

ENHANCING GUIDANCE FOR MISSING DATA IN DIFFUSION-BASED SEQUENTIAL RECOMMENDATION

Qilong Yan^{1,2}

Yifei Xing^{2,4}

Dugang Liu³

Jingpu Duan^{2,*}

Jian Yin^{1,2,*}

¹ School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China

² Peng Cheng Laboratory, Shenzhen, China

³ Shenzhen University, Shenzhen, China

⁴ University of Chinese Academy of Sciences, Beijing, China

{yanql, xingyf}@pcl.ac.cn, dugang.ldg@gmail.com

ABSTRACT

Contemporary sequential recommendation methods are becoming more complex, shifting from classification to a diffusion-guided generative paradigm. However, the quality of guidance in the form of user information is often compromised by missing data in the observed sequences, leading to suboptimal generation quality. Existing methods address this by removing locally similar items, but overlook “critical turning points” in user interest, which are crucial for accurately predicting subsequent user intent. To address this, we propose a novel Counterfactual Attention Regulation Diffusion model (CARD), which focuses on amplifying the signal from key interest-turning-point items while concurrently identifying and suppressing noise within the user sequence. CARD consists of (1) a Dual-side Thompson Sampling method to identify sequences undergoing significant interest shift, and (2) a counterfactual attention mechanism for these sequences to quantify the importance of each item. In this manner, CARD provides the diffusion model with a high-quality guidance signal composed of dynamically re-weighted interaction vectors to enable effective generation. Experiments show our method works well on real-world data without being computationally expensive. Our code is available at <https://github.com/yanqilong3321/CARD>.

Index Terms— Sequential Recommendation, Diffusion Models, Missing Data

1. INTRODUCTION

Sequential recommendation aims to predict the next item of interest based on a user’s historical interaction sequence [1,

*Corresponding authors. This work is supported by - Mobile Information Networks National Science and Technology Major Project under Grant No.2025ZD1303200, the National Natural Science Foundation of China (U22B2060), the Key-Area Research and Development Program of Guangdong Province (2024B0101050005), and the Research Foundation of Science and Technology Plan Project of Guangzhou City (2023B01J0001, 2024B01W0004).

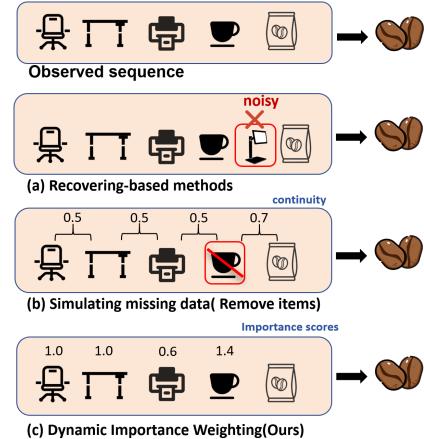


Fig. 1. Comparisons of existing methods for missing data: (a) Recovering-based methods risk introducing erroneous items. (b) Simulating missingness by removing items based on local continuity. (c) Our method by re-weighting items based on their predictive importance

2, 3]. Traditional discriminative recommendation [4, 5, 6] focuses on distinguishing the target item from negative samples, whereas emerging generative recommendation aims to directly generate “oracle items” that best match user preferences [7]. Compared to VAEs and GANs, which are limited in generation stability and quality, diffusion models [7, 8, 9] have emerged as the most promising technology, excelling in both stability and quality by iteratively noising and denoising items under the guidance of interaction history. However, the performance of these advanced generative models largely depends on an ideal assumption: the user’s interaction history is complete. In real-world applications, missing data [10, 11] is pervasive and unpredictable, creating unreliable guidance signals that severely constrain the practical effectiveness of diffusion models.

