arXiv:2302.03431v2 [cs.IR] 8 Feb 2023

推荐中潜在动作空间的探索与正则化

Shuchang Liu 快手科技 中国北京
liushuchang@kuaishou.com

Qingpeng Cai 快手科技 中国北京
caiqingpeng@@kuaishou.com

Bowen Sun 北京大学 中国北京
b
wzdbml@gmail.com

Yuhao Wang 香港城市大学 中国香港
yhwang25-c@my.cityu.edu.hk

Ji Jiang 北京大学 中国北京
johnji
m0816@gmail.com

Dong Zheng 快手科技 中国北京
zhengdong@kuaishou.com

Kun Gai 未隶属任何机构 中国北京
gai.kun@qq.com

Peng Jiang†快手科技 中国北京
jiangpeng@kuaishou.com

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

Yongfeng Zhang 罗格斯大学 美国新泽西州新不伦瑞克
yongfeng.zhang@rutgers.edu

摘要

在推荐系统中，强化学习解决方案能够捕捉长期用户-系统交互，因此有效提升了推荐性能。然而，推荐策略的动作空间是一个物品列表；当候选物品池动态变化时，该列表可能极其庞大。为应对这一挑战，我们提出了一种超动作器与评论家学习框架，其中策略将物品列表生成过程分解为超动作推断步骤和效果动作选择步骤。第一步将给定的状态空间映射到向量化的超动作空间，第二步则基于超动作选择物品列表。为了调节两个动作空间之间的差异，我们设计了一个对齐模块，并为物品设计核映射函数，以确保推断准确率；此外，还引入了一个监督模块来稳定学习过程。我们基于公共数据集构建了模拟环境，并通过实验证明，与标准强化学习基线相比，我们的框架在推荐方面表现更优。

CCS 概念

• **信息系统** → **推荐系统**； • **计算方法** →**序贯决策**。

关键词

推荐系统、强化学习、表示学习

ACM 参考格式：
Shuchang Liu、Qingpeng Cai、Bowen Sun、Yuhao Wang、Ji Jiang、Dong Zheng、Kun Gai、Peng Jiang†、Xiangyu Zhao 和 Yongfeng Zhang。2023。推荐中潜在动作空间的探索与正则化。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，12 页。https://doi.org/10. 1145/3543507.3583244

1 引言

推荐系统（RS）是电子商务、社交媒体、新闻和广告等众多网络服务的基础组成部分之一。近年来，研究表明，用户与推荐系统之间的长期交互构成了一个马尔可夫决策过程（MDP），在这一过程中，可以采用强化学习（RL）方法进一步提升预测性能 [1]，相比传统的学习排序解决方案 [32]。强化学习背后的核心思想并非优化即时用户响应，而是最大化随时间推移的所有交互所产生的累积奖励。

在实践中采用这一表述时，推荐问题与其他强化学习任务的区别在于规模庞大且动态变化的离散动作空间带来的挑战。例如，在电子商务系统或视频推荐系统中，每次请求都可能需要从数百万个物品中选出一个推荐列表，

† 通讯作者。

允许免费制作本作品的全部或部分内容的数字版或纸质版，供个人或课堂使用，但不得为营利或商业利益而复制或分发，且所有副本均须保留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方拥有版权的部分，必须尊重其版权。注明出处后允许摘录。以其他方式复制、再版、发布到服务器或重新分发至列表，须事先获得特别许可和/或支付费用。请通过 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/10.1145/3543507.3583244

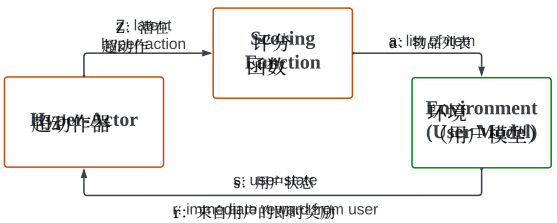

图 1：采用潜在超动作的马尔可夫决策过程建模

而候选池每天都在增长。这意味着，偏好小动作空间的基于表格的方法（例如 Q学习、SARSA、策略迭代）以及为固定动作空间设计的方法都不切实际。尽管人们尝试将推荐列表 [21] 分解为逐物品的子动作来缓解这一问题，但要在整个候选物品池上学习策略梯度仍然是一项艰巨的任务。

幸运的是，早期的非强化学习模型已经解决了这一挑战，例如潜因子模型 [26]和双塔协同过滤方法 [50]。它们学习一个共同的潜在空间，用于表示用户请求和物品（或物品列表），因此该潜在空间能够容纳任意数量的物品，且不受物品池动态变化的影响。在本研究中，我们将这一思路融入强化学习方法，并聚焦于列表推荐问题。推荐列表的潜在表示被概括为超动作，如图 1 所示。在前向推理中，策略首先提出一个向量化的超动作，然后该潜向量作为一个确定性函数，用于排序，最终从候选池中选出一个物品列表，记为效果动作。

需要注意的是，这一额外的隐式推理步骤也给强化学习解决方案带来了新的挑战：最重要的是，它导致两个动作空间之间出现不一致。具体而言，我们希望对超动作进行高效的端到端训练，但实际与用户交互的却是效果动作。另一方面，离散效果动作最准确的潜在表示，可能与用于选出该动作的超动作的表示并不完全相同，因此无法保证推断准确率。此外，这也引入了一个额外的探索阶段，而且尚不清楚应该在潜在空间中探索超动作，还是探索离散效果-动作空间。所有这些因素叠加在一起，增加了学习过程的不稳定性和不确定性，因此我们需要一种能够规范这两个动作空间并稳定强化学习过程的解决方案。

为解决上述挑战，我们提出了一种通用的超演员-评论家（HAC）学习框架，该框架包含四个组件：用户状态和超动作生成器；将超动作映射为效果动作（即推荐列表）的评分函数；评估超动作空间和效果-动作空间的评论家网络；以及根据效果动作反推超动作的逆映射模块。在训练过程中，我们在骨干演员-评论家学习范式的基础上增加了一个对齐模块，以确保两个动作空间之间的一致性，并增加了一个监督模块，以提升强化学习的稳定性和有效性。由此形成的框架推广了许多现有的推荐任务解决方案，例如 DDPG 和在线/离线监督学习（SL）。

我们的贡献总结如下：

- 我们提出了一种实用且高效的强化学习框架，通过潜在超动作学习如何从大型物品池中推荐一系列物品。
- 我们基于公共数据集构建了在线模拟器，并通过实验证明，与标准强化学习和监督学习方案相比，所提出的框架性能更优。
- 我们还指出，提供监督并规范超动作空间与效果-动作空间之间的一致性，有助于提高采样效率和推断准确率。

2 方法

2.1 问题定义

这里我们考虑基于会话的推荐场景：系统面对一个大小为 N 的物品池，记为 I；对于每位用户，我们观测到静态用户特征 $\boldsymbol{u}$ 以及一个会话的交互历史 $x_{1:t}=\{x_1, \ldots\ldots, x_t\}$，其中 $x_i \in \mathrm{I}(1 \leq i \leq t)$。目标是在整个交互会话中，通过交互式地向用户推荐物品列表，使用户的累积奖励最大化。如第 1 节所述，我们强调潜在动作空间的存在，并将推荐任务建模为一个经修改的马尔可夫决策过程（MDP），如图 1 所示。引入潜在动作空间后，MDP 的组成部分如下：

- S：用户状态的连续表示空间。
- A：最终的效果-动作空间对应于可能的推荐列表。不失一般性，我们考虑固定大小为 k 的列表，因此动作空间为 $\mathrm{A}=\mathrm{I}^k$。
- Z：潜在超动作空间，是一个连续向量空间，用于编码效果动作的选择方式。我们假设存在一个多对一映射函数 $f: \mathrm{Z} \rightarrow \mathrm{A}$。
- R：累积奖励函数，用于估计用户会话中的用户反馈；即时奖励$r(s, a)$则表示在状态 $s \in \mathrm{S}$ 上执行动作 $a \in \mathrm{A}$ 时获得的单步奖励。

请注意，其中隐含了一个转移模型 $\mathrm{P}(s_t|s_t, a_t)$，用于描述在状态 s_t 上执行动作 a_t 后到达某个新状态 s_t的概率。在推荐系统中，该转移函数被整合到用户状态编码器中，通常由一个以用户交互序列为输入的序列模型进行建模。

