for the network to infer that the ordering of products in the data is irrelevant for the approximation. Rather, we explicitly impose natural architectural constraints that are consistent with our model. This can lead to better approximation quality for a given computational budget for training the neural network. Once such permutation-invariant embeddings are obtained, we concatenate all the embedding vectors and other state variables into a single vector, which is then transformed through a separate feed-forward neural network with fully connected layers, followed by layer normalization and a ReLU activation. The resulting representation is finally projected onto a one-dimensional output with a linear layer to obtain the approximate value of the value function. Empirically, the use of layer normalization improves training performance and stability. Figure 2 visually summarizes our neural network architecture by illustrating how different input variables enter different components of the neural network, how the Deep Sets are used to achieve the permutation invariance, and how the neural network returns the approximation of the value function.

After the neural network training is complete at the beginning of a given MCMC iteration, we take the neural network parameters from the last gradient step of the training, denoted as $\hat{\boldsymbol{\psi}}$, to approximate the integrated value function as $\hat{V}_{\hat{\boldsymbol{\psi}}}(\Omega_{ith}; \boldsymbol{\theta}_i)$ for state Ω_{ith} and structural parameters $\boldsymbol{\theta}_i$ for any consumer i, session t, and step h. The approximate conditional choice probability can then be computed as

$$\hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith}; \boldsymbol{\theta}_i) = \frac{\exp\left(\hat{V}_{a_{ith},\hat{\boldsymbol{\psi}}}(\Omega_{ith}; \boldsymbol{\theta}_i)\right)}{\sum_{a' \in \mathcal{A}_{ith}} \exp\left(\hat{V}_{a',\hat{\boldsymbol{\psi}}}(\Omega_{ith}; \boldsymbol{\theta}_i)\right)}, \tag{14}$$

where the approximate action-specific value function $\hat{V}_{a,\hat{\boldsymbol{\psi}}}(\Omega_{ith}; \boldsymbol{\theta}_i)$ is computed as in Equation (8), but by replacing the value function $V(\Omega_{i,t,h+1})$ with its neural approximation. The integral with respect to the state transition probability measure defined in Equation (5) is also replaced with a Monte Carlo integration.

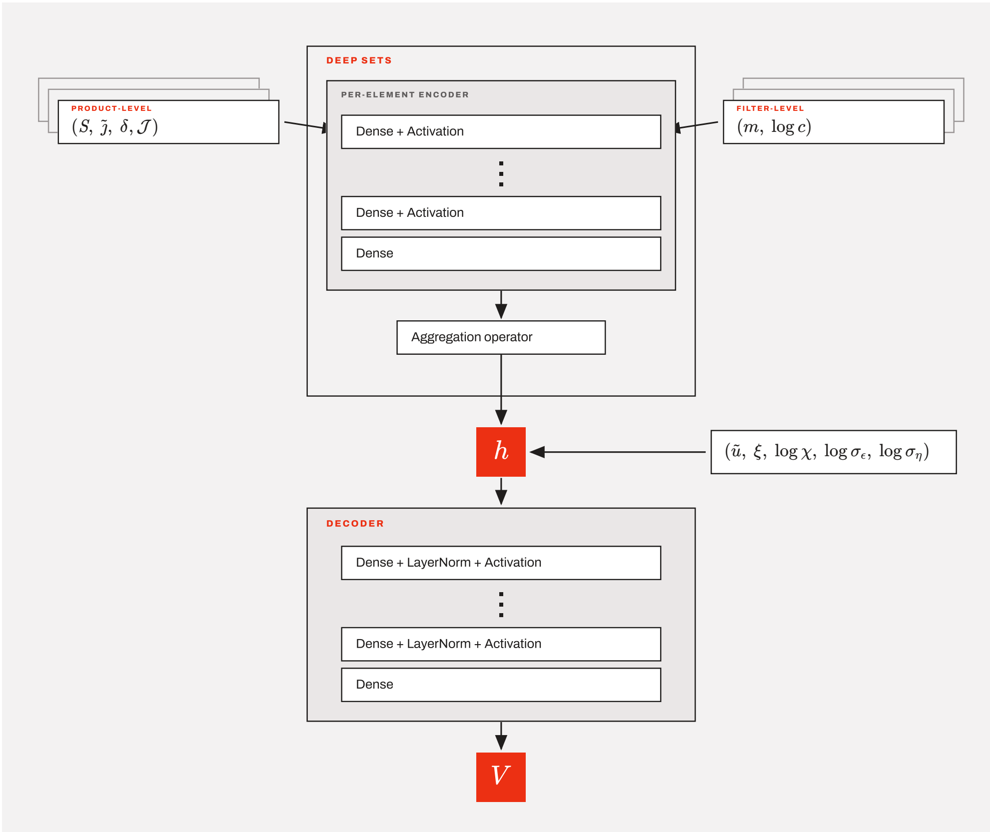

Figure 2: Neural network architecture

3.3 Data Inputs For Model Estimation

Before we describe our empirical application, we review and summarize the data required for model estimation. Let

$$\left(y_{it}, S_{it}, M_{it}, \{(\boldsymbol{x}_{ijt}, \boldsymbol{w}_{ijt})\}_{j=1}^{J}\right)$$

be the observed data for each session $t \in \{1, \ldots, T_i\}$, of consumer $i \in \{1, \ldots, n\}$. Here $y_{it} \in \mathcal{J}_{it} \cup \{0\}$ is the chosen (purchased) alternative. The set $\mathcal{J}_{it} = \mathcal{J}_{i,t,H_{it}}$ contains all the products that consumer i has seen in session t, where H_{it} is the number of steps (actions) recorded in the session. S_{it} is an ordered search set (tuple) such that $S_{it} = (s_{it1}, \ldots, s_{i,t,H_{it}})$, where $s_{ith} \in \mathcal{J}_{ith}$ represents which product is searched in step h if the action taken in step h is to search a product, otherwise $s_{ith} = \texttt{null}$. In particular $s_{it1} = s_{i,t,H_{it}} = \texttt{null}$ holds for all i and t. $M_{it} = (m_{it1}, \ldots, m_{itH_{it}})$ summarizes filter selection, or which filter is active in each step $h \in \{1, \ldots, H_{it}\}$, where $m_{it1} = \texttt{null}$ and $m_{ith} \in \mathcal{M}$ for $h \geq 2$.

As defined earlier, $\boldsymbol{x}_{ijt}$ is a vector of product attributes for each product $j \in \mathcal{J}$. These are observable to consumers prior to search (e.g., price). The vector $\boldsymbol{w}_{ijt}$ also denotes product attributes, but consumers can observe them only after they search (i.e., click on product j and see the corresponding product description page). Finally, we observe the structure of filters: which products belong to which filter(s), i.e., $\{\mathcal{J}_m\}_{m\in\mathcal{M}}$, and the row position $r(j,m)$ of each product $j \in \mathcal{J}_m$ under each filter $m \in \mathcal{M}$.

3.4 Simulation

We conduct a small Monte Carlo simulation to complement our identification discussion in Section 2.10 and verify the validity of our estimation method. We set the number of consumers to $n = 500$, the number of products to $J = 20$, and the number of filters to $M = 3$. We use one intercept and two normally distributed product attributes: one pre-search attribute and one post-search attribute. We use an approximate value function that is based on the same neural network architecture discussed in Section 3.2 to generate the data. We generate 10 datasets with different random number seeds and estimate our model on each by running the MCMC sampler. In the simulation exercise, we focus on the recovery of the structural parameters during estimation instead of assessing the quality of the value function approximation. We discuss the quality of the value function approximation in Appendix A.2 and diagnose this for our application. Table 2 reports the simulation results. The table shows for each parameter the true value, the average of the posterior mean and posterior standard deviation across the 10 simulated datasets, and the standard deviation of these posterior quantities across the same datasets. Thus, the "Average" columns summarize the typical estimate and the typical estimated uncertainty, while the "Standard deviation" columns show how much each varies across datasets. We see from the table that the model parameters are recovered well overall.

Table 2: Monte Carlo simulation results

		Average		Standard deviation	
Parameter	True	Posterior mean	Posterior SD	Posterior mean	Posterior SD
μ_0	-0.5	-0.53	0.07	0.12	0.02
μ_1	0.6	0.55	0.04	0.03	0.00
μ_2	-0.7	-0.64	0.05	0.04	0.00
Σ_{00}	0.8	0.68	0.12	0.28	0.08
Σ_{10}	0.1	0.00	0.07	0.09	0.02
Σ_{11}	0.8	0.62	0.06	0.06	0.01
Σ_{20}	0.1	0.12	0.07	0.16	0.03
Σ_{21}	0.1	0.10	0.05	0.03	0.01
Σ_{22}	0.8	0.68	0.08	0.09	0.01
γ_0	-0.1	-0.09	0.03	0.02	0.00
γ_1	1.5	1.49	0.03	0.02	0.00
σ_c^2	0.2	0.18	0.01	0.01	0.00
λ	2.0	1.97	0.02	0.01	0.00
σ_χ^2	0.1	0.09	0.01	0.00	0.00
σ_η^2	0.2	0.18	0.01	0.01	0.00
σ_ε^2	0.5	0.45	0.03	0.04	0.00

4 Empirical Application

4.1 Data Description

We use clickstream data from an online store that sells Japanese incense, a product category that has been deeply integrated into Japanese culture for centuries and is used in both religious ceremonies and everyday relaxation. It is available in a variety of forms, including sticks, cones, and coils, among others. The price of incense also varies widely across products, depending on factors such as the quality of ingredients, the fragrance, and the reputation of the brand. Given the highly subjective nature of the preferences over Japanese incense, consumers can carefully explore products and read product descriptions to find the alternative that matches their personal taste or a particular occasion.

Our clickstream data span two years—from September 2023 to August 2025. We use purchase observations from only the registered customers of the store to reliably track individual-level search and purchase sessions. A session is defined as a sequence of clicks and a customer is considered to have begun a different session if two consecutive clicks are at least 24 hours apart. This assumption results in 7,706 sessions involving $n = 5{,}352$ unique consumers in our sample. A quarter of the customers in the sample have multiple sessions, with a mean inter-session interval being 97 days and the median being 28 days. The mean number of steps per session is 4.22, with a standard deviation of 2.88. This translates into 2.41 filters applied per session on average, with a standard deviation of 2.31, and the average number of searches of 0.80, with a standard deviation of 1.10. Table 3 shows the session-level summary statistics of our data. We observe that a non-negligible number of sessions end with the consumer making a purchase without visiting the product description page corresponding to the purchased product. It is important to note that this behavior is not consistent with the Weitzman framework which allows for purchase of only a searched product.

