Recommendation in Offline Stores: A Gamification Approach for Learning the Spatiotemporal Representation of Indoor Shopping

Jongkyung Shin*
Changhun Lee*†
Chiehyeon Lim†
shinjk1156@unist.ac.kr
messy92@unist.ac.kr
chlim@unist.ac.kr
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea

Yunmo Shin
Junseok Lim
ymshin@retailtech.co.kr
sharpjs@retailtech.co.kr
Retailtech Co., Ltd.
Seoul, Republic of Korea

ABSTRACT

With the current advancements in mobile and sensing technologies used to collect real-time data in offline stores, retailers and wholesalers have attempted to develop recommender systems to enhance sales and customer experience. However, existing studies on recommender systems have primarily focused on e-commerce platforms and other online services. They did not consider the unique features of indoor shopping in real stores such as the physical environments and objects, which significantly affect the movement and purchase behaviors of customers, thereby representing the "spatiotemporal contexts" that are critical to identifying recommendable items. In this study, we propose a gamification approach wherein a real store is emulated in a pixel world and a recurrent convolutional network is trained to learn the spatiotemporal representation of offline shopping. The superiority and advantages of our method over existing sequential recommender systems are demonstrated through a real-world application in a hypermarket. We believe that our work can significantly contribute to promoting the practice of providing recommendations in offline stores and services.

CCS CONCEPTS

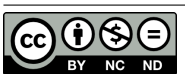
• **Information systems** → **Recommender systems**; **Location based services**.

KEYWORDS

interactive recommender system, offline stores, indoor shopping, spatiotemporal representation, gamification, recurrent convolutional network, reinforcement learning

ACM Reference Format:
Jongkyung Shin, Changhun Lee, Chiehyeon Lim, Yunmo Shin, and Junseok Lim. 2022. Recommendation in Offline Stores: A Gamification Approach for Learning the Spatiotemporal Representation of Indoor Shopping. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539199

*Equal contribution
†Corresponding author



KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08.
https://doi.org/10.1145/3534678.3539199

1 INTRODUCTION

Recommender systems are one of the most popular and successful applications of data science. By learning the purchase records, a recommender system can support customers in searching for diverse items based on their various needs and implicit preferences [14, 41]. Numerous studies have applied traditional and modern data science techniques to develop real-world recommender systems that can learn customer needs and preferences for movies, books, and other items listed in e-commerce services [34, 39]. Owing to the flexibility of online environments that interact with customers in real-time, online services can immediately recognize the context of a customer's needs and promptly offer personalized recommendations [15]. It has been demonstrated that these advanced systems significantly improve customer experience and engagement in online services, thereby increasing profits [26, 41]. However, this is rare in "offline" actual stores (i.e., retail and wholesale stores). Although some related studies have investigated the simple application of traditional collaborative filtering techniques to identify recommendations for retail customers [27, 31], to the best of our knowledge, no study has demonstrated the successful use of advanced data science to interactively identify recommendable items for customers in real-world offline stores.

Unlike the simplified "online environment" of e-commerce websites and mobile applications, a recommender system for offline stores must consider the "offline environment," wherein customers are required to make physical movements that are constrained to the dynamics of offline shopping.[1] Thus, indoor shopping in real stores involves the following three unique features that pose challenges to the collection and learning of data. First, traces of a focal customer in an offline store form a unique item purchase sequence *and* route. Although two customers may purchase the same sequence of items, they typically use different routes. Second, customers interact with a store environment dynamically and in

[1]Customers in offline stores must make unavoidable movements and are constantly and sequentially exposed to the items that are not included in their original purchase plans during the movement. The physical constraints cause delays in accessing recommended items, and constantly expose customers to other items during their movements through the stores. These delays and exposure represent the "contexts" crucial for recommending specific items to customers when they are moving to access the target items. Therefore, in offline stores, the items near to the customer's current location can be considered for identifying potential recommendations.

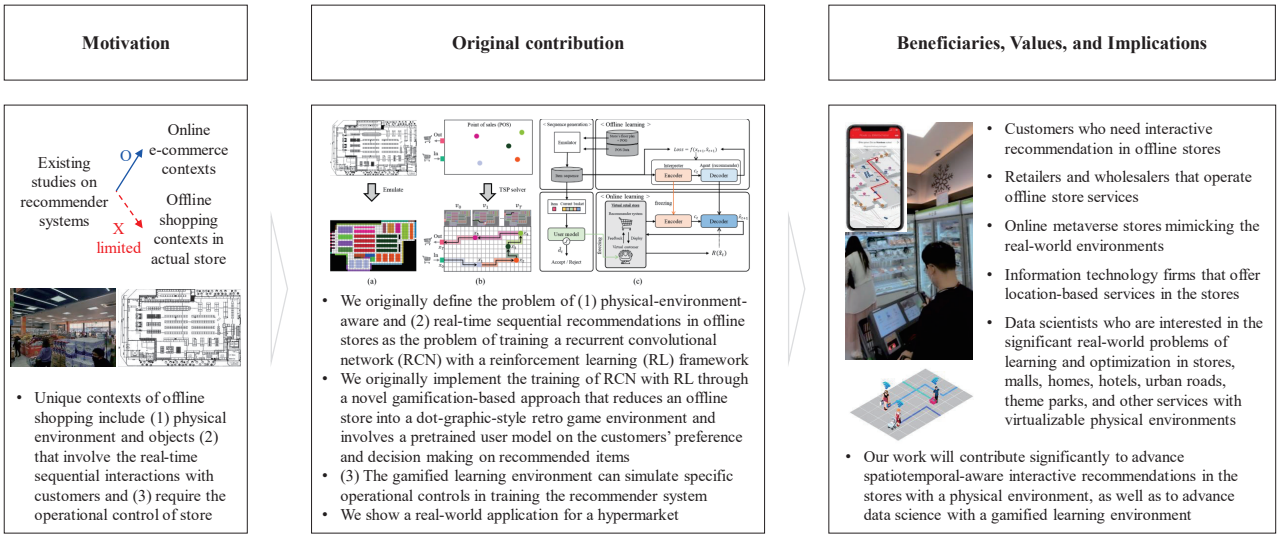


Figure 1: Overview and contribution of this work

real-time. For example, they pass by display shelves in a store environment while avoiding or creating crowded areas. Third, offline stores require operational control (e.g., control on congestion and item prices) depending on the time, item availability, and other factors of sales and operations. This operational control affects customer behavior, and offline stores have specific intentions to control indoor customers. Therefore, an offline store recommender system must be able to accommodate the contexts of (1) spatiotemporal patterns of customers and objects in physical environments, (2) real-time sequential interactions of customers constrained by the dynamics of offline shopping, and (3) operational control of sales from the perspective of retailers or wholesalers. However, existing studies on recommender systems have not fully considered these unique contexts of offline shopping.

Advancements in sensing technologies have enabled the collection of customer and physical environment data of offline stores. This includes the use of smart devices, such as indoor sensing systems [4] and smart shopping carts [28, 29], that interact with customers in the stores to enhance their shopping experience. Accordingly, interactive recommender systems mounted on smart devices are expected to improve the shopping experience of customers and enhance sales in offline stores [29]. However, the development of interactive recommender systems stores remains challenging because such smart devices are available only at a few stores.