Two distinct approaches currently exist to address this

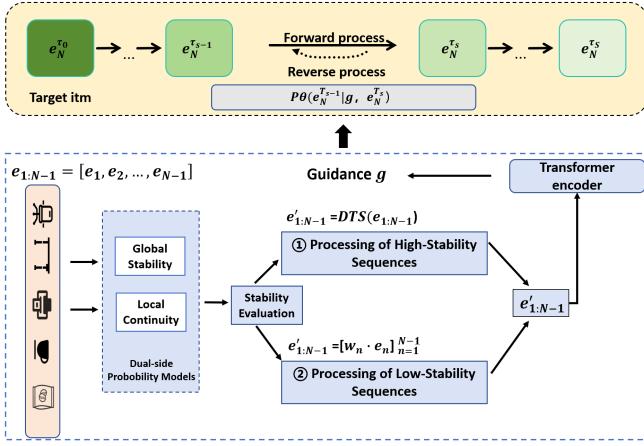


Fig. 2. The overall architecture follows a three-step process: (1) it first evaluates sequence stability to determine a processing path; (2) high-stability sequences are then simplified via DTS, while (3) low-stability ones are re-weighted using our counterfactual attention. The optimized sequence is then encoded to provide guidance for the diffusion model.

problem: recovering-based strategies [12, 13, 11] and strategies based on simulating missingness. The core idea of recovering-based strategies is to first “repair” or “complete” the incomplete sequence through interpolation [14, 15] or augmentation before guiding the diffusion model, as seen in PDRec [13] and SSDRec [11]. Although intuitive, these methods face a fundamental challenge: the “uncertainty” of missing data [16, 17]. Since the complete sequence as a “ground truth” is unavailable, the recovery process is essentially a guess, which can easily introduce new noise or distort true user preferences (as shown in Fig. 1(a)), potentially providing even more erroneous guidance to the diffusion model.

Strategies based on simulating missingness, in contrast, propose that rather than riskily “recovering” data, it is better to train the model by actively simulating missingness, enabling it to “extrapolate” and handle real-world missing data. For example, TDM [18] identifies and removes highly similar items under stable preferences by calculating the “local continuity” between items and the “global stability” of the sequence, thereby constructing training samples with missingness while preserving core user preferences. However, the existing simulation mechanisms have limitations. For example, as shown in Fig. 1(b), due to the high similarity between “coffee mug” and “coffee shop”, TDM would tend to remove one of them, believing it has minimal impact on the overall preference evolution, but this might destroy critical bridge information.

To this end, we propose CARD, a novel Counterfactual Attention Regulation Diffusion model. CARD employs a more robust scheme for handling sequence uncertainty. Instead of removing items, it dynamically evaluates and re-

shapes the importance of each item. This approach can both simulate missingness (when importance is very low) and amplify the signals of key items by increasing their attention. To achieve this, CARD introduces an attention mechanism based on counterfactual reasoning, motivated by the principle that an item’s importance should be measured by its “contribution” to predicting future user preferences, rather than its local similarity to adjacent items. Specifically, CARD quantifies this contribution by comparing prediction error in two scenarios: one including the item’s information and one excluding it (the counterfactual case). The “Prediction Error Reduction” brought by the item’s inclusion is used to adjust its importance weight. As shown in Fig. 1(c), removing the “coffee mug” would cause a sharp increase in prediction error for subsequent items like “coffee beans.” Therefore, CARD will increase the attention weight of the “coffee mug,” identifying it as a key item that signifies a user’s interest shift. Similarly, an item like a “printer” that might interfere with subsequent predictions would have its attention weight reduced. Finally, to enhance computational efficiency, CARD only performs attention calculations for low-stability sequences and for items that are potential turning points. For high-stability sequences, CARD adopts the mature DTS strategy, as it remains an effective means of reducing redundancy. In summary, our contributions are threefold: (1) a novel counterfactual attention mechanism that defines item importance by its predictive contribution (“Prediction Error Reduction”), rather than local similarity; (2) an efficient routing strategy that selectively applies this attention mechanism to significantly reduce computational overhead; and (3) extensive experiments validating CARD’s effectiveness and efficiency on real-world datasets.

