PARWiS：在极低预算下使用主动成对比较判定获胜者

Shailendra Bhandari 奥斯陆都会大学计算机科学系 挪威奥斯陆 电子邮箱：shailendra.bhandari@oslomet.no

arXiv:2603.01171v1 [cs.LG] 1 Mar 2026

摘要——**在基于偏好的学习中，如何在有限预算下通过主动成对比较从一组项目中判定获胜者，是一项颇具挑战的问题。本研究旨在实现并评估 PARWiS 算法。该算法通过谱排序和扰动性成对选择，在极低预算下识别最佳项目。本研究在 PARWiS 的基础上扩展出上下文变体（上下文 PARWiS）和基于强化学习的变体（强化学习 PARWiS），并将它们与包括双重汤普森采样和随机选择策略在内的基线方法进行比较。评估涵盖合成数据集和真实世界数据集（Jester 和 MovieLens），针对 20 个项目，采用 40、60 和 80 次比较的预算。性能通过恢复比例、报告获胜者的真实排名、真实获胜者的报告排名和累积遗憾进行衡量，同时考察分离度指标 $\Delta_{1,2}$。结果表明，PARWiS 和强化学习 PARWiS 在所有数据集上的表现均优于基线方法，尤其是在 Jester 数据集中，其 $\Delta_{1,2}$ 更高；而在更具挑战性的 MovieLens 数据集中，随着 $\Delta_{1,2}$ 较小，性能差距有所缩小。上下文 PARWiS 的表现与 PARWiS 相当，这表明上下文特征可能还需要进一步调优，才能带来显著收益。**

索引词—**对决老虎机、主动学习、基于偏好的学习、PARWiS、强化学习、极低预算**

I. 引言

通过成对比较进行基于偏好的学习，是一种强大的方法，可用于推荐系统、社会选择和信息检索等应用中的排名与获胜者判定 [1, 2]。然而，在现实场景中，比较次数通常受有限预算的约束，这种预算也称为极低预算。Sheth 和 Rajkumar [3]提出了 PARWiS（极低预算下的获胜者成对主动恢复），这是一种旨在此类约束下，利用谱排序和扰动性成对选择策略高效确定获胜者的算法。

在本研究中，我们实现了 PARWiS，并将其扩展为两个变体：受 Bengs 等人启发的上下文 PARWiS [4, 5], ，用于在可用时纳入项目特征；以及强化学习 PARWiS，这是一种基于强化学习的方法，利用 Q 学习来优化配对选择。这些算法与两个基线方法进行了比较：双重汤普森采样（双重汤普森采样） [6]以及随机配对选择策略。评估采用通过 Bradley-Terry（BT）模型生成的合成数据，以及真实世界数据集：Jester [7]和 MovieLens 20M [8]。我们在三种极低预算（40、60 和 80 次比较，每次针对 20 个项目）下，使用 [4, 5], 中的指标评估了性能：恢复比例、报告获胜者的真实排名、真实获胜者的报告排名以及累积遗憾。此外，还计算了分离度指标 $\Delta_{1,2}$ 以分析问题难度，并通过统计检验评估性能差异的显著性。

结果表明，PARWiS 和强化学习 PARWiS 在恢复率和累积遗憾方面始终优于基线方法，尤其是在 Jester 数据集上，此时较高的 $\Delta_{1,2}$ 表示问题较容易。在更具挑战性的 MovieLens 数据集上，由于 $\Delta_{1,2}$较小，PARWiS 和强化学习 PARWiS 仍表现更好，但优势幅度较小。上下文 PARWiS 的表现与 PARWiS 相近，这表明所用的上下文特征（仅在合成数据中可用）可能还需要进一步优化。统计检验证实，PARWiS 和强化学习 PARWiS 相较于基线方法的改进具有显著性；误差分析则显示，与其他智能体相比，强化学习 PARWiS 失败时更接近真实获胜者。

II. 相关工作

在对决老虎机框架中，通过成对比较进行获胜者判定和排名的问题已得到广泛研究。该框架衔接了多臂老虎机与基于偏好的学习，在推荐系统、社会选择和信息检索等应用中尤为相关。在这些应用中，通常无法获得直接的数值反馈，因此必须通过比较来推断偏好。下文将回顾对决老虎机、用于排名的主动学习、上下文扩展，以及使用真实世界数据集进行评估方面的重要进展。

A. 对决老虎机与遗憾最小化

对决老虎机框架旨在模拟反馈是相对而非绝对的场景，反馈通常以成对比较的形式呈现。Yue等人 [9] 将 k 臂对决老虎机问题形式化，重点研究遗憾最小化。他们提出了交错过滤算法等算法，通过维护一组潜在获胜者并借助比较逐步筛选，从而平衡探索与利用。这些算法的遗憾界随臂数和比较预算而变化，但该方法假设孔多塞获胜者存在，而这一假设并非适用于所有场景。