在用户会话的每个交互步骤中，给定用户当前的状态 $s_t \in \mathrm{S}$（例如用户画像和历史交互记录），推荐策略 $\pi_\theta(a_t|s_t)$ 首先推断出一个超动作表示 Z_t，然后生成一个物品列表作为效果动作 a_t。用户环境会返回用户反馈以及更新后的用户状态。奖励函数 $r(s_t, a_t)$ 被视为已知，因此也被视为环境的一部分。目标是找到一个最优推荐策略 $\pi^*(a_t|s_t): \mathrm{S} \rightarrow \mathrm{A}$，使其在整个会话期间最大化期望累积奖励：

$$\mathbb{E}_{\tau\sim\pi}[\mathcal{R}(\tau)] = \mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{|\tau|}\gamma^t r(s_t, a_t)\Big] \tag{1}$$

其中，$\tau = [(s_0, a_0, r_0), (s_1, a_1, r_1) \ldots]$ 表示采样得到的轨迹，$\gamma \in [0, 1)$ 表示未来奖励的折扣因子。

2.2 整体框架

我们在图2中展示了所提出的框架，并将其称为超动作评论家（HAC）学习方法。推荐策略 $\pi(a_t|s_t)$ 被分解为一个超演员网络 $P(Z_t|s_t)$，用于生成向量化超动作；以及一个排序评分器$P(a_t|Z_t)$，根据超动作选择最终的推荐列表。随后，我们提出在不同动作空间之间共享评论家网络，使其能够评估超动作或最终的效果动作（即推荐列表）。我们的框架以 DDPG [30] 为基础，但与之不同的是，我们强调了在演员学习和评论家学习中使用不同动作空间的重要性。具体而言，我们基于效果动作优化评论家，以确保动作/状态评估的准确性；并使用超动作优化演员，从而实现高效的端到端训练与探索。为确保两个不同动作空间之间的一致性，我们还利用物品核函数学习一个逆池化模块，以便从效果动作反推出超动作。这意味着，两个动作空间的评估将共用同一个评论家，并且从效果动作中学到的知识可以迁移到超动作。为稳定学习过程，我们还利用用户即时反馈对效果动作进行监督。

2.3 用户状态与超动作器

在推荐系统中，用户可观测的信息通常包括最新的用户交互历史$\boldsymbol{x}_{1:\ t}$ 以及静态的人口统计特征 $\boldsymbol{u}$。随后，可以使用任意序列模型对这些信息进行编码，并推断动态用户状态：

$$\boldsymbol{s}_t = \text{StateEnc}(\Phi(x_1), \ldots, \Phi(x_t), \boldsymbol{u}) \tag{2}$$

其中，物品核函数 Φ 将物品的原始特征映射到核空间中的稠密嵌入。我们还使用用户核函数将用户特征 $\boldsymbol{u}$ 映射到与物品相同的核空间，然后将历史记录中物品序列与用户嵌入拼接，作为新的序列。

为编码用户特征和历史记录，有几种可行的序列模型，例如 [23, 25, 40], ；我们使用 SASRec [25] 中的状态编码器作为主干网络，因为它能够通过自注意力机制捕获动态历史记录中的序列模式以及用户-物品相关性。状态编码完成后，通过一个多层感知机（MLP）模块推断出向量化表示（即超动作）：

$$Z_t = \text{MLP}(\boldsymbol{s}_t) \tag{3}$$

我们假设该超动作服从标准高斯分布 $\text{N}(Z_t, \sigma_Z^2)$，并且可以使用重参数化技巧进行端到端训练。

2.4 评分函数与效果动作

给定包含充分用户偏好的信息的 Z_t，我们可以假设物品的选择仅取决于该潜向量。换言之，我们具有条件独立性：

$$P(a_t|\boldsymbol{s}_t, Z_t) = P(a_t|Z_t) \tag{4}$$

请注意，这一设定还表明，推断出的 Z_t 是一个超动作，可以视为后续物品选择模块的参数。因此，整体推荐策略遵循 $\pi(a_t|s_t) \sim P(a_t|Z_t)$。

生成效果动作时，我们根据由 Z_t 参数化的评分函数，从候选池中选择物品。该评分函数为 I 中的每个候选物品提供一个排序分数，最终通过 top-k 选择或分类采样生成推荐列表。以 SASRec 模型为例，评分函数是核物品嵌入与编码后的用户状态之间的点积：

$$\text{score}(i|Z_t) = \Phi(i)^\top Z_t \tag{5}$$

请注意，物品核函数 Φ 的参数不视为超动作 Z_t 的一部分，因为它独立于给定的用户状态。为了实现高效学习和推理，可以假设每个物品的选择在条件上彼此独立：$P(a_t|Z_t) = \prod_{i \in a_t} P(i|Z_t)$，这与 [21] 中的列表分解类似。然后，我们可以将物品的选择或采样概率定义为 $P(i|Z_t) = \text{softmax}_I(\text{score}(i|Z_t))$。

2.5 共享评论器与逆模块

评论家的目的是准确评估状态-动作对的长期质量（例如 DDPG 中的 Q函数），或给定状态的预期价值（例如 A2C 中的 V函数），从而有效指导演员学习和动作探索。与标准强化学习框架相比，新的问题设置使我们能够评估带有 $Q(s_t, Z_t)$ 的超动作、带有 $Q(s_t, a_t)$ 的效果动作，或两者兼而有之。为了确保对来自不同空间的动作进行一致评估，我们提出通过共享评论器网络，在 $Q(s_t, a_t)$ 与 $Q(s_t, Z_t)$ 之间传递知识。如图2所示，这个共享评论器是一个映射函数 $g: \text{S} \times \text{Z} \to \mathbb{R}$，它接收用户状态 s_t 和核空间中的动作嵌入 Z_t。或者等价地，$Q(s_t, Z_t) = g(s_t, Z_t)$。为评估效果动作，我们引入了一个逆模块 h，以反向推断超动作：

$$\hat{Z}_t = h(a_t) = \text{pooling}(\Phi(i)|i \in a_t) \tag{6}$$

评估结果变为 $Q(s_t, a_t) = g(s_t, \hat{Z}_t)$。在实践中，我们发现，对核空间中的物品嵌入进行平均池化可以得到最稳定的结果，尽管有无穷多个潜在的 Z 都能生成相同的列表。与使用相邻状态推断潜在动作的现有方法（如 PG- RA [4], ）相比，我们认为，推荐任务中的效果动作包含足够的信息，可用于恢复超动作。此外，我们使用重构损失，通过对齐损失函数进一步约束超动作与效果动作之间的一致性。正如下一节将要介绍的，这可以确保生成的超动作 Z_t 位于核空间中靠近候选物品的有效区域内。

2.6 总体学习框架

整体优化过程是一种经过修改的演员-评论家学习框架，由评论家损失、演员损失、超演员损失和监督损失组成。经验回放缓冲区 D 将收集以下形式的样本记录：$(s_t, a_t, r(s_t, a_t)\ s_{t+1}, d)$，其中 d 是一个终止信号，用于指示用户是否已

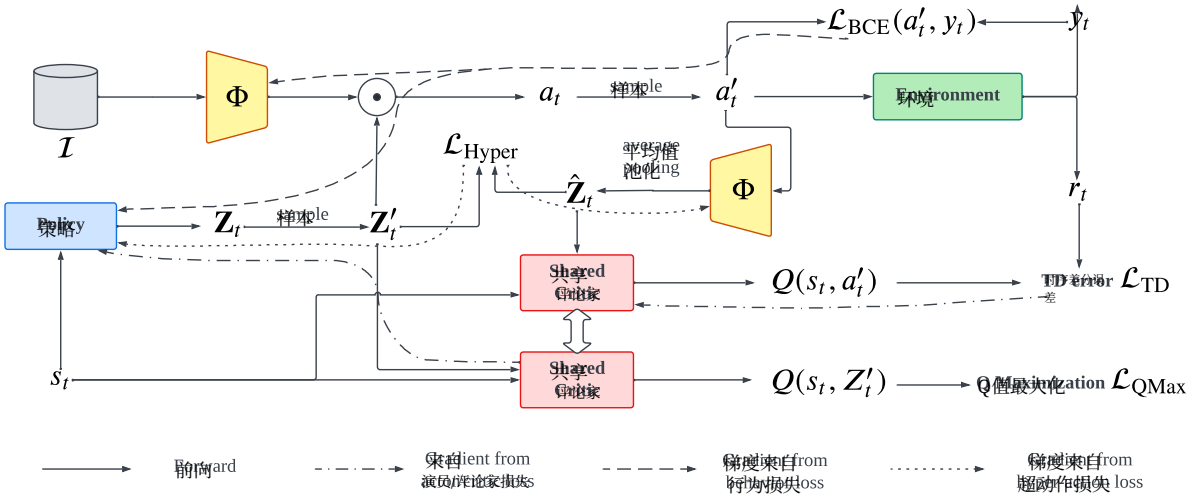

图2：超演员-评论家（ HAC）学习框架。⊙表示从中选择物品的评分函数，$\mathcal{I}$，

离开。评论家损失旨在训练一个能够捕捉动作质量模式的准确评估器：

$$\mathcal{L}_{\text{TD}} = \mathbb{E}_{\mathcal{D}}\Big[(r(s_t, a_t) + \gamma(1-d)Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2\Big] \tag{7}$$

