DENOISING-ORIENTED DEEP HIERARCHICAL REINFORCEMENT LEARNING FOR NEXT-BASKET RECOMMENDATION[⋆]

Qihan Du, Li Yu[†], Huiyuan Li, Youfang Leng, Ningrui Ou

Renmin University of China, Beijing, China

ABSTRACT

Next basket recommendation aims to provide users a basket of items on the next visit by considering the sequence of their historical baskets. However, since a user's purchase interests vary over time, historical baskets often contain many irrelevant items to his/her next choices. Therefore, it is necessary to denoise the sequence of historical baskets and reserve the indeed relevant items to enhance the recommendation performance. In this work, we propose a **H**ierarchical **R**einforcement **L**earning framework **for** next **Ba**sket recommendation, named HRL4Ba, which learns the personalized inter-basket and intra-basket contexts of the user for dynamic denoising. Specifically, the high-level and the low-level agent in the denoising module perform hierarchical decisions, i.e., revise baskets and remove items; the recommendation module serves as the environment to give feedback to agents and recommends the next basket. Extensive experiments on two e-commerce datasets show the HRL4Ba outperforms existing state-of-the-art methods, and our ablation studies further show the effectiveness of each component in HRL4Ba.

Index Terms— Recommender systems, Reinforcement learning, Deep learning, Next-basket recommendation.

1. INTRODUCTION

With the accelerated transformation of market from offline to online, E-commerce has become increasingly prevalent. Next Basket Recommendation (NBR) [1] is an essential online service for many E-commerce platforms (e.g., Amazon, Taobao and Instacart), which recommends a basket of items to users to satisfy their needs. In the NBR task, given a user's purchase history, i.e., a sequence of baskets, our goal is to recommend the basket that the user may purchase next [2].

Generally, a user's next choices are highly dependent on the items she has purchased previously, e.g., buying several *accessories* for a *cellphone*. However, purchase in basket is a complex user behavior. A historical basket may contain many items that are irrelevant to the user's next choice, which dilute the impact of items that are indeed relevant. Therefore, it is a critical issue for NBR to automatically identify those relevant items in historical baskets to guide recommendation. Many efforts have been made to solve it. To distinguish the influences of different items within a basket, attention-based models such as the neural attention model (ANAM) [3] and the intention network (IntNet) [4] can be used to assign attention coefficients to intra-basket items as their importances to the user's next choice. However, since the item relevance is not considered, attention-based approaches rigidly assign weights to irrelevant items, introducing noises in the basket representation. Unsupervised learning-based methods can be used to explicitly model item relevance. For example, the basket relevance network (Beacon) [5] extracts pairwise correlation from the co-occurrence frequency of item pairs to predict a more coherent next basket, and the contrast learning encoder (CLEA) [6] maps items within a basket into opposing clusters in the feature space to classify relevant and irrelevant items.

Although these works have achieved progress, they still have limitations. Specifically, they often determine the item relevance based on a fixed policy, such as co-occurrence frequency [5] and similarity clustering [6], which cannot adequately follow the user's personalized inter-basket and intra-basket contexts. For example, given a target item *broom*, and two historical baskets $\{$*cellphone, mask, mop*$\}$,$\{$*candle, pumpkin*$\}$ of a user. Existing methods only regard the *mop* relevant to the *broom*, and the rest are noise; However, following the context about Halloween in adjacent baskets and items, except for the *mop*, the *mask*, *candle* and *pumpkin* are also relevant to the *broom*, and only the *cellphone* is noise. Thus, in order to precisely remove historical items that are noise to the target item, we need to model the hierarchical personalized contexts from the basket-level and the item-level.