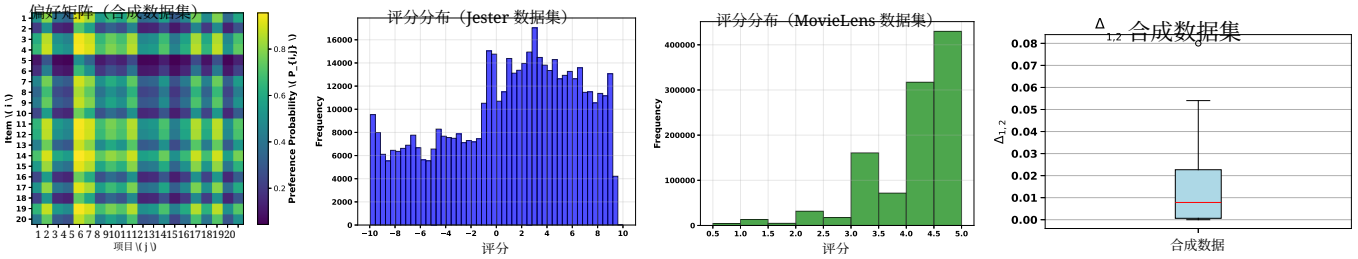


图 1. 数据集可视化：(a) 合成数据集 ($k = 20$) 的偏好矩阵热力图，展示成对概率 $P_{i,j}$。(b) Jester 中选出的 20 个笑话的评分直方图，评分范围为 -10 至 10。(c) MovieLens 中选出的 20 部电影的评分直方图，评分范围为 0.5 至 5。(d) 合成数据集在 30 次运行中的 $\Delta_{1,2}$ 箱线图，展示问题难度的分布。

后续研究扩展了这一框架，使其能够处理更复杂的偏好结构。Zoghi 等人 [10]提出了相对上置信界（RUCB）算法，将上置信界（UCB）策略应用于对决老虎机。RUCB 为成对偏好维护置信区间，并选择最有可能减少获胜者相关不确定性的臂对。尽管 RUCB 能实现次线性遗憾，但它需要大量比较才能收敛，因此不太适合极低预算的场景。

稀疏 Borda [11] 进一步改进了这种方法，通过在成对比较矩阵中纳入稀疏性条件，尽早剔除次优项。该算法旨在找出 Borda获胜者（平均偏好得分最高的项目），在我们的设定中，这与布拉德利-特里-卢斯（BTL）获胜者一致。然而，正如 Sheth 和 Rajkumar[3], 所示，由于比较次数不足，无法准确估计稀疏结构，因此该算法在极其有限的预算下表现受限。

B. 用于排名和获胜者判定的主动学习

主动学习方法旨在尽量减少对项目进行排名或识别前k个项目所需的比较次数。Jamieson 和 Nowak [12]提出了一种主动排序算法，将项目建模为 d维欧几里得空间中的嵌入，并利用几何性质选择信息量大的配对。该方法对无噪声比较有效，但在处理噪声模型（如本研究核心的 BTL）时表现不佳。

Mohajer 等人 [13] 提出了 SELECT，这是一种用于前-k 项排名聚合的单淘汰赛算法。该算法在每一轮随机配对项目，并将获胜者晋级至下一轮，直至确定前-k 个项目。该算法通过在各轮中重复进行比较，扩展为可处理极低预算的情形；但正如 [3]所指出的，其随机配对策略限制了效率。Heckel 等人 [14]提出了主动排序（AR），这是一个用于完全排序和获胜者判定的通用框架。AR 为项目维护置信度分数，并将它们划分到不同的分箱中，以便重点比较不确定的项目对。尽管 AR 在参数假设（例如 BTL）成立时表现良好，但其所需的比较次数超出了极低预算通常允许的范围，因此在我们的设定中竞争力较弱。

Sheth 和 Rajkumar [3]提出了 PARWiS（极低预算下的成对主动赢家恢复），专门用于预算受限情况下的获胜者判定。PARWiS 利用谱排序 [15] 估计 BTL 分数，并选择扰动最大的项目对，迭代更新排名。该算法包含初始化阶段（通过 $k-1$ 次比较构建初始排名）和更新阶段；更新阶段利用扰动量选择能够最大程度改变排名的项目对。在极低预算条件下，PARWiS 在合成数据和真实世界数据集上的表现优于 SELECT 和 AR 等基线方法。我们的工作以 PARWiS 为基础，通过实现并扩展其上下文变体，进一步推进了这一方法。

C. 上下文对决赌博机

上下文对决赌博机纳入项目特征以改进配对选择，尤其是在比较充足的场景中。Bengs 等人 [4, 5] 对上下文赌博机进行了综述，重点介绍了使用线性或非线性模型、基于特征预测比较结果的算法。例如，线性收益上下文对决赌博机（CBLP） [16] 将偏好建模为项目特征的线性函数，并选择能最大化期望信息增益的配对。这些方法虽然有效，但通常假设可以获得丰富的特征集，而在 Jester 或 MovieLens 等真实世界数据集中，这些特征集可能并不存在。

近期研究探索了非线性模型，以提升灵活性。Saha 和 Gopalan [17] 提出了一种基于核的方法，用于建模复杂的偏好关系，并在高维场景下实现了更紧的遗憾界。然而，这些方法需要大量计算资源和大量比较，因此不适用于极低预算。在我们的工作中，我们通过 Contextual PARWiS 扩展 PARWiS，利用逻辑回归根据特征预测比较结果。然而，由于真实世界的数据集缺少特征，Contextual PARWiS 在这种情况下会退回到非上下文行为。

D. 谱排序与偏好聚合

谱排序方法通过成对比较来估计物品得分，是 PARWiS 等算法的基础。Negahban 等人 [15] 提出了 Rank Centrality，

该方法根据比较结果构建增广马尔可夫链，并计算其平稳分布，以近似 BTL 分数。Rank Centrality 只需$\mathcal{O}(n\log\ n)$次比较，就能获得合理的误差界，因此适用于主动学习场景。谱最大似然估计[18] 使用逐坐标最大似然估计来改进这些估计值，从而提高排名前k 的准确性。然而，这些方法通常是被动的，假设比较集合固定；而 PARWiS 会主动选择配对以优化排名。

Agarwal 等人提出了谱排序的一种加速版本，将计算复杂度降低了数个数量级。尽管前景可期，但它在主动设置中的应用仍未得到充分探索；我们将其集成到 PARWiS 留待未来工作。其他偏好聚合方法，例如 Multisort 中使用的 Copeland 聚合，会结合多个来源的排名（例如带噪快速排序）来确定获胜者。这些方法在极低预算下效果较差，因为它们需要多个完全排序，超出了比较次数的限制。