其中，a_{t+1} 由推荐策略使用贪心方法生成（当Q函数准确时，等价于求 $Q(s_t, a_t)$的arg max）。这是标准的时序差分误差；在学习评论家时，我们仅计算效果动作的Q值，以确保评估的准确性。请注意，在基于DDPG的方法中，每个演员和评论家都与一个目标网络配对，用于估计未来的状态-动作对 $Q(s_{t+1}, a_{t+1})$。这些目标网络从演员和评论家处采用弹性更新，使其缓慢的优化过程能够稳定学习。

随后，以准确的评论家作为评估器，我们可以通过最大化推断动作的Q值来高效地学习演员，这等价于最小化以下演员损失：

$$\mathcal{L}_{\text{QMax}} = \mathbb{E}_{\mathcal{D}}\Big[Q(s_t, Z_t)\Big] \tag{8}$$

其中，Z_t 由第2.3节所述的超动作器推断得到。请注意，评论家和演员的学习使用不同的动作空间，因此我们需要对齐这两个空间，以避免生成的超动作 [39] 发生模式坍塌。在我们的方法中，使用L2正则化器来确保这种一致性：

$$\mathcal{L}_{\text{Hyper}} = \mathbb{E}_{D}\Big[\|Z_t - \hat{Z}_t\|^2\Big] \tag{9}$$

其中，Z_t 是根据状态 s_t生成的超动作，而 $\hat{Z}_t$ 则是先按照第2.4 节所述，使用 Z_t 贪婪地选择效果动作，再按照第 2.5 节所述，使用逆模块重构超动作而生成的。

此外，为了更好地稳定强化学习，并利用每个物品上的详细用户响应信号，我们还基于效果动作加入了一个监督学习目标。

$$\mathcal{L}_{\text{BCE}} = \mathbb{E}\Big[\sum_{i \in a_t} y_{t,i} \log P(i|Z_t) + (1 - y_{t,i}) \log(1 - P(i|Z_t))\Big] \tag{10}$$

算法 1 超演员评论家训练

1: **过程**HAC 2: 初始化行动者、评论家和物品核函数中的所有可训练参数。 3: 初始化经验回放缓冲区 B。 4: **当**尚未收敛时，在每次迭代中**执行**5: 在运行回合中应用当前策略，收集样本并将其存储到 B 中。 6: 从 B 中抽取小批量样本 $(s_t, a_t, r(s_t, a_t)\ s_{t+1}, d) \sim$。 7: 使用公式（8）和公式（7）的损失更新演员和评论家。 8: 如果存在动作对齐，则使用公式（9）的损失更新演员和核函数。 9: 如果存在监督，则使用公式（10）的损失更新演员和核函数。

10: **结束 while 循环**

这是一种二元交叉熵损失，其中$y_{t,\ i}$是曝光物品 i对应的真实用户响应。我们提醒读者，还有其他先进的监督和正则化方法可用于适配基于强化学习的模型，并有望扩展我们的框架。例如， [14] 可以为效果-动作空间提供监督，以进行离策略训练；而 [49] 则非常适用于超动作空间的距离控制。总之，我们将由此得到的学习范式概述为算法 1。请注意，逆模块的参数来自物品核，因此公式（9）和公式（10）都按照第 8–9 行的方式更新这些参数。

2.7 超动作空间与效果-动作空间中的探索

读者可能会注意到，如图2所示，引入潜在超动作也增加了一步额外的动作采样，因此，该框架同时支持在超动作空间中进行采样（例如添加高斯噪声），以及在效果-动作空间中进行采样（例如根据排序分数对物品进行分类采样

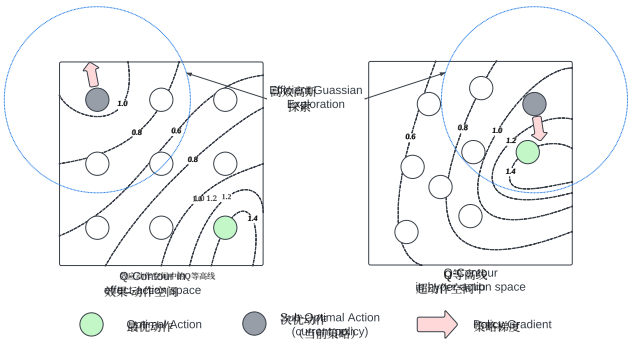

图3：不同动作空间中的探索

）。从理论上讲，这意味着效果动作的采样概率应描述如下：

$$P(a_t|s_t) = \int_{Z_t} P(a_t|Z_t)P(Z_t|s_t) \tag{11}$$

当效果动作生成是确定性的，探索仅取决于 Z_t的采样，与 [4] 中的情况类似；而如果超动作器是确定性的，探索则仅取决于效果动作采样，如标准策略梯度方法中的情况。请注意，超动作的方差 σ_Z^2 控制着推断出的潜在动作嵌入的不确定性，找到合适的值对于提升强化学习中的探索有效性至关重要。一方面，方差过小会限制对新动作的探索，导致结果次优；另一方面，方差过大则会引发不稳定的动作探索，使其难以收敛。

一般而言，我们希望利用超动作空间高效的学习与探索，因此，如第2.6节所述，使 Z_t 的分布与核空间中的嵌入物品对齐至关重要。如图3所示，物品核函数通过折叠动作空间来提升策略的表达能力，从而使探索更加高效。尽管我们不确定是否存在一种适用于所有探索潜在动作空间的强化学习方案的保证，但我们将在第3.4节通过实证展示：将用户和物品编码到同一个核空间，并使用公式（9）通过逆池化模块调节动作的有效性。

3 实验

3.1 实验设置

3.1.1 数据集：我们的实验使用了三个公共数据集：**RL4RS**[1] 是一个基于会话的数据集，[44] 最初于 BigData Cup 2021 年推出，旨在推动强化学习研究；**ML1M**[2] 是 MovieLens 数据，[20] 包含 100 万条记录，记录了用户对电影的评分；**KuaiRand1K**[3] 是一个用于序列短视频推荐的新近数据集，我们使用的是 1K 版本，[16]其中已移除无关视频。我们对所有数据集进行预处理，将其统一为如下格式：每条记录均由（用户特征、用户

[1]https://github.com/fuxiAIlab/RL4RS [2] https://grouplens.org/datasets/movielens/1m/ [3] https://kuairand.com/

数据集	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	#记录	k
RL4RS	-	283	781,367	9
MovieLens-1M	6400	3706	1,000,208	10
KuaiRand	986	11,643	969,831	10

表1：数据集摘要。 k 表示推荐列表的大小（即效果动作大小）。RL4RS数据集提供用户画像特征而非用户ID，因此没有用户集合数量。

历史记录、曝光物品、用户反馈和时间戳）构成，并按顺序排列。该过程的详细信息总结于附录A.1，所得数据集统计信息见表1。

3.1.2 在线环境模拟器。为了模拟在线用户交互，我们针对每个数据集训练一个用户响应模型 $\Psi : S \times A \to \mathbb{R}^k$。用户状态基于静态用户特征和动态历史交互得出。$\Psi$ 输出用户对推荐 a_t中每个物品给出正向反馈的概率，最终响应 $y_t \in \{0, 1\}^k$（例如点击）根据这些概率进行均匀采样。我们将奖励 $r(s_t, a_t) \in [-0.2, 1.0]$ 设计为逐物品奖励的平均值。我们在附录A.2中提供了环境的详细信息。

3.1.3 模型与基线方法。 我们采用 SASRec [25] 作为骨干策略，如第 2.3 节所述。它由基于 Transformer 的状态编码器、超动作生成器和点积评分器组成。我们还基于所提出的 HAC 框架实现了以下强化学习基线，以更好地展示 HAC 如何推广现有方法：