Therefore, the challenge arises of developing an interactive recommender system that is device-free but can learn the spatiotemporal representations of indoor shopping in real stores, considering the aforementioned contexts from (1) to (3). For this purpose, we propose a gamification approach in which a recommender system is trained in a virtual environment to learn the spatiotemporal representation of offline shopping. The working of our gamification approach is as follows:

- First, it converts an offline store to a virtual environment (a dot-graphic-style retro game environment similar to Atari) by creating a 2D pixel image that emulates the floor plan of stores. This virtual environment is denoted as a pixel world.
- Second, a virtual customer that simulates the customer shopping process in an actual store is introduced, which is called the user model. It consists of two functions: navigation and decision-making. The navigation function is implemented by the A^*-algorithm [13] and the decision-making function by a multi-layer perceptron (MLP).
- Third, a recommendation model is implemented based on a recurrent convolutional neural network (RCN) [2, 8] that represents the spatiotemporal nature of the customers' shopping. The spatial information is represented using 2D pixel images and the temporal information is constructed by overlapping these images. Each image is input into a convolution neural network (CNN) [19] and encoded as the latent context. Gated recurrent units (GRUs) [6] then sequentially decode the latent context into the recommendable items.
- Fourth, an interaction between the user and recommendation model was modeled to identify customer preferences and the dynamics of offline shopping. The derivation of interactive training is analogous to the form of REINFORCE algorithm [36]; therefore, the recommendation model can control the customers' shopping behaviors regarding sales operations, and can implement and test any operational scenario.

This is an original study to develop an interactive recommender system that fully considers the spatiotemporal nature of indoor shopping (see Figure 1). Its primary academic and technical contribution is to connect modern machine learning techniques with emerging systems and collectable data in the retail and wholesale industries (see Section 2). Specifically, we defined the development of a spatiotemporal-aware recommender system for offline stores as a problem of interactive RCN learning and successfully solved it through a novel gamification-based approach (see Sections 3 and 4). The proposed approach was validated through a comparative experiment with existing methods (see Sections 5 and 6).

2 BACKGROUND

The development of a recommender system that suggests suitable content based on a user's taste has attracted considerable attention from both industries and academia. However, studies on recommender systems have been mostly limited to online setups, and their extensions to offline setups have rarely been addressed. In this section, we briefly discuss the characteristics of recommender systems in offline stores, and introduce previous studies related to the key concepts of our work.

2.1 Recommender systems for offline stores

A critical factor in increasing customer satisfaction and store sales is to recommend items suitable for customers who are navigating around the store. Most of the previous studies used online logged data to develop recommender systems for an offline setup, ignoring the peculiarities of shopping in offline stores [31]. Although Dlugolinsky et al. [7] proposed a recommender system for brick-and-mortar retail stores without online logged data, it was limited to small retailers, where one-on-one communications with clerks are possible. In relation to this work, the indoor shopping support systems [3] have emerged as the breakthrough for the recommendations in an offline setup. These systems interact with customers in real-time based on portable devices such as smart shopping carts [28]. Meanwhile, a recommender system that suggests a product by analyzing product images taken through a mobile application or real-time video transmitted through AR devices has been proposed as well [1, 33]. These studies show that recommendations in offline stores should occur in real-time while shopping, and there is a need to suggest an item of interest based on the spatiotemporal information of customers currently navigating in stores.

2.2 Spatiotemporal-aware recommendations

As real-time and location-based services flourish, it becomes crucial to develop an interactive recommender system that considers the spatiotemporal context associated with user locations. For example, studies on the next point-of-interest recommendations have attempted to extract spatiotemporal features and exploit them to develop a recommender system [21, 23, 37, 38]. Most of these studies focused on addressing temporal information; therefore, most of them primarily used recurrent neural networks (RNNs), such as long short-term memory and GRUs. Other related studies include taxi route recommendation using CNNs and deep reinforcement learning [17], transportation recommendations based on graph convolution and self-attention [22], and spatiotemporal-aware app recommendations through a probabilistic framework [12].

2.3 Gamification of learning

Although there have been various attempts to introduce spatiotemporal awareness in a recommender system [23, 38], to the best of our knowledge, limited studies have investigated offline store recommendations. Moreover, in such studies, the recommendation itself is not derived as a result of learning, but rather depends on a smart device. For example, a recommender system is trained using point-of-sale data (POS data) and is mounted on a smart device provided in the store. Given that POS data only represent the purchase history of customers, the spatiotemporal aspect of the recommendation is fully dependent on the telecommunication technology of the smart device and not the learning of the recommender system.

In contrast, our study uses a learning-based approach by leveraging the concept of gamification. Specifically, we created an image that emulates the environment of a real offline store. The spatial information of an offline store was expressed in pixels, that is, the offline store was represented in a dot-graphic-style retro game environment like Atari (See Figure 2). Additionally, a user model that simulates the shopping behavior of customers in a store was introduced into the game environment. The user model identifies the change in customer location with a sequence of consecutive images, after which the temporal information is constructed by overlapping multiple images according to time, which is analogous to the structure of a video frame (see Figures 2 and 3). Finally, the recommender system is regarded as an agent that interacts with the game environment. Thus, the recommender system can be trained to suggest an item based on spatiotemporal information discovered from the "in-game" situation. For a better understanding of the concept of our gamification approach for learning, please refer to [24, 40].

3 METHOD

This section describes our proposed approach to provide recommendations in offline stores and presents the technical details of the specific modules.

3.1 Emulator

The essence of our study is that a recommender system must consider the spatiotemporal nature of offline shopping. Because shopping trajectories of customers are determined by changes in physical locations over time, it seems natural to train the recommender system using their shopping trajectory data. However, real shopping trajectories cannot be obtained owing to two challenges. First, we must install as many sensors as shelves in an offline store to collect reliable trajectories. However, installing new sensors has considerable costs. Second, we must identify which items are selected by customers and at which locations. Given that a recommendation occurs at the item level, the spatiotemporal information captured by sensors must be combined with item information, which is available only through the use of a smart shopping cart that communicates with the sensors in real-time. However, only a few stores use smart-shopping carts. To overcome these practical challenges, we propose building an emulator, that is, a virtual environment that imitates a real offline store, and use it to train the recommender system. The emulator consists of the following three components:

- *Pixel world*: A virtual environment that imitates the physical environment of a real offline store, such as the types of shelves, absolute positions of items, and relative distances between them.
- *Item sequence*: A sequence of items generated by the travelling salesman problem (TSP) solver implemented using Google's OR-tools [9]. When a POS dataset is available, the TSP solver can align the items such that the total pixel distances between items in the pixel world are minimized. This represents the shopping trajectories of customers.

