基于决策变换器的面向用户留存的推荐

Kesen Zhao 香港城市大学
kesenzhao2-c@my.cityu.edu.hk

Lixin Zou†武汉大学
zoulixin15
@gmail.com

Xiangyu Zhao†香港城市大学
xianzhao@cityu.edu.hk

Maolin Wang 香港城市大学
MorinWang@foxmail.com

Dawei Yin 百度
yinda
wei@acm.org

arXiv:2303.06347v1 [cs.IR] 11 Mar 2023

摘要

由于强化学习（RL）在提升用户参与度方面具有重要作用，使用强化学习改善用户留存率日益受到关注。然而，由于需要进行试错搜索，在不损害用户体验的情况下从头训练强化学习策略不可避免。此外，旨在不进行在线交互的情况下优化策略的离线方法，面临价值估计稳定性问题这一顽疾，或反事实策略评估中无界方差的问题。为此，我们提出使用决策变换器（DT）优化用户留存，通过将强化学习转化为自回归问题来规避离线方法的困难。然而，由于以下挑战，将DT 应用于推荐并非易事：（**1**）数值奖励的建模能力不足；（**2**）策略学习与推荐生成之间存在数据差异；（**3**）离线性能评估不可靠。因此，本研究提出了一系列策略来解决上述问题。首先，我们通过对元嵌入进行加权聚合，构建高效的奖励提示，以获得信息丰富的奖励嵌入。随后，我们采用加权对比学习方法来解决训练与推理之间的差异。此外，我们设计了两种稳健的离线指标来衡量用户留存。最后，在基准数据集上取得的显著提升证明了所提方法的优越性。实现代码可在 https://github.com/kesenzhao/DT4Rec.git 获取。

CCS 概念

• **信息系统 → 推荐系统**。

关键词

决策 Transformer、序列推荐系统、提示、对比学习

† 通讯作者。

允许为个人或课堂使用而免费制作本作品的全部或部分数字副本或纸质副本，前提是这些副本不得用于营利或商业利益，也不得为此进行分发，并且副本须附有本声明及首页的完整引文。对于本作品中由作者以外的其他人拥有版权的部分，必须尊重其版权。注明出处后可以摘录。若要以其他方式复制、重新出版、发布到服务器或分发至列表，须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。*WWW '23，2023*年*5*月*1*日至*5*日，美国德克萨斯州奥斯汀© 2023 版权所有归所有者/作者所有。出版权已授权给 ACM。
ACM ISBN 978-1-4503-9416-1/23/04...$15.00
https://doi.org/10.1145/3543507.3583418

ACM 引用格式：
Kesen Zhao、Lixin Zou†、Xiangyu Zhao†、Maolin Wang 和 Dawei Yin。2023。使用决策变换器实现面向用户留存的推荐。载于*2023* 年 *ACM Web* 会议论文集（*WWW '23*），*2023*年*5*月*1*日至*5*日，美国德克萨斯州奥斯汀。ACM，美国纽约州纽约市，9 页。https://doi.org/10.1145/ 3543507.3583418

1 引言

序列推荐系统（SRSs）对用户的历史交互进行建模，并向用户推荐可能感兴趣的物品。由于其在现实世界系统中发挥着不可替代的作用，例如 Netflix[1] 的电影推荐和 Amazon[2] 的电子商务推荐，因此在学术界和工业界都受到了广泛关注。序列推荐系统的成功很大程度上依赖于用户在平台上的参与度，而这在一定程度上体现在用户的即时反馈（如点击 [8, 40]）上。然而，这些即时反馈并不能完全揭示用户的偏好 [37]。例如，一些标题和封面引人注目、但内容质量低下的物品可能会吸引用户点击，进而损害用户对平台的信任 [35]。因此，优化用户在平台上的长期参与度至关重要 [47], ，例如提升用户留存率；后者是衡量用户满意度的一个更理想指标。

作为优化长期／延迟指标的工具， [24],近年来，强化学习（RL）已被广泛用于优化用户留存率 [6]。尽管现有基于强化学习的序列推荐系统能够探索并建模用户的动态兴趣 [39], ，但受离线学习挑战的影响，仍有很大的改进空间。与游戏场景不同，在游戏场景中，强化学习智能体可通过试错搜索 [29], 取得巨大成功；而在在线序列推荐系统中从头开始训练，则因推荐不合适的物品可能导致用户流失，因而难以承受。因此，整个领域近来都将关注重点放在基于离线强化学习的序列推荐系统上。然而，无论是基于价值的方法还是基于策略的方法，将离线强化学习付诸实践都令人头疼。对于基于价值的方法，臭名昭著的不稳定性问题（即“致命三元组”）推动了基于模型的方法的发展 [5]。然而，由于推荐场景中的状态空间庞大，转移概率难以估计，进而导致性能不尽如人意 [42]。对于基于策略的方法，反事实策略评估的无界方差促使研究者对反事实权重进行截断或舍弃 [3],，这可能导致权重不准确，并影响性能 [36]。

[1] https://www.netflix.com/ [2] https://www.amazon.com/

为探索基于强化学习的推荐的潜力，我们提出使用决策变换器 [2]（DT）优化用户留存推荐。该方法将离线强化学习转化为自回归问题，从而解决上述离线学习挑战。具体而言，DT需要根据特定的奖励（即用户留存率）和状态生成推荐。在以最优奖励为条件时，DT能够在推荐阶段生成可实现预期回报的未来动作。尽管DT在推荐领域前景广阔，但将其应用于序列推荐系统并非易事，面临以下挑战：**（1）奖励建模不足**。奖励是DT最关键的组成部分，直接影响推荐质量。然而，在DT中，将奖励转换为嵌入时忽略了其偏序关系，导致模型训练不足。**（2）推荐生成存在差异**。在DT框架下，推荐以最大化奖励为目标生成，这与训练阶段遇到的多样化奖励不一致。因此，模型无法利用奖励较低的数据中的知识（如图2所示）。**（3）性能评估不可靠**。尽管DT解决了离线学习问题，我们仍需通过重要性加权离线评估来衡量学习到的策略的性能，但这会导致无界方差，使评估结果不可靠。

为了解决这些问题，我们提出了一种新颖的框架DT4Rec，首次将**决策Transformer**用于**推荐**。具体而言，我们通过对离散化奖励值所生成的元嵌入进行自动加权聚合，生成奖励嵌入，从而保持奖励之间的偏序关系。随后，我们引入加权对比学习来弥补推理与训练之间的差异，通过对比大量和少量奖励样本，利用较少的奖励样本。此外，我们提出了两种新颖且可靠的指标，即基于模型的用户留存得分和基于相似度的用户留存得分，用于全面评估策略。与离策略评估方法相比，该方法在性能评估方面的方差更低，即稳定性更高（如图4所示）。我们的主要贡献总结如下：

- 我们提出了一种新颖的基于决策变换器的序列推荐系统模型DT4Rec，解决了基于强化学习的推荐系统中离线学习的难题。
- 我们提出了自动离散化奖励提示和对比监督策略学习，分别有效解决了奖励建模不足以及训练与推理之间存在差异的问题。
- 我们设计了基于模型和基于相似度的用户留存率评分。与离策略评估方法相比，它们能够公平地评估模型性能。
- 在两个基准数据集上进行的实验表明，我们提出的模型具有优越性。