- **在线监督学习**：SASRec 策略直接从即时用户反馈中学习，而非从长期累积奖励中学习。
- **A2C**：A3C [34] 的同步版本，在效果-动作空间上应用策略梯度。
- **DDPG**：一种深度 DPG 框架，演员和评论家 [30]均使用超动作空间。该方法等同于不使用监督的 HAC 模型。
- **TD3**：通过双重 Q 学习改进 DDPG，使评论家的训练更加稳定 [15]。
- **DDPG-RA**：采用 [4]中的动作表示学习的 DDPG 框架。该方法与我们的工作最为接近，它对效果动作进行正则化，而我们的 HAC 模型则对齐超动作。

为了更好地比较强化学习和监督学习方法，我们还纳入了**离线监督学习**，它基于离线数据而非在线环境，使用公式（10）优化策略。这些模型的架构和具体说明见附录 A.3。

3.1.4 评估。 对于所有数据集，我们根据记录时间戳，将前 80% 划分为训练集，后 20% 划分为评估集。随后，我们在训练集上预训练在线环境，并在整个数据集上预训练另一个在线环境，以供后续评估。我们在第一个环境中训练推荐策略，并在第二个环境中评估这些策略。训练期间，所有实验的奖励折扣设为 $\gamma = 0.9$，交互深度限制为 ≤ 20。我们发现，大多数基于强化学习的方法在 50,000 次迭代内收敛并趋于稳定。对于长期评估指标，我们考虑**总奖励**

模型	RL4RS 总奖励	RL4RS 交互深度	ML1M 总奖励	ML1M 交互深度	KuaiRand 总奖励	KuaiRand 交互深度
离线监督学习	6.721	8.163	18.559	18.717	<u>14.394</u>	<u>14.982</u>
在线监督学习	<u>9.502</u>	<u>10.571</u>	<u>18.629</u>	<u>18.780</u>	13.456	14.147
A2C	7.789	9.140	16.158	16.556	12.460	13.250
DDPG	8.337	9.588	17.205	17.508	11.394	12.313
TD3	8.553	9.791	17.545	17.814	11.777	12.664
PG-RA	8.561	9.728	18.466	18.633	10.859	11.814
HAC	**10.059**	**11.102**	**18.863**	**18.988**	**14.789**	**15.335**

表2：在线性能。最佳结果以粗体标出，次佳结果以下划线标出

它表示整个用户会话中的奖励总和；**交互深度** 则表示用户与模拟在线环境进行交互的次数，该环境会与学习得到的策略进行交互，从而观测每个会话。对于这两项指标，值越高表示性能越好。为评估学习得到的策略的稳定性，我们纳入**奖励方差**这一指标，用于估计策略面对不同用户状态时处理方式的不一致程度，因此值越低表示策略越稳定。请注意，该指标描述的是跨状态的方差，而非跨随机种子的方差。在每项实验中，我们评估上述所有指标，并报告不同用户会话上的平均值。

3.2 有效性

对于所有任务，推荐系统的目标都是最大化由总奖励和平均交互深度体现的长期满意度。对于每个模型，我们通过网格搜索调整超参数，并选取结果最佳的设置，在表2中报告。

主要结果： 我们可以看到，所提出的HAC框架在所有数据集上均持续取得最佳表现，在两项长期指标上均优于最佳基线方法：在RL4RS上提升6%，在ML1M上提升1%，在KuaiRand上提升3%。这表明所提出的超动作具有较强的表达能力，且该学习方法有效。值得注意的是，其他强化学习方法仅在RL4RS任务上优于离线监督学习；而在ML1M和KuaiRand中，随着动作空间增大，它们的表现反而更差。这表明，我们的HAC框架能够更好地捕捉较大动作空间中的模式。

强化学习基线方法： 在强化学习方案中，A2C 的性能始终最差，且似乎是最不稳定的学习框架；但在动作空间较大的数据集（ML1M 和 KuaiRand）中，A2C 与 DDPG 之间的差距往往较小，而 A2C 在动作空间最大的 KuaiRand 中，A2C 的性能甚至优于 DDPG。由于 A2C 直接优化效果动作，而 DDPG 使用超动作，这一差距的缩小可能表明，在更大的效果-动作空间中，学习一致且准确的超动作表示可能会变得更加困难。这也可能说明，确保两个动作空间之间的一致性对于实现有效的强化学习至关重要。TD3 的性能略优于 DDPG，但整体表现仍然相似。

动作空间正则化： 除了我们的方法之外，DDPG-RA 方法也解决了动作空间的对齐问题，其行为与我们的方法最为接近。不同的是，它不像 HAC 那样对超动作进行正则化，而是对效果动作空间进行对齐，而该空间并未直接用于引导演员。此外，DDPG-RA 在学习评论家时使用超动作而非效果动作，因此其效果优于 A2C 和 DDPG，但不及我们的方法，甚至不及监督方法。

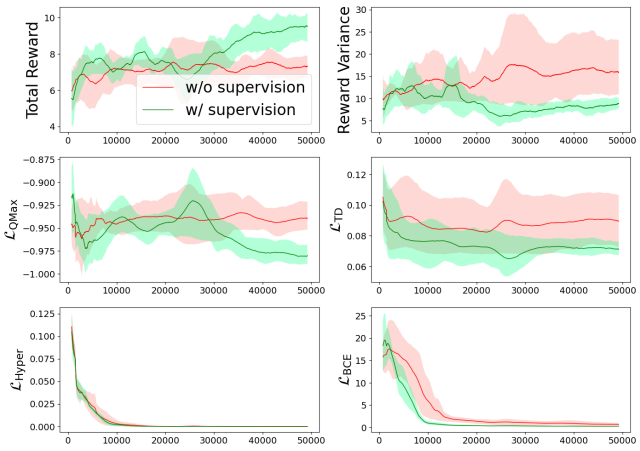

图 4：RL4RS 上无监督 HAC 的训练曲线。横轴表示迭代次数。图中以对数刻度展示了四种损失。

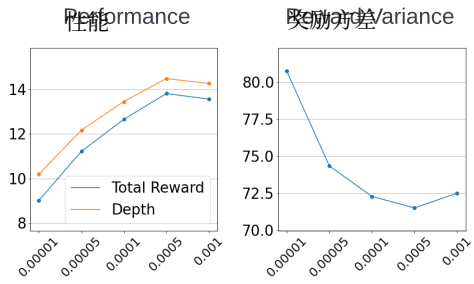

图 5：监督损失对 KuaiRand 的影响。横轴表示公式(10)中的学习率。

3.3 HAC学习

监督与稳定学习： 为进一步说明 HAC 模型的训练行为，我们在图4中绘制了 HAC 的学习曲线，并与"无监督的 HAC"进行比较；后者没有公式（10）中监督损失的梯度。所有方法在性能指标上都趋于饱和

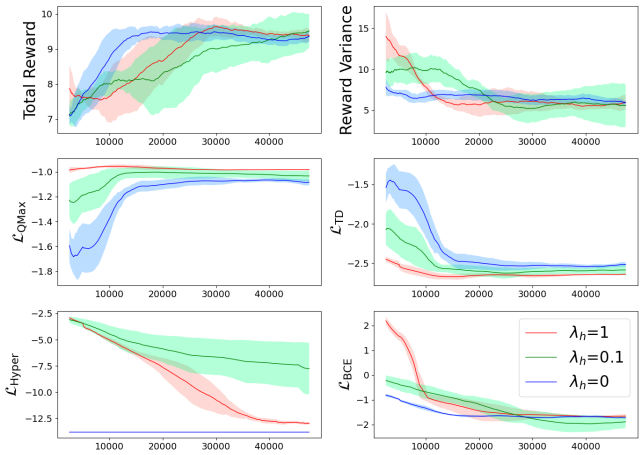

图 6：HAC 在 RL4RS 上的训练曲线。奖励方差对应于不同用户的总奖励方差。四种损失以对数尺度呈现。 λ_h表示超动作对齐的幅度。 $\lambda_h = 0$不包含此对齐损失，因此具有$\mathcal{L}_{超} = 0$。