Recently, reinforcement learning has made great success on various challenging tasks that require sequential modeling [7] and dynamic interaction [8]. Motivated by the above concerns, in this work, we propose a **H**ierarchical **R**einforcement **L**earning framework **for** next **Ba**sket recommendation named **HRL4Ba**, which consists of a denoising module and a recommendation module. Specifically, we formalize the denoising task as a hierarchical decision process. In the denoising module, the high-level agent observes the inter-basket context to decide which basket contains noise, while the low-level agent observes the intra-basket context to decide which items in that basket are noises and remove them. Then, the pre-trained rec-

[†] Li Yu is the corresponding author.

[⋆]This work is supported by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China, grant number: 21XNH178.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

ommendation module serves as the environment to give feedback to the agents and recommends the next basket based on the denoised user purchase history. Finally, two modules are trained jointly utilizing DPG [9], to unify denoising and recommendation. Our contributions are summarized as follows:

- We propose HRL4Ba, a hierarchical reinforcement learning-based denoising framework to improve the performance of the NBR. To the best of our knowledge, this is the first work for NBR empowered by reinforcement learning (RL).
- We formalize denoising as a hierarchical decision process that incorporates personalized inter-basket and intra-basket contexts, and design two agents to cooperate at basket-level and item-level for better denoising.
- Extensive experiments on two public E-commerce datasets show the effectiveness of the proposed HRL4Ba.

2. PROBLEM FORMULATION

Assume that an E-commerce platform with a set of users $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ and items $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$. For each user u, given his/her purchase history, i.e., a sequence of baskets $\mathcal{B}^u = (B_1^u, B_2^u, ..., B_t^u)$, where $B_t^u \subseteq \mathcal{V}$ is the basket he/she purchased at time t, our goal is to recommend the basket B_{t+1}^u to him/her at time $t+1$. Specifically, the $B_t^u = \{v_1^{u,t}, v_2^{u,t}, ..., v_m^{u,t}\}$ is a collection of items, where m is the size of the basket. For the denoising process, given the target item $v_j \in \mathcal{V}$ predicted by the pre-trained recommender, the denoised sequence of baskets for v_j is denoted as $\widetilde{\mathcal{B}^{u,j}}$. For convenience, we omit the user superscript u in the rest of the paper. We define the key notions of RL [10] as follows:

- **State**: The high-level state $s^h \in \mathbb{R}^d$ and the low-level state $s^l \in \mathbb{R}^d$ are dense vectors containing the observation of the current inter-basket and intra-basket context, respectively.
- **Action**: The high-level action a_t^h is a binary scalar {0,1} that indicates whether the basket B_t^u needs to be revised, and the low-level action a_m^l is a binary scalar {0,1} that indicates whether the item v_m^t should be removed from B_t^u.
- **Reward**: The agents will be given a delayed reward $R(s, a)$ after the entire sequence of baksets are denoised, i.e., the reward will be given equally to all non-zero actions a^h/a^l.

3. PROPOSED FRAMEWORK: HRL4BA

The overall architecture of HRL4Ba is shown in Fig.1. The denoising module takes the original sequence of baskets $\mathcal{B}$ and the target item v_j as input, then outputs the denoised sequence of baskets $\widetilde{\mathcal{B}^j}$ for v_j. The recommendation module gives the agents a delayed reward for denoising (i.e., the difference between $\widetilde{\mathcal{B}^j}$ and $\mathcal{B}$), and feeds the next target item v_{j+1} from the M candidates into the deniosing module. After M denoising iterations, the M candidates will be re-ranked due to the update of the recommendation probability, and we treat the top-m $(m \ll M)$ items from M candidates with the highest new recommendation probability as the next basket.

We first represent each item $v_i \in \mathcal{V}$ as a dense embedding vector $e_i \in \mathbb{R}^d$. Then, the user u's profile such as id, age, gender, etc., is also embedded as a dense vector $f \in \mathbb{R}^d$. Finally, each basket B_t be represented as a dense vector $b_t \in \mathbb{R}^d$, which will be detailed in Section 3.2.