2 方法

2.1 问题定义

在典型的序列推荐中，给定第 t 次推荐中由 $U_t = \{v_i^t\}_{i=1}^M$ 表示的M用户感兴趣的物品序列，我们旨在预测一组 N用户可能感兴趣的物品 $a_t = [v_{t,1}, \ldots, v_{t,N}]$。形式上，我们的目标是生成推荐结果 a_t^*，并最大化特定指标 Θ，即

$$a_t^* = \arg\max_{a_t} \Theta(a_t|U_t). \tag{1}$$

根据序列推荐系统的目标，我们可以通过优化用户的即时反馈（即预测准确率）或长期用户参与度（例如用户留存率）这一指标来具体化 Θ。对于用户留存，我们将其定义为接下来 K 个时间间隔内（例如接下来 K 天、接下来 K 个月）登录平台的用户数量

$$e_t = \sum_{k=1}^{K} \mathbb{1}\left[\text{log-in}_{t+k}\right], \tag{2}$$

其中，$\mathbb{1}$ 是指示函数，log-in_{t+k} 是一个二元函数，用于表示用户是否在下一个第（$t+k$）个时间步登录平台。

序列推荐的马尔可夫决策过程建模。为了通过强化学习优化指标 Θ，我们在马尔可夫决策过程（MDP）框架下对序列推荐进行形式化建模，其中包括一系列状态、动作和奖励。具体而言，状态 $s_t = U_t$ 定义为历史交互。动作 a_t 是推荐列表，由推荐策略 π： $s_t \rightarrow a_t$ 生成。每次推荐之前， s_t 都会通过附加用户的交互进行更新

末尾包含的物品为 $s_t = s_{t-1} \oplus \{U_t - U_{t-1}\}$。给定特定的评估指标 Θ，奖励通常定义为该指标的函数。具体而言，对于用户留存，奖励设定为与 e_t 相同。通过正式定义的 s_t、 a_t、 r_t，用户系统的交互构成一条轨迹

$$\tau = [s_1, a_1, r_1, \ldots, s_t, a_t, r_t]. \tag{3}$$

2.2 基于决策变换器的推荐

不同于传统框架通过近似状态-动作价值来进行规划 [17] 或直接优化基于梯度的策略函数 [4]，，决策变换器将强化学习问题转化为自回归问题，利用先前的奖励、状态和动作直接预测期望的动作（参见第 2.7 节）。作为一项监督学习任务，决策 Transformer 避免了“致命三元组”和无界方差问题，并展现出离线学习方面的优势 [2]。为了在序列推荐系统中实例化决策 Transformer，我们将其正式拆分为以下四个模块（如图1所示）：

- **嵌入模块**：嵌入模块旨在将奖励、状态和动作映射为稠密向量。具体而言，在将原始的 s_t、 a_t 和 r_t 转换为嵌入之前，我们需要按照奖励、状态和动作的顺序排列轨迹中的数据，以简化动作预测：

$$\tau_t' = [\widehat{r}_1, s_1, a_1, \ldots, \widehat{r}_t, s_t, a_t], \tag{4}$$

其中 $\widehat{r}_t = \sum_{t'=t}^{T} e_{t'}$ 表示累积奖励，即回合回报 [2]。 T表示推荐总轮数。然后，嵌入模块使用编码器模型（第 2.4 节）将原始特征序列 τ_t依次转换为特征向量序列 $\boldsymbol{\tau}'_t = [\widehat{\boldsymbol{r}}_1, \boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \widehat{\boldsymbol{r}}_t, \boldsymbol{s}_t, \boldsymbol{a}_t]$。特别是，由于奖励十分重要，我们将用于奖励的嵌入模块称为

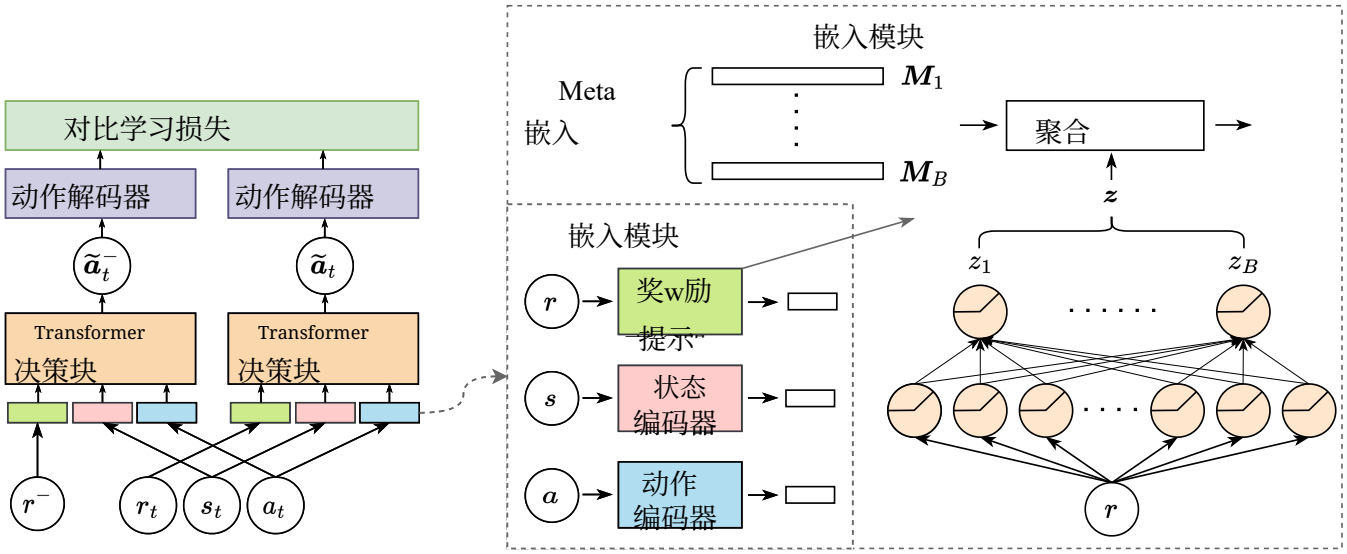

图1：DT4Rec 的框架概览。负样本与正样本具有相同的状态和动作，而奖励则替换为不同的值。正样本和负样本共享相同的模型参数。

奖励提示，其旨在提示决策 Transformer 生成所需的推荐列表。

• **决策块**：决策块是模型的核心，它将上下文信息的稠密向量 $\tau_t = \{\widehat{r}_1, s_1, a_1, \ldots, \widehat{r}_t, s_t\}$ 转换为上下文特征 A_t，以生成下一时间步所需的响应，其中 $\widehat{r}_t$ 是生成的奖励提示。

• **动作解码器**：给定上下文信息 A_t，动作解码器需要生成与真实动作 a_t匹配的动作序列 $\hat{a}_t$。

• **监督策略学习**：监督策略学习的目标是使用特定的损失函数，最小化生成的动作 $\hat{a}_t$ 与真实值 a_t 之间的差异。因此，它将常规的强化学习任务转化为监督学习问题。在推理时，将最优奖励指定为提示，决策 Transformer 旨在推荐能够最大化用户留存的物品。