2. METHODOLOGY

In this section, we first briefly introduce the preliminaries of diffusion models in recommendation, then detail CARD’s components. An overview of the architecture is shown in Fig. 2.

2.1. Preliminaries: Diffusion Models for Recommendation

A diffusion model generates a target item embedding \mathbf{e}_N^0 from Gaussian noise $\mathbf{e}_N^{\tau_s} \sim \mathcal{N}(0, \mathbf{I})$ through a reverse process. This process is conditioned on a guidance vector \mathbf{g} , which is derived from the user’s historical interaction sequence $\mathbf{e}_{1:N-1}$. The training objective is to predict the original item embedding \mathbf{e}_N^0 from its noised version $\mathbf{e}_N^{\tau_s}$ at step τ_s . During inference, we employ classifier-free guidance [19] to combine conditional and unconditional predictions, enhancing generation quality:

$$\tilde{f}_\theta(\cdot) = (1 + w)f_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s) - w f_\theta(\mathbf{e}_N^{\tau_s}, \phi, \tau_s), \quad (1)$$

where w is the guidance strength and ϕ is a dummy token for unconditional generation.

2.2. Dynamic Guidance Optimization via Routing

The core of CARD is its ability to create a high-quality guidance vector \mathbf{g} by dynamically optimizing the user sequence based on its stability.

2.2.1. Stability Evaluation.

Following TDM [18], we first quantify the local continuity between adjacent items \mathbf{e}_n and \mathbf{e}_{n+1} using a softmax-normalized cosine similarity:

$$\text{con}_n = \frac{\exp(\text{sim}(\mathbf{e}_n, \mathbf{e}_{n+1}))}{\sum_{n'=1}^{N-2} \exp(\text{sim}(\mathbf{e}_{n'}, \mathbf{e}_{n'+1}))}. \quad (2)$$

The overall stability of a sequence is then measured by the entropy of this distribution, denoted as a stability score s_k , where a higher value indicates lower stability (i.e., a significant interest shift):

$$s_k = - \sum_{n=1}^{N-2} \text{con}_n \log(\text{con}_n). \quad (3)$$

Based on a predefined threshold λ_{stb} , we route the sequence. For high-stability sequences ($s_k \leq \lambda_{stb}$), we adopt the DTS strategy from TDM [18] to remove redundant items. Low-stability sequences ($s_k > \lambda_{stb}$) are processed as follows.

2.2.2. Processing of Low-Stability Sequences.

For low-stability sequences, which signal a user’s interest shift, naively applying DTS can be detrimental. It may catastrophically damage the guidance signal by removing the very items that signify the interest shift. To address this critical flaw, CARD introduces its core contribution: a **counterfactual attention mechanism**.

Instead of removing items, we dynamically re-weight them. The importance of each item is not determined by local similarity, but by its causal impact on future predictions. Specifically, we quantify an item’s importance by the **Prediction Error Reduction (PER)**. To achieve this, we introduce an auxiliary prediction task: for each position n , we predict the average embedding of the next W items, denoted as $\bar{\mathbf{e}}_{>n}$. We then calculate the prediction loss under two conditions: one using the full history up to item n (encoded as hidden state \mathbf{h}_n), and a counterfactual one using only the history up to item $n-1$ (encoded as \mathbf{h}_{n-1}). The PER is the difference between these two losses:

$$\text{PER}_n = \underbrace{\|\text{MLP}_{aux}(\mathbf{h}_{n-1}) - \bar{\mathbf{e}}_{>n}\|^2}_{\text{Loss without } \mathbf{e}_n} - \underbrace{\|\text{MLP}_{aux}(\mathbf{h}_n) - \bar{\mathbf{e}}_{>n}\|^2}_{\text{Loss with } \mathbf{e}_n} \quad (4)$$

A high PER value indicates a critical “bridge” item. This results in a weighted sequence:

$$\tilde{\mathbf{e}}_{1:N-1} = [\mathbf{w}_n \cdot \mathbf{e}_n]_{n=1}^{N-1}, \quad \text{if } s_k > \lambda_{stb}, \quad (5)$$

where the weight $\mathbf{w}_n = 1 + \tanh(\text{PER}_n/T)$ is designed to amplify the signal of key turning points ($w_n > 1$) and suppress the influence of noisy or irrelevant items ($w_n < 1$). This provides a far more robust and nuanced guidance signal for the diffusion model.