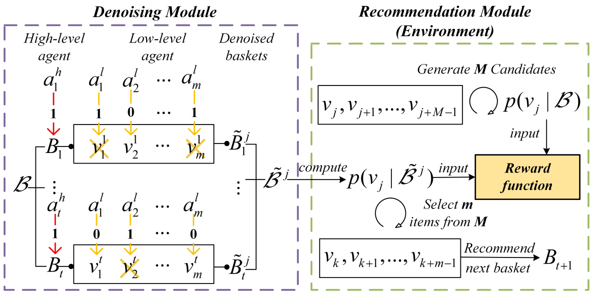

Fig. 1: Architecture of HRL4Ba.

3.1. Denoising Module

As we mentioned before, it is important to jointly model inter-basket and intra-basket contexts in order to denoise from two levels including basket-level (high-level) and item-level (low-level). So, the high-level agent and the low-level agent are designed to make hierarchical decisions for denoising.

High-level agent (HA). Given a sequence of baskets $\mathcal{B} = [B_1, B_2, ..., B_t]$ and a target item v_j, the high-level agent first determines which basket contains the noises.

(1)*High-level state encoder*. The RNN with gated recurrent unit (GRU) [11] is introduced to learn the inter-basket context. With the sequence of basket vectors $[b_1, b_2, ..., b_t]$ as input, the GRU^h generates the corresponding hidden states $[h_1^h, h_2^h, ..., h_t^h]$, i.e., $h_t^h = GRU^h(h_{t-1}^h, b_t)$. A linear layer is utilized to merge the current inter-basket context h_t^h with the target item vector e_j to produce the high-level state:

$$s_t^h = W_{s1}^h h_t^h + W_{s2}^h e_j + \phi_s^h \tag{1}$$

(2) *High-level Actor*. The high-level Actor generates the action to decide whether to revise the basket B_t. We feed s_t^h into a nonlinear layer, which serves as the high-level policy [12] function $\mu^h(s_t^h; \Theta^h)$ to obtain the action a_t^h:

$$\widehat{a}_t^h = \sigma(W_a^h s_t^h + \phi_a^h) \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function, and Θ^h is the parameters of the high-level actor network. Note that $\widehat{a}_t^h \in [0, 1]$ is a virtual action, we set a hard threshold to map it to a real action $a_t^h \in \{0, 1\}$, i.e., we set a_t^h =1 when $\widehat{a}_t^h \geqq 0.5$, it means that the high-level agent revises the basket, and a_t^h =0 otherwise.

(3) *High-level Critic*. The high-level Critic estimates Q values which represents the expected return after revising/not revising a basket. A nonlinear layer is treated as an approximator [13] for the high-level action-value function as follows:

$$\begin{aligned} q^h(s_t^h, a_t^h; \Psi^h) = ReLU\Big(& a_t^h \cdot (W_{q1}^h s_t^h + \phi_{q1}^h) \\ & + (1 - a_t^h) \cdot (W_{q2}^h s_t^h + \phi_{q2}^h)\Big) \end{aligned} \tag{3}$$

where Ψ^h is the parameters of the high-level critic network.

Low-level agent (LA). If a_t^h=1, it means that the basket $B_t = \{v_1^t, v_2^t, ..., v_m^t\}$ contains noises, and the low-level agent is required to determine which items are noise and remove them.

(1) *Low-level state encoder*. Similar to the state encoder of HA, another GRU^l is introduced to learn the intra-basket context via $h_m^l = GRU^l(h_{m-1}^l, e_m^t)$, where e_m^t denotes the m-th item vector in the t-th basket. Then, another linear layer is utilized to merge the current intra-basket context with the target item vector e_j to produce the low-level state:

$$s_m^l = W_{s1}^l h_m^l + W_{s2}^l e_j + \phi_s^l \tag{4}$$

(2) *Low-level Actor*. The low-level Actor generates the action to decide whether to remove v_m^t from B_t as noise. We feed s_m^l into another nonlinear layer, which serves as the low-level policy function $\mu^l(s_m^l; \Theta^l)$ to obtain the action a_i^l:

$$\widehat{a}_m^l = \sigma(W_a^l s_m^l + \phi_a^l) \tag{5}$$

where Θ^l is the parameters of the low-level actor network. Note that the virtual action $\widehat{a}_i^l$ will be mapped to the real action a_m^l in the similar way as the Actor of HA. Specifically, if $a_m^l = 1$, the low-level agent removes the item e_m^t from B_t; else $a_m^l = 0$, e_m^t is not removed.