2.3 自动离散化奖励提示

奖励作为强化学习的目标，通过数值区分不同策略的表现。因此，决策 Transformer 生成的提示应保持奖励之间的偏序关系，即如果两个奖励相近，那么它们生成的提示之间的欧几里得距离就更小。

为此，我们提出采用自动离散化方法生成更高效的提示，如图1所示。该方法包括对数值进行离散化，并对多层感知机学习到的元嵌入进行自动加权聚合。它能让相似的奖励共享嵌入。具体而言，我们根据奖励值，将其转换为一批 B 可学习嵌入$\boldsymbol{M}_b \in \mathbb{R}$的加权分数 $\forall b \in [1, B]$，如下所示：

$$z = \phi\left(W\sigma\left(w\widehat{r}\right) + \alpha\sigma\left(w\widehat{r}\right)\right), \tag{5}$$

其中，$w \in \mathbb{R}^{1\times B}$ 和 $W \in \mathbb{R}^{B\times B}$ 是可学习变量。σ 是 Leaky ReLU [11]。ϕ 是 Softmax函数。给定加权分数 z，奖励嵌入 $\widehat{r}$ 设为元嵌入的聚合，具体可表示为：

$$\widehat{r} = \sum_{b=1}^{B} z_b \boldsymbol{M}_b, \tag{6}$$

其中，z_b 是 z的第 b个元素。由于直接将 r 的值作为神经网络的输入，以确保奖励之间的偏序关系，因此只要神经网络是平滑的，相似的 $\widehat{r}$就会共享相似的嵌入。

2.4 状态-动作编码器

动作编码器将动作映射为向量。难点在于推荐动作的长度是动态变化的，因为用户每次可能与不同数量的物品交互。因此，我们使用门控循环单元（GRU）对交互序列进行建模。事实证明，在为推荐系统建模动态物品序列时，GRU的表现优于长短期记忆网络（LSTM）[14]。此外，与基于庞大的 Transformer [19] 的方法相比，GRU是兼顾效率与效果的更经济的替代方案。具体而言，我们将序列的最大长度设为 N。对于长度小于 N的序列，我们在序列中填充**0** 向量 [32]，使其达到 N。形式上，序列信息可提取为

$$\boldsymbol{H}_n = GRU_e\left(\boldsymbol{v}_n, \boldsymbol{H}_{n-1}\right) \tag{7}$$

$$\boldsymbol{a}_t = \boldsymbol{H}_N, \tag{8}$$

其中，GRU_e是循环层，N是最大序列长度，$\boldsymbol{H}_n$是隐藏状态。此外，我们将 a_t 的嵌入设为最后一个时间步的隐藏状态，即 $\boldsymbol{a}_t$。

2.5 Transformer决策模块

先前的研究已证明，Transformer 和强化学习在序列推荐任务中具有优势。一方面，Transformer 通过自注意力平等地处理序列中的各个元素，从而避免循环神经网络中的长序列信息丢失，并进一步减轻建模长序列时的梯度消失或梯度爆炸问题 [19, 46, 51]。另一方面，强化学习擅长动态建模用户的兴趣画像并捕捉用户的兴趣变化。为兼取二者之长，我们提出通过 Transformer决策模块，从用户的交互轨迹中学习用户兴趣 [2]。

具体而言，对于用户留存推荐，我们需要对动态上下文信息进行建模，以生成推荐决策，这与生成任务类似。因此，我们选择单向 Transformer 层作为骨干模型，用于建模复杂的特征交互；该模型在生成任务中已展现出相较于现有方法的显著优势 [10]。此外，我们采用跳跃连接来缓解过拟合 [12]，并使用前馈神经网络层对特征进行线性映射 [20]。因此，推荐决策的上下文信息可表示为

$$\widetilde{\boldsymbol{A}} = FFN\left(MultiHeadAttention\left(\boldsymbol{\tau}' - \{\boldsymbol{a}_t\}\right)\right), \tag{9}$$

其中，$\boldsymbol{A}$是预测动作嵌入矩阵，其第 t 行为 $\widetilde{\boldsymbol{a}}_t$，对应于 $t \in [1, T]$。$FFN(x) = GELU(x\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2$是带有跳跃连接的前馈神经网络层。这里，$GELU$ [13] 是 Transformer 模型中常用的激活函数。$MultiHeadAttention$ 表示 [33], 中定义的多头自注意力机制，该机制已被证明能够有效地联合学习来自不同表示子空间的信息，因此已广泛应用于推荐系统中 [18]。

2.6 动作解码器

给定每个时间步与物品交互的上下文信息，动作解码器旨在通过 GRU [7]解码用户感兴趣的物品序列。当前交互物品在解码时未知。我们仅使用先前交互和上下文信息来预测该物品，具体如下

$$\widehat{\boldsymbol{v}}_n = \boldsymbol{v}_n \oplus \widetilde{\boldsymbol{a}}_t \tag{10}$$

$$\widetilde{\boldsymbol{v}}_{n+1} = GRU_d\left(\widetilde{\boldsymbol{v}}_n, \widehat{\boldsymbol{v}}_n\right), \tag{11}$$

其中，$\boldsymbol{v}_n$是 v_n的嵌入，$\widetilde{\boldsymbol{v}}_{n+1}$是对第 $n+1$ 个位置物品的预测，$\oplus$表示拼接操作。由于没有可用于预测第一个物品的信息，我们使用'bos'标记作为起始标记，并随机初始化 $\widetilde{\boldsymbol{v}}_0$。完整序列可解码为

$$\begin{aligned}\widetilde{\boldsymbol{V}} &= \text{decoder}\left(\text{bos}, \widehat{\boldsymbol{v}}_1, \ldots, \widehat{\boldsymbol{v}}_n, \ldots, \widehat{\boldsymbol{v}}_{N-1}\right) \\ &= [\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_n, \ldots, \widetilde{\boldsymbol{v}}_N],\end{aligned} \tag{12}$$

其中，$\boldsymbol{V}$是第t个时间步交互物品序列的预测矩阵，N是最大序列长度。

请注意，序列的长度可能并不固定。进行预测时，我们不知道解码器中序列的长度，因此会在每个序列末尾添加一个'eos'标记作为结束标记，然后仍用零填充至 N 的长度。预测出'eos'后，解码器便不会继续预测。

2.7 对比监督策略学习

在决策变换器中，由于其设计目标是生成奖励最大的动作，因此推理时只会使用最大奖励。这样一来，奖励较低的样本可能无法得到充分利用（已在第 3.7 节中验证）。为了充分利用样本中的知识，我们提出采用加权对比学习方法，将奖励较低的动作视为负样本，以避免推荐奖励较低的动作。因此，我们的目标函数由两部分组成：交叉熵损失和加权对比学习损失。

2.7.1 加权对比损失。对于每个样本，我们使用状态和动作相同、奖励不同的样本作为负样本，将其预测动作嵌入记为 $\widetilde{\boldsymbol{V}}^-$。加权对比学习损失可表示为

$$\mathcal{L}_{\text{CL}} = -\sum_{\widetilde{\boldsymbol{V}}^- \in \Upsilon} \kappa\left(\widetilde{\boldsymbol{V}}^-\right) f_s\left(\widetilde{\boldsymbol{V}}, \widetilde{\boldsymbol{V}}^-\right), \tag{13}$$