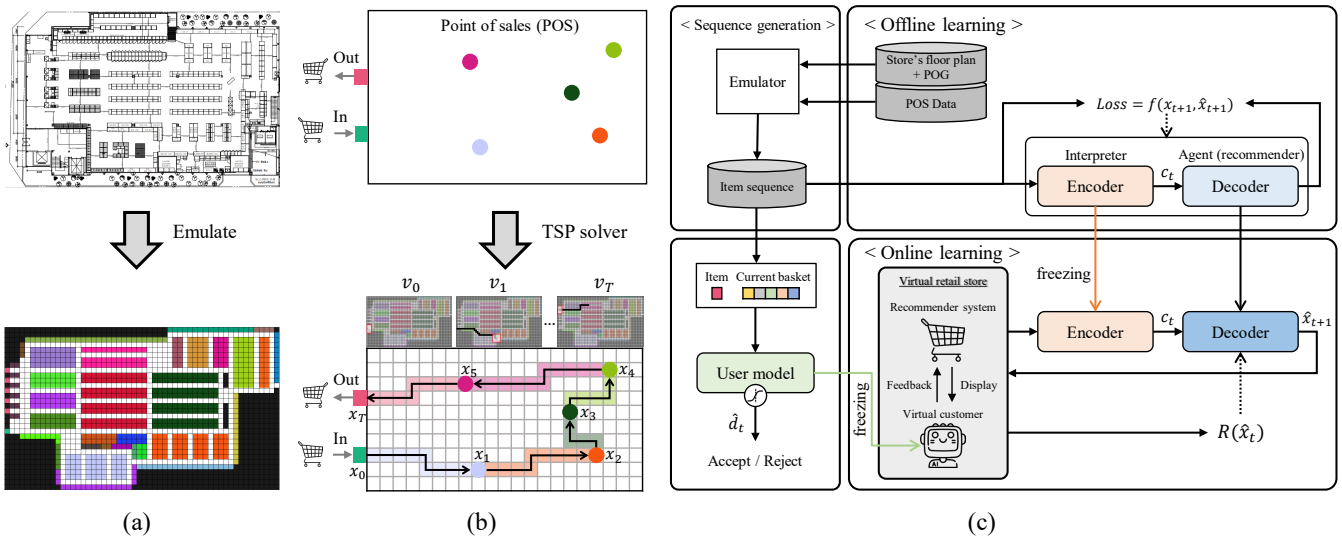


Figure 2: Details of emulator and recommender system. (a) Illustration of a pixel-world that imitates the floor plan of offline stores. The colors and coordinates represent the type of shelves and locations of items. (b) Creation of item sequence and video stream that connect the entrance and exit using the shortest path. (c) The proposed framework of a recommender system for offline stores consisting of two stages: offline and online learning.

- *User model*: A virtual user that simulates the shopping behavior of a customer inside the store. This component consists of two functions: navigation and decision-making. The user model moves to the adjacent pixel based on the A*-algorithm, and when it discovers an item that it planned to purchase or faces a recommended item, it decides whether to accept or reject it.

When the emulator is used as the training environment for a recommender system, the system is regarded as an agent that interacts with the three components described above and learns to identify an item while considering the spatiotemporal nature of offline stores. Further details of the learning mechanisms are provided in the following sections.

3.2 Item sequence

Considering that $\mathbf{x}^{(i)} = \{x_1, x_2, ..., x_P\}$ is a set of total P items purchased by customer i, it can be transformed into an item sequence $\mathbf{x}^{(i)}_{1:T} = [x_1, x_2, ..., x_T]$ by aligning the items in an order such that the distance between items in the pixel world is minimized. x_t is the t-th item represented as the token, and T is a fixed sequence length based on customer j who purchased the maximum number of items. Therefore, $\mathbf{x}^{(i)}_{1:T}$ always satisfies $P \leq T$ for any $i \neq j$, whereas a part of the sequence corresponding to $T - P$ is filled with $<$ pad $>$ tokens. As described in the previous section, the alignment of items is executed by a TSP solver, which marks the location of items in the pixel world and connects them with the shortest path from the entrance (In) to the exit (Out), as illustrated in Figure 2(b). The generated item sequence suggests an efficient shopping trajectory for customers.

3.3 Video stream data format

Because the purpose of our study is to develop a spatiotemporal-aware recommender system for offline stores, a data format that can contain both spatial and temporal information is required. Thus, the item sequence $\mathbf{x}_{1:T}$ is represented by a stream of frames. Each tth frame $v_t \in \mathbb{R}^{H \times W \times D}$ is an image, where H, W, and D denote the height, width, and depth of the image, respectively. v_t illustrates a partial shopping trajectory by marking consecutive pixels that connect the locations of x_{t-1} and x_t using the shortest path (see Figure 2(b)). Hence, v_t can be considered as a function that maps x_{t-1} to x_t, and $\mathbf{v}_{1:T} = [v_1, v_2, ..., v_T]$ indicates the video stream that describes a shopping trajectory of length T (see Figure 3), in which spatiotemporal information is expressed by changes in pixel values.

3.4 User model

The user model consists of two functions (navigation and decision making) and simulates the customer shopping experience in an actual store. The navigation function enables pixel-wise movement based on the A*algorithm:

$$\hat{\tau}_t \sim \mathrm{A}^*(\hat{x}_t, x_{t-1}) \,,$$

where $\hat{x}_t$ is a random item located some pixels away from x_{t-1}, and $\hat{\tau}_t$ is the shortest trajectory connecting $\hat{x}_t$ and x_{t-1}. In our gamification setting, $\hat{x}_t$ can be given as the recommended item. It means that $\hat{\tau}_t$ is stochastic according to $\hat{x}_t$, so is the pixel distribution of v_t:

$$\hat{v}_t \sim \mathop{\mathbb{E}}_{\hat{\tau}_t \sim A^*}[v_t|\hat{\tau}_t] = \int p(\hat{\tau}_t) p(v_t|\hat{\tau}_t)\, d\hat{\tau}_t$$
$$\propto \int p(\hat{x}_t) p(v_t|\hat{x}_t)\, d\hat{x}_t \,. \quad (1)$$

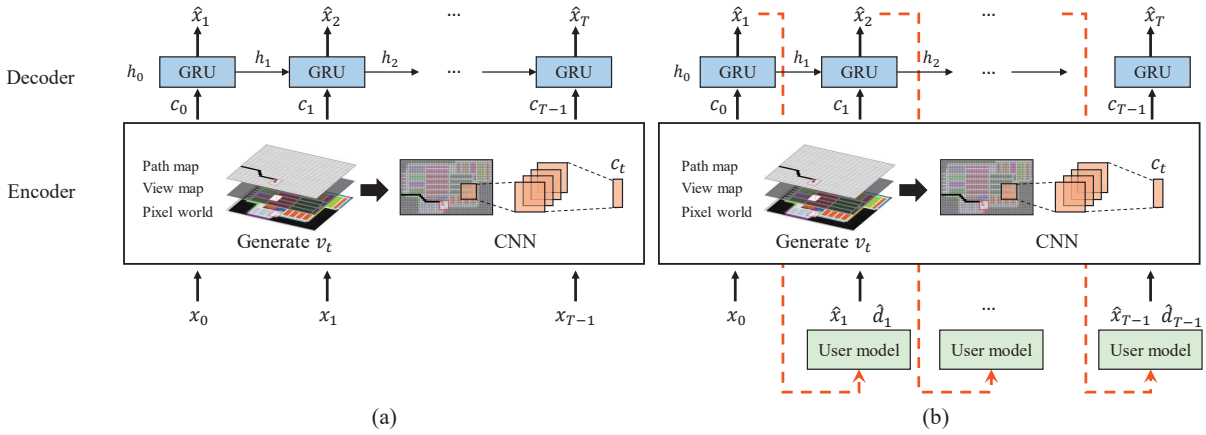


Figure 3: The training process: (a) Offline learning for spatiotemporal-awareness. (b) Online learning for interactive training.