(3) *Low-level Critic*. The low-level Critic estimates Q values which represents the expected return after removing/not removing an item. The low-level action-value function is approximated by the nonlinear layer as follows:

$$\begin{aligned} q^l(s_m^l, a_m^l; \Psi^l) = ReLU\Big(& a_m^l \cdot (W_{q1}^l s_m^l + \phi_{q1}^l) \\ & +(1 - a_m^l) \cdot (W_{q2}^l s_m^l + \phi_{q2}^l\Big) \end{aligned} \tag{6}$$

where Ψ^l is the parameters of the low-level critic network.

3.2. Recommendation Module (Environment)

The pre-trained recommendation module first recalls top-M ($M \gg m$) candidates $\{v_j, v_{j+1}, ..., v_{j+M-1}\}$ with the highest probability based on the original sequence of baskets $\mathcal{B}$:

$$p(v_j|\mathcal{B}; \Phi) = \sigma(e_j^\top \cdot c_t) \tag{7}$$

where Φ is the pre-trained parameter for recommendation and $c_t \in \mathbb{R}^d$ is the user's current preferences extracted from $\mathcal{B}$ by the aggregator. Specifically, the aggregator merges user preferences at the item-level and basket-level. For the former, we apply DOT to compute attention coefficient [13] between user vector f and item vector e_m^t, then, the attention coefficient of all items within the basket B_t are normalized by: $\omega_{pe_m^t} = \frac{exp(f^\top \cdot e_m^t)}{\sum_{k=1}^m exp(f^\top \cdot e_k^t)}$, thus, the basket vector is computed by: $b_t = \sum_{k=1}^m \omega_{pe_m^t} \cdot e_m^t$. For the latter, we feed the basket vector b_t to a new GRU^R and regard the hidden state h_t^R as the user's preference vector c_t, i.e., $c_t = GRU^R(h_{t-1}^R, b_t)$.

Reward function: Given a denosied sequence of baskets $\widetilde{\mathcal{B}^j}$ for the target item v_j, the environment gives the agents a delayed reward based on the difference in the recommendation probability before and after denoising:

$$\begin{aligned} R(s, a) &= \lambda\Big(p(v_j|\widetilde{\mathcal{B}^j}; \Phi) - p(v_j|\mathcal{B}; \Phi)\Big) \\ &= \lambda\Big(\sigma(e_j^\top \cdot \widetilde{c}_t^j) - \sigma(e_j^\top \cdot c_t)\Big) \end{aligned} \tag{8}$$

where λ is the scale factor to expand the utility of $R(s, a)$, and $\widetilde{c}_t^j \in \mathbb{R}^d$ is the denoised user's current preferences vector extracted from $\widetilde{\mathcal{B}^j}$ by the aggregator. A positive difference means that the denoising gains positive utility [14] and the reward $R(s, a)$ will be given equally to all denoised actions $a^h/a^l = 1$, while all actions $a^h/a^l = 0$ will have no reward.

Meanwhile, the recommended probability of the target item v_j will be updated from $p(v_j|\mathcal{B}; \Phi)$ to $p(v_j|\widetilde{\mathcal{B}^j}; \Phi)$. After the last target item is updated, the recommender will select the top-m items $\{v_k, v_{k+1}, ..., v_{k+m-1}\}$ from M candidates according to the highest $p(v_k|\widetilde{\mathcal{B}^k}; \Phi)$ as the next basket.