The data contain $J = 35$ products that were sold on a regular basis in the store during the data period. The store allows consumers to filter products using product attributes, such as a price range, by clicking on an icon associated with each filter shown on the webpage. Products on the webpage are displayed in a grid format. The row position of a product is defined at the product and filter level since different filters can result in different product arrangements. Row position refers to the row number counted from the top of the page with the topmost row defined as 1, the next row as 2, and so forth. Here, we focus on seven filters so that $|\mathcal{M}| = 7 + 1 = 8$ (including the no-filtering option). Table 4 summarizes these filtering mechanisms. Four of the filters are constructed based on price ranges with thresholds 3,000 yen (approximately 20 USD), 5,000 yen (approximately 33 USD), and 10,000 yen (approximately 66 USD). The other three filters are based on non-price attributes. The first non-price filter contains only products that are wrapped in *furoshiki*, a traditional Japanese wrapping cloth, which can be considered a form of packaging that some people prefer for gifting purposes. Another non-price filter returns those products that come with free delivery. The third filter selects incense products that are bundled with candles.

In our application, we define the pre-search attributes $\boldsymbol{x}_{ijt}$ as a five-dimensional vector whose first element is an intercept for inside products that captures the baseline tendency to purchase an incense product. Its second element is a binary variable that indicates whether product j is wrapped in a *furoshiki*. The third element is also binary indicating whether the product comes with free delivery. The fourth element of $\boldsymbol{x}_{ijt}$ takes a value of 1 if the product is bundled with candles and 0 otherwise. The fifth and last element measures the price in 10,000 yen. Thus the elements of $\boldsymbol{x}_{ijt}$ directly correspond to the attribute-based filters in Table 4 and the prices. Consumers can observe these attributes and the prices without clicking on a product description page (i.e., without searching) as they can infer the attributes either from which filter is

Table 3: Summary statistics at the session level

	Mean	SD	25%	50%	75%
Steps (number of actions)	4.22	2.88	2	3	5
Number of searches	0.80	1.10	0	1	1
Purchase	0.42	0.49	0	0	1
Purchase without search	0.09	0.28	0	0	0
Filters selected	2.41	2.31	1	2	3

The number of observations (sessions) is 7,706.

Purchase without search means a purchase of a product that has not been searched in a given session.

Number of filters selected counts duplicate filters applied multiple times in a session.

Table 4: Summary statistics for filters

Filter description	# products	Mean price	Mean # searches	Share
Main page (no criteria)	35	5,071	0.08	0.27
Price range: $[0, 3000)$	10	1,936	0.52	0.11
Price range: $[3000, 5000)$	9	3,923	0.53	0.28
Price range: $[5000, 10000)$	13	6,575	0.81	0.17
Price range: $[10000, \infty)$	3	12,450	0.45	0.06
Wrapped in a *furoshiki*	3	6,930	0.47	0.03
Free delivery	5	5,236	0.46	0.09
Bundled with candles	3	5,060	0.11	0.00

Prices are shown in yen.

Share in the last column denotes the fraction of times each filter is applied in the data.

being applied or from the product listing page.

For the post-search attributes, we let $\boldsymbol{w}_{ijt}$ be a binary scalar variable that indicates whether product j produces a high amount of smoke when burned. This attribute-level information is observable to consumers only if they engage in search and read the product description page, and is therefore included in the $\boldsymbol{w}_{ijt}$ variable.

4.2 Parameter Estimates

We ran the MCMC sampler for a total of 820,000 iterations, of which the first 80,000 iterations also involved the neural network training for approximating the value function. We stopped training the neural network after it was exposed to a wide range of different state and parameter values across the MCMC iterations. We then took every 100th iteration among the last 100,000 MCMC iterations (i.e., we retained 1,000 draws) to compute the posterior quantities.

Table 5 shows the posterior means and standard deviations of the model parameters, grouped into a few parameter blocks. The first block in the table presents the population mean of the preference parameters, including those for the pre- and post-search attributes, as described in Section 4.1. Immediately below are the population variances of these preference parameters, indicating the degree of consumer heterogeneity in the data. We also provide estimates of the cost related parameters and the variances of pre- and post-search shocks. Several qualitative takeaways emerge from the results in the table. First, consumers tend to prefer traditional, stick type of incense and an average customer does not like products that include candles, based on the negative sign of the posterior mean of μ_3, which aligns with the descriptive statistics reported in the last line of Table 4. The post-search product attribute that captures whether a product produces a high amount of smoke matters substantially to consumers. While they tend to prefer a low smoke level on average, there is a considerable degree of preference heterogeneity as can be seen from the relatively large population variance parameter for this attribute.

The estimates for the cost parameters indicate that the product's row position significantly affects consumer search costs, a finding consistent with the literature (Ursu, 2018). The posterior mean of γ_1 is estimated to be 0.25 with the posterior mean estimate of the unobserved heterogeneity $\sigma_c^2 = 0.01$. This translates to a roughly 20% increase in the search cost on average from the first row to the second row of a product's position on the webpage, an approximately 12% increase in search cost when a product moves from the second to the third row, and a 9% increase when the product's position changes from the third to the fourth row. The estimates of filtering cost parameters conform with intuition. As the posterior mean of the population mean parameter, λ, is 1.15, consumers on average are estimated to have a slightly lower cost of selecting a filter than searching a product by clicking and landing on the product's description page.

The estimates of the variances of the two random shocks are also informative. The posterior mean of the post-search shock variance is substantially higher than that of the pre-search shock variance. This result implies that the post-search shocks, which represent the information that consumers see only when they search and land on a product *description* page, can be much more informative for their decisions than the directly visible product information they see on a product *list* page (net of what the researcher observes such as price). This finding is intuitive as the product description pages typically contain more detailed textual or visual information about products.

Table 5: Estimation results

	Posterior mean	Posterior SD
Preference Parameters (population mean)		
Intercept (μ_0)	-0.13	0.02
Wrapped in a *furoshiki* (μ_1)	-0.74	0.04
Log free delivery (μ_2)	-10.59	0.04
Bundled with candles (μ_3)	-0.31	0.03
Log price sensitivity (μ_4)	-4.05	0.06
High smoke (μ_5)	-0.70	0.05
Preference Parameters (population variance)		
Intercept (Σ_{00})	1.45	0.06
Wrapped in a *furoshiki* (Σ_{11})	0.29	0.03
Free delivery (Σ_{22})	0.19	0.06
Bundled with candles (Σ_{33})	0.07	0.01
Price (Σ_{44})	0.21	0.04
High smoke (Σ_{55})	4.78	0.19
Search costs		
Intercept (γ_0)	1.30	0.01
Slope (γ_1)	0.25	0.00
Unobserved heterogeneity (σ_c^2)	0.01	0.00
Filter costs		
Population mean (λ)	1.15	0.00
Population variance (σ_χ^2)	0.00	0.00
Shock variances		
Post-search shock variance (σ_ε^2)	1.31	0.06
Pre-search shock variance (σ_η^2)	0.01	0.00

4.3 Counterfactual Analyses

4.3.1 Overview

We study two counterfactual scenarios to assess the impact of the filtering mechanisms on consideration sets, product search, and purchase decisions. Our model enables us to predict in a structurally consistent fashion how consumers would behave under alternative filtering regimes. We contrast the consumer behavior in counterfactual scenarios with that under an "observed" data scenario that mimics the structure of the actual dataset. We simulate consumer behavior under a given scenario using 1,000 random datasets that are based on 1,000 MCMC draws from our estimation. In particular, we sample the individual-level parameters from their population distribution, where the population-level parameters are obtained from the MCMC draws and then generate the data according to our model. We perform this procedure for each of the counterfactual scenarios we study as well as the "observed" scenario to ensure that the comparison is based on the same random number seed.

In the first counterfactual scenario, we study the role of the price filters by considering a context in which the online store offers a less granular set of price filters alongside the other existing non-price filters. The price range is partitioned into two groups. The first price filter has products that cost less than 5,000 yen, and the second filter corresponds to those that are 5,000 yen or higher in price. We therefore have two fewer filters in this scenario than in our empirical setting.

The second counterfactual scenario is designed to examine a common assumption in the search literature that consumers must search a product to purchase it. Most of the literature assumes this mainly to rely on the Weitzman framework. Tractable, index-based optimal policies are no longer readily available without this assumption. However, this assumption may not reflect many empirical contexts in which consumers can purchase a product by adding it to a shopping cart directly from a product listing page, that is, without searching that product. This could be rational behavior if consumers do not need to resolve product uncertainty fully to make a purchase decision or if their (relatively high) search costs do not justify product search. Indeed, in our application, the proportion of sessions in which consumers purchase an inside product without searching it is 21% of all the sessions that involve the purchase of an inside good. While our model allows for purchase without search as can be seen from Section 2, we are able to predict how consumers would behave should they be prohibited from purchasing products without searching them through counterfactual analyses. We now present the results of our counterfactual scenarios.

4.3.2 Role of Price Filters

As shown in Table 4, the focal online store features four price-related filters in our empirical context. In this counterfactual scenario, we simplify the filter structure by consolidating these into two, separated by a threshold of 5,000 yen. We find that these two price filters are used only in 44% collectively out of all the steps in which a filter action is taken. This is a substantial decrease from the observed scenario where consumers use the price related filters more than half of the steps (52%) when a filter action is taken. Instead, the non-price filters are estimated to have a higher usage rate, 42%, as opposed to 35% under the "observed" data scenario. The composition of the filter usage shares, however, is estimated to be largely preserved. The predicted share of the low price range filter relative to the high price range filter in this counterfactual scenario is roughly equal to the combined share of the two lowest price range filters relative to that of the two highest in our empirical context. This suggests that having less granular price filters would not result in substantial substitutions between the higher-end price filters and the lower-end ones.

We also find that the number of product searches slightly decreases from 0.86 to 0.84 searches per session, although this difference is not substantial considering the posterior uncertainty. However, consumers would be slightly more likely to purchase products without searching them first in this counterfactual scenario. The purchase probability is estimated to stay roughly the same when the posterior uncertainty is taken into account. While the changes in some quantities are marginal, the overall results are intuitive: consumers would rely on non-price filters to narrow their consideration sets under the simplified price filter structure.