Meanwhile, the decision-making function is implemented based on the multi-layer perceptron (MLP) network:

$$\mathbf{e}_t = \text{Embed}\,(x_t|\phi)$$
$$\mathbf{e}_{1:t} = \text{Concat}\,(\mathbf{e}_{1:t-1}, \mathbf{e}_t)$$
$$d_t = \text{MLP}\,(\mathbf{e}_{1:t}|\phi)\ ,$$

where ϕ is the learnable parameter, $\mathbf{e}_t \in \mathbb{R}^d$ is the d-dimensional embedding vector of x_t, and $\mathbf{e}_{1:t} \in \mathbb{R}^{d\times t}$ is the embedding matrix in which the embedding vectors are stacked up to x_t. Thus, the embedding matrix (referring to a stack of items) indicates the status of the customer basket. Moreover, d_t is a binary indicator that is equal to 1 if the t-th item is concatenated right, and 0 if not.[2] Right concatenation means that x_t is stacked right after $\mathbf{x}_{1:t-1}$, and not elsewhere. The MLP is then trained to predict $\hat{d}_t$ using fake sequences, item sequences, and corresponding targets $d_t = \{0, 1\}$.[3] A fake sequence is created by randomly shuffling the tokens from the item sequence. The following binary cross-entropy loss is applied to optimize the user model:

$$J(\phi) = \sum_{t=1}^{T} d_t \log \hat{d}_t = \sum_{t=1}^{T} \log p_\phi(d_t|x_t, \mathbf{e}_{1:t-1})\ . \tag{2}$$

3.5 Recommendation model

To ensure that the recommender system is aware of spatiotemporal representations, we built a recommendation model based on RCN, which extracts the spatial feature c and temporal feature h through CNN and GRU, respectively:

$$c_{t-1} = \text{CNN}\,(v_{t-1}|\theta)$$
$$o_{t-1}, h_{t-1} = \text{GRU}\,(c_{t-1}, h_{t-2}|\theta)$$
$$x_t = \text{Softmax}\,(o_{t-1})\ ,$$

where θ is the learnable parameter, and o_{t-1} is an output vector that captures the spatiotemporal information of a partial shopping trajectory from x_{t-2} to x_{t-1}. The softmax function then converts the output vector into a probability. The following categorical cross-entropy loss (XE-loss) is applied to optimize the recommendation model:

$$J(\theta) = \sum_{t=1}^{T} x_t \log \hat{x}_t = \sum_{t=1}^{T} \log p_\theta(x_t|v_{t-1}, h_{t-2})\ , \tag{3}$$

where $|I|$ is the number of items to consider and $\hat{x}_t \in \mathbb{R}^{|I|}$ is the recommendation probability of x_t. From a sampling perspective, $\hat{x}_t$ can be identified such that the recommended item is x_t. Additionally, we used the backpropagation-through-time (BPTT) algorithm [35] to optimize the model at the stream (sequence) level and not at the frame (token) level.

3.6 Interactive training

The ideal situation for a customer shopping in offline stores is that the recommendations occur in real time based on their basket information. The status of a basket not only represents their preference but also the dynamic conditions of offline shopping. As explained in Section 3.4, the decision-making function of the user model depends on the basket. Therefore, the interaction between the user and recommendation model is considered. The user model p_ϕ and recommendation model p_θ can be described as path integrals,

$$\hat{d}_t \sim \int p(x_{1:t}) p_\phi(d_t|x_{1:t})\, dx_{1:t} \tag{4}$$
$$\hat{x}_t \sim \int p(v_{t-1}) p_\theta(x_t|v_{t-1})\, dv_{t-1}\ . \tag{5}$$

If p_ϕ is provided to interact with p_θ, we can train p_θ such that the recommender system is executed as in the real situation. The interaction between p_ϕ and p_θ is then modeled as

$$J(\phi, \theta) = \int p_\phi(\hat{d}_t = 1|x_{1:t-1}, \hat{x}_t) \circ p_\theta(\hat{x}_t|v_{t-1})\, d\hat{x}_t\ ,$$

by integrating Equations (4) and (5) into the recursive mechanism and marginalizing it with respect to $\hat{x}_t$.[4] Note that we only consider

[2] d_t indicates a decision of user model of whether to accept or reject an item given t-th status of basket.

[3] The MLP network is designed to receive a full basket with the $\mathbf{e}_{1:T} \in \mathbb{R}^{d\times T}$ dimension. Note that masking is applied to not consider the time step from $t+1$ to T when the t-th item is stacked into the basket.

[4] A "recursive mechanism" refers to an online learning setup that utilizes predicted values as target values.

the case when $\hat{d}_t = 1$; thus, the recommendation model is updated only if the user model accepts its prediction. The objective function of interactive training is then derived as,

$$\nabla_\theta J(\phi, \theta) = \mathop{\mathbb{E}}_{\hat{x}_t \sim p_\theta} [\sigma(\hat{x}_t) \nabla_\theta \log p_\theta(\hat{x}_t | v_{t-1})] \tag{6}$$

where

$$\sigma(\hat{x}_t) = p_\phi(\hat{d}_t = 1 | x_{1:t-1}, \hat{x}_t) \ .$$

The recommendation model adapts to the preferences of a customer through fine-tuning by $\sigma(\cdot)$, as reflected in the user model. Specifically, p_θ and p_ϕ depend on the conditional and joint probability distributions of the item sequences, respectively, and $\sigma(\cdot)$ mixes tokens by feeding $\hat{x}_t$ to p_θ with an agreement of p_ϕ. To implement interactive training, we devised an algorithm called *Generative Token Mixing* (see Appendix for details). Because the algorithm is executed online, the length of shopping T varies depending on the acceptance rate of the user model. Moreover, according to Equation (1), $\nabla_\theta J(\phi, \theta)$ is inversely proportional to A^*. This suggests that the recommendation model changes the behavior of the user model. For example, by accepting a recommendation, the user model may no longer follow the shortest path. Note that the decision-making function of the user model was set to follow the A^* algorithm with 80% probability and random decision with 20% probability during interactive training.

4 SALES OPERATIONS

Customer stay time and total purchases are key factors related to sales and operations in offline stores. Thus, a recommender system can function as a store management tool by reducing the length of the shopping trajectory, promoting the purchase of expensive items, or implementing other control strategies. This section describes how reinforcement learning (RL) can control recommender systems for such sales operations purposes. We demonstrate how our proposed model can influence the shopping behavior of customers related to sales operations.

The proposed recommendation model considers the spatiotemporal nature of offline shopping in a dynamic manner, based on interactions with the user model. This means that recommendations can be controlled to influence shopping behaviors of customers from both spatial and temporal perspectives. Because RL is a framework in which an agent interacts with an environment repeatedly, gamification settings are suitable for using RL.[5] As in [5, 20], we extend Equation (6) to the REINFORCE algorithm [36], which is the on-policy RL method,

$$\nabla_\theta J(\phi, \theta) = \mathop{\mathbb{E}}_{\hat{x}_t \sim p_\theta} [R(\hat{x}_t) \sigma(\hat{x}_t) \nabla_\theta \log p_\theta(\hat{x}_t | v_{t-1})] \ ,$$

by adding $R(\cdot)$, which is a reward function that returns the numerical value as a control signal. For example, a reward function can be designed to provide higher scores with a decrease in shopping time or an increase in the total purchase amount. In this case, the reward is observed at the end of the shopping, while the BPTT algorithm allows the parameters to be updated such that rewards can be observed at every t-th step. As a result, the recommender system learns to offer as few items (to minimize shopping time) or as many items as possible (to maximize the total purchase). Thus, we can implement and test any operational scenario based on the proposed gamification-based approach. We conducted an analysis with regard to the controlled recommendation, and its results are presented in Section 6.

[5] Assume that the pixel world and user model comprise an environment, and the recommendation model is an agent.