E. 用于偏好学习的真实世界数据集

真实世界数据集为评估基于偏好的学习算法提供了实用基准。Jester 数据集收集于 1999 年至 2003 年，包含来自 73,421 名用户对 100 个笑话给出的 410 万条评分，评分范围为 -10 至 。由于其评分矩阵较为稠密，该数据集广泛用于推荐系统研究，尤其是其中一组被大多数用户评分的“标尺”笑话。MovieLens 20M 是另一个标准基准，包含来自 138,000 名用户对 27,000 部电影给出的 2000 万条评分；其评分矩阵较为稀疏，对算法处理缺失数据的能力构成了挑战。在 中，这两个数据集都曾用于评估 PARWiS：通过对评分差值应用逻辑函数，将评分转换为成对概率。我们在实验中采用了这种方法。

其他数据集（如 Sushi） [22], 直接提供偏好排名，可将其转换为成对比较。众包方法（如 Crowd-BT） [23], 扩展了 BTL 模型，以处理恶意用户行为，并同时对投票者可靠性和物品得分进行建模。这些数据集和方法凸显了真实世界应用中偏好数据的多样性，促使我们在合成数据集和真实世界数据集上进行评估。

F. 极低预算下的挑战

极低预算给对抗老虎机和主动排序带来了独特挑战。RUCB 和稀疏 Borda 等算法需要进行大量比较才能降低遗憾，而在 $B = 2k$ 这样的约束下，这往往难以实现。SELECT 和 AR 等主动学习方法则难以进行随机或次优的配对选择，导致恢复率较低。PARWiS 通过关注能够最大限度促进排序更新的扰动对来应对这些挑战，我们在工作中复现并拓展了这一策略。然而，问题难度的影响（以$\Delta_{1,\ 2}$ 衡量）仍是一个关键因素，因为顶层项目之间的差距越小，确定获胜者就越困难；我们在实验中探讨了这一现象。

总之，本研究以 PARWiS 奠定的基础为出发点，融入 Contextual 和强化学习扩展，并在多种数据集上评估性能。通过解决现有方法在极低预算下的局限，本研究将有助于深入理解基于偏好学习中的主动获胜者判定。

III. 方法论

A. 问题设定

考虑一个包含 k 个项目的集合，并在布拉德利-特里-卢斯（BTL）模型下进行成对比较。在 BTL 模型中，每个项目 i 都有一个得分 w_i，项目 i 击败项目 j 的概率为:

$$P_{i,j} = \frac{w_i}{w_i + w_j}.$$

给定预算 B，目标是通过最多进行 B 次比较，找出得分最高的项目（获胜者）。我们参照 [3], 在极低预算 $B = 2k,\ 3k,\ 4k$的情况下评估性能。

B. 算法

这里实现了五种算法:

- **双重汤普森采样（双重汤普森采样）** [6]: 使用两步汤普森采样来选择项目对，并为成对偏好维护贝塔先验。
- **随机**: 均匀随机地选择项目对，作为基线。
- **PARWiS** [3]: 通过包含 $k-1$ 次比较的初始化阶段构建谱排序，随后选择扰动最大的项目对来更新排名。
- **上下文 PARWiS**: PARWiS 的扩展版本，在可用时纳入项目特征，并借助逻辑回归预测比较结果，灵感源自 [4, 5]。
- **强化学习 PARWiS**: 一种基于强化学习的 PAR WiS 扩展版本，使用 Q 学习来学习配对选择策略。状态包括当前排名和比较次数，动作是选择一对项目进行比较，奖励则结合了每一步的遗憾减少，以及最终恢复真实获胜者时获得的奖励。

C. 数据集

该算法在三个数据集上进行了评估，并提供了详细的特征描述和可视化结果，以突出其特性。

- **合成数据**: 使用 BTL 模型生成，包含 $k = 20$ 个项目和 $d = 5$ 个特征维度。物品得分 w_i从正态分布中采样，成对概率按 $P_{i,\ j} = w_i/(w_i + w_j)$ 计算。特征为随机正态向量，仅供上下文

PARWiS 使用。图 1(a) 展示了偏好矩阵的热力图$P_{i,j}$，呈现了成对偏好的变化。图 1(d) 展示了 $\Delta_{1,2}$ 在 30 次运行中的箱线图，证实了所报告的变化（$\Delta_{1,2} = 0.0152 \pm 0.0190$），并表明其难度适中，但有些运行较容易或较困难。

- **Jester**：Jester 数据集 1（子集 1） [7]，包含 73,421 位用户在 1999 年至 2003 年间对 100 个笑话给出的 410 万条评分。评分范围为 -10 至+10，其中 99 表示评分缺失。为与 [3](Δ1,2 = 0.0043)报告的难度相匹配，我随机选取了 20 个笑话（使用固定随机种子以确保结果可复现）。然而，选出的子集得到 $\Delta_{1,2} = 0.0946 \pm 0.0000$，这表明问题相对容易，可能是评分分布所致。图 1(b) 展示了所选笑话评分的直方图，分布略呈右偏，以 0 为中心，大多数评分介于 -5 至 5 之间。这个稠密的评分矩阵有助于准确估计成对概率。

- **MovieLens**：MovieLens 20M [8],包含 138,000 位用户在 1995 年至 2015 年间对 27,000 部电影给出的 2,000 万条评分。评分范围为 0.5 至 5（以 0.5 为增量）。我们按评分数量选取排名前 20 的电影，得到一个稀疏的评分矩阵，其 $\Delta_{1,2} = 0.0008 \pm 0.0000$，表明这是一个具有挑战性的问题，其中评分最高的两部电影几乎难以区分。图 1(c) 展示了所选电影评分的直方图，分布呈左偏，大多数评分介于 3 至 5 之间，反映出用户倾向于给热门电影较高的评分。