其中，Υ 是负样本集合。$f_s(\cdot)$通过对矩阵 $\widetilde{\boldsymbol{V}}$和 $\widetilde{\boldsymbol{V}}^-$每一行的嵌入点积取平均，计算两个序列的相似度。$\kappa\left(\widetilde{\boldsymbol{V}}^-\right)$是根据负样本的奖励值设定的加权超参数，即权重与奖励成反比。原因是，较低的奖励会降低用户留存率，因此我们希望与这类样本不相似。

除了加权对比损失外，我们还使用原始交叉熵损失来优化决策 Transformer，具体如下：

$$\widehat{\boldsymbol{Y}} = \psi\left(\boldsymbol{W}_v \widetilde{\boldsymbol{V}} + \boldsymbol{b}_v\right) \tag{14}$$

$$\mathcal{L}_{CE} = CE\left(\widehat{\boldsymbol{Y}}, \boldsymbol{Y}\right), \tag{15}$$

其中，$\boldsymbol{Y}$是标签矩阵，每一行都包含一个独热标签；$\boldsymbol{Y}$是相应的预测标签分布。这里，交叉熵 是交叉熵损失函数，ψ是Softmax函数。$\boldsymbol{W}_v$和 $\boldsymbol{b}_v$是可学习参数。

最后，总体损失函数由交叉熵损失和加权对比损失组合而成，表示为：

$$\mathcal{L} = \mathcal{L}_{CE} + \beta \mathcal{L}_{CL}, \tag{16}$$

其中 β是一个超参数。

3 实验

本节在两个基准数据集——爱奇艺和 ML-1m——上开展实验，以回答以下研究问题。

- **研究问题1**：我们提出的 DT4Rec 与其他最先进的基线方法相比，性能如何?
- **研究问题 2**：我们的自动离散化奖励提示对奖励建模有何贡献?
- **研究问题3**：我们的对比监督策略学习方法如何缓解推荐训练与推理之间的分布外（OOD）问题?
- **RQ4**：我们设计的评估方法的稳定性如何?
- **RQ5**：我们的 DT4Rec 的泛化能力如何?

表 1：数据集统计。UR 是平均用户留存率的简称。

数据集	用户	物品	交互	UR	密度
爱奇艺	3,000,000	4,000,000	71,046,026	4.80	5.8e-6
ML-1m	6,014	3,417	1,000,000	4.13	4.84%

3.1 数据集

我们在两个真实数据集上进行了实验：爱奇艺用户留存数据（爱奇艺）和 ML-1m。表 1 介绍了这两个数据集的详细信息。具体而言，**爱奇艺数据集**[3] 记录了约 400 万个视频的 7,000 万条用户交互记录，例如观看次数和评论。该数据集非常稀疏且含有噪声。因此，为了不失一般性，我们选取至少有 20 条交互记录的用户进行实验。此外，我们不区分交互类型，而是将它们视为相同的交互。用户留存率定义为用户在下一周登录的平均天数，即 $K = 7$（这一设置采用自 WSDM Cup 2022[3]）。**ML-1m 数据集** [4] 是序列推荐系统中常用的基准数据集，记录了 6,014 位用户对 3,417 部电影的评分。由于 ML-1m 的记录跨越较长的时间范围，我们按月计算用户留存率。

3.2 评估

我们使用以下评估指标，评估所提出方法在即时反馈（即预测准确率）和长期参与度（即用户留存率）方面的表现。具体而言，这些指标定义如下。

3.2.1 预测准确率。 为评估预测准确率，我们采用序列预测问题中四种广泛使用的常见指标：BLEU [26], ROUGE [23],NDCG、命中率 [31]:

• BLEU 用于评估序列到序列推荐中动态长度推荐的精确率 [27], , 这也是自然语言处理中常用的指标。

• ROUGE 指动态长度推荐的召回率，是序列到序列推荐系统中常用的指标 [21]。

• HR@K用于衡量真实物品出现在前K个推荐中的概率。

• NDCG@K用于衡量前K个推荐中的累积增益（CG）分数，并考虑位置对推荐的影响 [34], , 其中CG分数表示物品与真实值之间的相似度。

3.2.2 用户留存。为了评估生成的推荐序列的有效性，我们设计了两个指标MB-URS和SB-URS，并采用了两个常用指标：改进用户留存（IUR）和未返回计数（NRC） [35]。

• **MB-URS**。基于**模**型的**用**户**留**存**得**分**评**估推荐物品序列的效果，方法是直接返回用户留存得分。具体而言，我们使用独立于训练数据集的 30% 验证数据集，训练一个监督模型来预测特定状态-动作对的奖励

[3]http://challenge.ai.iqiyi.com/ [4] https://grouplens.org/datasets/movielens/1m/

。因此，通过最小化预测奖励与真实值之间的均方误差损失来优化模型。

• **SB-URS**。基于**相似度的用户留存分数**通过对真实用户留存分数进行加权求和，评估推荐物品序列的有效性。具体而言，我们根据真实奖励将样本分为8个类别，并计算每个类别中预测序列与真实值之间的BLEU-1分数，作为其相似度。随后，按如下方式计算SB-URS：

$$\text{SB-URS} = \sum_{k=0}^{K} s_k \cdot \left(g_k - \frac{K}{2}\right) \cdot N_k, \tag{17}$$

其中， s_k 是类别 k 的相似度。 g_k 是类别 k 的真实奖励。对于真实奖励较小的样本，我们希望相似度尽可能小；对于真实奖励较大的样本，我们希望相似度尽可能大。 N_k 是奖励为 k的样本数量。

• **改进用户留存（IUR）**。它衡量相较于离线数据中记录的用户平均留存率，用户平均留存率的相对提升百分比，直接反映用户对系统的参与度。IUR 百分比越高，表明推荐列表能让更多用户参与到系统中。

• **未返回计数（NRC）**。它表示用户在收到推荐后离开系统的比例。未返回计数越低，表明推荐列表能让更多用户参与到系统中。

3.3 基线方法

我们将模型与不同类型推荐方法中的最先进方法进行比较，包括：

• **BERT4Rec [30]**: 它使用双向 Transformer 学习序列信息。此外，它采用掩码语言模型来增加任务难度并增强模型的训练能力。

• **SASRec [18]**: 它采用从左到右的单向 Transformer 来捕捉用户的序列偏好。

• **LIRD [45]**: 它结合了基于策略的梯度和基于价值的 DQN，并采用演员-评论家（AC）框架。在每个推荐步骤中，系统会推荐多个物品，并由模拟器自动学习最优推荐策略，以最大化来自用户的长期奖励。

• **TopK [3]**: 它在离线学习中采用基于策略的强化学习方法，并使用重要性权重解决分布失配问题。它假设一组不重复物品的奖励等于其中每个物品的奖励之和，并在每个时间步向用户推荐多个物品。本文将此模型称为 TopK。