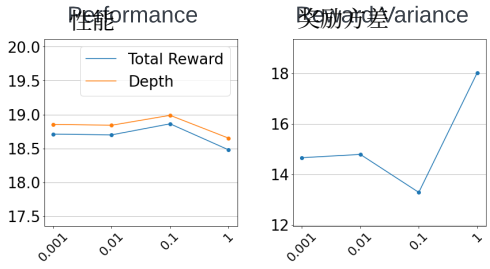

图 7：超动作对齐对 ML1M 的影响。X 轴表示公式(9)的幅度，即图 6 中的 λ_h。

（如面板“总奖励”所示），并且都能成功降低第 2.6 节提到的四种损失函数。值得注意的是，“无监督的 HAC”仍能成功降低每个物品上的二元交叉熵损失，这表明基于汇总逐项信号的奖励进行强化学习是有效的。我们还可以看到，加入监督有助于提升模型性能，并降低演员损失 L_{QMax}，从而帮助探索更好的动作。值得注意的是，与“无监督的 HAC”相比，HAC 的评论家损失 L_{TD}更低，这表明其学习过程更稳定。我们还可以通过观察用户之间总奖励的方差来验证这一点，因为奖励方差越高，说明学习到的策略越难以针对不同用户状态提供良好的动作。如图5所示，在 KuaiRand 环境中，提高监督损失的重要性有助于改善推荐性能并降低方差。然而，如果为监督模块设置的学习率过高（KuaiRand 中为 0.001），则可能过度利用用户意图并损害性能。我们在另外两个数据集上也观察到类似的模式，结果见附录B。

超动作调节： 在比较不同超动作幅度的对齐效果时，我们进行了相同的消融实验

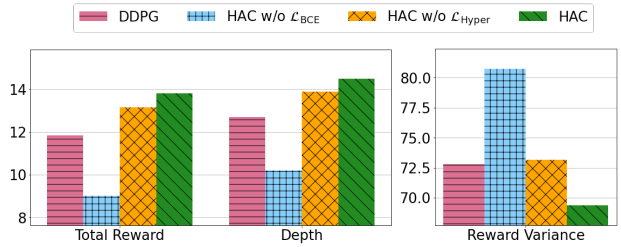

图8：模型组件对 ML1M 的影响

，针对损失公式（9），并在图 6 中绘制了 HAC 的学习曲线。总体而言，与仅使用监督（$\lambda_h = 0$）的 HAC 相比，加入超动作对齐（$\lambda_h = 0.1$和 $\lambda_h = 1$）会使 HAC 框架在总奖励和奖励方差的收敛方面学习得更慢。相反，更一致的动作空间有助于模型更好地学习并探索动作策略。此外，提高该对齐模块的重要性会导致 L_{QMax} 变差、L_{TD}变好，这表明评论家在捕捉动作质量方面更加准确。请注意，$\lambda_h = 1$ 比 $\lambda_h = 0.1$ 更稳定，但在探索方面可能效果较差。为进一步验证这一点，我们在图 7 中展示了评估结果，结果显示存在一个最佳点 $\lambda_h = 0.1$，在该点上，推荐的奖励更高且更稳定。

3.4 消融实验

模型组件： 为了更好地研究框架中不同学习模块的作用，我们比较了本方法的几种变体：1）DDPG：不包含监督和动作对齐的 HAC，且演员学习和评论家学习均使用超动作空间；2）不含 $L_{二元交叉熵}$ 的 HAC：去除 HAC 的监督；3）不含 $L_{超动作}$ 的 HAC：去除超动作对齐模块。我们在图8中总结了ML1M数据集上的结果。可以看到，去除监督或对齐模块都会降低性能并增大奖励方差。这表明这两个模块都有助于提升模型的准确性和学习稳定性。其他数据集上也观察到类似结果，详见附录B。值得注意的是，DDPG 的性能与不含 $L_{二元交叉熵}$ 的 HAC 基本相同，这表明像 HAC 那样为演员学习和评论家学习采用彼此分离的动作空间可能会降低性能，而仅仅加入动作对齐模块并不能弥补这一差距。从这个意义上说，HAC 同时需要超动作对齐和监督。这也意味着，两个动作空间之间的不一致性小于聚合奖励与逐项用户响应信号之间的不一致性。

在不同动作空间上的探索： 对于 HAC 模型的探索，我们可以在超动作空间和效果-动作空间上进行探索。为了比较不同幅度的超动作探索所产生的影响，我们在学习过程中改变高斯噪声的方差，同时固定 HAC 的所有其他超参数，并在图9中展示结果。对推荐性能的比较表明，搜索空间中间存在一个最优点，这说明应谨慎设计探索，使采样方差既不能太小，也不能太大。正如我们在第2.7节所讨论的，较小的

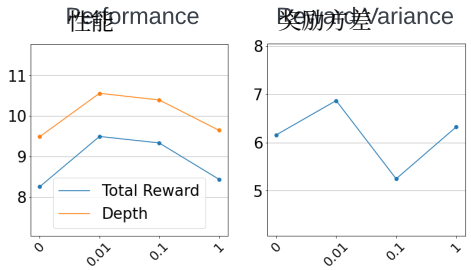

图9：HAC 在 RL4RS 数据集上超动作探索幅度的影响。X 轴表示 HAC 模型中超动作的高斯噪声方差。

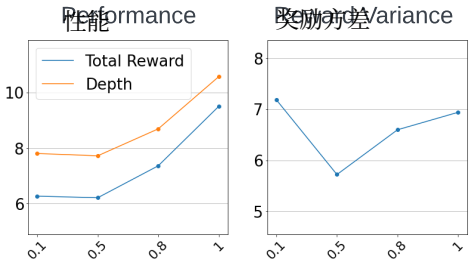

图10：HAC 在 RL4RS 数据集上效果动作探索幅度的影响。X 轴表示 HAC 模型中效果动作采用贪心 top-k 选择而非分类采样的比例，并且我们将超动作噪声设为 0。

方差可能会限制对新动作的探索，而较大的方差可能导致不稳定、难以收敛的探索。根据经验，我们发现，在效果动作上进行采样不如在超动作上进行探索有效。如图10所示，采用 top-k 贪心选择可取得最佳结果，而加入分类采样则会使学习到的策略变得次优。

4 相关工作

4.1 序列推荐与基于会话的推荐

基于会话的推荐（SBR）问题与序列推荐（SR）任务 [45]密切相关。SR 任务旨在学习一种策略，根据给定用户的历史交互推断下一次推荐（物品或列表）。相比之下，SBR 考虑了交互会话的开始与结束。我们认为，我们的设定介于这两种概念之间：将列表大小设为 1 后，它几乎与 SR 完全相同，区别仅在于奖励设定；然而，我们的目标是优化用户会话的全部未来奖励，这更接近 [45] 中定义的下一部分 SBR。在这两种问题设定下，最常用的解决方案都是采用马尔可夫链假设来建模用户—系统交互的动态转移。这类研究面临的主要挑战是，如何基于长期历史构建具有代表性的用户状态。早期的推荐问题解决方案采用了协同过滤技术 [12, 27, 29, 36] ，后续方法则采用了深度神经网络，例如循环网络 [23]、卷积网络 [43]，记忆网络 [10]，自注意力 [25, 40]，图卷积网络 [5, 46]，机器推理 [7, 24, 38] 和基础模型 [19] ，以提升模型的表达能力，从而更好地捕捉用户/物品特征和交互历史中丰富而复杂的信息。这些方法背后的核心思想是准确编码长程历史，但这种范式并未优化用户长期奖励。

4.2 推荐中的强化学习

基于强化学习的推荐系统 [1, 37, 41] 也遵循马尔可夫决策过程的形式化定义，并强调优化累积奖励的重要性，因为累积奖励代表长期用户满意度。在最简单的设置中，使用基于表格的方法 [33] 优化评估表，但这类方法仅适用于固定的一组状态-动作对。随后，人们提出了基于价值的方法 [35, 42, 51, 56] 和策略梯度方法 [6, 8, 17, 18, 28, 47] ，根据采样得到的长期奖励，学习评估和优化动作策略。演员-评论家范式 [48, 53, 54] 通过同时学习动作评估器和动作生成器，整合了这两种方法。基于强化学习的推荐系统面临的主要挑战包括庞大的组合状态/动作空间 [13, 21, 31]、不稳定的学习行为 [2, 9]，以及为异质用户行为寻找最优奖励函数 [3, 11]。我们的工作聚焦于动作空间表示学习和强化学习的稳定性。我们认为 PG-RA [4] 是最接近的相关工作，它同样强调潜在动作表示的学习。正如我们在第 3.1 节所述，PG-RA 旨在通过共享评分函数迁移知识，并在效果动作空间上应用动作对齐，但这并不适合用户和物品的潜因子分解。我们已在第 3.2 节通过实验验证了其性能较差。此外，我们通过在不同模拟环境中的评估展示了我们方法的有效性，但也提醒读者，实际的在线环境仍有可能具有更复杂、动态的机制。从这个意义上说，一些研究致力于为推荐系统构建更真实的在线用户环境 [22, 52, 55] ，这在实践中可以补充我们的工作。