2.2.3. Guidance Vector Generation.

Regardless of the path taken, the processed sequence $\tilde{\mathbf{e}}_{1:N-1}$ is fed into a Transformer encoder to produce the final guidance vector: $\mathbf{g} = \text{Transformer}(\tilde{\mathbf{e}}_{1:N-1})$.

2.3. Model Training and Inference

The denoising network f_θ is trained to predict the original item \mathbf{e}_N^0 from a noised version, guided by the final guidance vector \mathbf{g} , and optimized via a mean squared error objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{e}_N^0, \mathbf{g}, \epsilon, \tau_s} \left[\|\mathbf{e}_N^0 - f_\theta(\mathbf{e}_N^{\tau_s}, \mathbf{g}, \tau_s)\|^2 \right]. \quad (6)$$

During inference, generation starts from random noise and iteratively denoises it using classifier-free guidance (Eq. 1). The final generated embedding is used to retrieve top-K recommendations.

3. EXPERIMENTS

3.1. Experimental Setup

Datasets. We conduct experiments on two widely-used real-world datasets: Zhihu[20] and KuaiRec[21]. Following standard preprocessing, we filter out users and items with fewer than 5 interactions and ensure all sequences have a minimum length of 3. We follow the standard data splitting and evaluation protocols as in previous works [18, 9].

Evaluation Metrics. We adopt the standard leave-one-out strategy for evaluation, ranking the ground-truth next item against random negative samples. We report Hit Ratio (HR@20) and Normalized Discounted Cumulative Gain (NDCG@20), averaged over five runs.

Baselines. We compare CARD against three groups of models: (i) Conventional Recommenders: GRU4Rec [4], SAS-Rec [1], BERT4Rec [22], and AdaRanker [23]; (ii) Generative Recommenders: DiffRec [7], DreamRec [9], and TDM [18]; and (iii) Recovering-based Models: PDRec [13], SS-DRRec [11], STEAM [12], and DiffuASR [24].

Implementation Details. All models are implemented in PyTorch and trained on an NVIDIA RTX 4090 GPU. The embedding dimension is 64. We use the Adam optimizer with a learning rate tuned from $\{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. For CARD,

Table 1. Main performance comparison (%). Best in **bold**, second-best underlined. Improv. is vs. runner-up.

Methods	KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20
<i>Conventional Recommenders</i>				
GRU4Rec	3.32±0.11	1.23±0.08	1.78±0.12	0.67±0.03
SASRec	3.92±0.18	1.53±0.11	1.62±0.01	0.60±0.03
BERT4Rec	3.77±0.09	1.73±0.04	2.01±0.06	0.72±0.04
AdaRanker	4.14±0.09	1.89±0.05	1.70±0.04	0.61±0.02
<i>Recovering-based Models</i>				
STEAM	4.98±0.05	2.90±0.02	1.75±0.02	0.69±0.02
SSDRec	4.19±0.08	3.28±0.06	2.03±0.06	0.72±0.03
DiffuASR	4.53±0.02	3.30±0.03	2.05±0.02	0.71±0.02
PDRec	4.42±0.03	3.55±0.04	2.10±0.03	0.74±0.02
<i>Generative Recommenders</i>				
DiffRec	3.74±0.08	1.77±0.05	1.82±0.03	0.65±0.09
DreamRec	5.16±0.05	4.11±0.02	2.26±0.07	0.79±0.01
TDM	<u>5.53±0.03</u>	<u>4.82±0.04</u>	<u>2.33±0.04</u>	0.75±0.03
CARD (Ours)	5.78±0.04	5.02±0.03	2.57±0.05	0.83±0.02
Improv.	4.52%	4.15%	10.30%	5.06%

we tune the stability threshold λ_{stb} in $[0.5, 2.0]$ and future window size W in $\{1, 3, 5\}$. Baselines use their officially reported hyperparameters.