• **DT4Rec-R**：为说明奖励在引导模型方面的重要性，我们训练了一个不考虑奖励的 DT模型，记为 DT4Rec-R。

3.4 超参数设置

在提示生成器中，我们设置桶数量 $B = 10$、跳跃连接 $\alpha = 0.1$。在动作编码器和解码器中，我们将最大序列长度设为 $N = 20$，RNN 层数设为 1。在决策变换器中，

表2：预测准确率的整体性能比较。最佳性能以粗体标出。

数据集	模型	指标			
		BLEU↑	ROUGE↑	NDCG↑	HR↑
爱奇艺	BERT4Rec	0.7964	0.7693	0.7384	0.7673
	SASRec	0.8009	0.7906	0.7827	0.7815
	DT4Rec	**0.8249***	**0.8172***	**0.8139***	**0.8044***
ML-1m	BERT4Rec	0.3817	0.3806	0.5286	0.2769
	SASRec	0.4052	0.3983	0.5409	0.3123
	DT4Rec	**0.4331***	**0.4185***	**0.5679***	**0.3342***

“*”表示相较于最佳基线取得了统计显著性提升（即，采用 $p < 0.05$ 进行双侧 t 检验）。↑：数值越高越好；↓：数值越低越好。

表 3：用户留存方面的整体性能比较。最佳性能以粗体标出。

数据集	模型	指标			
		MB-URS↑	SB-URS↑	IUR↑	NRC↓
爱奇艺	DT4Rec-R	5.16	52131	7.5%	2.6%
	TopK	5.41	61045	12.7%	2.0%
	LIRD	5.63	63572	17.3%	1.9%
	DT4Rec	**6.05***	**72270***	**26.0%***	**1.4%***
ML-1m	DT4Rec-R	5.42	13050	31.2%	9.7%
	TopK	5.71	13964	38.3%	6.9%
	LIRD	5.86	14627	41.9%	4.8%
	DT4Rec	**5.93***	**15562***	**43.6%***	**3.6%***

“*”表示相较于最佳基线取得了统计显著性提升（即，使用 $p < 0.05$ 进行双侧 t 检验）。↑：越高越好；↓：越低越好。

我们将最大轨迹长度 T 设为 {10,20, 30, 40, 50}，Transformer 层数设为 2，注意力头数设为 8，嵌入维度设为 128。我们选择 AdamW 作为优化器，并将学习率设为 0.01。为降低计算成本，我们仅将最近交互过的 30 个物品保存为状态。对于其他基线方法，超参数按照其原作者的建议设为最优值，或在与我们的模型相同的范围内进行搜索。各模型的结果均在其最优超参数设置下报告。实现代码可在线获取[5]。

3.5 研究问题1：总体比较

3.5.1 预测准确率。如表2所示，我们记录了所有模型在预测准确率任务上的最佳结果。从表2中，我们得出以下观察：

• 我们的模型在两个数据集上的表现均优于所有基线方法。这体现了我们的模型在优化即时用户反馈方面的有效性。与现有方法相比，我们的模型充分利用强化学习，对用户在每个时间步的奖励和状态进行建模。这使我们的模型能够动态建模用户的兴趣特征，以满足用户不断变化的需求，因此我们的模型在预测准确率任务上能够取得更优异的表现。

表4：爱奇艺数据集上的消融研究。

架构	指标		
	SB-URS	MB-URS	IUR
DT4Rec	72270	6.05	26.0%
不使用对比学习	65175 (-9.82%)	5.68 (-6.11%)	18.3% (-29.6%)
不使用权重	69316 (-4.09%)	5.79 (-4.30%)	20.6% (-20.8%)
不使用自动判别	70953 (-1.82%)	5.96 (-1.49%)	24.2% (-6.9%)

• 在两个数据集上，SASRec 的表现都优于 BERT4Rec。我们的模型和 SASRec 都采用单向 Transformer，而 BERT4Rec 采用双向 Transformer。由于我们的任务是端到端生成任务，单向 Transformer 更适合该任务。

3.5.2 用户留存。表 3 记录了所有模型在用户留存最大化任务上的最佳结果。我们可以得出以下结论：
• 我们的模型在 MB-URS 和 SB-URS 上均优于其他基线方法。DT4Rec 在爱奇艺和 ML-1m 上的表现均大幅优于最佳基线 LIRD。• 传统离线方法（即 TopK 和 LIRD）仍然受到“Dead Traid”问题和折扣因子导致的短视问题的影响，因为它们的表现明显逊于所提出的方法。

• 此外，基于策略梯度的 TopK 模型容易收敛到次优解，而且确定合适的学习率颇具挑战。由于序列推荐系统中的动作空间很大，基于演员-评论家框架的 LIRD 模型在估计 Q 值时更加困难。
• 然而，我们的模型通过序列建模学习奖励、状态与动作之间的映射，无需自举，从而避免了“死三元组”问题。与传统的强化学习策略不同，我们的模型也无需估计 Q 值，从而规避了序列推荐系统中动作空间过大的问题。与 DT4Rec-R 的比较说明，奖励对于学习用户的长期偏好至关重要。

3.6 研究问题 2：自动离散化奖励提示的有效性

通过比较 DT4Rec 和 DT4Rec-R 的结果，我们证明了奖励对指导模型训练的重要性。在本小节中，我们通过消融研究进一步展示了所提出的奖励偏序建模改进的有效性。具体而言，我们将 DT4Rec 与使用朴素提示的模型进行比较，该模型由单层前馈神经网络实现，即表4中的“不使用自动判别”。实验结果表明，自动离散化奖励提示是有效的。若不使用自动离散化奖励提示，MB-URS 和 SB-URS 上的性能分别下降 1.82% 和 1.49 %，这表明自动离散化奖励提示能够使生成的嵌入保持其对应奖励值之间的偏序关系。

[5]https://github.com/kesenzhao/DT4Rec.git

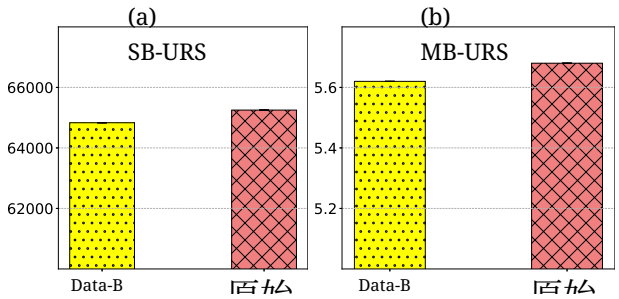

图 2：在原始爱奇艺数据集上训练的模型与移除奖励值较小部分后的数据集（即 Data-B）上训练的模型之间的比较。

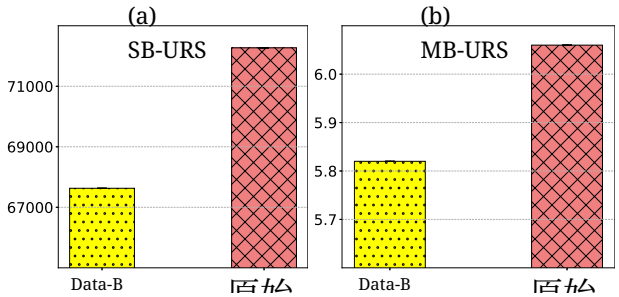

图 3：在原始爱奇艺数据集上训练的 DT4Rec 与移除奖励值较小部分后的数据集（即 Data-B）上训练的 DT4Rec 之间的比较。

3.7 研究问题3：对比监督策略学习的有效性

我们进一步揭示了决策 Transformer 存在的分布外问题，并通过以下分析实验有力地说明了这些改进的有效性。

分布外问题分析。在 DT 模型中，由于生成推荐时仅输入奖励的最大值，模型会利用高奖励值样本中的知识。因此，奖励较低样本中的知识不会被纳入最终推荐。具体而言，我们评估了在原始数据集和移除奖励值低于 4 的样本后的 Data-B 上训练、且未采用对比学习的模型。图 2 展示了爱奇艺数据集上的结果。从图 2 可以看出，模型在 Data-B 和原始数据集上的表现差异可以忽略不计。这表明模型几乎没有利用高奖励样本中的知识，从而验证了我们关于分布外问题的论断。