4.3.3 Role of Purchase without Search

In this counterfactual scenario, we study the role of the "no purchase without search" assumption that is common in the literature. Specifically, we replace the definition of Equation (3) with $u^*_{ith} = \tilde{u}_{ith}$ so that consumers would have to search a product to purchase it in a given session. This way, we are able to use the same set of structural parameter estimates from the full dataset, including observations where consumers purchase products without searching, to predict consumer behavior under scenarios in which purchasing without prior search is either allowed or prohibited. We can interpret the counterfactual results as a prediction of consumer behavior under the assumption of no purchase without search.

The (inside product) purchase probability is estimated to decrease substantially from 33% to 26%, which makes intuitive sense since consumers would now have to engage in costly search for the product in order to purchase it in a given session. We find little change in how consumers would apply filters when they are not allowed to purchase a product without search. They are predicted to use price related filters roughly half of the steps when they apply a filter (including the no filter option). The number of searches is also estimated to be roughly the same with a slight increase from 0.86 to 0.88 although this difference is marginal after we take the posterior uncertainty into account.

5 Conclusion

We developed a dynamic programming framework to model filter-assisted consumer search that accommodates two features of e-commerce filtering interfaces that existing sequential search models do not handle, namely, (a) overlapping filters, whereby applying one filter can reveal information about products reachable through other filters, and (b) purchase without prior search, whereby consumers can purchase an alternative without having resolved all uncertainty about it. As these features break the independence assumption that underlies index policies (e.g., Weitzman (1979), Gittins et al. (2011), and Greminger (2022)), we specified the consumer's problem as a dynamic program and estimated it using inverse reinforcement learning. The resulting MDP's state space is large and includes discrete and continuous states, some of which are unobserved. This renders standard estimation approaches for dynamic discrete choice models computationally prohibitive. We addressed this challenge by developing a Bayesian MCMC estimation scheme that approximates the value function with a neural network that is retrained within each MCMC iteration during an initialization period. The neural network uses Deep Sets-based embeddings to exploit the permutation invariance of the product- and filter-level state variables. A Monte Carlo simulation confirmed that the sampler recovers the structural parameters well.

We applied our model to two years of clickstream data from an online incense retailer. Our results show that search costs increase substantially with row position, that filtering is on average slightly less costly than searching, and that the post-search shocks exhibit much higher variability than the pre-search shocks. We also conducted two counterfactual analyses to find that consolidating price filters shifts filter use toward non-price filters rather than reducing filter use overall, and prohibiting purchase without search, the assumption that is ubiquitous in the search literature, lowers predicted purchase probability by about seven percentage points. The latter result showcases the importance of accommodating purchase without search in empirical contexts.

Our results have implications for both research and practice. For researchers, our findings show that restrictive assumptions regarding the sequential search process based on non-overlapping filters and no purchase without search can be consequential. Relaxing these assumptions can change substantive conclusions about consideration set formation and demand. Methodologically, we showed how inverse reinforcement learning combined with neural value function approximation can extend dynamic discrete choice estimation to settings involving large state spaces for which the nested fixed point algorithm, CCP-based estimation, or kernel-based approaches are not easily applicable. For practitioners, our results show that the design of a filtering interface is not neutral: consolidating or restructuring filters redistributes how consumers narrow consideration sets across filter types rather than reducing filter use uniformly. In addition, retailers who wish to project demand accurately should account for the possibility of purchases made without search rather than assuming, as the literature usually does, that all purchases are preceded by search to fully resolve product uncertainty.

Our modeling framework can be extended in multiple directions. We used a parametric approach to specify consumer utilities and the heterogeneity in search and filtering costs. Such parametric assumptions could be relaxed, for example, by incorporating Bayesian nonparametric specifications, as in Onzo and Ansari (2025). This would shed light on how a more flexible specification influences model results. This, however, would come at a greater computational cost. We allowed search costs to vary across consumers as well as across the row positions in which a product is displayed on a webpage. It could also be interesting to study settings where search costs could vary across column positions when consumers scroll the webpage horizontally. Finally, we assumed that price and other product attributes are exogenous, but one could account for endogeneity, for example, through a control function approach using appropriate instruments, where they are available.

6 Declarations: Funding and Competing Interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no funding to report.

References

Aguirregabiria, V. and Mira, P. (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1):38–67.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Arcidiacono, P. and Miller, R. A. (2011). Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79(6):1823–1867.

Bloom, S., Brumberg, J. C., Fisk, I., Harrison, R. J., Hull, R., Ramasubramanian, M., Vliet, K. V., and Wing, J. (2025). Empire AI: A new model for provisioning AI and HPC for academic research in the public good. In *Practice and Experience in Advanced Research Computing (PEARC '25)*, page 4, Columbus, OH, USA. ACM.

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

Choi, H. and Mela, C. F. (2019). Monetizing online marketplaces. *Marketing Science*, 38(6):948–972.

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

Doval, L. (2018). Whether or not to open Pandora's box. *Journal of Economic Theory*, 175:127–158.

Fershtman, D. and Pavan, A. (2024). Searching for "arms": Experimentation with endogenous consideration sets. *Working paper*.

Gardete, P. M. and Hunter, M. (2024). Multiattribute search: Empirical evidence and information design. *Marketing Science*, 43(5):1052–1080.

Gittins, J., Glazebrook, K., and Weber, R. (2011). *Multi-armed Bandit Allocation Indices*. Wiley, 2nd edition.

Greminger, R. P. (2022). Optimal search and discovery. *Management Science*, 68(5):3904–3924.

Greminger, R. P. (2025). Trade-offs between ranking objectives: Reduced-form evidence and structural estimation. *arXiv working paper*.

Hauser, J. R. and Wernerfelt, B. (1990). An evaluation cost model of consideration sets. *Journal of Consumer Research*, 16(4):393–408.

Hodgson, C. and Lewis, G. (2025). You can lead a horse to water: Spatial learning and path dependence in consumer search. *Econometrica*, 93(4):1299–1332.

Honka, E., Hortaçsu, A., and Vitorino, M. A. (2017). Advertising, consumer awareness, and choice: Evidence from the U.S. banking industry. *RAND Journal of Economics*, 48(3):611–646.

Hotz, V. J. and Miller, R. A. (1993). Conditional choice probabilities and the estimation of dynamic models. *Review of Economic Studies*, 60(3):497–529.

Hotz, V. J., Miller, R. A., Sanders, S., and Smith, J. (1994). A simulation estimator for dynamic models of discrete choice. *Review of Economic Studies*, 61(2):265–289.

Imai, S., Jain, N., and Ching, A. (2009). Bayesian estimation of dynamic discrete choice models. *Econometrica*, 77(6):1865–1899.

Jacoby, J., Speller, D. E., and Kohn, C. A. (1974). Brand choice behavior as a function of information load. *Journal of Marketing Research*, 11(1):63–69.

Jiang, B. and Zou, T. (2020). Consumer search and filtering on online retail platforms. *Journal of Marketing Research*, 57(5):900–916.

Kang, E. H., Yoganarasimhan, H., and Jain, L. (2026). An empirical risk minimization approach for offline inverse RL and dynamic discrete choice model.

Kim, J. B., Albuquerque, P., and Bronnenberg, B. J. (2010). Online demand under limited consumer search. *Marketing Science*, 29(6):1001–1023.

Kim, J. B., Albuquerque, P., and Bronnenberg, B. J. (2017). The Probit choice model under sequential search with an application to online retailing. *Management Science*, 63(11):3911–3929.

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

Kuksov, D. and Villas-Boas, J. M. (2010). When more alternatives lead to less choice. *Marketing Science*, 29(3):507–524.

McCulloch, R. and Rossi, P. E. (1994). An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64(1):207–240.

McCulloch, R. E., Polson, N. G., and Rossi, P. E. (2000). A Bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of Econometrics*, 99(1):173–193.

Morozov, I. (2023). Measuring benefits from new products in markets with information frictions. *Management Science*, 69(11):6988–7008.

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

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548, Chia Laguna Resort, Sardinia, Italy. PMLR.

Norets, A. (2012). Estimation of dynamic discrete choice models using artificial neural network approximations. *Econometric Reviews*, 31(1):84–106.

Onzo, K. and Ansari, A. (2025). Bayesian nonparametric sequential search. *Journal of Marketing Research*, 62(2):362–385.

Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica*, 55(5):999–1033.

Scheibehenne, B., Greifeneder, R., and Todd, P. M. (2010). Can there ever be too many options? A meta-analytic review of choice overload. *Journal of Consumer Research*, 37(3):409–425.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Ursu, R., Seiler, S., and Honka, E. (2025). The sequential search model: A framework for empirical research. *Quantitative Marketing and Economics*, 23(1):165–213.

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

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

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

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In *Advances in Neural Information Processing Systems*, volume 30.

Zhang, X., Ursu, R., Honka, E., and Yao, Y. O. (2023). Product discovery and consumer search routes: Evidence from a mobile app. *SSRN Working Paper*.

A Estimation Details

We describe further details of the estimation method beyond what is described in Section 3.

A.1 Gibbs Sampling

We highlight how the full conditionals facilitate computationally tractable posterior sampling in our model, as depicted in Figure 1 by explaining the full conditional for the pre-search attribute preference parameters $\boldsymbol{\beta}_i$ for consumer i. This can be written as

$$\begin{aligned} &p\left(\boldsymbol{\beta}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\ &\quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\{\delta_{ijt}\} \mid \boldsymbol{\beta}_i, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i \mid \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right). \end{aligned} \tag{15}$$

Here, $p(\cdot)$ denotes a generic probability density. In Equation (15), the density $p(\boldsymbol{\beta}_i \mid \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1})$ derived from Equation (10) serves as a prior for $\boldsymbol{\beta}_i$. This prior can be combined with two components of the likelihood, $p\left(\{\delta_{ijt}\} \mid \boldsymbol{\beta}_i, \sigma_\eta^2\right)$ and $p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)$, to make inferences about $\boldsymbol{\beta}_i$. The former is the density of the normal distribution of the augmented pre-search utility δ_{ijt} with mean $\boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i$ and variance σ_η^2. The latter is given by Equation (9), which will be approximated as Equation (14), conditional on the state. This approximation is tractable because unobserved state variables are augmented in our Bayesian approach. For algorithmic efficiency, we reformulate the full conditional given in Equation (15) as

$$\begin{aligned} &p\left(\boldsymbol{\beta}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\ &\quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right). \end{aligned} \tag{16}$$

Here, we exploit the fact that there are two sources of information for the preference parameter for the pre-search attributes and reformulate the full conditional for $\boldsymbol{\beta}_i$ as in Equation (16) to construct a prior $p\left(\boldsymbol{\beta}_i \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)$ relative to $\boldsymbol{\beta}_i$ that is likely more informative than the prior $p\left(\boldsymbol{\beta}_i \mid \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)$ before the reformulation. This reformulation can help obtain a candidate draw that is accepted in a Metropolis step, which will be described below. Consequently, it may lower the overall number of forward passes of the trained neural network required to evaluate the first component of the likelihood in Equation (15) across MCMC iterations.