5 EXPERIMENT

In this section, we describe the extensive experiments performed using the proposed approach. Our source code and datasets can be accessed from our GitHub repository for reproduction.[6]

5.1 Dataset and learning environment

In this study, we used two types of data obtained from a hypermarket in South Korea. The first were the logged POS data, which were randomly collected from 2021/7/1 to 2021/12/31. The total number of transaction logs and products was 73,014 and 32,940, respectively. After sorting the logs chronologically to create trajectories, they were divided for use in training, validation, and test in offline learning in a 10:1:1 ratio. The maximum length of each trajectory was set to 20 and padding was added if the number of items in the trajectory was less than 20. To train the decision-making function of the user model, trajectories that contained frequent purchase itemsets were used to reproduce the general purchasing behavior of customers.[7] Thirty frequent purchase itemsets were extracted using the FP-growth algorithm [11], with the minimum support set to 0.001 and the number of items set to 3. It should be noted that the user model was trained on data that were not used to train the recommender system in an offline learning setup.

The second type of data, which include the floor plan of the store and a plan-o-gram, were used to design a pixel world that emulated the real hypermarket. Particularly, the floor plan was used to recreate the store structure and placement of shelves in a pixel world. To emulate the real store as realistically as possible, its overall size ratio and the distances between shelves were considered. Meanwhile, information unrelated to customer purchase behaviors, such as the location of storage and stairs, was excluded. Consequently, the grid world consisted of 32×54 pixels. Using the plan-o-gram, all products included in the POS data were mapped to the pixels considered as shelves, and each product was set such that it could be located in only one pixel.

5.2 Experimental setting

We performed both offline and online learning for our framework (see (a) and (b) in Figure 3). First, the proposed model was compared with baselines for offline evaluation. For training efficiency, pairs of video frames and item labels were created beforehand using an emulator and then offline learning was performed. Next, the results of online learning were evaluated to confirm whether the recommendation model could be learned through interactions with the user model in the pixel world and controlled through reward shaping. See the Appendix for details of the parameter and simulation settings.

[6] https://github.com/JK-SHIN-PG/gamification-offrec

[7] A frequent purchase itemset is defined as a frequently observed pattern that consists of some items.

Model	Item-brand-level relevance							Item-product-level relevance		
	HR@1	HR@5	Prec@5	NG@5	Prec@20	NG@20	MAP@20	NG@5	NG@20	MAP@20
PoP	0.0001	0.0175	0.0035	0.0073	0.0019	0.0137	0.0025	0.0073	0.0149	0.0124
SeqPoP	0.0044	0.0312	0.0062	0.0172	0.0040	0.0308	0.0042	0.0185	0.0385	0.0322
GRU4Rec	0.0073	0.0360	0.0072	0.0209	0.0044	0.0311	0.0056	0.0773	0.1855	0.0905
Caser	0.0014	0.0051	0.0018	0.0035	0.0021	0.0090	0.0023	0.1509	0.2727	0.1775
SASRec	0.0237	0.0374	0.0076	0.0303	0.0036	0.0389	0.0067	**0.4670**	**0.4799**	0.3085
Ours	**0.0296**	**0.0918**	**0.0196**	**0.0611**	**0.0107**	**0.0873**	**0.0161**	0.4203	0.4660	**0.5733**

Table 1: Performance comparison on offline learning

Baselines. Given their different focuses, it could be unreasonable to compare the proposed approach, which recommends items based on spatiotemporal information, with existing sequential recommendation methods, which use information on previously purchased items. Nonetheless, to confirm the recommendation performance and utility of our approach, we performed a comparison experiment with existing methods as baselines. Considering that it is difficult to collect behavioral data from identified users owing to privacy concerns and that many users utilize the smart devices (e.g., smart shopping carts) without logging-in, our model was designed to estimate recommendation policies for general customers in offline stores. Thus, user characteristics were not considered in the implementation of the following baselines:

- POP: All items are ranked in a descending order based on their popularity in whole sequences in the training dataset.
- SeqPOP: Items are ranked in a descending order based on their popularity in the sequence. The popularity of an item is updated sequentially.
- GRU4Rec [14]: This model utilizes GRU to capture long-term sequential behaviors for session-based recommendations.
- Caser [30]: This model employs the convolution operations on the embedding matrix to capture sequential information.
- SASRec [18]: This model leverages the Transformer architecture to capture the semantics in item purchase sequences.

Metrics. To evaluate our model and the baselines for offline learning, four metrics were employed: hit ratio (HR), precision (Prec) [10], mean average precision (MAP) [32], and normalized discounted cumulative gain (NG) [16]. These metrics evaluate whether the recommended items match the item-brand-level. Additionally, we calculated NG and MAP with modified relevance to confirm whether the recommended items matched at the item-product level. Note that the item-brand level identifies the specific producer of an item (e.g., the 250 milliliter chocolate milk produced by the company A vs. company B), whereas the item-product level includes items of a broad product category (e.g., milk vs. snack). The relevance of each recommended item can have a maximum of four points, and if all product categories (large, medium, and small) and item brand match, a full score is obtained, and if a level does not match, one point is deducted.[8] The evaluation was conducted by providing items individually in the trajectory and checking the ranks of the following items. Note that we executed the evaluation after $t = 3$; the first three steps were provided as the initial state of the recommender system, thereby informing it of users' context, such as the current location and route. Each experiment was repeated five times, and the average performance was calculated.

[8]For example, if only the large category of recommended product is matched with the target product, one point is awarded.

6 ANALYSIS OF RESULTS

6.1 Offline evaluation

Table 1 presents the results of the comparative study for both the item-brand and item-product levels. First, it indicates that the consideration of sequential patterns improves the recommendation performance. For example, sequential recommendation methods (SeqPoP, GRU4Rec, and SASRec) outperformed the popularity-based method (PoP) for all metrics. The only exception is that Caser performed worse than PoP at the item-brand level comparison. We believe that this is because the demographic information addressed in Caser was not available in the dataset with the privacy concerns and technical issues in the hypermarket. Second, our model outperformed all baselines at the item-brand level, at which customers actually identify a specific item and decide to purchase it. This result suggests that our gamification-based approach, which represents the spatiotemporal nature of indoor shopping, works more effectively than the models that consider only a sequential pattern of purchases. Meanwhile, it is noteworthy that SASRec outperformed our model at the item-product level for NG@5 and NG@20. This suggests that SASRec may focus on product-level patterns, while our model captures brand-level patterns by exploiting the information when items of different brands are displayed in different locations.

6.2 Online evaluation

This section demonstrates how a recommendation model influences a user model through interactive training and reward shaping. With the goal of reducing in-store congestion and increasing profits, the following reward function was designed to provide higher scores for lower shopping times or higher purchase amounts,

$$R(\hat{\mathbf{x}}_{1:T}) = (1-\lambda)\log \mathrm{TPR}_{\mathrm{scale}}(\hat{\mathbf{x}}_{1:T}) - \lambda \log \mathrm{LOS}_{\mathrm{scale}}(\hat{\mathbf{x}}_{1:T}) \,,$$

where λ is the control parameter, and TPR and LOS are the total price of (accepted) recommended items and length of shopping, respectively. We can change the degree of control according to λ. For training stability, the min-max scaler and logarithm are applied to TPR and LOS. As a result, we observed that an item was successfully recommended in a way that partially changes the behavior of the user model.

Figure 4 shows the changes in TPR and LOS during interactive training. It describes that the value of TPR (LOS) was successfully controlled to increase (decrease) when $\lambda = 0$ ($\lambda = 1$). Interestingly, the control effect over TPR was more significant if we set $\lambda = 0.5$. This implies that (1) the shopping time can be easily controlled because it is directly related to the spatiotemporal representations, (2) and the total purchase amount gets more controllable when the variance of shopping trajectories increases by control over shopping time. In this case, in order to increase TPR, the recommendation model suggests as many expensive items as possible.