5 结论

我们提出了一种实用的演员-评论家学习框架，可在离散效果-动作空间和向量化超动作空间中优化推荐策略。在基于公共数据集构建的在线环境中进行的实证结果验证了所提方法的有效性和稳定性。我们的设计主要着眼于泛化能力，能够兼容包括演员-评论家、DDPG 以及监督学习在内的各种现有解决方案框架。此外，我们将超动作的概念视为一种初步尝试，旨在应用比简单点积更复杂的评分函数。

致谢

本研究部分由 APRC—CityU 新研究计划（编号：9610565，香港城市大学新教师启动基金）、SIRG—CityU 战略跨学科研究基金（编号：7020046、7020074）以及 CityU—HKIDS 青年学者研究基金（编号：9360163）资助。

参考文献

[1] M Mehdi Afsar、Trafford Crump 和 Behrouz Far。2021年。基于强化学习的推荐系统：综述。ACM计算调查（CSUR）（2021年）。[2] Xueying Bai、Jian Guan 和 Hongning Wang。2019年。采用对抗训练的在线推荐模型化强化学习。神经信息处理系统进展 32（2019年）。[3] Qingpeng Cai、Ruohan Zhan、Chi Zhang、Jie Zheng、Guangwei Ding、Pinghua Gong、Dong Zheng 和 Peng Jiang。2022年。短视频推荐中的约束强化学习。arXiv 预印本 arXiv:2205.13248（2022年）。[4] Yash Chandak、Georgios Theocharous、James Kostas、Scott Jordan 和 Philip Thomas。2019年。学习强化学习的动作表示。载于《国际机器学习会议》。PMLR，941–950。[5] Jianxin Chang、Chen Gao、Yu Zheng、Yiqun Hui、Yanan Niu、Yang Song、Depeng Jin 和 Yong Li。2021年。基于图神经网络的序列推荐。载于《第44届国际ACM SIGIR信息检索研究与发展会议论文集》。378–387。[6] Haokun Chen、Xinyi Dai、Han Cai、Weinan Zhang、Xuejian Wang、Ruiming Tang、Yuzhou Zhang 和 Yong Yu。2019年。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。[7] Hanxiong Chen、Shaoyun Shi、Yunqi Li 和 Yongfeng Zhang。2021年。神经协同推理。载于《2021年网络会议论文集》。1516–1527。[8] Minmin Chen、Alex Beutel、Paul Covington、Sagar Jain、Francois Belletti 和 Ed H Chi。2019年。REINFORCE 推荐系统的 Top-k 离策略校正。载于《第十二届 ACM 网络搜索与数据挖掘国际会议论文集》。456–464。[9] Minmin Chen、Bo Chang、Can Xu 和 Ed H Chi。2021年。用于改进 REINFORCE 推荐系统的用户响应模型。载于《第14届 ACM 网络搜索与数据挖掘国际会议论文集》。121–129。[10] Xu Chen、Hongteng Xu、Yongfeng Zhang、Jiaxi Tang、Yixin Cao、Zheng Qin 和 Hongyuan Zha。2018年。基于用户记忆网络的序列推荐。载于《第十一届 ACM 网络搜索与数据挖掘国际会议论文集》。108–116。[11] Xiaocong Chen、Lina Yao、Aixin Sun、Xianzhi Wang、Xi wei Xu 和 Liming Zhu。2021年。用于在线推荐的生成式逆向深度强化学习。载于《第30届 ACM 信息与知识管理国际会议论文集》。201–210。[12] Heng-Tze Cheng、Levent Koc、Jeremiah Harmsen、Tal Shaked、Tushar Chandra、Hrishi Aradhye、Glen Anderson、Greg Corrado、Wei Chai、Mustafa Ispir 等。2016年。用于推荐系统的宽深度学习。载于《推荐系统深度学习研讨会论文集》。7–10。[13] Gabriel Dulac-Arnold、Richard Evans、Hado van Hasselt、Peter Sunehag、Timothy Lillicrap、Jonathan Hunt、Timothy Mann、Theophane Weber、Thomas Degris 和 Ben Coppin。2015年。大规模离散动作空间中的深度强化学习。arXiv 预印本 arXiv:1512.07679（2015年）。[14] Scott Fujimoto 和 Shixiang Shane Gu。2021年。离线强化学习的极简方法。神经信息处理系统进展 34（2021年），20132–20145。[15] Scott Fujimoto、Herke Hoof 和 David Meger。2018年。解决演员-评论家方法中的函数逼近误差。载于《国际机器学习会议》。PMLR，1587–1596。[16] Chongming Gao、Shijun Li、Yuan Zhang、Jiawei Chen、Biao Li、Wenqiang Lei、Peng Jiang 和 Xiangnan He。2022年。KuaiRand：一个包含随机曝光视频的无偏序列推荐数据集。载于《第31届 ACM 信息与知识管理国际会议论文集》（美国佐治亚州亚特兰大）（CIKM '22）。5页。https://doi.org/10.1145/3511808.3557624[17] Yingqiang Ge、Shuchang Liu、Ruoyuan Gao、Yikun Xian、Yunqi Li、Xiangyu Zhao、Changhua Pei、Fei Sun、Junfeng Ge、Wenwu Ou 和 Yongfeng Zhang。2021年。迈向推荐中的长期公平性。载于《第14届 ACM 网络搜索与数据挖掘国际会议论文集》。445–453。[18] Yingqiang Ge、Xiaoting Zhao、Lucia Yu、Saurabh Paul、Diane Hu、Chu-Cheng Hsieh 和 Yongfeng Zhang。2022年。通过强化学习实现推荐中帕累托效率的公平性—效用权衡。载于《第十五届 ACM 网络搜索与数据挖掘国际会议论文集》。316–324。[19] Shijie Geng、Shuchang Liu、Zuohui Fu、Yingqiang Ge 和 Yongfeng Zhang。2022年。推荐即语言处理（RLP）：统一的预训练、个性化提示与预测范式（P5）。载于《第16届 ACM推荐系统会议论文集》。299–315。[20] F Maxwell Harper。2015年。MovieLens数据集：历史与背景。ACM交互式智能系统汇刊（TIIS）5 4（2015年）1–19。F Maxwell Harper 和 Joseph A Konstan。2015年。MovieLens数据集：历史与背景。ACM交互式智能系统汇刊（TIIS）5 4（2015年）1–19。[21] Eugene Ie、Vihan Jain、Jing Wang、Sanmit Narvekar、Ritesh Agarwal、Rui Wu、Heng-Tze Cheng、Tushar Chandra 和 Craig Boutilier。2019年。SlateQ：用于推荐集合强化学习的易处理分解。（2019年）。[22] Eugene Ie、Chih wei Hsu、Martin Mladenov、Vihan Jain、Sanmit Narvekar、Jing Wang、Rui Wu 和 Craig Boutilier。2019年。RecSim：面向推荐系统的可配置仿真平台。（2019年）。arXiv:1909。04847 [cs.LG]