加权对比学习方法的有效性（研究问题*3*）。在本段中，我们从提升性能和充分利用低奖励样本中的知识两个方面，验证加权对比学习的有效性。对于前者，我们将 DT4Rec 与不包含对比学习损失的模型进行比较，并将其记为“不使用对比学习”，以证明加权对比学习的优越性。表4中显著的性能下降验证了加权

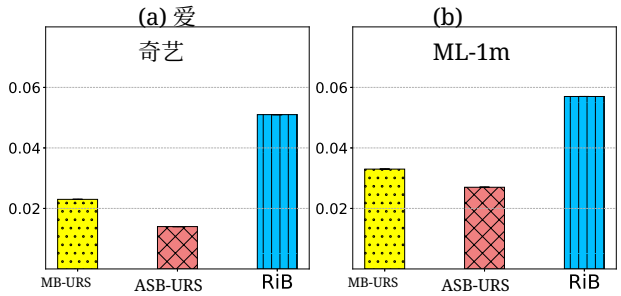

图4：不同评估方法在爱奇艺和 ML-1m 数据集上的方差比较。

对比学习方法的有效性。为了利用低奖励样本中的知识，我们在图3中将 DT4Rec 与不使用低奖励样本的变体进行比较。具体而言，我们绘制了原始模型以及不使用低奖励样本的模型在 SB-URS 和 MB-URS 上的性能。从图中可以清楚地看到，在原始爱奇艺数据集上，我们的模型在 SB-URS 和 MB-URS 上的表现均优于不使用低奖励样本的模型，表明我们的模型充分利用了低奖励样本。原因在于，我们的模型会避免推荐与低奖励样本相似的物品序列，这与我们对加权对比学习方法的直觉一致。

3.8 RQ4：评估效率

我们通过将 MB-URS 和 SB-URS 与 Wu等人提出的评估方法进行比较，来说明 MB-URS 和 SB-URS 的有效性，并将后者记为“RiB”。我们将测试集划分为 5 个子集，并计算这 5 个测试集上的性能方差，作为评估方法稳定性的指标。为确保比较时规模相同，我们用 ASB-URS 替换 SB-URS。ASB-URS 通过将公式（17）中的 N_k 除以**平**均**相**似**用**户**回**报**分**数来进行评估。图4展示了评估方法在爱奇艺和 ML-1m 数据集上的方差。结果表明，我们的评估方法在两个数据集上的方差都很小，证实了所提方法的有效性。

3.9 RQ5：模型泛化分析

在推理阶段，生成推荐时不使用真实奖励，而是仅依据当前状态使用最大奖励。因此，训练样本中可能不存在最大奖励，从而导致训练与测试不匹配。因此，模型需要具备较强的泛化能力，而不能只是在原始数据集上针对某个特定回报进行模仿学习。我们在爱奇艺数据集上开展了行为克隆分析实验。具体而言，我们将奖励为 6 或 7 的样本视为高奖励部分，并将数据集重新划分为高奖励部分和非高奖励部分。此外，我们控制高奖励样本占整个数据集的比例，以研究模型的泛化能力。在图 5 中，我们绘制了 DT4Rec 和 TopK 在高奖励样本比例不同的数据集上的 MB-URS 和 SB-URS，即 [10, 25, 40, 100]。从图中，我们观察到以下

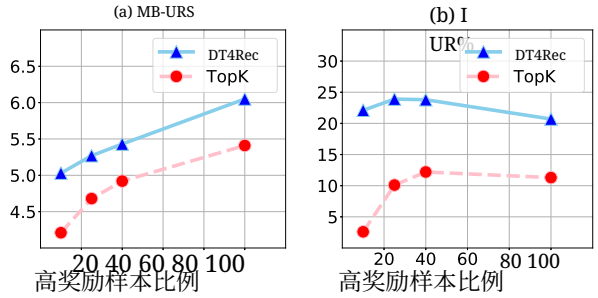

图 5：爱奇艺数据集上的行为克隆分析。

事实：（1）在这两个新数据集上，我们的模型表现均优于 TopK，表明所提出的方法具有较强的泛化能力。（2）即使只使用 10% 的高奖励样本，DT4Rec 的 SB-URS 仍高于 20%。这表明所提出的方法在提升用户留存率方面效果显著。当高奖励样本较少时，对高奖励样本进行策略评估会变得不准确且困难。然而，我们的模型以监督方式训练，避免了策略评估的高方差，并提升了模型在测试集上的表现。

4 相关工作

我们将与本研究相关的早期文献，即序列推荐和基于强化学习的推荐系统，总结如下。

4.1 序列推荐系统

序列推荐系统旨在处理交互行为的顺序问题 [22, 38]。早期方法主要基于马尔可夫链和马尔可夫过程。例如，FPMC 结合矩阵分解（MF）和马尔可夫链（MC），为每位用户学习一个转移矩阵 [28]；Hosseinzadeh 等人通过层次马尔可夫模型捕捉用户偏好的变化 [16]。除了基于 MC 的方法外，还有基于循环神经网络的方法。例如，GRU4Rec [32]率先提出充分利用循环神经网络模型学习序列信息来构建推荐系统；GRU4RecF 则提出并行循环神经网络，以利用物品特征并提高推荐效率 [15]。然而，这些研究都没有针对长期参与度进行优化，例如用户回访次数 [9], 和浏览深度 [47], ，而这对提高推荐质量至关重要。

4.2 基于强化学习的推荐

由于在推荐系统中进行试错搜索成本过高，因此这里仅介绍离线强化学习方法。一般而言，基于强化学习的推荐系统主要分为两类：基于模型和无模型。

基于模型的模型。 基于模型的模型通过学习固定模型 [6, 43, 48, 50] 来表示整个环境。然而，在推荐场景中，由于环境不断变化，模型对环境的表示会产生偏差。如果持续更新模型以适应环境变化，将会产生巨大的开销。此外，用户行为发生的概率无法估计，转移概率函数也难以确定。

无模型方法。 在无模型方法中，状态转移概率函数未知，且无需使用 [6, 49]。DEERS [44] 指出，负反馈也会对推荐系统产生影响。RML [25]通过直接优化相关反馈指标，改进了以往的启发式方法。TPGR [1] 提出了一种基于树结构策略的模型，以解决离散动作空间过大的问题。LIRD[45]利用演员-评论家框架进行列表级推荐。DeepPage [41] 进一步考虑了推荐物品序列的二维位置。尽管无模型方法的效果优于基于模型的方法，但仍存在许多问题。首先，在推荐系统中，由于动作空间很大，价值函数难以近似估计，奖励函数也难以确定。其次，策略难以评估，而且评估方法的方差通常没有上界。