对于真实世界数据集，按照 [3] 的方法，对平均评分之差应用逻辑函数，将评分转换为成对概率。由于 Jester 和 MovieLens 缺少项目特征，上下文 PARWiS 在这些数据集上会退回到非上下文行为。

D. 评估指标

以下指标定义见 [3]:

- **恢复比例**：推荐真实获胜者的运行所占比例。

- **报告获胜者的真实排名**：推荐项目在 BTL 排序中的真实排名（越低越好）。
- **真实获胜者的报告排名**：智能体内部排名中赋予真实获胜者的排名（仅适用于 PARWiS、上下文 PARWiS 和强化学习 PARWiS；越低越好）。
- **累积遗憾**：非最优项目赢得对决的次数。

- $\Delta_{1,2}$：排名前两位项目之间的差距，计算方式为 $(P_{1,2} - 0.5)_2$，用于表示问题难度。

我们还进行了统计检验（成对 t 检验），以评估性能差异的显著性，并通过错误分析评估失败率，以及算法未能找回获胜者时的平均真实排名。

IV. 实验

A. 设置

本研究还使用 $k = 20$个项目、 $B \in \{40, 60, 80\}$预算，对算法进行了评估；每个数据集和每种预算均运行 30 次。每次模拟的时域与预算相匹配。对于合成数据，特征由随机正态向量生成。对于真实世界数据集，由于没有可用特征，上下文 PARWiS 会退化为非上下文行为。强化学习 PARWiS 使用基于遗憾和最终奖励的组合训练 5000 回合。

B. 结果与讨论

我在表 I、II 和 III 中分别汇总了所有数据集和预算下的结果，重点关注恢复比例、报告的获胜者的真实排名以及累积遗憾。其他指标，包括真实获胜者的报告排名、统计检验和误差分析，见附录C。图 2、3 和 4 展示了 $B = 40$ 下随对决次数变化的性能趋势；其他预算下的图表以及最终指标分布的箱线图见附录 A。

结果表明，在合成数据和 Jester 数据集上，PARWiS 和强化学习 PARWiS 始终取得最高的恢复比例，在所有预算下均约为 0.467，这体现了它们在中等到大型问题上的稳健性 $\Delta_{1,2}$。在 MovieLens 上，$\Delta_{1,2} = 0.0008$，所有智能体都表现不佳，恢复比例降至 0.100–0.167，凸显了区分排名靠前项目的难度。报告的获胜者的真实排名也呈现类似趋势：PARWiS 和强化学习 PARWiS 在 Jester 上的排名最低（2.067），说明推荐结果更接近真实获胜者；而在 MovieLens 上，真实排名则较高（约为 6.633–6.667）。在所有数据集上，PARWiS 的累积遗憾最低；强化学习 PARWiS 在 MovieLens 上的遗憾略高（例如，在 $B = 80$ 时为 56.967），这可能是由于该数据集具有较大挑战性。

图 2、3 和 4 分别展示了合成数据、Jester 和 MovieLens 数据集在 $B = 40$ 下的性能趋势。累积遗憾图表明，PARWiS 和强化学习 PARWiS 的遗憾累积速度比双重汤普森采样和随机方法更慢，并在初始化阶段（$k - 1 = 19$ 次比较）后趋于稳定。恢复比例图显示，PARWiS 随时间推移始终保持较高的恢复比例，而双重汤普森采样和随机方法难以识别获胜者。PARWiS 和强化学习 PARWiS 的报告获胜者的真实排名下降得更快；强化学习 PARWiS 中真实获胜者的报告排名更高（例如，在合成数据上约为 5.3），而 PARWiS 则约为 4.4，这表明其排序机制仍有改进空间。附录 A 中展示的 $B = 60$ 和 $B = 80$也呈现出类似趋势，PARWiS 在不同预算下始终保持领先。附录 A 中通过箱线图展示了不同运行之间的变化。

C. 统计检验与误差分析

通过成对 t 检验评估性能差异的显著性，重点比较

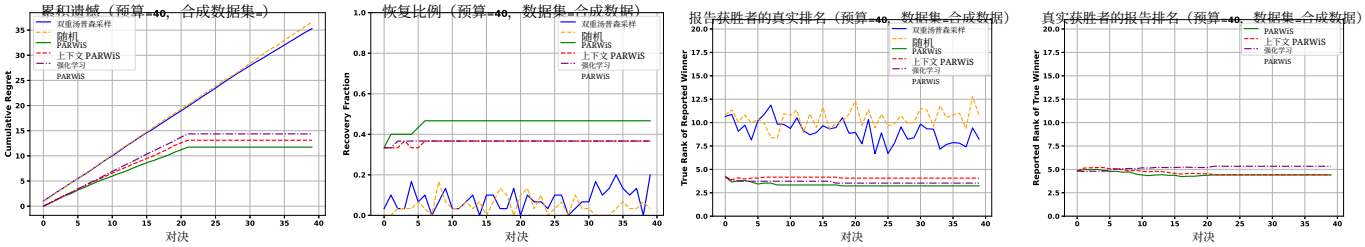


图 2. $B = 40$ 时合成数据集上的性能。从左到右依次为：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。 $B = 60$，80 的图表见附录 A，图 5 和图 6。