3.3. Training Methodology

In our proposed HRL4Ba, the Actor-Critic structure is applied to both HA and LA. Since the policies of HA and LA are deterministic, we utilize DPG [9] based algorithm to train the parameters. The gradient of the objective function is:

$$\nabla_\Theta J(\Theta) = \mathbb{E}_{\tau \backsim \mu_\Theta}[\nabla_\Theta \mu(s; \Theta) \nabla_a q(s, a; \Psi)|_{a=\mu(s;\Theta)}] \tag{9}$$

for simplicity, Θ=$\{\Theta^h, \Theta^l\}$, Ψ=$\{\Psi^h, \Psi^l\}$, s, a denote states and actions, respectively. τ is the sampled sequence like $\{s_1^h, a_1^h, ..., s_t^h, a_t^h\}$ for HA or $\{s_1^l, a_1^l, ..., s_m^l, a_m^l\}$ for LA. The gradients of the parameters are as follow:

$$\begin{aligned} \Delta\Theta &= \alpha \nabla_\Theta \mu(s; \Theta) \nabla_a q(s, a; \Psi)|_{a=\mu(s;\Theta)} \\ \Delta\Psi &= \beta \delta \nabla_\Psi q(s, a; \Psi) \end{aligned} \tag{10}$$

where $\delta = R(s, a) + \gamma q(s', \mu(s'; \Theta); \Psi) - q(s, a; \Psi)$ is the TD-error [15], s' is the next state, and γ is the discount factor, α and β are the learning rates of actors and critics. The gradient ascent is performed to update the parameters of actor and critic networks respectively : $\Theta \leftarrow \Theta + \Delta\Theta$; $\Psi \leftarrow \Psi + \Delta\Psi$.

4. EXPERIMENT AND ANALYSIS

4.1. Experimental Settings

Datasets. We conduct experiments on two public e-commerce datasets: Tafeng and Instacart After removing rare items and users (fewer than 10 purchases) [16] from datasets, the statistics of the preprocessed datasets are listed in Table 1.

Evaluation Metrics. We evaluate the performance of HRL4Ba with the metrics: **F1@K** (F1-score) and **NDCG@K** (Normalized Discounted Cumulative Gain). Note that K denotes a list of items which is equal to the basket size m. To ensure generality, we set $K(m)$ as a fixed number 5 or 10 for testing.

Implementation Details. For each dataset, we randomly choose 10% of the users for testing while the rest for training. We fix the length of the basket sequence to t, i.e., 8 and 20 for Tafeng and Instacart respectively according to their statistical characteristics in Table 1. We apply a sliding window [17] for those sequences longer than t and padding [6] for those shorter than t, respectively. Thus, the maximum step in each episode for both the high-level and low-level agent is t and m respectively. The other hyper-parameters are defined as follows: we set the dimensions of all vectors d=256, the reward scaling factor λ=2.87, the discount factor γ=0.95, and the learning rate α=0.005 and β=0.01. Our experimenst were completed on NVIDIA RTX3090 GPU with 24GB memory.

Table 1: Statistics of datasets.

Dataset	Tafeng	Instacart
# Users	9,238	206,209
# Items	7,982	42,987
# Baskets	67,964	3,345,786
# Avg.basket size	7.4	10.1
# Avg.sequence length	5.9	16.2

Table 2: Overall performance comparison. The best and second results are bolded and underlined, respectively.