To draw $\boldsymbol{\beta}_i$ from the reformulated full conditional given in Equation (16), we consider the second term on the right-hand side of Equation (16) as a prior for $\boldsymbol{\beta}_i$ and the first term as a likelihood under the reformulation done in Equation (16). Let $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^{(r)}$ be a sample from the full conditional obtained at the r-th MCMC iteration. At the $(r+1)$-th iteration, we obtain a candidate value $\boldsymbol{\beta}_i^*$ by applying a single elliptical slice sampling step (Murray et al., 2010) to $\boldsymbol{\beta}_i^{(r)}$, which targets the prior under the reformulated specification. Because of the transformation from $\boldsymbol{\beta}_i$ to $\boldsymbol{\xi}_i$ done in Equation (10), the prior distribution is not analytically available or a known distribution (e.g., Gaussian) from which it is easy to sample in many cases, for example, when there is a negativity constraint on the price coefficient. Onzo and Ansari (2025) also use elliptical slice sampling in a sequential search model to draw from a complicated distribution whose density is known up to a normalizing constant.

We then compute the acceptance probability using Equation (16) as follows:

$$
\begin{aligned}
\rho_i &= \min\left\{1, \frac{p\left(\boldsymbol{\beta}_i^* \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i^{(r)} \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)}{p\left(\boldsymbol{\beta}_i^{(r)} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i^* \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i^*, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i^{(r)}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}\right\}.
\end{aligned}
$$

We accept the candidate value $\boldsymbol{\beta}_i^*$ and set $\boldsymbol{\beta}_i^{(r+1)} = \boldsymbol{\beta}_i^*$ with probability ρ_i. Otherwise, we set $\boldsymbol{\beta}_i^{(r+1)} = \boldsymbol{\beta}_i^{(r)}$. Note that there is no tuning parameter in the sampling algorithm we described above unlike a more common random-walk Metropolis-Hastings step. Sampling performance can be highly sensitive to the tuning parameter in a highly complicated and hierarchical setup, like ours, so this is a distinct advantage.

We take a similar approach for drawing the full conditionals for the other parameters in a non-conjugate case with multiple components of the likelihood relative to the parameters.

A.2 Neural Network Training

We describe how we train a neural network to approximate the value function. We first encode product- and filter-level input to the neural network to use Deep Sets (Zaheer et al., 2017). For example, the set S in a state Ω, which specifies the products that have been searched in a session, is multi-hot encoded to a J-dimensional binary vector and then further converted alongside other product-level state variables. Let v_j be a concatenated vector for the state corresponding to product j, we can write the Deep Sets-based embedding as $\rho\left(\sum_{j=1}^{J} \phi(v_j)\right)$, where $\rho(\cdot)$ and $\phi(\cdot)$ are specified through feedforward neural networks.

Here, one can see that the output of this transformation is invariant to the permutation of the product indices while maintaining its expressivity. A similar treatment is implemented for the filter-level (i.e., m-specific) state variables since the order within $\mathcal{M}$ present in the data is also arbitrary.

Having specified the architecture of $\hat{V}_{\psi}$ as shown in Figure 2, we describe how the neural network parameters in the training phase are updated. The loss function for the value function approximation is given by

$$\mathcal{L}\left(\psi;\bar{\psi}\right)=\frac{1}{B}\sum_{b=1}^{B}\left(\hat{V}_{\psi}(\Omega_{(b)};\boldsymbol{\theta}_{(b)})-\mathcal{T}\hat{V}_{\bar{\psi}}(\Omega_{(b)};\boldsymbol{\theta}_{(b)})\right)^{2}, \tag{17}$$

where $\Omega_{(b)}$ and $\boldsymbol{\theta}_{(b)}$ are state and parameters drawn from the distributions centered at the current MCMC draws for each $b \in \{1,\ldots,B\}$, where B is set to 2^{19}, although only randomly selected 2^{9} datapoints are processed in each batch to compute the gradient of the loss function. We perform a stochastic gradient descent with the Adam optimizer for a maximum of two epochs in each MCMC iteration. The operator $\mathcal{T}$ in Equation (17) is the (soft) Bellman operator in the dynamic programming defined from Equation (7). Since we generally do not know the true value function values, the minimization of the loss function given in Equation (17) is considered non-standard and different from supervised learning. Thus, consistent with the reinforcement learning literature, we take a semi-gradient approach and take the gradient of the loss function $\mathcal{L}\left(\psi;\bar{\psi}\right)$ with respect to ψ when updating the neural network parameters while fixing $\bar{\psi}$ (Sutton and Barto, 2018). The parameter $\bar{\psi}$ is instead updated to the latest ψ every some gradient step during the training.

We assess the quality of the value function approximation by computing the neural network output $\hat{V}_{\psi}$ and the Bellman operator to it, $\mathcal{T}\hat{V}_{\psi}$ evaluated at various state and structural parameter values. The difference between $\hat{V}_{\psi}$ and $\mathcal{T}\hat{V}_{\psi}$ would be exactly zero for any state or structural parameter values if the neural network parameter ψ were such that $\hat{V}_{\psi}$ coincides with the true value function. Figure 3 shows a scatter plot in which the horizontal axis shows $\hat{V}_{\psi}$ and the vertical axis shows $\mathcal{T}\hat{V}_{\psi}$ with the neural network parameter ψ being the one we use from the last MCMC iteration. We also provide in Figure 4 another scatter plot with the same horizontal axis but whose vertical axis represents the neural network output $\hat{V}_{\psi'}$ but with another neural network parameter ψ', which is the parameter from an additional iteration of neural network training. This scatter plot shows how stable the neural network training is and resembles what is reported in Hodgson and Lewis (2025). The two scatterplots visually suggest a good approximation of the value function. The correlation between $\hat{V}_{\psi}$ and $\mathcal{T}\hat{V}_{\psi}$ from the first scatter plot is roughly 0.995 and that between $\hat{V}_{\psi}$ and $\hat{V}_{\psi'}$ from the second plot is 0.999.

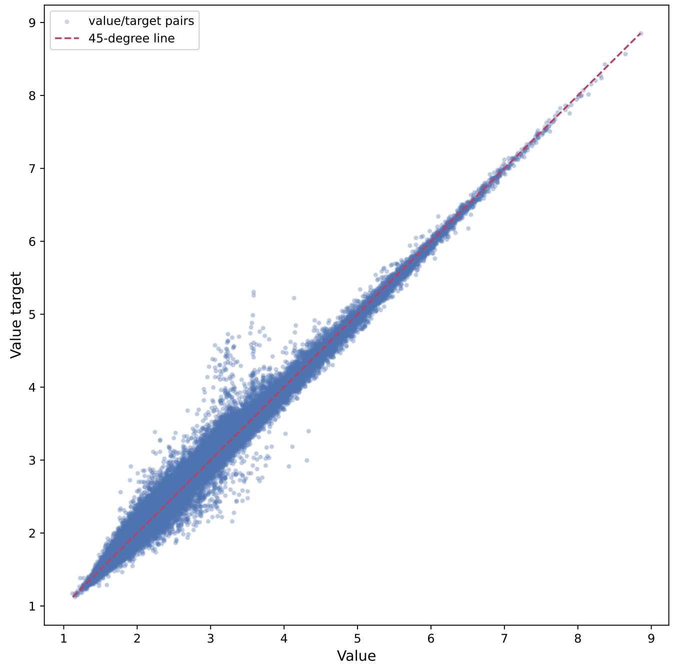

Figure 3: The scatter plot of the two functions $\hat{V}_{\psi}$ and $\mathcal{T}\hat{V}_{\psi}$ using the neural network parameter ψ from the last MCMC iteration

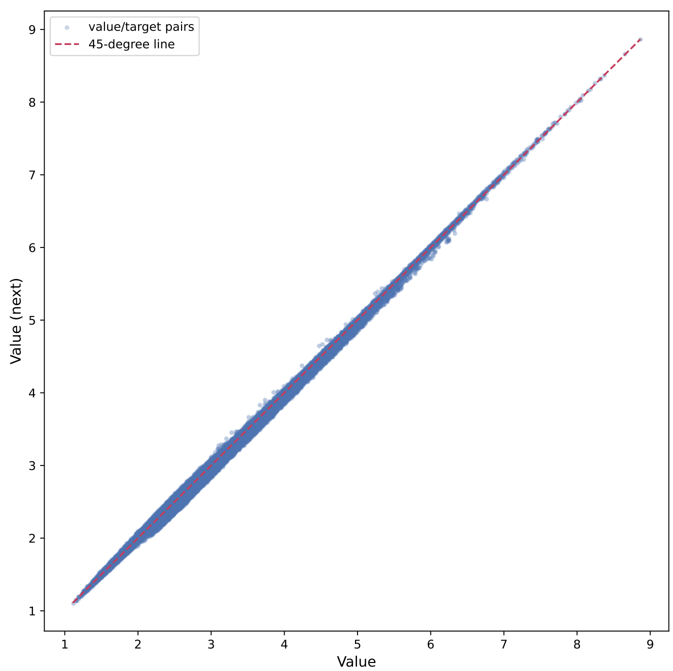

Figure 4: The scatter plot of the two functions $\hat{V}_{\psi}$ and $\hat{V}_{\psi'}$ using the neural network parameter ψ and another parameter ψ' from another iteration of neural network training

Web Appendix

W.A MCMC

We present the full conditionals for our model in this section. We iteratively draw from these full conditionals to compute posterior quantities. To do so, we use a combination of Gibbs sampling, data augmentation, elliptical slice sampling, and the independent Metropolis algorithm. The directed acyclic graph in Figure 1 is useful for understanding the dependence structure of each of the full conditionals of the parameters.