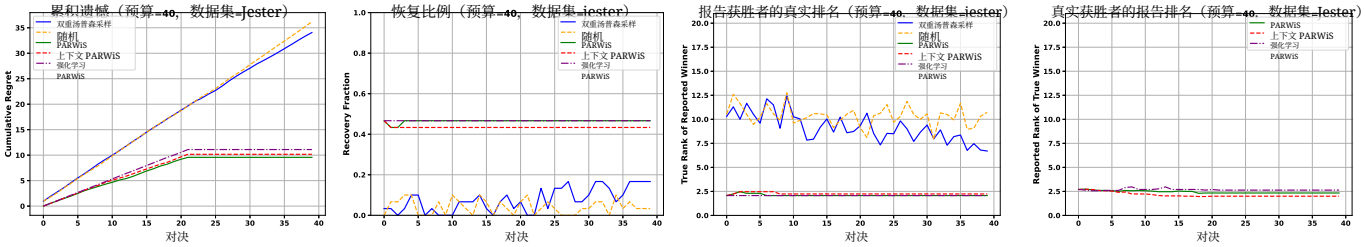


图 3. Jester 数据集上的性能 $B = 40$。从左到右：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名真实获胜者。 $B = 60$，80 的图表见附录 A 的图 7 和图 8。

表 I 不同数据集和预算下的恢复比例。

	合成数据（$\Delta_{1,\ 2} = 0.0152 \pm 0.0190$）			Jester（$\Delta_{1,2} = 0.0946 \pm 0.0000$）			MovieLens（$\Delta_{1,2} = 0.0008 \pm 0.0000$）		
智能体	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$
双重汤普森采样	0.200	0.067	0.267	0.167	0.233	0.467	0.133	0.067	0.067
随机	0.033	0.067	0.000	0.033	0.000	0.067	0.033	0.000	0.067
PARWiS	**0.467**	**0.467**	**0.467**	**0.467**	**0.467**	**0.467**	**0.167**	**0.167**	**0.167**
上下文 PARWiS	0.367	0.367	0.367	0.433	0.433	0.433	**0.167**	**0.167**	**0.167**
强化学习 PARWiS	0.367	0.367	0.367	**0.467**	**0.467**	**0.467**	0.100	0.100	0.100

表II 跨数据集和预算的报告获胜者的真实排名。

	合成数据 ($\Delta_{1,\ 2} = 0.0152 \pm 0.0190$)			Jester ($\Delta_{1,2} = 0.0946 \pm 0.0000$)			MovieLens ($\Delta_{1,2} = 0.0008 \pm 0.0000$)		
智能体	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$
双重汤普森采样	8.233	6.933	4.767	6.700	4.700	3.133	9.233	10.300	11.500
随机	10.767	10.367	10.733	10.733	9.367	10.733	9.233	11.033	10.767
PARWiS	**3.233**	**3.233**	**3.233**	**2.067**	**2.067**	**2.067**	**6.633**	**6.633**	**6.633**
上下文 PARWiS	3.900	4.067	4.067	2.233	2.233	2.233	**6.633**	**6.633**	**6.633**
强化学习 PARWiS	3.533	3.533	3.533	**2.067**	**2.067**	**2.067**	6.667	6.667	6.667

双重汤普森采样与 PARWiS 在恢复比例方面的差异（附录 C 中的表 VII）。在 $B = 40$ 的合成数据上，PARWiS 的表现显著优于双重汤普森采样（t 统计量 =-2.246，p 值= 0.029），且差距在 $B = 60$ 时进一步扩大（t 统计量 = -3.862，p 值 = 0.000）。在 Jester 上，$B = 40$ 时差异显著（t 统计量 = -2.594，p 值 = 0.012），但到 $B = 80$ 时差异减小（t 统计量 = 0.000，p 值 = 1.000），此时双重汤普森采样迎头赶上（恢复比例为 0.467）。在 MovieLens 上，差异并不显著（例如，$B = 40$ 时 p 值为 = 0.723），这反映出该数据集的难度。

误差分析（附录 C 中的表 IV 和表 VI）显示，PARWiS 和强化学习 PARWiS 的失败率最低在 Jester 上（0.533），且失败时更接近真实获胜者（平均真实排名为 3.000）。在 MovieLens 上，失败率较高（0.833–0.900），但与双重汤普森采样相比，强化学习 PARWiS 失败时更接近真实获胜者（平均真实排名为 7.296，而双重汤普森采样为 12.250，位于 $B = 80$）。

V. 讨论与结论

结果与 Sheth 和 Rajkumar[3], 的发现一致：PARWiS 在极低预算下表现出色，尤其是在 $\Delta_{1,2}$ 较大的数据集上（例如 Jester）。与 Jester（0.0946）和合成数据（0.0152）相比，MovieLens 中较小的$\Delta_{1,2}$ （0.0008）解释了性能下降的原因，因为顶级项目更难区分，这与 {

表 III 不同数据集和预算下的累积遗憾。

	合成数据 ($\Delta_{1,\ 2} = 0.0152 \pm 0.0190$)			Jester ($\Delta_{1,2} = 0.0946 \pm 0.0000$)			MovieLens（$\Delta_{1,2} = 0.0008 \pm 0.0000$）		
智能体	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$
双重汤普森采样	35.300	52.933	67.267	34.067	51.167	67.667	36.733	55.767	74.800
随机	36.633	54.833	73.200	36.167	54.233	72.600	37.733	56.767	75.800
PARWiS	**11.733**	**22.000**	**33.133**	**9.567**	**17.600**	**25.633**	**18.067**	**35.100**	**52.333**
上下文 PARWiS	13.067	24.333	35.467	10.167	18.533	27.200	18.100	35.133	52.367
强化学习 PARWiS	14.367	26.300	42.300	11.100	21.667	32.400	19.567	38.633	56.967