5 结论

本文提出了一种新颖的、基于强化学习的序列推荐系统 DT4Rec。具体而言，该方法将强化学习转化为自回归问题，从而避免了不稳定性问题和无界方差带来的困扰。此外，我们还提出了一系列技术，以促进 DT4Rec 的成功应用。具体来说，自动离散化奖励提示的设计能够有效建模奖励的数值，并引导模型训练，使其着眼于长期用户参与度。我们提出的对比式监督策略学习，减轻了朴素决策变换器在推理与训练之间的不一致性。为了评估模型，我们提出了两个稳定的指标，即 MB-URS 和 SB-URS，并验证了它们比现有指标更加稳定。在基准数据集上开展的大量实验表明，所提方法行之有效。

致谢

本研究部分获得了 APRC - CityU 新研究计划（编号：9610565，香港城市大学新教员启动基金）、SIRG - CityU 战略跨学科研究基金（编号：7020046、7020074）、HKIDS 早期职业研究基金（编号：9360163）、华为创新研究计划和蚂蚁集团（CCF-蚂蚁研究基金）的资助。

参考文献

[1] Haokun Chen、Xinyi Dai、Han Cai、Weinan Zhang、Xuejian Wang、Ruiming Tang、Yuzhou Zhang 和 Yong Yu。2019。基于树结构策略梯度的大规模交互式推荐。载于*AAAI* 人工智能大会论文集，第 33 卷。3312–3320。[2] Lili Chen、Kevin Lu、Aravind Rajeswaran、Kimin Lee、Aditya Grover、Misha Laskin、Pieter Abbeel、Aravind Srinivas 和 Igor Mordatch。2021。决策变换器：通过序列建模进行强化学习。神经信息处理系统进展 34（2021），15084–15097。[3] Minmin Chen、Alex Beutel、Paul Covington、Sagar Jain、Francois Belletti 和 Ed H Chi。2019。用于 REINFORCE 推荐系统的 Top-k 离策略校正。载于第十二届 *ACM* 国际网络搜索与数据挖掘会议论文集。456–464。[4] Minmin Chen、Bo Chang、Can Xu 和 Ed H Chi。2021。用于改进 REINFORCE 推荐系统的用户响应模型。载于第 *14* 届 *ACM* 国际网络搜索与数据挖掘会议论文集。121–129。[5] Xinshi Chen、Shuang Li、Hui Li、Shaohua Jiang、Yuan Qi 和 Le Song。2019。用于基于强化学习的推荐系统的生成对抗用户模型。载于国际机器学习会议。PMLR，1052–1061。