Full Conditional for $\boldsymbol{\beta}_i$ For notational convenience, we let

$$\boldsymbol{\xi}_i = \left[\tilde{\boldsymbol{\beta}}_i', \tilde{\boldsymbol{\zeta}}_i'\right]' \tag{18}$$

and

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{\beta}} \\ \boldsymbol{\mu}_{\boldsymbol{\zeta}} \end{bmatrix}, \; \Sigma^{-1} = \begin{bmatrix} \Lambda_{\boldsymbol{\beta\beta}} & \Lambda_{\boldsymbol{\beta\zeta}} \\ \Lambda_{\boldsymbol{\zeta\beta}} & \Lambda_{\boldsymbol{\zeta\zeta}} \end{bmatrix}. \tag{19}$$

Note that $\tilde{\boldsymbol{\beta}}_i$ and $\tilde{\boldsymbol{\zeta}}_i$ are elementwise transformations from $\boldsymbol{\beta}_i$ and $\boldsymbol{\zeta}_i$, respectively.

As briefly described in Section 3, the full conditional for $\boldsymbol{\beta}_i$ is given by

$$\begin{aligned} & p\left(\boldsymbol{\beta}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\ & \quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\{\delta_{ijt}\} \mid \boldsymbol{\beta}_i, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i \mid \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right), \end{aligned}$$

as stated in Equation (15). From Equation (10), the distribution of $\tilde{\boldsymbol{\beta}}_i$ conditional on $(\boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1})$ is given by $\tilde{\boldsymbol{\beta}}_i \sim \mathrm{N}(\tilde{\boldsymbol{\mu}}_{\boldsymbol{\beta}}, \tilde{\Sigma}_{\boldsymbol{\beta}})$, which serves as a prior, where

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{\boldsymbol{\beta}} &= \boldsymbol{\mu}_{\boldsymbol{\beta}} - \Lambda_{\boldsymbol{\beta\beta}}^{-1} \Lambda_{\boldsymbol{\beta\zeta}} \left(\tilde{\boldsymbol{\zeta}}_i - \boldsymbol{\mu}_{\boldsymbol{\zeta}}\right) \\ \tilde{\Sigma}_{\boldsymbol{\beta}} &= \Lambda_{\boldsymbol{\beta\beta}}^{-1} \end{aligned}.$$

The likelihood has two components: $\delta_{ijt} \sim \mathrm{N}(\boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i, \sigma_\eta^2)$ for all $j \in \mathcal{J}$ and $t \in \{1, \ldots, T_i\}$, and $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$ and $t \in \{1, \ldots, T_i\}$. To achieve good performance in mixing and sampling speed, we first reformulate the full conditional to obtain

$$
\begin{aligned}
&p\left(\boldsymbol{\beta}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right).
\end{aligned}
$$

as in Equation (16). After this reformulation, we draw a candidate value of $\boldsymbol{\beta}_i$ using elliptical slice sampling (Murray et al., 2010) and then accept or reject it with certain probability in a Metropolis step. It is worth noting that no tuning parameters are required in the entire process of drawing $\boldsymbol{\beta}_i$, in contrast to more common sampling algorithms such as the random-walk Metropolis-Hastings algorithm. This feature of no tuning is especially useful in complex, hierarchical models like ours because sampling performance can depend crucially on the tuning parameters.

Drawing a candidate value Let $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^{(r)}$ be a sample from the full conditional obtained at the r-th MCMC iteration and $\tilde{\boldsymbol{\beta}}_i^{(r)}$ be its elementwise transformed vector. We perform elliptical slice sampling to sample from $p\left(\boldsymbol{\beta}_i \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)$. Given a uniform draw $\upsilon \sim \mathrm{Uniform}(0,1)$, the threshold is computed as

$$
\tau = \log \upsilon + \sum_{t=1}^{T_i} \sum_{j=1}^{J} \log \mathrm{N}\left(\delta_{ijt} \mid \boldsymbol{x}_{ijt}' \boldsymbol{\beta}_i^{(r)}, \sigma_\eta^2\right).
$$

Here, $\mathrm{N}(\cdot \mid \cdot, \cdot)$ denotes the density of a normal distribution evaluated at a particular value. A proposal value for a candidate value is $\check{\boldsymbol{\beta}}_i$, which is obtained from its elementwise transformed vector

$$
\check{\tilde{\boldsymbol{\beta}}}_i = \tilde{\boldsymbol{\mu}}_{\boldsymbol{\beta}} + \left(\tilde{\boldsymbol{\beta}}_i^{(r)} - \tilde{\boldsymbol{\mu}}_{\boldsymbol{\beta}}\right) \cos \vartheta + \boldsymbol{\varphi} \sin \vartheta, \tag{20}
$$

where $\boldsymbol{\varphi} \sim \mathrm{N}(\mathbf{0}, \tilde{\Sigma}_{\boldsymbol{\beta}})$ and $\vartheta \sim \mathrm{Uniform}(0, 2\pi)$. The bracket is defined as $(\vartheta_{\min}, \vartheta_{\max}) = (\vartheta - 2\pi, \vartheta)$. If

$$
\sum_{t=1}^{T_i} \sum_{j=1}^{J} \log \mathrm{N}\left(\delta_{ijt} \mid \boldsymbol{x}_{ijt}' \check{\boldsymbol{\beta}}_i, \sigma_\eta^2\right) > \tau,
$$

then we accept the proposal value and consider it as a candidate value so that $\boldsymbol{\beta}_i^* = \check{\boldsymbol{\beta}}_i$. Otherwise, the bracket is updated as follows: $\vartheta_{\min} = \vartheta$ if $\vartheta < 0$, or $\vartheta_{\max} = \vartheta$ otherwise. We then update $\vartheta \sim \mathrm{Uniform}(\vartheta_{\min}, \vartheta_{\max})$ and redefine a proposal value $\check{\boldsymbol{\beta}}_i$ based on Equation (20) using the updated ϑ. The process of updating $(\vartheta_{\min}, \vartheta_{\max})$, ϑ, and $\check{\boldsymbol{\beta}}_i$ continues until a proposal value is accepted.

Computing an acceptance probability We now describe how a candidate value $\boldsymbol{\beta}_i^*$, which is an accepted proposal value in elliptical slice sampling, is either accepted or rejected probabilistically in a Metropolis step. The acceptance probability is given by

$$\begin{aligned}
\rho_i &= \min\left\{1, \frac{p\left(\boldsymbol{\beta}_i^* \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i^{(r)} \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)}{p\left(\boldsymbol{\beta}_i^{(r)} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\beta}_i^* \mid \{\delta_{ijt}\}, \sigma_\eta^2, \boldsymbol{\zeta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i^*, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid \boldsymbol{\beta}_i^{(r)}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}\right\} \\
&= \min\left\{1, \frac{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith}(\boldsymbol{\beta}_i^*))}{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\boldsymbol{\beta}_i^{(r)}\right)\right)}\right\}.
\end{aligned}$$

Note that we need the trained neural network parameter $\hat{\boldsymbol{\psi}}$ when computing $\hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith})$. As stated in Section 3, the (ex-ante) value function and the conditional choice probability are functions of the state Ω_{ith} as well as the structural parameters $\boldsymbol{\theta}_i$. We denote $\Omega_{ith}(\boldsymbol{\beta}_i^*)$ as a concise expression for these state and structural parameters evaluated at the candidate value $\boldsymbol{\beta}_i^*$. We define $\Omega_{ith}\left(\boldsymbol{\beta}_i^{(r)}\right)$ similarly. We accept the candidate value $\boldsymbol{\beta}_i^*$ and let $\boldsymbol{\beta}_i^{(r+1)} = \boldsymbol{\beta}_i^*$ with probability ρ_i, otherwise we reject it and let $\boldsymbol{\beta}_i^{(r+1)} = \boldsymbol{\beta}_i^{(r)}$. Finally, $\boldsymbol{\beta}_i^{(r+1)}$ is considered as a sample from the full conditional at the $(r+1)$-th MCMC iteration.

Full Conditional for $\boldsymbol{\zeta}_i$ The full conditional for $\boldsymbol{\zeta}_i$ is given by

$$\begin{aligned}
&p\left(\boldsymbol{\zeta}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\zeta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\{u_{ijt}\} \mid \boldsymbol{\zeta}_i, \{\delta_{ijt}\}, \sigma_\varepsilon^2\right) p\left(\boldsymbol{\zeta}_i \mid \boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \boldsymbol{\zeta}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\boldsymbol{\zeta}_i \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \sigma_\varepsilon^2, \boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right).
\end{aligned}$$

Here, we describe the full conditional for $\boldsymbol{\zeta}_i$ when $\boldsymbol{\zeta}_i = \tilde{\boldsymbol{\zeta}}_i$ in Equation (18), as in our empirical context. This results in a conjugate step for drawing from the prior

$$p\left(\boldsymbol{\zeta}_i \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \sigma_\varepsilon^2, \boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right) \tag{21}$$

under the reformulation done similarly to that in the full conditional for $\boldsymbol{\beta}_i$. Otherwise, we can take a similar approach using elliptical slice sampling, as is done for $\boldsymbol{\beta}_i$, as described above.