3.2. Results

Main Results. Table 1 presents the overall performance comparison on the Zhihu and KuaiRec datasets. We can draw several key observations. Firstly, generative models, particularly diffusion-based approaches like TDM and DreamRec, generally surpass conventional and recovering-based methods, underscoring the strength of the generative paradigm in recommendation. Secondly, our proposed model, CARD, consistently achieves state-of-the-art performance across all metrics on both datasets. Specifically, on the challenging Zhihu dataset, CARD outperforms the strongest baselines, leading to significant relative improvements of 10.30% in HR@20 (over TDM) and 5.06% in NDCG@20 (over DreamRec). This substantial margin validates the effectiveness of our dynamic guidance optimization strategy. By differentiating between high- and low-stability sequences and applying tailored processing—counterfactual attention for the latter—CARD successfully captures critical interest shifts, leading to a higher quality guidance signal and superior recommendation accuracy.

Ablation Study. To validate the contributions of CARD’s key components, we conduct an ablation study with two variants: (1) CARD w/o Routing, which applies the counterfactual attention mechanism to all sequences, regardless of their stability score; and (2) CARD w/o Attention, which disables the counterfactual attention and processes all sequences using the standard DTS strategy. As shown in Table 2, removing the counterfactual attention (w/o Attention) leads to a signif-

icant performance degradation, proving the effectiveness of this mechanism. Removing the routing module (w/o Routing) also impacts performance and, more critically, results in substantially higher computational costs. Their combination makes CARD both effective and efficient.

Table 2. Ablation study results (%) on Zhihu and KuaiRec.

Methods	KuaiRec		Zhihu	
	HR@20	NDCG@20	HR@20	NDCG@20
w/o Routing	5.66	4.89	2.28	0.77
w/o Attention	5.53	4.82	2.34	0.75
CARD	5.78	5.02	2.57	0.83

Efficiency comparison. To evaluate the efficiency of our model, we compare the training time per epoch and inference time per batch against several baselines, with results shown in Table 3. Notably, CARD significantly reduces training time compared to TDM—especially on KuaiRec—by selectively applying intensive counterfactual attention to low-stability sequences via our routing mechanism. During inference, CARD maintains a speed comparable to TDM as their core diffusion processes are identical. Thus, CARD enhances accuracy with minimal overhead, showcasing its practical efficiency.

Table 3. Running time comparison of CARD with baselines.

Methods	KuaiRec		Zhihu	
	Train	Inference	Train	Inference
SASRec	02m 07s	00m 08s	00m 10s	00m 01s
AdaRanker	03m 38s	00m 09s	00m 14s	00m 01s
DreamRec	03m 59s	32m 40s	00m 14s	01m 31s
TDM	03m 30s	00m 34s	00m 20s	00m 02s
CARD	02m 40s	00m 34s	00m 13s	00m 02s

4. CONCLUSION

In this paper, we addressed the critical challenge of handling missing data in diffusion-based sequential recommendation, particularly for sequences involving significant interest shifts. We proposed CARD, a novel model that dynamically optimizes the guidance signal for the diffusion process. By introducing a routing strategy to differentiate sequence stability and a counterfactual attention mechanism to adjust the weights of key items in low-stability sequences, this approach amplifies the signals of key items while suppressing potential noise. Extensive experiments demonstrated that CARD not only achieves state-of-the-art recommendation accuracy but also improves training efficiency. Our work presents a robust and efficient paradigm for enhancing the guidance quality in generative recommendation systems.