Datasets	Model	F1@K(%)		NDCG@K(%)	
		@5	@10	@5	@10
Tafeng	POP	4.83	4.06	8.76	7.18
	FPMC	10.21	9.63	19.93	17.27
	DREAM	11.58	10.92	20.44	18.21
	ANAM	12.63	11.17	21.85	20.07
	IntNet	12.86	11.33	21.92	20.81
	Beacon	14.88	13.41	27.06	26.54
	CLEA	14.93	14.17	27.42	27.35
	R4Ba	10.22	9.06	18.73	17.42
	H4Ba	15.04	14.31	27.76	27.41
	L4Ba	<u>15.11</u>	<u>14.68</u>	<u>28.34</u>	<u>27.68</u>
	HRL4Ba	**16.71**	**15.96**	**29.63**	**28.74**
Instacart	POP	6.68	5.79	11.24	9.47
	FPMC	26.67	25.63	41.43	39.18
	DREAM	27.36	26.93	42.18	41.22
	ANAM	35.63	33.17	58.30	54.64
	IntNet	29.74	28.11	45.39	45.01
	Beacon	35.93	34.41	59.67	53.21
	CLEA	38.68	37.39	61.31	58.92
	R4Ba	27.73	27.14	42.17	41.29
	H4Ba	38.95	<u>38.05</u>	61.49	<u>59.27</u>
	L4Ba	<u>39.19</u>	37.66	<u>62.23</u>	59.03
	HRL4Ba	**40.81**	**39.26**	**63.58**	**61.34**

4.2. Comparison to Baselines

We compare HRL4Ba with a variety of NBR methods, including two unsupervised-based methods CLEA [6] and Beacon [5], two attention-based based methods IntNet [4] and ANAM [3], three traditional methods DREAM [18], FPMC [19] and POP. Note that R/H/L4Ba are variants of HRL4Ba, which will be detailed in Section 4.3. All hyper-parameters of the baselines are carefully tuned based on cross-validation. From Table 2 we can observe that HRL4Ba outperforms all baselines on all datasets. Specifically, HRL4Ba is superior to CLEA and Beacon that follow fixed denoising policy, which shows the advancement of our dynamic denoising policy that fuses personalized hierarchical contexts for the NBR task. Compared with attention-based and traditional methods, the proposed denoising module could assist HRL4Ba to focus more on relevance baskets/items, enabling learning more robust inter-/intra-context of users, which effectively improves the overall performance. Moreover, although attention-based methods outperform traditional methods, they are still inferior to the denoising-oriented methods. This trend supports our idea of the necessity of denoising in the NBR task.

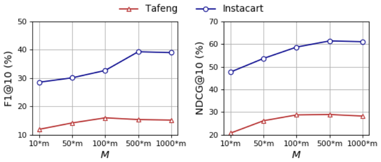

Fig. 2: Impact of M on the performance of HRL4Ba.

4.3. Model Analysis

Q1: Does the RL-based denoising module work?

A1: To answer this question, we designed a variant of HRL4Ba codenamed R4Ba, which has only the recommendation module. In R4Ba, the M candidates will be directly recommended to the user as the next basket without denoising. From Table 2, we find that the performance of R4Ba decreases awfully, i.e., the SOTA performance of HRL4Ba is mainly benefited from the RL-based denoising module.

Q2: Does the hierarchical structure for denoising helpful?

A2: We designed two variants codenamed H4Ba and L4Ba respectively. They share the recommendation module, but H4Ba has only the high-level agent, while L4Ba has only the low-level agent. In Table 2, we find that the two variants show competitive performance, but are mutually restricted to incomplete contexts and thus cannot achieve the SOTA performance of HRL4Ba. This confirms that the hierarchical intra-basket/inter-basket decision structure we designed is helpful.

Q3: What is the effect of the number of candidate M?

A3: As we mentioned before, the m items in the next basket are selected from M candidates ($M \gg m$), so the quality of M candidates affects the final recommendation performance. Here, we set M to be different multiples of m, i.e., M/m = $[10, 50, 100, 500, 1000]$. In Figure 2, the perceptual field of HRL4Ba increases with M and the performance gradually improves. However, an oversized M will harm the performance, e.g., the whole item pool $\mathcal{V}$ is treated as candidate when M=1000m in Tafeng. Morever, We find that the optimal M differs among datasets, e.g., Instacart with a large item pool suited for 500m and Tafeng with a small item pool suited for 100m. Thus, the general paradigm is $M = \lceil |\mathcal{V}|/80 \rceil m$.