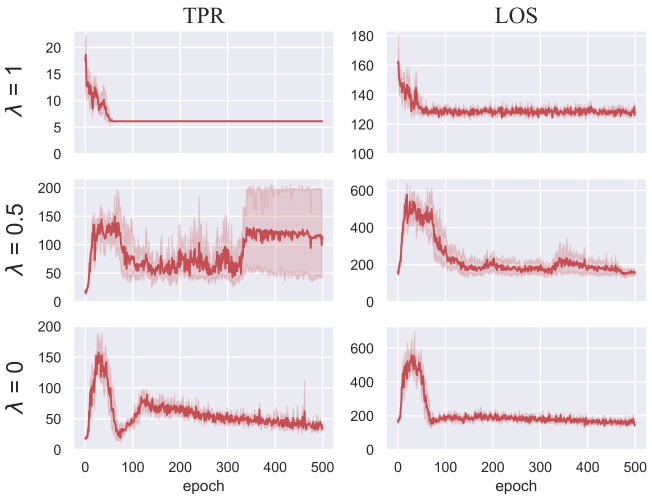


Figure 4: Changes in TPR and LOS according to λ

Table 2 reports the result of offline and online evaluation on the synthetic purchase plans. The synthetic purchase plans were obtained by sampling 100 frequent purchase itemsets at random (see Evaluation setting in Appendix A.1). We compared the performance of different recommendation models with respect to four metrics. Acceptance rate is the ratio of accepted recommendation, LOS is the number of pixels passed by user model, TPR marks user's total purchase in dollars, and log p indicates the diversity of recommendation.[9] The result shows that the LOS control works for the synthetic purchase plans as expected and reproduces that TPR is maximized at $\lambda = 0.5$.

	λ	Metric			
		Acceptance Rate (%)	LOS	TPR ($)	logP
Offline	-	**3.650**	153.13	19.50	-24.64
Online	1.0	2.288	**124.11**	6.14	**-0.003**
	0.75	2.288	**124.11**	6.14	-0.006
	0.5	3.215	157.78	**95.97**	-15.59
	0.25	2.033	137.96	29.16	-29.81
	0.0	2.083	126.38	31.46	-28.45

Table 2: The result of controlled recommendation

[9]The smaller log p, the more diverse recommendation.

7 DISCUSSION

In this study, we developed a novel approach for item recommendation in offline stores, which captures spatiotemporal contexts and considers sales operations, by using the RCN model and reinforcement learning under gamification. Our approach can be employed in offline stores where the location of customers can be traced in real time; this approach can be used even when the identification of customers is not possible. However, several limitations and issues need to be addressed for future research and applications. For example, we projected a three-dimensional real world onto a two-dimensional pixel world and placed each item in one pixel. As a result, multiple items were placed on one pixel, so our approach could not distinguish the spatial contexts between these items. Considering that heterogeneous spatial contexts between items might lead to improved recommendations, we will deal with this limitation in our future work. In addition, reinforcement-learning-based recommender systems generally have scalability issues due to the large action space. In our case, the number of unique items was not as large as that in public datasets collected from e-commerce websites[10], so we did not have to address the scalability issue seriously. However, to develop a recommender system for large offline stores where numerous items are displayed, the training efficiency and performance of our approach may need to be improved.

Meanwhile, in the real world, retailers change their sales operations frequently for various reasons, such as the changes in macroeconomic, managerial, and social conditions. However, we did not discuss whether our approach is adaptable to such a changing situation. In many cases, recommender systems are merely retrained when the sales operations change. However, this is inefficient because the model can forget all the existing trained information during the retraining process. In our future work, we will deal with this problem in a continuous learning manner without retraining the model to adjust to the changes in sales operations. Moreover, other practical challenges need to be addressed as well, such as the personalization of recommendations and the development of a streamlined emulator. The result of our work is based on a pretrained user model. The user model represents the taste of general customers, but an essence of recommendation is personalization. Therefore, extending our gamification approach to address the personalization issue would be an interesting topic. Additionally, implementing a streamlined engine that emulates the real world must be addressed in future research and application of our work. In consideration of the human efforts required to create a game environment, the development of an emulator that analyzes an input (e.g., a floor plan) and automatically returns an environment (e.g., a pixel world) will be increasingly important in the future.

8 CONCLUDING REMARKS

To the best of our knowledge, this study is the first to solve the unique data science problem of developing an interactive recommender system *specifically designed for offline stores*. An experiment using the real-world dataset collected from a South Korean hypermarket clearly demonstrated the superiority and advantages of the proposed approach compared with existing methods. Furthermore,

[10]For example, the number of unique items in Retailrocket and Tmall datasets is about 200K and 2M, respectively.

this study will create a real-world impact as the proposed approach will be used in a hypermarket chain in South Korea. Finally, our work can promote the use of data science in gamified learning environments, such as learning and controlling the patterns of people and objects in virtualized physical environments. As a result, we believe that our work will significantly contribute to many location-based services [3, 25] for offline stores, shopping malls, event venues, theme parks, production yards, and other physical environments that can be transformed into virtual environments.

REFERENCES

[1] Jesús Omar Álvarez Márquez and Jürgen Ziegler. 2020. In-store augmented reality-enabled product comparison and recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 180–189.

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432* (2015).

[3] Anahid Basiri, Elena Simona Lohan, Terry Moore, Adam Winstanley, Pekka Peltola, Chris Hill, Pouria Amirian, and Pedro Figueiredo e Silva. 2017. Indoor location based services challenges, requirements and usability of current solutions. *Computer Science Review* 24 (2017), 1–12.

[4] Luka Batistić and Mladen Tomic. 2018. Overview of indoor positioning system technologies. In *Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 0473–0478.

[5] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[7] Stefan Dlugolinsky, Giang Nguyen, Martin Seleng, and Ladislav Hluchy. 2017. Decision influence and proactive sale support in a chain of convenience stores. In *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*. IEEE, 000277–000284.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

[9] Inc. Google. 2021. Google's optimization tools (or-tools). https://developers.google.com/optimization/routing/tsp.

[10] Asela Gunawardana and Guy Shani. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* 10, 12 (2009).

[11] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining Frequent Patterns without Candidate Generation. *SIGMOD Rec.* 29, 2 (may 2000), 1–12. https://doi.org/10.1145/335191.335372

[12] Yong-Jin Han, Seong-Bae Park, and Se-Young Park. 2017. Personalized app recommendation using spatio-temporal app usage log. *Inform. Process. Lett.* 124 (2017), 15–20.

[13] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. arXiv:1511.06939 [cs.LG]

[15] Chao Huang, Xian Wu, Xuchao Zhang, Chuxu Zhang, Jiashu Zhao, Dawei Yin, and Nitesh V Chawla. 2019. Online purchase prediction via multi-scale modeling of behavior dynamics. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2613–2622.

[16] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[17] Shenggong Ji, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2020. Spatio-temporal feature fusion for dynamic taxi route recommendation via deep reinforcement learning. *Knowledge-Based Systems* 205 (2020), 106302.

[18] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. https://doi.org/10.48550/ARXIV.1808.09781

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[20] Changhun Lee, Soohyeok Kim, Chiehyeon Lim, Jayun Kim, Yeji Kim, and Minyoung Jung. 2021. Diet Planning with Machine Learning: Teacher-forced REINFORCE for Composition Compliance with Nutrition Enhancement. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3150–3160.

[21] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2009–2019.

[22] Hao Liu, Jindong Han, Yanjie Fu, Jingbo Zhou, Xinjiang Lu, and Hui Xiong. 2020. Multi-modal transportation recommendation with unified route representation learning. *Proceedings of the VLDB Endowment* 14, 3 (2020), 342–350.