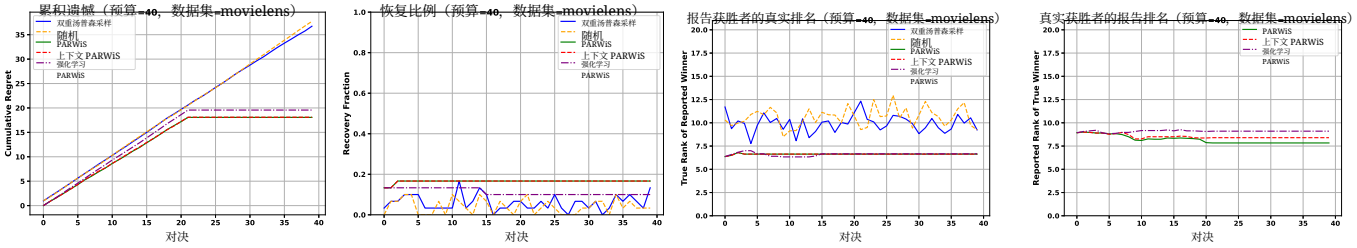


图 4. $B = 40$ 时 MovieLens 数据集上的表现。从左到右：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。 $B = 60$，80 的图见附录 A 图 9 和图 10。

v1}中的理论复杂度界限一致。值得注意的是，针对 Jester 更新后的预处理（随机选取 20 个笑话）仍得到了高于 [3] 报告值（0.0043）的$\Delta_{1,2}$ （0.0946），这表明可能需要进一步调整笑话的选取方式，才能达到其报告的难度。

强化学习 PARWiS 在 Jester 和合成数据集上的表现与 PARWiS 相当，在 Jester 上的恢复比例与 PARWiS 持平（0.467），且失败时距离真实获胜者更近（例如，Jester 上的平均真实排名为 3.000，合成数据集上为 5.000）。然而，它在 MovieLens 上的表现较弱（恢复比例为 0.100，而 PARWiS 为 0.167），这可能是由于具有挑战性的$\Delta_{1,\ 2}$ 以及需要更多训练或更丰富的状态表征。统计检验证实，PARWiS 和 RLPARWiS 相较于基线方法的改进具有显著性（$p < 0.05$），而 PARWiS 、上下文 PARWiS 和强化学习 PARWiS 之间的差异通常不显著（$p > 0.05$），表明它们的效果相当。

上下文 PARWiS 在真实世界数据集上的表现与 PARWiS 相似，因为缺少特征，它会退回到非上下文行为。在具备特征的合成数据上，其表现略逊一筹，这表明随机特征（$d = 5$）可能不够有信息量。未来的工作可以探索从真实世界数据集中提取特征（例如，使用 MovieLens 中的标签数据），以提升上下文 PARWiS。

总之，我们实现并评估了 PARWiS 算法，用于在极低预算下进行获胜者判定，并将其扩展为上下文和强化学习变体。实验确认，PARWiS 和强化学习 PARWiS 的表现优于基线方法，尤其是在较简单的问题上（较高的$\Delta_{1,\ 2}$）。强化学习扩展展现出良好的前景，但仍需针对具有挑战性的数据集进行进一步优化。未来的工作包括改进上下文 PARWiS 的特征工程、增强强化学习 PARWiS 的状态表示，以及探索在极低预算下的前k 恢复。

参考文献

[1]H. Aziz、F. Brandt、E. Elkind 和 P. Skowron，计算社会选择：前十年及未来。柏林，海德堡：Springer-Verlag，2022年，第48–65页。 [在线]。获取地址：https://doi.org/10.1007/ 978-3-319-91908-9 4[2]E. Si mpson 和 I. Gurevych,–“面向群体的可扩展贝叶斯偏好学习”，机器学习，第109卷，第4期，第689–718页，2020年4月。 [在线]。获取地址：https://doi.org/10.1007/s10994-019-05867-2[3]D. Y. Sheth 和 A. Rajkumar，“Parwis：在极其有限的预算下通过主动成对比较确定获胜者”，载于*2021*年 *IEEE* 国际数据挖掘会议（*ICDM*），2021年，第569–578页。[4]Y. Xu、W. Lu 和 R. Song，“存在干扰的线性上下文赌博机”，2024年。 [在线]。获取地址：https://arxiv.org/abs/2409.15682[5]Y. D. Tsai 和 S. D. Lin，“基于生成对抗网络的非线性上下文赌博机快速在线推断”，2022年。 [在线]。获取地址：https: // arxiv.org/abs/2202.08867[6]H. Wu 和 X. Liu，“对决赌博机的双重汤普森采样”，载于神经信息处理系统进展，D. Lee、M. Sugiyama、U. Luxburg、I. Guyon 和 R. Garnett 编，第29卷。Curran Associates,Inc.,2016。 [在线]。获取地址：https://proceedings.neurips.cc/paper files/paper/2016/file/9de6d14fff9806d4bcd1ef555be766cd-Paper.pdf [7]K. Goldberg、T. Roeder、D. Gupta 和 C. Perkins，“Eigentaste：一种常数时间协同过滤算法”，信息检索，第4卷，第2期，第133–151页，2001年。 [在线]。获取地址：https://doi.org/10. 1023/A:1011419012209