[23] Dietmar Jannach 和 Malte Ludewig。2017。循环神经网络与邻域相遇：用于基于会话的推荐。载于第11届 ACM 推荐系统会议论文集。306–310。[24] Jianchao Ji、Zelong Li、Shuyuan Xu、Max Xiong、Juntao Tan、Yingqiang Ge、Hao Wang 和 Yongfeng Zhang。2023。反事实协同推理。载于第16届 ACM 网络搜索与数据挖掘国际会议论文集。[25] Wang-Cheng Kang 和 Julian McAuley。2018。自注意力序列推荐。载于2018年 IEEE 国际数据挖掘会议（ICDM）。IEEE，197–206。[26] Yehuda Koren、Robert Bell 和 Chris Volinsky。2009。推荐系统的矩阵分解技术。Computer 42，8（2009），30–37。[27] Yehuda Koren、Steffen Rendle 和 Robert Bell。2022。协同过滤的进展。《推荐系统手册》（2022），91–142。[28] Zelong Li、Jianchao Ji、Yingqiang Ge 和 Yongfeng Zhang。2022。AutoLossGen：推荐系统的自动损失函数生成。载于第45届国际ACM SIGIR信息检索研究与发展会议论文集。1304–1315。[29] Dawen Liang、Rahul G Krishnan、Matthew D Hoffman 和 Tony Jebara。2018。用于协同过滤的变分自编码器。载于2018年万维网大会论文集。689–698。[30] Timothy P. Lillicrap、Jonathan J. Hunt、Alexander Pritzel、Nicolas Heess、Tom Erez、Yuval Tassa、David Silver 和 Daan Wierstra。2016。深度强化学习与连续控制。。载于 ICLR（海报）。http://arxiv.org/abs/1509.02971 [31] Feng Liu、Ruiming Tang、Xutao Li、Weinan Zhang、Yunming Ye、Haokun Chen、Huifeng Guo、Yuzhou Zhang 和 Xiuqiang He。2020。基于深度强化学习的推荐中的状态表示建模。Knowledge-Based Systems 205（2020），106170。[32] Tie-Yan Liu 等。2009。信息检索中的学习排序。Foundations and Trends in Information Retrieval 3，3（2009），225–331。[33] Tariq Mahmood 和 Francesco Ricci。2007。交互式推荐系统中的学习与自适应性。载于第九届国际电子商务会议论文集。75–84。[34] Volodymyr Mnih、Adria Puigdomenech Badia、Mehdi Mirza、Alex Graves、Timothy Lillicrap、Tim Harley、David Silver 和 Koray Kavukcuoglu。2016。深度强化学习的异步方法。载于国际机器学习会议。PMLR，1928–1937。[35] Changhua Pei、Xinru Yang、Qing Cui、Xiao Lin、Fei Sun、Peng Jiang、Wenwu Ou 和 Yongfeng Zhang。2019。基于强化学习利润最大化的价值感知推荐。载于万维网大会论文集。3123–3129。[36] Steffen Rendle、Christoph Freudenthaler 和 Lars Schmidt-Thieme。2010。为下一篮子推荐构建个性化马尔可夫链的因子分解。载于第19届万维网国际会议论文集。811–820。[37] Guy Shani、David Heckerman、Ronen I Brafman 和 Craig Boutilier。2005。一种基于马尔可夫决策过程的推荐系统。Journal of Machine Learning Research 6，9（2005）。[38] Shaoyun Shi、Hanxiong Chen、Weizhi Ma、Jiaxin Mao、Min Zhang 和 Yongfeng Zhang。2020。神经逻辑推理。载于第29届 ACM 国际信息与知识管理会议论文集。1365–1374。[39] Akash Srivastava、Lazar Valkov、Chris Russell、Michael U Gutmann 和 Charles Sutton。2017。Veegan：利用隐式变分学习减少生成对抗网络中的模式崩溃。神经信息处理系统进展 30（2017）。[40] Fei Sun、Jun Liu、Jian Wu、Changhua Pei、Xiao Lin、Wenwu Ou 和 Peng Jiang。2019。BERT4Rec：基于 Transformer 双向编码器表示的序列推荐。载于第28届 ACM 国际信息与知识管理会议论文集。1441–1450。[41] Richard S Sutton 和 Andrew G Barto。2018。《强化学习：导论》。麻省理工学院出版社。[42] Nima Taghipour、Ahmad Kardan 和 Saeed Shiry Ghidary。2007。基于使用情况的网络推荐：一种强化学习方法。载于2007年 ACM 推荐系统会议论文集。113–120。[43] Jiaxi Tang 和 Ke Wang。2018。通过卷积序列嵌入实现个性化 top-n 序列推荐。载于第11届 ACM 网络搜索与数据挖掘国际会议论文集。565–573。[44] Kai Wang、Zhene Zou、Yue Shang、Qilin Deng、Minghao Zhao、Runze Wu、Xudong Shen、Tangjie Lyu 和 Changjie Fan。2021。RL4RS：基于强化学习的推荐系统真实世界基准。arXiv abs/2110.11073（2021）。[45] Shoujin Wang、Longbing Cao、Yan Wang、Quan Z Sheng、Mehmet A Orgun 和 Defu Lian。2021。基于会话的推荐系统综述。ACM计算调查（CSUR）54，7（2021），1–38。[46] Shu Wu、Yuyuan Tang、Yanqiao Zhu、Liang Wang、Xing Xie 和 Tieniu Tan。2019。基于图神经网络的会话推荐。载于 AAAI 人工智能大会论文集，第33卷。346–353。[47] Yikun Xian、Zuohui Fu、Shan Muthukrishnan、Gerard De Melo 和 Yongfeng Zhang。2019。用于可解释推荐的强化学习知识图谱推理。载于第42届国际 ACM SIGIR 信息检索研究与发展会议论文集。285–294。

[48] Xin Xin、Alexandros Karatzoglou、Ioannis Arapakis 和 Joemon M Jose。2020。面向推荐系统的自监督强化学习。载于第*43*届国际 *ACM SIGIR* 信息检索研究与发展会议。931–940。[49] Xin Xin、Alexandros Karatzoglou、Ioannis Arapakis 和 Joemon M. Jose。2022。面向推荐系统的监督式优势演员-评论家（*WSDM '22*）。计算机协会，美国纽约州纽约市，1186–1196。https: //doi.org/10.1145/3488560.3498494[50] Xinyang Yi、Ji Yang、Lichan Hong、Derek Zhiyuan Cheng、Lukasz Heldt、Aditee Kumthekar、Zhe Zhao、Li Wei 和 Ed Chi。2019。面向大规模语料项目推荐的采样偏差校正神经建模。载于第*13*届 *ACM* 推荐系统会议论文集。269–277。[51] Xiangyu Zhao、Changsheng Gu、Haoshenglun Zhang、Xiwang Yang、Xiaobing Liu、Hui Liu 和 Jiliang Tang。2021。DEAR：推荐系统中的在线广告曝光深度强化学习。载于*AAAI* 人工智能大会，第35卷。750–758。[52] Xiangyu Zhao、Long Xia、Jiliang Tang 和 Dawei Yin。2019。《Xiangyu Zhao、Long Xia、Jiliang Tang 和 Dawei Yin 撰写，Martin Vesely 协调的〈用于搜索、推荐和在线广告的深度强化学习：综述〉》。《*ACM Sigweb* 通讯》春季刊（2019年），1–15。[53] 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。[54] 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* 国际信息与知识管理会议论文集。1883–1891。[55] 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*年网络会议论文集。3582–3589。[56] Xiangyu Zhao、Liang Zhang、Zhuoye Ding、Long Xia、Jiliang Tang 和 Dawei Yin。2018年。基于成对深度强化学习的带负反馈的推荐。载于第*24*届 *ACM SIGKDD* 国际知识发现与数据挖掘会议论文集。1040–1048。

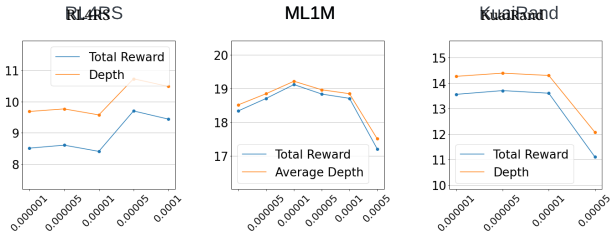

图11：控制 HAC 的演员学习率，X 轴表示$\mathcal{L}_{\text{QMax}}$的学习率。

A 实验设置

A.1 数据集处理

对于 ML1M数据集，我们将用户评分高于 3 的电影视为正样本（表示“喜欢”），其余视为负样本。对于 KuaiRand[16] 数据集，我们首先进行 50核过滤，移除出现次数少于 50 次的视频。观看时长比例大于 0.8 的视频被视为正样本（表示“有效观看”或“消费”）。对于 ML1M 和 KuaiRand 数据集，我们按时间顺序将每个用户会话分割为由长度为 10 的物品列表组成的序列。对于每个分割后的列表，仅将该列表之前的正样本视为历史行为。因此，最终数据集中的每条记录都由用户ID、历史行为序列、物品列表和标签列表组成，其格式与 RL4RS数据集相同。