[23] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the Web Conference 2021*. 2177–2185.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[25] Dillip Mohapatra and SB Suma. 2005. Survey of location based wireless services. In *Proceedings of the 2005 IEEE International Conference on Personal Wireless Communications, 2005 (ICPWC 2005)*.

[26] Changhua Pei, Xinru Yang, Qing Cui, Xiao Lin, Fei Sun, Peng Jiang, Wenwu Ou, and Yongfeng Zhang. 2019. Value-aware recommendation based on reinforcement profit maximization. In *The World Wide Web Conference*. 3123–3129.

[27] Bayu Yudha Pratama, Indra Budi, and Arlisa Yuliawati. 2020. Product recommendation in offline retail industry by using collaborative filtering. *International Journal of Advanced Computer Science and Applications* 11, 9 (2020), 635–643.

[28] Pranavi Satheesan, Jesuthasan Alosius, Rajaratnam Thisanthan, Priyanka Raveendran, and Janani Tharmaseelan. [n.d.]. Enhancement of Supermarket using Smart Trolley. *International Journal of Computer Applications* 975 ([n. d.]), 8887.

[29] Ryosuke Takada, Kenya Hoshimure, Takuya Iwamoto, and Jun Baba. 2021. POP Cart: Product Recommendation System by an Agent on a Shopping Cart. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 59–66.

[30] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. arXiv:1809.07426 [cs.IR]

[31] Kutuzova Tatiana and Melnik Mikhail. 2018. Market basket analysis of heterogeneous data sources for recommendation system improvement. *Procedia Computer Science* 136 (2018), 246–254.

[32] Andrew Turpin and Falk Scholer. 2006. User performance versus precision measures for simple search tasks. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 11–18.

[33] Georg Waltner, Michael Schwarz, Stefan Ladstätter, Anna Weber, Patrick Luley, Horst Bischof, Meinrad Lindschinger, Irene Schmid, and Lucas Paletta. 2015. MANGO-mobile augmented reality with functional eating guidance and food awareness. In *International Conference on Image Analysis and Processing*. Springer, 425–432.

[34] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, ShaoZhang Niu, and Jimmy Huang. 2020. KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 209–218.

[35] Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[36] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[37] Taofeng Xue, Beihong Jin, Beibei Li, Weiqing Wang, Qi Zhang, and Sihua Tian. 2019. A spatio-temporal recommender system for on-demand cinemas. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1553–1562.

[38] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhixu Li, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[39] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 95–103.

[40] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[41] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2810–2818.

A APPENDIX

This section provides further information on the experimental settings of this study. In addition, Figures 5 and 6 illustrate the convergence and the result of online learning by λ, respectively.

A.1 Implementation Details

Parameter setting. All models, including the baselines, were optimized using the Adam optimizer by setting the batch size to 1024. The learning rate was set to 0.001 and 0.0005 for offline and online learning, respectively. For the CNN of our model, the sizes of the padding, stride, and kernel were set to 1, 1, and 2, respectively. For our model and GRU4Rec, the number of hidden states and layers of the GRU were set to 256 and 2, respectively. For Caser, the Markov order was set to 3. The embedding size of the MLP in the user model was 32. Other hyperparameters were tuned according to the references of original studies.[11][12][13]

Simulation setting. In each episode for training, the user model was initialized to be located in a pixel corresponding to the entrance of the store, and three items sampled from frequent itemsets were regarded as the planned purchase items of the user model. For stable training, 10 episodes were conducted per epoch, and the recommendation model was updated by the average value from the result of the episodes. The training was iterated for 500 epochs.

Evaluation setting. For evaluation, 100 itemsets were constructed by sampling three items without duplication from 21 unique items extracted in frequent itemsets.[14] For each episode, one of 100 itemsets was considered to be planned as purchase items of the user model. After 100 episodes, we reported the average value of each metric.

A.2 Algorithm for interactive training

The goal of interactive training is to induce the recommender system to learn user preferences and adjust recommendation policies accordingly. To achieve this, three components must be prepared: a pre-trained recommendation model p_θ, a pre-trained user decision model p_ϕ, and customers' planned purchase $\mathbf{x}_{1:T} = [x_1, x_2, ..., x_T]$. Conceptually, p_ϕ is designed to capture user preferences, but it is trained to learn the joint probability distribution of items, that is, $p_\phi(\hat{d}_t|x_{1:t-1}, \hat{x}_t) \simeq p_\phi(x_1, ..., x_{t-1}, \hat{x}_t)$. Because p_θ learns the conditional probability distribution of the items, that is, $p_\theta(\hat{x}_t|v_{t-1}) \simeq p_\theta(\hat{x}_t|x_1, ..., x_{t-1})$, it is mathematically evident that $p_\theta(\hat{x}_t|x_1, ..., x_{t-1})$ is always greater than or equal to $p_\phi(x_1, ..., x_{t-1}, \hat{x}_t)$.[15]

However, in an online setup wherein the recommendation model interacts with the user model in real time, $p_\theta(\hat{x}_t|x_1, ..., x_{t-1})$ often decreases to a degree less than $p_\phi(x_1, ..., x_{t-1}, \hat{x}_t)$ because of the user's unexpected behavior. Specifically, because the data used in online learning are different from those used in offline learning, the user model may visit shelves that were not considered in offline learning. The recommendation model then faces a pixel distribution that is never observed during pretraining and recommends unlikely items. If we use all unlikely items for interactive training, the recommendation policy is corrected toward an undesirable distribution. However, if we exclude all unlikely items, the recommendation model will rarely be updated via interactive training.

Thus, we devised a method that uses only the recommended items that are considered acceptable by the user model. The proposed method is presented in Algorithm 1. Because the user model acts as a filter that mixes a sequence of planned items by inserting recommended items (i.e., recommended tokens), we refer to this *Generative Token Mixing* (GTM). Using this, interactive training can be achieved with maximal desirability and the recommendation policy can be fine-tuned to adapt to user preferences.

Algorithm 1 Generative Token Mixing (GTM)

Input: a set of items $\mathbf{x}$, an emulator $E(\cdot)$, pre-trained recommendation model p_θ, pre-trained user model p_ϕ and A*-algorithm $A^*(\cdot)$, learning rate α, the number of planned items T, basket size δ, the number of epochs N

1: $T = 3$, $\mathbf{x}_{1:T} = E(\mathbf{x})$ ▷ Section 3.2
2: // $\mathbf{x}_{1:T} = [x_1, ..., x_T]$; items planned to purchase.
3: $K = 0$, $\delta = 20$
4: **for** $t = 2, \ldots, T + K$ **do**
5: $v_{t-1} = E(A^*(x_{t-1}, x_{t-2}))$ ▷ Section 3.3 and Equation (1)
6: $\hat{x}_t = p_\theta(\cdot|v_{t-1})$
7: $\hat{x}_{1:t} = \text{Concat}(x_{1:t-1}, \hat{x}_t)$
8: $\hat{d}_t = p_\phi(\cdot|\hat{x}_{1:t})$
9: **if** $\hat{d}_t == 1$ **then**
10: $x_{t+1:T+1} \leftarrow x_{t:T}$ ▷ Push a sequence forward
11: $x_t \leftarrow \hat{x}_t$ ▷ Insert a recommended item in a sequence
12: $K = K + 1$
13: // Obtain new longer sequence $\mathbf{x}_{1:T+K}$ with K insertions,
14: // mixing tokens filtered by p_ϕ into a sequence of p_θ.
15: // K increases as p_θ adapts to p_ϕ.
16: **else**
17: **if** $t == T + K$ **then**
18: break;
19: **end if**
20: **end if**
21: **end for**
22: $J(\theta, K) \stackrel{\text{def}}{=} \sum_{t=1}^{T+K} \log p_\theta(x_t|v_{t-1}, h_{t-2})$ ▷ Equation (3)
23: $\nabla_\theta J(\phi, \theta) \leftarrow \nabla_\theta J(\theta, K)$ ▷ Instead of Equation (6)
24: $\theta \leftarrow \theta - \alpha\,(-\nabla_\theta J(\phi, \theta))$
25: // Repeat lines from 4 to 28 for N epochs.