5. CONCLUSION

In this work, we propose a hierarchical reinforcement learning framework named HRL4Ba for the NBR task. Specifically, in the denoising module, the high-level and the low-level agents automatically learn the hierarchical personalized inter-basket and intra-basket contexts to perform denoising at the basket-level and the item-level, respectively. Then, the recommendation module refines the recommendation probabilities of the candidates based on the denoised basket sequence to produce a more relevant next basket. To the best of our knowledge, this is the first work on the NBR empowered by RL. In the future, we will import external knowledges as supervised signals to guide agents for further improvement.

6. REFERENCES

[1] A Mantha, Y Arora, S Gupta, P Kanumala, Z.W. Liu, S Guo, and K Achan, "A large-scale deep architecture for personalized grocery basket recommendations," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3807–3811.

[2] W Wang and L.B. Cao, "Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 3, pp. 1–26, 2021.

[3] T Bai, J.Y. Nie, X. Zhao, Y.T. Zhu, P Du, and J.R. Wen, "An attribute-aware neural attentive model for next basket recommendation," in *Proceedings of the 41st International Conference on Research & Development in Information Retrieval (SIGIR)*, 2018, pp. 1201–1204.

[4] S.J. Wang, L Hu, Y Wang, Q.Z. Sheng, M Orgun, and L.B. Cao, "Intention nets: psychology-inspired user choice behavior modeling for next-basket prediction," *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 04, pp. 6259–6266, 2020.

[5] H.W. Lauw D.T. Le and Y Fang, "Correlation-sensitive next-basket recommendation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 2808–2814.

[6] Y.Q. Qin, P.F. Wang, and C.L. Li, "The world is binary: Contrastive learning for denoising next basket recommendation," in *Proceedings of the 44th International Conference on Research & Development in Information Retrieval (SIGIR)*, 2021, pp. 859–868.

[7] R.B. Xie, S.L. Zhang, R Wang, F Xia, and L.Y. Lin, "Hierarchical reinforcement learning for integrated recommendation," in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[8] T Xiao and D.L. Wang, "A general offline reinforcement learning framework for interactive recommendation," in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[9] D Silver, G Lever, N Heess, T Degris, D Wierstra, and M Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML)*, 2014, pp. 387–395.

[10] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan 2016.

[11] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[12] D.Y. Zhao, L Zhang, B Zhang, L.Z. Zheng, Y.J. Bao, and W.P. Yan, "Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations," in *Proceedings of the 43rd International Conference on Research & Development in Information Retrieval (SIGIR)*, 2020, pp. 871–880.

[13] Y Lei, Z.T. Wang, W.J. Li, H.B. Pei, and Q.Y. Dai, "Social attentive deep q-networks for recommender systems," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2020.

[14] J Zhang, B.W. Hao, B Chen, C.P. Li, H Chen, and J.M. Sun, "Hierarchical reinforcement learning for course recommendation in moocs," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2019, vol. 33, pp. 435–442.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. V eness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Aug 2015.

[16] Y.F. Leng, L Yu, J Xiong, and G.Y. Xu, "Recurrent convolution basket map for diversity next-basket recommendation," in *Proceedings of the International Conference on Database Systems for Advanced Applications (DASFAA)*, 2020, pp. 638–653.

[17] S.J. Wang, L Hu, Y Wang, Q.Z. Sheng, M.A. Orgun, and L.B. Cao, "Intention2basket: A neural intention-driven approach for dynamic next-basket planning.," in *Proceedings of the Twenty-ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 2333–2339.

[18] F Yu, Q Liu, S Wu, L Wang, and T.N. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International Conference on Research & Development in Information Retrieval (SIGIR)*, 2016, pp. 729–732.

[19] S Rendle, C Freudenthaler, and L Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World Wide Web (WWW)*, 2010, pp. 811–820.