[8]F. Harper 和 J. Konstan，“MovieLens 数据集：历史与背景”，*ACM*交互式智能系统汇刊，第5卷，第4期，2015年12月，出版者版权：© 2015 ACM。[9]Y. Yue、J. Broder、R. Kleinberg 和 T. Joachims，“k臂对决老虎机问题”，计算机与系统科学杂志，第78卷，第5期，第1538–1556页，2012年，jCSS特别专题：云计算2011。 [在线]。获取地址：https://www.sciencedirect.com/science/article/pii/S0022000012000281[10]M. Zoghi、S. Whiteson、R. Munos 和 M. Rijke，“k臂对决赌博机问题的相对上置信界”，载于第*31*届国际机器学习会议论文集，机器学习研究论文集系列，E. P. Xing 和 T. Jebara 编，第32卷，第2期。北京，中国：PMLR，2014年6月22–24日，第10–18页。 [在线]。获取地址：https://proceedings.mlr.press/v32/zoghi14.html[11]K. G. Jamieson、S. Katariya、A. Deshpande 和 R. D. Nowak，“稀疏对决赌博机”，*ArXiv*，第abs/1502.00133卷，2015年。 [在线]。获取地址：https://api.semanticscholar.org/CorpusID:10900487[12]K. G. Jamieson 和 R. D. Nowak，“利用成对比较进行主动排序”，2011年。 [在线]。获取地址：https://arxiv.org/abs/1109.3701[13] S. Mohajer、C. Suh 和 A. Elmahdy，“基于带噪比较的前k 名聚合的主动学习”，载于第*34*届国际机器学习会议论文集，机器学习研究论文集系列，D. Precup 和 Y. W. Teh 编，第70卷。PMLR，2017年8月6–11日，第2488–2497页。[在线]。获取地址：https://proceedings.mlr.press/v70/ mohajer17a.html[14] R. Heckel、N. B. Shah、K. Ramchandran 和 M. J. Wainwright，“基于成对比较的主动排序，以及参数假设何时无济于事”，2016年。[在线]。获取地址：https://arxiv.org/abs/1606.08842[15]S. N. Negahban、S. Oh 和 D. Shah，“Rank Centrality：基于成对比较的排序”，运筹学，第65卷，第266–287页，2012年。 [在线]。获取地址：https://api.semanticscholar.org/CorpusID:3602049[16]Y. Yue 和 T. Joachims，“将信息检索系统作为对决赌博机问题进行交互式优化”，载于第*26*届年度国际机器学习会议论文集，ICML ’09系列。美国纽约州纽约市：美国计算机协会，2009年，第1201–1208页。 [在线]。获取地址：https://doi.org/10.1145/1553374.1553527[17]M. Dud´ık、K. Hofmann、R. E. Schapire、A. Slivkins 和 M. Zoghi，“上下文对决赌博机”，载于第*28*届学习理论会议论文集，机器学习研究论文集系列，P. Gr¨unwald、E. Hazan 和 S. Kale 编，第40卷。巴黎，法国：PMLR，2015年7月3–6日，第563–587页。 [在线]。获取地址：https://proceedings.mlr.press/ v40/Dudik15.html[18]Y. Chen 和 C. Suh，“谱最大似然估计：基于成对比较的 Top-k 排名聚合”，2015 年。 [在线]。可访问：https://arxiv.org/abs/1504.07218[19]A. Agarwal、P. Patil 和 S. Agarwal，“加速谱排序”，载于机器学习研究论文集，第 *35* 届国际机器学习会议论文集，机器学习研究论文集丛书，J. Dy 和 A. Krause 编，第 80 卷。PMLR，2018 年 7 月 10–15 日，第 70–79 页。 [在线]。可访问：https://proceedings.mlr.press/v80/agarwal18b.html[20] D. G. Saari 和 V. R. Merlin，“Copeland 方法 (*)”，载于《经济理论》，第 8 卷，第 1 期，第 51–76 页，1996 年。[在线]。可访问：https://ideas.repec.org/a/spr/joecth/v8y1996i1p51-76.html[21]L. Maystre 和 M. Grossglauser，“直接排序！一种简单有效的主动偏好学习方法”，2017 年。 [在线]。可访问：https://arxiv.org/abs/1502.05556[22]T. Kamishima，“Nantonac 协同过滤：基于顺序响应的推荐”，载于第九届 *ACM SIGKDD* 国际知识发现与数据挖掘会议论文集，KDD ’03 丛书。美国纽约州纽约市：美国计算机协会，2003 年，第 583–588 页。 [在线]。可访问：https://doi.org/10.1145/956750.956823[23]X. Chen、P. N. Bennett、K. Collins-Thompson 和 E. Horvitz，“众包环境中的成对排名聚合”，载于第六届 *ACM* 国际网络搜索与数据挖掘会议论文集，WSDM ’13 丛书。美国纽约州纽约市：美国计算机协会，2013 年，第 193–202 页。 [在线]。可访问：https://doi.org/10.1145/2433396.2433420

附录

A. 补充可视化

本节提供了一组全面的补充可视化图表，以补充主文档中的性能分析。具体而言，其中包括合成数据、Jester 和 MovieLens 数据集在预算 $B = 40$、60、80 下的性能图和箱线图。这些性能图通过展示 $B = 60$ 和 $B = 80$ 的结果，扩展了图 2、3 和 4 中的分析，从而更深入地了解算法在预算增加时的表现。每张性能图由四个子图组成：累积遗憾、恢复比例、报告获胜者的真实排名和真实获胜者的报告排名，这些子图共同展示了这些指标随对决次数的变化。这些图突显了各算法的动态表现，显示它们以多快的速度收敛到真实获胜者，以及遗憾如何随时间累积。箱线图则通过展示 30 次运行中最终指标值的分布，与这些图相辅相成，详细呈现了每次模拟结束时性能的变异性和一致性。综合来看，这些图表全面展示了