5. REFERENCES

- [1] Wang-Cheng Kang and Julian McAuley, “Self-attentive sequential recommendation,” in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.
- [2] Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He, “Customizing language models with instance-wise lora for sequential recommendation,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 113072–113095, 2024.
- [3] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He, “Llara: Large language-recommendation assistant,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 1785–1795.
- [4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, “Session-based recommendations with recurrent neural networks,” *arXiv preprint arXiv:1511.06939*, 2015.
- [5] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang, “Contrastive learning for representation degeneration problem in sequential recommendation,” in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 813–823.
- [6] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen, “Unbiased sequential recommendation with latent confounders,” in *Proceedings of the ACM web conference 2022*, 2022, pp. 2195–2204.
- [7] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua, “Diffusion recommender model,” in *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 2023, pp. 832–841.
- [8] Yong Niu, Xing Xing, Zhichun Jia, Ruidi Liu, Mindong Xin, and Jianfu Cui, “Diffusion recommendation with implicit sequence influence,” in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 1719–1725.
- [9] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He, “Generate what you prefer: Reshaping sequential recommendation via guided diffusion,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 24247–24261, 2023.
- [10] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al., “Patch diffusion: Faster and more data-efficient training of diffusion models,” *Advances in neural information processing systems*, vol. 36, pp. 72137–72154, 2023.
- [11] Chi Zhang, Qilong Han, Rui Chen, Xiangyu Zhao, Peng Tang, and Hongtao Song, “Ssdrec: Self-augmented sequence denoising for sequential recommendation,” in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 803–815.
- [12] Yujie Lin, Chenyang Wang, Zhumin Chen, Zhaochun Ren, Xin Xin, Qiang Yan, Maarten de Rijke, Xiuzhen Cheng, and Pengjie Ren, “A self-correcting sequential recommender,” in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1283–1293.
- [13] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang, “Plug-in diffusion model for sequential recommendation,” in *Proceedings of the AAAI conference on artificial intelligence*, 2024, vol. 38, pp. 8886–8894.
- [14] Erna Hikmawati, Heru Nugroho, and Kridanto Surendro, “Improve the quality of recommender systems based on collaborative filtering with missing data imputation,” in *Proceedings of the 2024 13th International Conference on Software and Computer Applications*, 2024, pp. 75–80.
- [15] Harald Steck, “Evaluation of recommendations: rating-prediction and ranking,” in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 213–220.
- [16] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu, “Sequential recommendation via stochastic self-attention,” in *Proceedings of the ACM web conference 2022*, 2022, pp. 2036–2047.
- [17] Chenxu Wang, Fuli Feng, Yang Zhang, Qifan Wang, Xunhan Hu, and Xiangnan He, “Rethinking missing data: Aleatoric uncertainty-aware recommendation,” *IEEE Transactions on Big Data*, vol. 9, no. 6, pp. 1607–1619, 2023.
- [18] Wenyu Mao, Zhengyi Yang, Jiancan Wu, Haozhe Liu, Yancheng Yuan, Xiang Wang, and Xiangnan He, “Addressing missing data issue for diffusion-based recommendation,” in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 2152–2161.
- [19] Jonathan Ho and Tim Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [20] Bin Hao, Min Zhang, Weizhi Ma, Shaoyun Shi, Xinxing Yu, Houzhi Shan, Yiqun Liu, and Shaoping Ma, “A large-scale rich context query and recommendation dataset in online knowledge-sharing,” *arXiv preprint arXiv:2106.06467*, 2021.
- [21] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua, “Kuairec: A fully-observed dataset and insights for evaluating recommender systems,” in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 540–550.
- [22] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.
- [23] Xinyan Fan, Jianxun Lian, Wayne Xin Zhao, Zheng Liu, Chaozhuo Li, and Xing Xie, “Ada-ranker: A data distribution adaptive ranking paradigm for sequential recommendation,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1599–1610.
- [24] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian, “Diffusion augmentation for sequential recommendation,” in *Proceedings of the 32nd ACM International conference on information and knowledge management*, 2023, pp. 1576–1586.