From Equation (19), the distribution of $\boldsymbol{\zeta}_i$ conditional on $(\boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1})$ is

$$\boldsymbol{\zeta}_i = \tilde{\boldsymbol{\zeta}}_i \sim \mathrm{N}\left(\boldsymbol{\mu}_{\boldsymbol{\zeta}} - \Lambda_{\boldsymbol{\zeta}\boldsymbol{\zeta}}^{-1} \Lambda_{\boldsymbol{\zeta}\boldsymbol{\beta}}\left(\tilde{\boldsymbol{\beta}}_i - \boldsymbol{\mu}_{\boldsymbol{\beta}}\right), \Lambda_{\boldsymbol{\zeta}\boldsymbol{\zeta}}^{-1}\right)$$

The likelihood has two components: $u_{ijt} - \delta_{ijt} \sim \mathrm{N}(\boldsymbol{w}'_{ijt}\boldsymbol{\zeta}_i, \sigma^2_\varepsilon)$ for all $j \in \mathcal{J}$ and $t \in \{1, \ldots, T_i\}$, and $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$ and $t \in \{1, \ldots, T_i\}$. Thus, we draw $\boldsymbol{\zeta}_i$ from its conditional as follows. We first draw $\boldsymbol{\zeta}^*_i$ from Equation (21), which is $\mathrm{N}(\hat{\boldsymbol{\mu}}_{\boldsymbol{\zeta}}, \hat{\Sigma}_{\boldsymbol{\zeta}})$, where

$$
\begin{aligned}
\hat{\Sigma}^{-1}_{\boldsymbol{\zeta}} &= \Lambda_{\boldsymbol{\zeta}\boldsymbol{\zeta}} + \sigma^{-2}_\varepsilon \sum_{t=1}^{T_i} \sum_{j=1}^{J} \boldsymbol{w}_{ijt}\boldsymbol{w}'_{ijt} \\
\hat{\boldsymbol{\mu}}_{\boldsymbol{\zeta}} &= \hat{\Sigma}_{\boldsymbol{\zeta}} \left(\Lambda_{\boldsymbol{\zeta}\boldsymbol{\zeta}} \left(\boldsymbol{\mu}_{\boldsymbol{\zeta}} - \Lambda^{-1}_{\boldsymbol{\zeta}\boldsymbol{\zeta}} \Lambda_{\boldsymbol{\zeta}\boldsymbol{\beta}} \left(\tilde{\boldsymbol{\beta}}_i - \boldsymbol{\mu}_{\boldsymbol{\beta}} \right) \right) + \sigma^{-2}_\varepsilon \sum_{t=1}^{T_i} \sum_{j=1}^{J} \boldsymbol{w}_{ijt}(u_{ijt} - \delta_{ijt}) \right),
\end{aligned}
$$

and consider it as a candidate value. We then have an independent Metropolis step, where we accept the candidate value with probability

$$
\begin{aligned}
\rho_i &= \min \left\{ 1, \frac{p\left(\boldsymbol{\zeta}^*_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma^2_\varepsilon, \sigma^2_\eta\right) p\left(\boldsymbol{\zeta}^{(r)}_i \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \sigma^2_\varepsilon, \boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)}{p\left(\boldsymbol{\zeta}^{(r)}_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \boldsymbol{\mu}, \Sigma^{-1}, \sigma^2_\varepsilon, \sigma^2_\eta\right) p\left(\boldsymbol{\zeta}^*_i \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \sigma^2_\varepsilon, \boldsymbol{\beta}_i, \boldsymbol{\mu}, \Sigma^{-1}\right)} \right\} \\
&= \min \left\{ 1, \frac{p\left(\{a_{ith}\} \mid \boldsymbol{\zeta}^*_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \sigma^2_\varepsilon, \sigma^2_\eta\right)}{p\left(\{a_{ith}\} \mid \boldsymbol{\zeta}^{(r)}_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \{c_{ir}\}, \chi_i, \sigma^2_\varepsilon, \sigma^2_\eta\right)} \right\} \\
&= \min \left\{ 1, \frac{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith}(\boldsymbol{\zeta}^*_i))}{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\boldsymbol{\zeta}^{(r)}_i\right)\right)} \right\}
\end{aligned}
$$

and set $\boldsymbol{\zeta}^{(r+1)}_i = \boldsymbol{\zeta}^*_i$ at the $(r+1)$-th MCMC iteration. Otherwise, we set $\boldsymbol{\zeta}^{(r+1)}_i = \boldsymbol{\zeta}^{(r)}_i$.

Full Conditional for c_{ir} The full conditional for the search cost parameters $\{c_{ir}\}_{r=1}^R$ is given by

$$
\begin{aligned}
&p\left(\{c_{ir}\} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \boldsymbol{\gamma}, \sigma^2_c, \sigma^2_\varepsilon, \sigma^2_\eta\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \{c_{ir}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \sigma^2_\varepsilon, \sigma^2_\eta\right) p\left(\{c_{ir}\} \mid \boldsymbol{\gamma}, \sigma^2_c\right).
\end{aligned}
$$

We use the independent Metropolis algorithm to sample from the full conditional. We first sample a candidate value from $p\left(\{c_{ir}\} \mid \boldsymbol{\gamma}, \sigma_c^2\right)$, which is given in Equation (11). Then, we accept it with probability

$$\begin{aligned}
\rho_i &= \min\left\{1, \frac{p\left(\{c_{ir}^*\} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \boldsymbol{\gamma}, \sigma_c^2, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\left\{c_{ir}^{(r)}\right\} \mid \boldsymbol{\gamma}, \sigma_c^2\right)}{p\left(\left\{c_{ir}^{(r)}\right\} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \boldsymbol{\gamma}, \sigma_c^2, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\{c_{ir}^*\} \mid \boldsymbol{\gamma}, \sigma_c^2\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \{c_{ir}^*\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid \left\{c_{ir}^{(r)}\right\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}\right\} \\
&= \min\left\{1, \ \frac{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith}(\{c_{ir}^*\}))}{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\left\{c_{ir}^{(r)}\right\}\right)\right)}\right\}
\end{aligned}$$

and set $c_{ir}^{(r+1)} = c_{ir}^*$ for each subscript $r \in \{1, \ldots, R\}$, representing a row position. Otherwise, we set $c_{ir}^{(r+1)} = c_{ir}^{(r)}$ for each subscript r. Note that the superscripts r and $r+1$ in $c_{ir}^{(r)}$ and $c_{ir}^{(r+1)}$ respectively denote the MCMC iterations.

Full Conditional for χ_i The full conditional for the filter cost parameter χ_i is given by

$$\begin{aligned}
&p\left(\chi_i \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \lambda, \sigma_\chi^2, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \chi_i, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\chi_i \mid \lambda, \sigma_\chi^2\right).
\end{aligned}$$

We use the independent Metropolis algorithm to sample from the full conditional. We first sample a candidate value from $p\left(\chi_i \mid \lambda, \sigma_\chi^2\right)$, which is given in Equation (12). Then, we accept it with probability

$$\begin{aligned}
\rho_i &= \min\left\{1, \frac{p\left(\chi_i^* \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \lambda, \sigma_\chi^2, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\chi_i^{(r)} \mid \lambda, \sigma_\chi^2\right)}{p\left(\chi_i^{(r)} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \lambda, \sigma_\chi^2, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\chi_i^* \mid \lambda, \sigma_\chi^2\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \chi_i^*, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \sigma_\varepsilon^2, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid \chi_i^{(r)}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \sigma_\varepsilon^2, \sigma_\eta^2\right)}\right\} \\
&= \min\left\{1, \ \frac{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}(a_{ith} \mid \Omega_{ith}(\chi_i^*))}{\prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\chi_i^{(r)}\right)\right)}\right\}
\end{aligned}$$

and set $\chi_i^{(r+1)} = \chi_i^*$. Otherwise, we set $\chi_i^{(r+1)} = \chi_i^{(r)}$. Finally, we consider $\chi_i^{(r+1)}$ as a draw from the full conditional at the $(r+1)$-th MCMC iteration.

Full Conditional for σ_ε^2 Let the prior for the post-search shock variance σ_ε^2 be $\sigma_\varepsilon^2 \sim \mathrm{IG}(a_\varepsilon, b_\varepsilon)$ with $a_\varepsilon = b_\varepsilon = 0.01$. The full conditional for σ_ε^2 is given by

$$
\begin{aligned}
&p\left(\sigma_\varepsilon^2 \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \sigma_\varepsilon^2, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) p\left(\{u_{ijt}\} \mid \sigma_\varepsilon^2, \{\delta_{ijt}\}, \{\boldsymbol{\zeta}_i\}\right) p\left(\sigma_\varepsilon^2\right).
\end{aligned}
$$

The likelihood relative to σ_ε^2 has two components: $u_{ijt} \sim \mathrm{N}(\delta_{ijt} + \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i, \sigma_\varepsilon^2)$ for all $j \in \mathcal{J}$, $t \in \{1, \ldots, T_i\}$, and $i \in \{1, \ldots, n\}$; and $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$, $t \in \{1, \ldots, T_i\}$, and $i \in \{1, \ldots, n\}$. We reformulate the full conditional above to obtain

$$
\begin{aligned}
&p\left(\sigma_\varepsilon^2 \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \sigma_\varepsilon^2, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) p\left(\sigma_\varepsilon^2 \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \{\boldsymbol{\zeta}_i\}\right).
\end{aligned}
$$

We conduct a Metropolis step with the prior $p\left(\sigma_\varepsilon^2 \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \{\boldsymbol{\zeta}_i\}\right)$ under the reformulation as the distribution from which a candidate value is drawn. This prior is given by $\sigma_\varepsilon^2 \sim \mathrm{IG}(\hat{a}_\varepsilon, \hat{b}_\varepsilon)$, where

$$
\hat{a}_\varepsilon = a_\varepsilon + \frac{1}{2} J \sum_{i=1}^{n} T_i
$$

and

$$
\hat{b}_\varepsilon = b_\varepsilon + \frac{1}{2} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \sum_{j=1}^{J} \left(u_{ijt} - \delta_{ijt} - \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i\right)^2.
$$

Then, we accept a candidate draw σ_ε^{2*} with probability

$$
\begin{aligned}
\rho &= \min\left\{1, \frac{p\left(\sigma_\varepsilon^{2*} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) p\left(\sigma_\varepsilon^{2(r)} \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \{\boldsymbol{\zeta}_i\}\right)}{p\left(\sigma_\varepsilon^{2(r)} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right) p\left(\sigma_\varepsilon^{2*} \mid \{u_{ijt}\}, \{\delta_{ijt}\}, \{\boldsymbol{\zeta}_i\}\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \sigma_\varepsilon^{2*}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid \sigma_\varepsilon^{2(r)}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\eta^2\right)}\right\} \\
&= \min\left\{1, \frac{\prod_{i=1}^{n} \prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\sigma_\varepsilon^{2*}\right)\right)}{\prod_{i=1}^{n} \prod_{t=1}^{T_i} \prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\sigma_\varepsilon^{2(r)}\right)\right)}\right\}
\end{aligned}
$$

and set $\sigma_\varepsilon^{2(r+1)} = \sigma_\varepsilon^{2*}$. Otherwise, we set $\sigma_\varepsilon^{2(r+1)} = \sigma_\varepsilon^{2(r)}$. Finally, we consider $\sigma_\varepsilon^{2(r+1)}$ as a draw from the full conditional for σ_ε^2 at the $(r+1)$-th MCMC iteration.