A.3 Acknowledgements

Here we acknowledge the research grants used for this work[16].

[11] https://github.com/graytowne/caser_pytorch
[12] https://github.com/hungthanhpham94/GRU4REC-pytorch
[13] https://github.com/kang205/SASRec
[14] The number of combinations is ${}_{21}C_3 = 1,330$
[15] Consider $p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)$. Since $p(x_1)$ and $p(x_2|x_1) \in [0, 1]$, $p(x_3|x_1, x_2)$ is always greater than or equal to $p(x_1, x_2, x_3)$.

[16] This work was supported by the Industrial Technology Innovation Program (20009841, A Development on the Integrated Management System of Small and Medium Retail Stores for Service Productivity Innovation) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1I1A4A01049121). This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-01336: Artificial Intelligence Graduate School Program - UNIST, No. 2021-0-02068, Artificial Intelligence Innovation Hub).

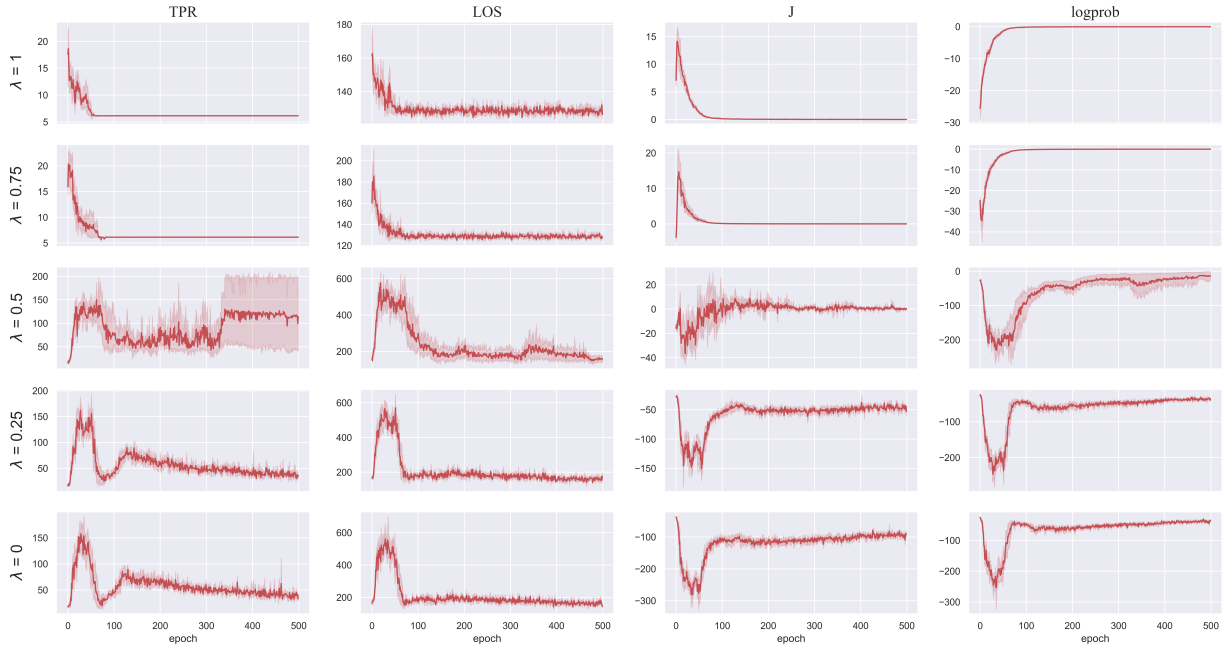


Figure 5: Recommendation control and policy convergence through online learning

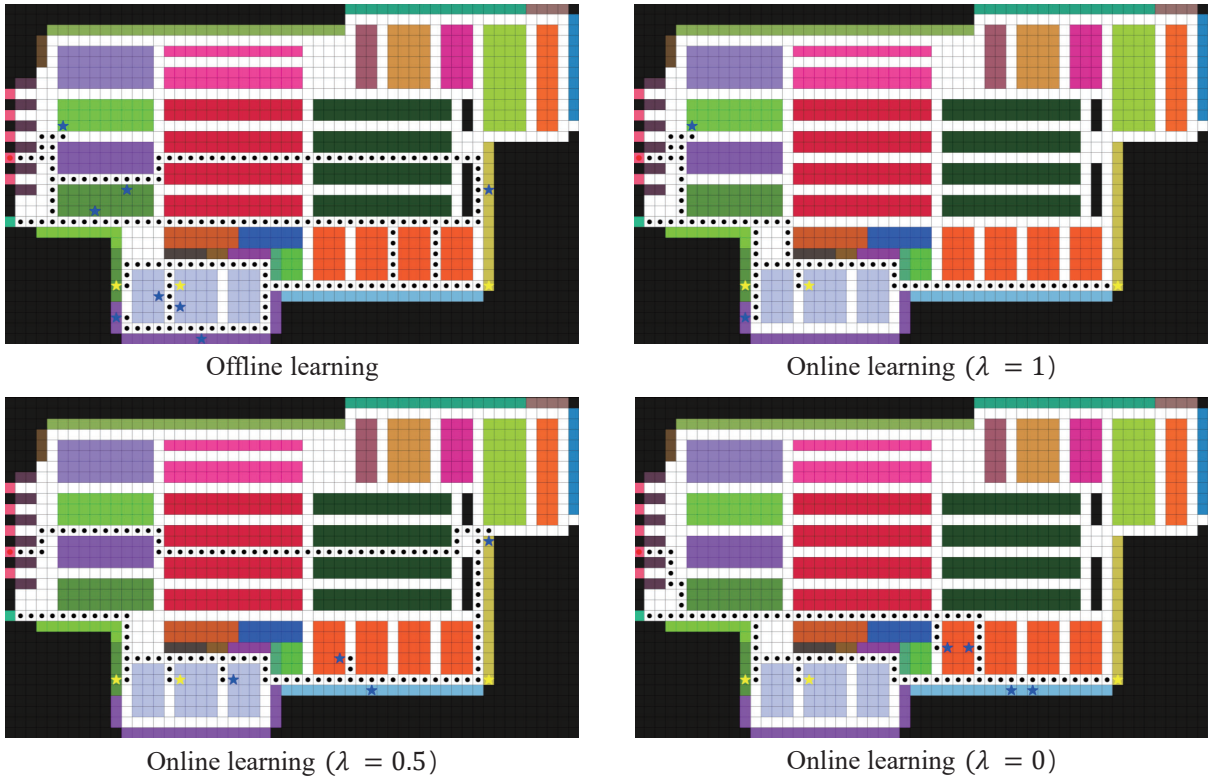


Figure 6: Shopping trajectories per λ: The yellow and blue stars represent the location of items that are planned to purchase and recommended items that are accepted, respectively. The black dot shows the customers' movement during their shopping.