A.2 在线环境

该环境基于用户响应模型构建，该模型由物品嵌入层、基于注意力的历史编码器以及用于对动作中物品进行评分的评分模块组成。嵌入层负责将稀疏特征转换为稠密特征，并由后续的历史编码器和动作物品评分模块共享。历史编码器层由两个多头注意力模块组成，并输出用户嵌入。第一个网络对历史记录中物品的序列信息进行编码，第二层则同时编码来自前一层的序列信息和用户信息。最终的评分模块计算目标/动作物品嵌入与用户嵌入之间的点积，并在训练过程中采用二元交叉熵损失。如第 3.1 节所述，用户反馈根据这些最终得分进行采样。在接收推荐列表并得到用户反馈后，正向物品用于更新用户历史记录，而静态特征保持不变。我们将奖励 $r(s_t, a_t)$ 设计为逐物品奖励的平均值，其中正向响应的奖励为 1，负向响应的奖励/代价为 -0.2。

我们还构建了一个基于耐心值的用户离开模型，用于提供会话结束信号 d。当系统耗尽用户的耐心，使其没有剩余耐心继续进行下一轮交互时，用户便会离开系统。用户离开模型会为用户设定耐心值预算；每当用户收到一个推荐列表，预算就会减少。耐心值预算的减少幅度取决于推荐质量，而且推荐质量越差，耐心值消耗得越快。此外，我们将所有环境的最大会话深度限制为 20。

A.3 模型规格

本节介绍模型的架构和超参数。我们的代码使用 PyTorch 实现，源代码见 https://github.com/CharlieMat/Hyper-Actor-Critic-for-Recommendation。

SASRec 骨干策略： 该模块在表3中的所有方法之间共享。对于任意给定的物品嵌入（出现在历史序列、曝光物品和候选物品池中），我们使用物品编码器将其映射为一个 32 维向量。随后，将用户历史记录的编码向量与可训练的位置嵌入相加，并与编码后的用户向量（通过一个隐藏维度为 128 的两层网络获得）拼接。然后，将最终输入序列送入一个包含 4 个注意力头、丢弃率为 0.1 的两层 Transformer 编码器。输出序列中的最后一个嵌入被视为编码后的用户状态。最后，我们使用线性映射根据用户状态生成超动作。最终效果动作则依据第 2.4 节所述的点积排序分数进行选择。

评论家： 评论家是一个多层感知机，以（编码后的用户状态向量、超动作向量）为输入，包含两个维度分别为 256 和 64 的隐藏层，并输出一个表示 Q值 的标量。不同的是，在 A2C 中使用的是一个价值网络，其隐藏层维度相同，但仅以用户状态作为输入。

离线监督： 我们基于观测曝光（推荐物品列表）及相应的用户反馈，使用离线训练集通过公式（10）中的二元交叉熵损失来优化策略。我们采用批量大小 64，并在 [0.0005、0.0001、0.00005、0.00001] 中搜索学习率，同时在 [0.0001、0.00005、0.00001、0.000005] 中搜索 L2正则化系数。在训练期间评估在线性能时，我们考虑对每个交互步骤采用小批量训练。

在线监督： 在线版本的监督学习与离线监督学习不同，它直接使用通过与用户进行在线交互而收集的采样轨迹，而非来自离线数据集的小批量样本。

对于以下所有基于强化学习的方法，我们使用容量为 100,000 的经验回放缓冲区，并在缓冲区中的样本超过 2,000 个时开始小批量训练。缓冲区中的每个样本由（用户静态特征、当前历史记录、后续历史记录、超动作嵌入、效果动作（推荐列表）、用户响应、即时奖励和“完成”信号）组成。我们同时运行 32 个用户会话/回合，因此每个交互步骤都会向缓冲区采样并存入 32 条记录。当前用户状态由（用户静态特征、当前历史记录）组成，后续用户状态也由（用户静态特征、当前历史记录）组成。演员学习率的通用超参数搜索空间为 $[0.0005, 0.0001, 0.00005, 0.00001, 0.000005, \ldots 0.000001]$，评论家学习率的搜索空间为 $[0.001, 0.0001, 0.00001] \ldots$，高斯超动作探索的方差为 $\sigma \in [0.1, 0.01, 0.001]$。除表3中提到的模型差异外，我们将实验设置总结如下：

A2C： 该方法结合了前述 SASRec 骨干网络与价值（评论家）网络，并采用由价值的优势分数引导的策略梯度。A2C 的探索

方法	演员学习	评论家学习	监督学习	动作对齐
SL	None	None	BCE	None
A2C	在 $P(a\|s)$ 上使用 $V(s)$ 进行策略梯度	$V(s)$上的时序差分误差	None	None
DDPG	$\max Q(s,Z)$，其中包含 $P(Z\|s)$	$Q(s,Z)$上的时序差分误差	None	None
TD3	$\max \min(Q_1(s,Z)\ Q_2(s,Z))$，其中包含 $P(Z\|s)$	$\min(Q_1(s,Z)\ Q_2(s,Z))$上的时序差分误差	None	None
DDPG-RA	关于 $Q(s,\ Z)$ 与 $P(Z\|s)$ 的最大值	$Q(s,\ Z)$ 上的时序差分误差	None	$-\mathbb{E}\left[\ln(P(a_t\|s_t,\ s_{t+1}))\right]$
HAC	关于 $Q(s,\ Z)$ 与 $P(Z\|s)$ 的最大值	$Q(s,\ a)$ 上的时序差分误差	可选	公式 (9)

表3：方法组件比较

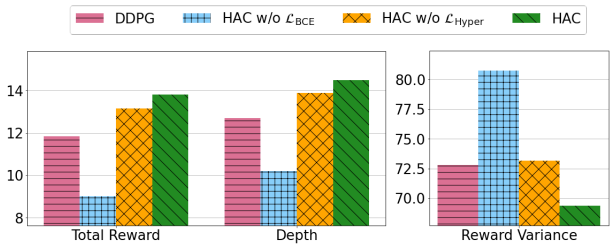

图14：HAC 模型在 KuaiRand 在线环境中的消融实验结果

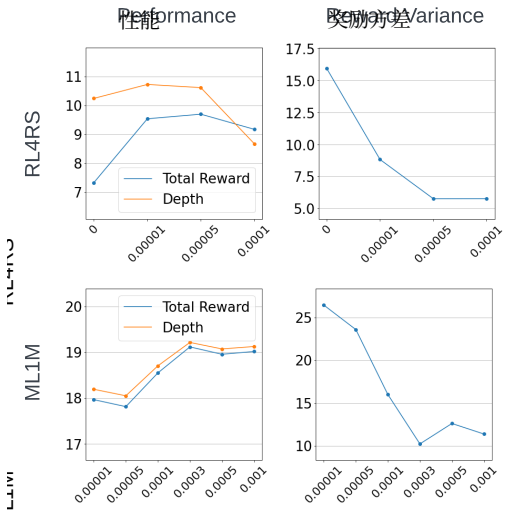

图12：公式(10) 在 ML1M 和 RL4RS 数据集上的监督学习率。

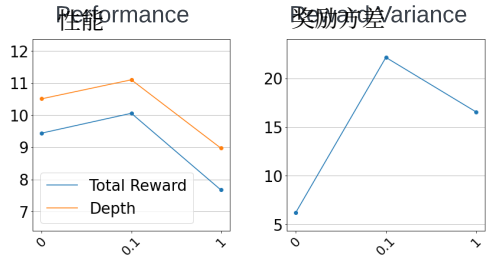

图13：超动作对齐对 RL4RS 在线环境的影响。X 轴表示公式(9)的幅度。

不会在超动作上进行探索，因此存在噪声方差 $\sigma = 0$；其动作探索则基于效果动作概率进行分类采样。

DDPG: 该模型使用 Q网络作为评论家，并在超动作空间上采用高斯采样。

DDPG-RA： 该模型是在 DDPG 模型的基础上增加了效果动作对齐模块，如表3所示。对于对齐模块，我们在 [0 00005, 0 00001，中搜索学习率。0 000005, 0 000001]

HAC： 该方法使用效果-动作空间进行评论家学习，并使用超动作空间进行演员学习。因此，我们可以在两种动作空间中进行探索。与 DDPG 和监督学习相比，该方法额外包含一个动作对齐模块，我们按照第3.3节所述，在 $\lambda_h \in [0.001, 0.01, 0.1, 1]$ 中搜索其幅度。

B 辅助实验结果

我们在图11中展示了 HAC 不同演员学习率的对比结果。在图12中，我们展示了 HAC 不同监督学习率的对比结果，作为第3.3节实验结果的补充。在图14中，我们展示了 HAC 不同模型消融设置的对比，作为第3.4节消融实验结果的补充。所有结果均与主要实验结果一致，进一步验证了我们的论点。