Full Conditional for σ_η^2 Let the prior for the pre-search shock variance σ_η^2 be $\sigma_\eta^2 \sim \mathrm{IG}(a_\eta, b_\eta)$ with $a_\eta = b_\eta = 0.01$. The full conditional for σ_η^2 is given by

$$\begin{aligned}
&p\left(\sigma_\eta^2 \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \sigma_\eta^2, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) p\left(\{u_{i0t}\} \mid \sigma_\eta^2\right) p\left(\{\delta_{ijt}\} \mid \sigma_\eta^2, \{\boldsymbol{\beta}_i\}\right) p\left(\sigma_\eta^2\right).
\end{aligned}$$

As can be seen, the likelihood relative to σ_η^2 has three components: $\delta_{ijt} \sim \mathrm{N}(\boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i, \sigma_\eta^2)$ for all $j \in \mathcal{J}$, $t \in \{1, \ldots, T_i\}$, and $i \in \{1, \ldots, n\}$; $u_{i0t} \sim \mathrm{N}(0, \sigma_\eta^2)$ for all $t \in \{1, \ldots, T_i\}$ and $i \in \{1, \ldots, n\}$; and $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$, $t \in \{1, \ldots, T_i\}$, and $i \in \{1, \ldots, n\}$. We reformulate the full conditional above to obtain

$$\begin{aligned}
&p\left(\sigma_\eta^2 \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \sigma_\eta^2, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) p\left(\sigma_\eta^2 \mid \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}\right).
\end{aligned}$$

We conduct a Metropolis step with the prior $p\left(\sigma_\eta^2 \mid \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}\right)$ under the reformulation as the distribution from which a candidate value is drawn. This prior is given by $\sigma_\eta^2 \sim \mathrm{IG}(\hat{a}_\eta, \hat{b}_\eta)$, where

$$\hat{a}_\eta = a_\eta + \frac{1}{2}(J+1)\sum_{i=1}^{n} T_i$$

and

$$\hat{b}_\eta = b_\eta + \frac{1}{2}\sum_{i=1}^{n}\sum_{t=1}^{T_i}\left(\sum_{j=1}^{J}(\delta_{ijt} - \boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i)^2 + u_{i0t}^2\right).$$

Then, we accept a candidate draw σ_η^{2*} with probability

$$\begin{aligned}
\rho &= \min\left\{1, \frac{p\left(\sigma_\eta^{2*} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) p\left(\sigma_\eta^{2(r)} \mid \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}\right)}{p\left(\sigma_\eta^{2(r)} \mid \{a_{ith}\}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right) p\left(\sigma_\eta^{2*} \mid \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}\right)}\right\} \\
&= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \sigma_\eta^{2*}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right)}{p\left(\{a_{ith}\} \mid \sigma_\eta^{2(r)}, \{u_{ijt}\}, \{u_{i0t}\}, \{\delta_{ijt}\}, \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \{c_{ir}\}, \{\chi_i\}, \sigma_\varepsilon^2\right)}\right\} \\
&= \min\left\{1, \frac{\prod_{i=1}^{n}\prod_{t=1}^{T_i}\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\sigma_\eta^{2*}\right)\right)}{\prod_{i=1}^{n}\prod_{t=1}^{T_i}\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\sigma_\eta^{2(r)}\right)\right)}\right\}
\end{aligned}$$

and set $\sigma_\eta^{2(r+1)} = \sigma_\eta^{2*}$. Otherwise, we set $\sigma_\eta^{2(r+1)} = \sigma_\eta^{2(r)}$. Finally, we consider $\sigma_\eta^{2(r+1)}$ as a draw from the full conditional for σ_η^2 at the $(r+1)$-th MCMC iteration.

Full Conditional for δ_{ijt} The full conditional for the pre-search utility δ_{ijt} for product $j \in \mathcal{J}$, session $t \in \{1, \ldots, T_i\}$, and consumer $i \in \{1, \ldots, n\}$ is given by

$$
\begin{aligned}
&p\left(\delta_{ijt} \mid \{\delta_{-(ijt)}\}, \{a_{ith}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \delta_{ijt}, \{\delta_{-(ijt)}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(u_{ijt} \mid \delta_{ijt}, \boldsymbol{\zeta}_i, \sigma_\varepsilon^2\right) p\left(\delta_{ijt} \mid \boldsymbol{\beta}_i, \sigma_\eta^2\right).
\end{aligned}
$$

Here, the prior relative to δ_{ijt} is given by $\delta_{ijt} \sim \mathrm{N}(\boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i, \sigma_\eta^2)$, and the likelihood has two components: $u_{ijt} - \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i \sim \mathrm{N}(\delta_{ijt}, \sigma_\varepsilon^2)$, and $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$. We reformulate the full conditional to obtain

$$
\begin{aligned}
&p\left(\delta_{ijt} \mid \{\delta_{-(ijt)}\}, \{a_{ith}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\
&\quad \propto p\left(\{a_{ith}\} \mid \delta_{ijt}, \{\delta_{-(ijt)}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(\delta_{ijt} \mid u_{ijt}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)
\end{aligned}
$$

and use the independent Metropolis algorithm, in which we draw a candidate value δ_{ijt}^* from a truncated normal distribution $\mathrm{trN}(\hat{\mu}_{ijt}, \hat{\sigma}_{ijt}^2)_{\text{lower}_{ijt}}^{\text{upper}_{ijt}}$, where

$$
\hat{\sigma}_{ijt}^{-2} = \sigma_\eta^{-2} + \sigma_\varepsilon^{-2}
$$

$$
\hat{\mu}_{ijt} = \hat{\sigma}_{ijt}^2 \left(\sigma_\eta^{-2} \boldsymbol{x}_{ijt}'\boldsymbol{\beta}_i + \sigma_\varepsilon^{-2}\left(u_{ijt} - \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i\right)\right),
$$

$$
\text{lower}_{ijt} = \begin{cases} \max\left\{\max_{j' \in S_{it} \cup \{0\}}\{u_{ij't}\},\ \max_{j' \in (\mathcal{J}_{it} \setminus S_{it}) \setminus \{j\}}\{\delta_{ij't} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} - \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i & \text{if } [j = y_{it}] \wedge [j \in \mathcal{J}_{it} \setminus S_{it}] \\ -\infty & \text{otherwise} \end{cases},
$$

and

$$
\text{upper}_{ijt} = \begin{cases} \max\left\{\max_{j' \in S_{it} \cup \{0\}}\{u_{ij't}\},\ \max_{j' \in (\mathcal{J}_{it} \setminus S_{it}) \setminus \{j\}}\{\delta_{ij't} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} - \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i & \text{if } [j \neq y_{it}] \wedge [j \in \mathcal{J}_{it} \setminus S_{it}] \\ \infty & \text{otherwise} \end{cases}.
$$

Here, $\overline{\boldsymbol{w}}$ is the mean of the distribution $F^{\boldsymbol{w}}$. The lower and upper bounds are derived from the construction of Equation (3) and the support of the conditional choice probability in Equation (9) to facilitate the efficient implementation of the independent Metropolis algorithm. For example, if consumer i purchases product j in session t without search, then the corresponding pre-search utility δ_{ijt} must be sufficiently high to rationalize this purchase behavior in the model because $u_{itH_{it}}^* = \overline{u}_{ijt}$ in Equation (3). With this proposal distribution, the acceptance probability can be computed as

$$\begin{aligned}\rho_{ijt} &= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid \delta^{*}_{ijt}, \{\delta_{-(ijt)}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right)}{p\left(\{a_{ith}\} \mid \delta^{(r)}_{ijt}, \{\delta_{-(ijt)}\}, \{u_{ijt}\}, u_{i0t}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right)}\right\} \\ &= \min\left\{1,\ \frac{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}(\delta^{*}_{ijt})\right)}{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(\delta^{(r)}_{ijt}\right)\right)}\right\}.\end{aligned}$$

We set $\delta^{(r+1)}_{ijt} = \delta^{*}_{ijt}$ with probability ρ_{ijt}, and $\delta^{(r+1)}_{ijt} = \delta^{(r)}_{ijt}$ with probability $1 - \rho_{ijt}$. We then consider $\delta^{(r+1)}_{ijt}$ as a draw from the full conditional for δ_{ijt} at the $(r+1)$-th MCMC iteration.

Full Conditional for u_{i0t} The full conditional for the utility for the outside option, u_{i0t}, in session $t \in \{1, \ldots, T_i\}$ and consumer $i \in \{1, \ldots, n\}$ is given by

$$\begin{aligned}&p\left(u_{i0t} \mid \{a_{ith}\}, \{u_{ijt}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right) \\ &\qquad \propto p\left(\{a_{ith}\} \mid u_{i0t}, \{u_{ijt}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right) p\left(u_{i0t} \mid \sigma^2_{\eta}\right).\end{aligned}$$

Here, the prior for u_{i0t} is $u_{i0t} \sim \mathrm{N}(0, \sigma^2_{\eta})$, and the likelihood is $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$. We use the independent Metropolis algorithm to draw from the full conditional, in which we first draw a candidate value u^{*}_{i0t} from a truncated normal distribution $\mathrm{trN}(0, \sigma^2_{\eta})^{\text{upper}_{it}}_{\text{lower}_{it}}$, where

$$\text{lower}_{it} = \begin{cases} \max\left\{\max_{j \in S_{it}}\{u_{ijt}\},\ \max_{j \in \mathcal{J}_{it} \setminus S_{it}}\{\delta_{ijt} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} & \text{if } y_{it} = 0 \\ -\infty & \text{if } y_{it} \neq 0 \end{cases},$$

and

$$\text{upper}_{it} = \begin{cases} \max\left\{\max_{j \in S_{it}}\{u_{ijt}\},\ \max_{j \in \mathcal{J}_{it} \setminus S_{it}}\{\delta_{ijt} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} & \text{if } y_{it} \neq 0 \\ \infty & \text{if } y_{it} = 0 \end{cases}.$$

The lower and upper bounds are derived from the construction of Equation (3) and the support of the conditional choice probability in Equation (9) to facilitate the efficient implementation of the independent Metropolis algorithm. For example, if consumer i does not purchase any (inside) product in session t, then the utility for the outside option u_{i0t} must be sufficiently high to rationalize this behavior in the model. With this proposal distribution, the acceptance probability can be computed as

$$\begin{aligned}\rho_{it} &= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid u_{i0t}^{*}, \{u_{ijt}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}{p\left(\{a_{ith}\} \mid u_{i0t}^{(r)}, \{u_{ijt}\}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right)}\right\} \\ &= \min\left\{1, \frac{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}(u_{i0t}^{*})\right)}{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(u_{i0t}^{(r)}\right)\right)}\right\}.\end{aligned}$$

We set $u_{i0t}^{(r+1)} = u_{i0t}^{*}$ with probability ρ_{it}, and $u_{i0t}^{(r+1)} = u_{i0t}^{(r)}$ with probability $1 - \rho_{it}$. We then consider $u_{i0t}^{(r+1)}$ as a draw from the full conditional for u_{i0t} at the $(r+1)$-th MCMC iteration.