[6] Xiaocong Chen、Lina Yao、Julian McAuley、Guanglin Zhou 和 Xianzhi Wang。2021。推荐系统中的深度强化学习综述：系统性回顾与未来方向。arXiv 预印本 arXiv:2109.03540（2021）。[7] Kyunghyun Cho、Bart Van Merriënboer、Caglar Gulcehre、Dzmitry Bahdanau、Fethi Bougares、Holger Schwenk 和 Yoshua Bengio。2014。使用用于统计机器翻译的 RNN编码器-解码器学习短语表示。arXiv 预印本 arXiv:1406.1078（2014）。[8] Abhinandan S Das、Mayur Datar、Ashutosh Garg 和 Shyam Rajaram。2007。谷歌新闻个性化：可扩展的在线协同过滤。载于第 16 届万维网国际会议论文集。271–280。[9] Nan Du、Yichen Wang、Niao He、Jimeng Sun 和 Le Song。2015。基于用户重复活动的时间敏感型推荐。神经信息处理系统进展 28（2015）。[10] Luciano Floridi 和 Massimo Chiriatti。2020。GPT-3：其本质、范围、局限与影响。Minds and Machines 30，4（2020），681–694。[11] Xavier Glorot、Antoine Bordes 和 Yoshua Bengio。2011。深度稀疏整流神经网络。载于第十四届人工智能与统计学国际会议论文集。JMLR Workshop and Conference Proceedings，315–323。[12] 何恺明、张祥雨、任少卿和孙剑。2016。用于图像识别的深度残差学习。载于电气电子工程师学会计算机视觉与模式识别会议论文集。770–778。[13] Dan Hendrycks 和 Kevin Gimpel。2016。高斯误差线性单元（GELU）。arXiv 预印本 arXiv:1606.08415（2016）。[14] Balázs Hidasi、Alexandros Karatzoglou、Linas Baltrunas 和 Domonkos Tikk。2015。基于循环神经网络的会话推荐。arXiv 预印本 arXiv:1511.06939（2015）。[15] Balázs Hidasi、Massimo Quadrana、Alexandros Karatzoglou 和 Domonkos Tikk。2016。用于富特征的基于会话推荐的并行循环神经网络架构。载于第 10 届 ACM 推荐系统会议论文集。241–248。[16] Mehdi Hosseinzadeh Aghdam、Negar Hariri、Bamshad Mobasher 和 Robin Burke。2015。使用分层隐马尔可夫模型使推荐适应情境变化。载于第 9 届 ACM 推荐系统会议论文集。241–244。[17] Eugene Ie、Vihan Jain、Jing Wang、Sanmit Narvekar、Ritesh Agarwal、Rui Wu、Heng-Tze Cheng、Tushar Chandra 和 Craig Boutilier。2019。SlateQ：一种用于推荐集合强化学习的易处理分解方法。（2019）。[18] Wang-Cheng Kang 和 Julian McAuley。2018。自注意力序列推荐。载于 2018 年电气电子工程师学会国际数据挖掘会议（ICDM）。电气电子工程师学会，197–206。[19] Shigeki Karita、Nanxin Chen、Tomoki Hayashi、Takaaki Hori、Hirofumi Inaguma、Ziyan Jiang、Masao Someki、Nelson Enrique Yalta Soplin、Ryuichi Yamamoto、Xiaofei Wang 等。2019。语音应用中 Transformer 与 RNN 的比较研究。载于 2019 年电气电子工程师学会自动语音识别与理解研讨会（ASRU）。电气电子工程师学会，449–456。[20] Jacob Devlin、Ming-Wei Chang、Kenton 和 Lee Kristina Toutanova。2019。BERT：用于语言理解的深度双向 Transformer 预训练。载于北美计算语言学协会人类语言技术分会论文集。4171–4186。[21] Mikhail A Khovrichev、Marina A Balakhontceva 和 Mikhail V Ionov。2018。评估用于模拟医务人员随时间流动的序列到序列模型。Procedia计算机科学 136（2018），425–432。[22] Muyang Li、Xiangyu Zhao、Chuan Lyu、Minghao Zhao、Runze Wu 和 Ruocheng Guo。2022。MLP4Rec：一种用于序列推荐的纯MLP架构。载于第 31 届国际人工智能联合会议和第 25 届欧洲人工智能会议（IJCAI-ECAI 2022）。国际人工智能联合会议，2138–2144。[23] Chin-Yew Lin。2004。ROUGE：一种用于摘要自动评估的工具包。载于《文本摘要新进展》。74–81。[24] Yuanguo Lin、Yong Liu、Fan Lin、Pengcheng Wu、Wenhua Zeng 和 Chunyan Miao。2021。推荐系统强化学习综述。arXiv 预印本 arXiv:2109.10665（2021）。[25] Ali Montazeralghaem、Hamed Zamani 和 James Allan。2020。相关性反馈的强化学习框架。载于第 43 届信息检索研究与发展国际会议论文集。59–68。[26] Kishore Papineni、Salim Roukos、Todd Ward 和 Wei-Jing Zhu。2002。BLEU：一种用于机器翻译自动评估的方法。载于计算语言学协会第 40 届年会论文集。311–318。[27] Matt Post。2018。呼吁清晰报告 BLEU分数。载于第三届机器翻译会议：研究论文集。186–191。[28] Steffen Rendle、Christoph Freudenthaler 和 Lars Schmidt-Thieme。2010。个性化马尔可夫链因子分解，用于下一购物篮推荐。载于第 19 届万维网国际会议论文集。811–820。[29] David Silver、Aja Huang、Chris J Maddison、Arthur Guez、Laurent Sifre、George Van Den Driessche、Julian Schrittwieser、Ioannis Antonoglou、Veda Panneershelvam、Marc Lanctot 等。2016。运用深度神经网络和树搜索掌握围棋。《自然》529，7587（2016），484–489。[30] Fei Sun、Jun Liu、Jian Wu、Changhua Pei、Xiao Lin、Wenwu Ou 和 Peng Jiang。2019 年。BERT4Rec：基于 Transformer 双向编码器表征的序列推荐。载于第28届 ACM 信息与知识管理国际会议论文集。1441–1450。[31] Ilya Sutskever、Oriol Vinyals 和 Quoc V Le。2014年。利用神经网络进行序列到序列学习。神经信息处理系统进展 27（2014年）。[32] Yong Kiam Tan、Xinxing Xu 和 Yong Liu。2016年。用于基于会话的推荐的改进型循环神经网络。载于第1届推荐系统深度学习研讨会论文集。17–22。[33] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。注意力就是你所需要的一切。神经信息处理系统进展 30（2017年）。[34] Yining Wang、Liwei Wang、Yuanzhi Li、Di He、Wei Chen 和 Tie-Yan Liu。2013年。NDCG 排序指标的理论分析。载于第26届学习理论年度会议论文集（COLT 2013），第8卷。6。[35] Qingyun Wu、Hongning Wang、Liangjie Hong 和 Yue Shi。2017年。回访即是信任：优化推荐系统中的长期用户参与度。载于2017年 ACM 信息与知识管理会议论文集。1927–1936。[36] Xin Xin、Tiago Pimentel、Alexandros Karatzoglou、Pengjie Ren、Konstantina Christakopoulou 和 Zhaochun Ren。2022年。重新思考推荐中的强化学习：提示视角。载于第45届国际 ACM 信息检索研究与发展会议论文集。1347–1357。[37] Xing Yi、Liangjie Hong、Erheng Zhong、Nanthan Nan Liu 和 Suju Rajan。2014年。超越点击：用于个性化的停留时间。载于第8届 ACM 推荐系统会议论文集。113–120。[38] Chi Zhang、Yantong Du、Xiangyu Zhao、Qilong Han、Rui Chen 和 Li Li。2022年。用于序列推荐中序列去噪的分层物品不一致信号学习。载于第31届 ACM 信息与知识管理国际会议论文集。2508–2518。[39] Weinan Zhang、Xiangyu Zhao、Li Zhao、Dawei Yin、Grace Hui Yang 和 Alex Beutel。2020年。信息检索中的深度强化学习：基础与进展。载于第43届国际 ACM 信息检索研究与发展会议论文集。2468–2471。[40] Xiangyu Zhao、Long Xia、Jiliang Tang 和 Dawei Yin。2019年。面向搜索、推荐和在线广告的深度强化学习：综述。ACM SIGWEB 通讯 春季（2019年），1–15。[41] Xiangyu Zhao、Long Xia、Liang Zhang、Zhuoye Ding、Dawei Yin 和 Jiliang Tang。2018年。用于逐页推荐的深度强化学习。载于第12届 ACM 推荐系统会议论文集。95–103。[42] Xiangyu Zhao、Long Xia、Lixin Zou、Hui Liu、Dawei Yin 和 Jiliang Tang。2020年。全链路推荐。载于第29届 ACM 信息与知识管理国际会议论文集。1883–1891。[43] Xiangyu Zhao、Long Xia、Lixin Zou、Hui Liu、Dawei Yin 和 Jiliang Tang。2021年。Usersim：通过监督式生成对抗网络进行用户模拟。载于2021 年 Web 会议论文集。3582–3589。[44] Xiangyu Zhao、Liang Zhang、Zhuoye Ding、Long Xia、Jiliang Tang 和 Dawei Yin。2018年。通过成对深度强化学习利用负反馈进行推荐。载于第24届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。ACM，1040–1048。[45] Xiangyu Zhao、Liang Zhang、Zhuoye Ding、Dawei Yin、Yihong Zhao 和 Jiliang Tang。2017年。用于列表式推荐的深度强化学习。arXiv 预印本 arXiv:1801.00209（2017年）。[46] Lixin Zou、Weixue Lu、Yiding Liu、Hengyi Cai、Xiaokai Chu、Dehong Ma、Daiting Shi、Yu Sun、Zhicong Cheng、Simiu Gu 等。2022年。基于预训练语言模型的网页搜索检索与排序。ACM 网页学报 17，1（2022年），1–36。[47] Lixin Zou、Long Xia、Zhuoye Ding、Jiaxing Song、Weidong Liu 和 Dawei Yin。2019年。通过强化学习优化推荐系统中的长期用户参与度。载于第25届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。2810–2818。[48] Lixin Zou、Long Xia、Pan Du、Zhuo Zhang、Ting Bai、Weidong Liu、Jian-Yun Nie 和 Dawei Yin。2020年。Pseudo Dyna-Q：用于交互式推荐的强化学习框架。载于第13届国际网页搜索与数据挖掘会议论文集。816–824。[49] Lixin Zou、Long Xia、Yulong Gu、Xiangyu Zhao、Weidong Liu、Jimmy Xiangji Huang 和 Dawei Yin。2020年。神经交互式协同过滤。载于第43届国际 ACM 信息检索研究与发展会议论文集。749–758。[50] Lixin Zou、Long Xia、Linfang Hou、Xiangyu Zhao 和 Dawei Yin。2021年。用于疟疾防控的数据高效强化学习。arXiv 电子印本（2021年），arXiv–2105。[51] Lixin Zou、Shengqiang Zhang、Hengyi Cai、Dehong Ma、Suqi Cheng、Shuaiqiang Wang、Daiting Shi、Zhicong Cheng 和 Dawei Yin。2021年。百度搜索中基于预训练语言模型的排序。载于第27届 ACM SIGKDD 知识发现与数据挖掘会议论文集。4014–4022。