Full Conditional for u_{ijt} The full conditional for the utility u_{ijt} defined in Equation (1) for product $j \in \mathcal{J}$, session $t \in \{1, \ldots, T_i\}$, and consumer $i \in \{1, \ldots, n\}$ is given by

$$\begin{aligned}&p\left(u_{ijt} \mid \{u_{-(ijt)}\}, \{a_{ith}\}, u_{i0t}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) \\ &\quad \propto p\left(\{a_{ith}\} \mid u_{ijt}, \{u_{-(ijt)}\}, u_{i0t}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma_\varepsilon^2, \sigma_\eta^2\right) p\left(u_{ijt} \mid \delta_{ijt}, \boldsymbol{\zeta}_i, \sigma_\varepsilon^2\right).\end{aligned}$$

The prior for u_{ijt} is $u_{ijt} \sim \mathrm{N}\left(\delta_{ijt} + \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i,\ \sigma_\varepsilon^2\right)$, and the likelihood is $a_{ith} \sim \Pr(a_{ith} \mid \Omega_{ith})$ as given in Equation (9) for all $h \in \{1, \ldots, H_{it}\}$. We use the independent Metropolis algorithm, in which we draw a candidate value u_{ijt}^{*} from a truncated normal distribution $\mathrm{trN}\left(\delta_{ijt} + \boldsymbol{w}_{ijt}'\boldsymbol{\zeta}_i, \sigma_\varepsilon^2\right)_{\text{lower}_{ijt}}^{\text{upper}_{ijt}}$, where

$$\text{lower}_{ijt} = \begin{cases} \max\left\{\max_{j' \in (S_{it} \cup \{0\}) \setminus \{j\}}\{u_{ij't}\},\ \max_{j' \in \mathcal{J}_{it} \setminus S_{it}}\{\delta_{ij't} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} & \text{if } [j = y_{it}] \wedge [j \in S_{it}] \\ -\infty & \text{otherwise} \end{cases},$$

and

$$\text{upper}_{ijt} = \begin{cases} \max\left\{\max_{j' \in (S_{it} \cup \{0\}) \setminus \{j\}}\{u_{ij't}\},\ \max_{j' \in \mathcal{J}_{it} \setminus S_{it}}\{\delta_{ij't} + \overline{\boldsymbol{w}}'\boldsymbol{\zeta}_i\}\right\} & \text{if } [j \neq y_{it}] \wedge [j \in S_{it}] \\ \infty & \text{otherwise} \end{cases}.$$

The lower and upper bounds are derived from the construction of Equation (3) and the support of the conditional choice probability in Equation (9) to facilitate the efficient implementation of the independent Metropolis algorithm. For example, if consumer i purchases a searched product j in session t, then the corresponding utility u_{ijt} must be sufficiently high to rationalize this purchase behavior in the model. With this proposal distribution, the acceptance probability can be computed as

$$\begin{aligned}\rho_{ijt} &= \min\left\{1, \frac{p\left(\{a_{ith}\} \mid u^{*}_{ijt}, \{u_{-(ijt)}\}, u_{i0t}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right)}{p\left(\{a_{ith}\} \mid u^{(r)}_{ijt}, \{u_{-(ijt)}\}, u_{i0t}, \{\delta_{ijt}\}, \boldsymbol{\beta}_i, \boldsymbol{\zeta}_i, \{c_{ir}\}, \chi_i, \sigma^2_{\varepsilon}, \sigma^2_{\eta}\right)}\right\} \\ &= \min\left\{1, \frac{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}(u^{*}_{ijt})\right)}{\prod_{h=1}^{H_{it}} \hat{\Pr}_{\hat{\boldsymbol{\psi}}}\left(a_{ith} \mid \Omega_{ith}\left(u^{(r)}_{ijt}\right)\right)}\right\}.\end{aligned}$$

We set $u^{(r+1)}_{ijt} = u^{*}_{ijt}$ with probability ρ_{ijt}, and $u^{(r+1)}_{ijt} = u^{(r)}_{ijt}$ with probability $1 - \rho_{ijt}$. We then consider $u^{(r+1)}_{ijt}$ as a draw from the full conditional for u_{ijt} at the $(r+1)$-th MCMC iteration.

Full Conditional for $\boldsymbol{\mu}$ The full conditional for the population mean parameter $\boldsymbol{\mu}$ in Equation (10) is given by

$$p\left(\boldsymbol{\mu} \mid \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \Sigma^{-1}\right) = \mathrm{N}\left(\hat{\boldsymbol{m}}_{\boldsymbol{\mu}}, \hat{C}_{\boldsymbol{\mu}}\right),$$

where

$$\begin{aligned}\hat{C}^{-1}_{\boldsymbol{\mu}} &= C^{-1}_{\boldsymbol{\mu}} + n\Sigma^{-1} \\ \hat{\boldsymbol{m}}_{\boldsymbol{\mu}} &= \hat{C}_{\boldsymbol{\mu}}\left(C^{-1}_{\boldsymbol{\mu}}\boldsymbol{m}_{\boldsymbol{\mu}} + \Sigma^{-1}\sum_{i=1}^{n}\boldsymbol{\xi}_i\right)\end{aligned}.$$

and $\boldsymbol{\xi}_i$ is a transformation from a vector $[\boldsymbol{\beta}'_i, \boldsymbol{\zeta}'_i]'$. We set the prior for $\boldsymbol{\mu}$ to $\mathrm{N}(\boldsymbol{m}_{\boldsymbol{\mu}}, C_{\boldsymbol{\mu}})$ with $\boldsymbol{m}_{\boldsymbol{\mu}} = \mathbf{0}$ and $C_{\boldsymbol{\mu}} = 10^2\mathbf{I}$, where $\mathbf{I}$ is an identity matrix.

Full Conditional for Σ^{-1} The full conditional for the precision matrix Σ^{-1} (i.e., the inverse of the covariance matrix Σ) in Equation (10) is given by

$$p\left(\Sigma^{-1} \mid \{\boldsymbol{\beta}_i\}, \{\boldsymbol{\zeta}_i\}, \boldsymbol{\mu}\right) = \mathrm{Wishart}\left(\hat{v}_{\Sigma}, \hat{R}_{\Sigma}\right),$$

where

$$\begin{aligned}\hat{v}_{\Sigma} &= v_{\Sigma} + n \\ \hat{R}^{-1}_{\Sigma} &= R^{-1}_{\Sigma} + \sum_{i=1}^{n}(\boldsymbol{\xi}_i - \boldsymbol{\mu})(\boldsymbol{\xi}_i - \boldsymbol{\mu})'\end{aligned}.$$

The prior for Σ^{-1} is set to $\mathrm{Wishart}(v_{\Sigma}, R_{\Sigma})$ with $v_{\Sigma} = \dim(\boldsymbol{\xi}_i) + 1$ and $R_{\Sigma} = \mathbf{I}$.

Full Conditional for γ The full conditional for γ in Equation (11) is given by a normal distribution because the likelihood relative to γ is normal and we set the prior to $\gamma \sim \mathrm{N}(\boldsymbol{m}_\gamma, C_\gamma)$ with $\boldsymbol{m}_\gamma = \mathbf{0}$ and $C_\gamma = 10^2\mathbf{I}$. In particular, we have $p(\gamma \mid \{c_{ir}\}, \sigma_c^2) = \mathrm{N}(\hat{\boldsymbol{m}}_\gamma, \hat{C}_\gamma)$, where

$$\begin{aligned} \hat{C}_\gamma^{-1} &= C_\gamma^{-1} + \sigma_c^{-2} \sum_{i=1}^{n} \sum_{r=1}^{R} \boldsymbol{z}_{ir} \boldsymbol{z}_{ir}' \\ \hat{\boldsymbol{m}}_\gamma &= \hat{C}_\gamma \left(C_\gamma^{-1} \boldsymbol{m}_\gamma + \sigma_c^{-2} \sum_{i=1}^{n} \sum_{r=1}^{R} \boldsymbol{z}_{ir} \log c_{ir} \right) \end{aligned}.$$

Full Conditional for σ_c^2 The full conditional for σ_c^2 in Equation (11) is given by $p(\sigma_c^2 \mid \{c_{ir}\}, \gamma) = \mathrm{IG}(\hat{a}_c, \hat{b}_c)$, where

$$\begin{aligned} \hat{a}_c &= a_c + \frac{1}{2} nR \\ \hat{b}_c &= b_c + \frac{1}{2} \sum_{i=1}^{n} \sum_{r=1}^{R} \left(\log c_{ir} - \boldsymbol{z}_{ir}' \gamma \right)^2 \end{aligned}.$$

We set the prior for σ_c^2 to $\mathrm{IG}(a_c, b_c)$ with $a_c = 0.01$ and $b_c = 0.01$.

Full Conditional for λ The full conditional for λ in Equation (12) is a normal distribution because the likelihood relative to λ is normal and we set the prior to $\lambda \sim \mathrm{N}(m_\lambda, s_\lambda^2)$ with $m_\lambda = 0$ and $s_\lambda^2 = 10^2$. Specifically, $p(\lambda \mid \{\chi_i\}, \sigma_\chi^2) = \mathrm{N}(\hat{m}_\lambda, \hat{s}_\lambda^2)$, where

$$\begin{aligned} \hat{s}_\lambda^{-2} &= s_\lambda^{-2} + n\sigma_\chi^{-2} \\ \hat{m}_\lambda &= \hat{s}_\lambda^2 \left(s_\lambda^{-2} m_\lambda + \sigma_\chi^{-2} \sum_{i=1}^{n} \log \chi_i \right) \end{aligned}.$$

Full Conditional for σ_χ^2 The full conditional for σ_χ^2 in Equation (12) is given by $p(\sigma_\chi^2 \mid \{\chi_i\}, \lambda) = \mathrm{IG}(\hat{a}_\chi, \hat{b}_\chi)$, where

$$\begin{aligned} \hat{a}_\chi &= a_\chi + \frac{1}{2} n \\ \hat{b}_\chi &= b_\chi + \frac{1}{2} \sum_{i=1}^{n} (\log \chi_i - \lambda)^2 \end{aligned}.$$

We set the prior for σ_χ^2 to $\mathrm{IG}(a_\chi, b_\chi)$ with $a_\chi = 0.01$ and $b_\chi = 0.01$.