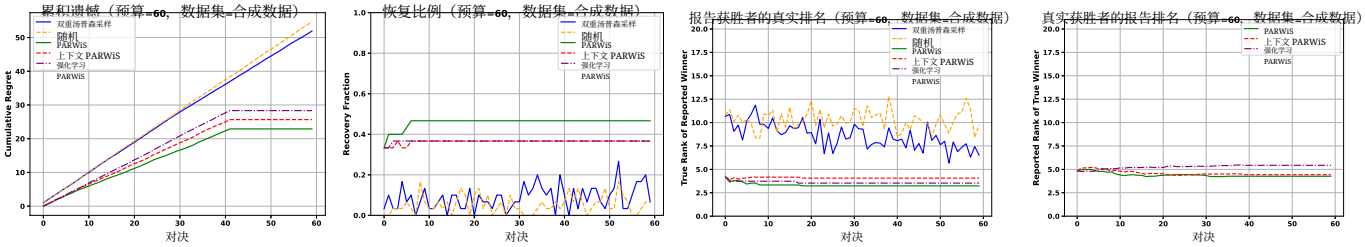


图 5。在 $B = 60$ 的合成数据集上的性能。从左到右依次为：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

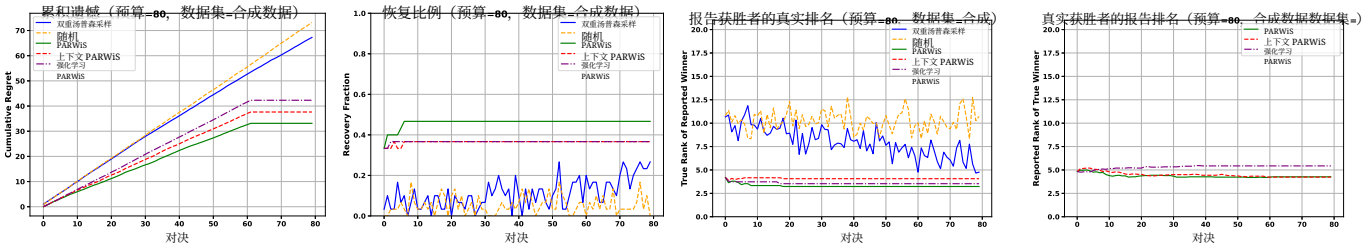


图 6。在 $B = 80$ 的合成数据集上的性能。从左到右依次为：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

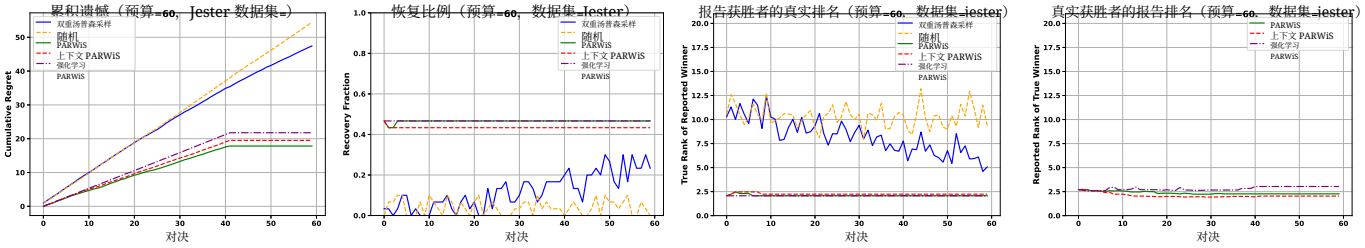


图 7。在 $B = 60$ 的 Jester 数据集上的性能。从左到右依次为：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

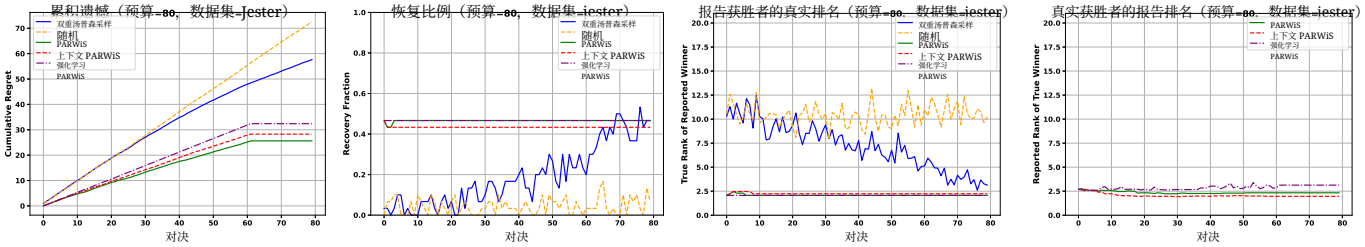


图 8。在 $B = 80$ 的 Jester 数据集上的性能。从左到右依次为：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

算法的行为，既呈现了随时间变化的趋势，也体现了它们在不同数据集和预算下最终结果的稳健性。

$B = 60$ 和 $B = 80$ 的性能图揭示了算法如何适应更大的预算，并延续了正文中 $B = 40$ 观察到的趋势。对于合成数据集（图 5 和图 6），PARWiS 和强化学习 PARWiS 在累积遗憾和恢复比例方面继续优于基线方法（DoubleTS 和随机），并随着预算增加保持领先。PARWiS 和强化学习 PARWiS 报告的赢家的真实排名仍然较低，这表明能够稳定识别出接近真实获胜者的项目，而强化学习 PARWiS 对真实获胜者的报告排名略有波动，表明其排序机制仍有进一步优化的空间。在 Jester 数据集上（图 7 和图 8），较大的 $\Delta_{1,2} = 0.0946$使胜者识别更容易，PARWiS 和强化学习 PARWiS 在所有预算下均实现了约 0.467 的稳定恢复比例，并且在初始阶段之后累积遗憾极少。正如正文所述，双重汤普森采样在 $B = 80$ 时有所改善，其恢复比例达到 0.467，追上了 PARWiS 和强化学习 PARWiS。对于 MovieLens 数据集（图 9 和图 10），

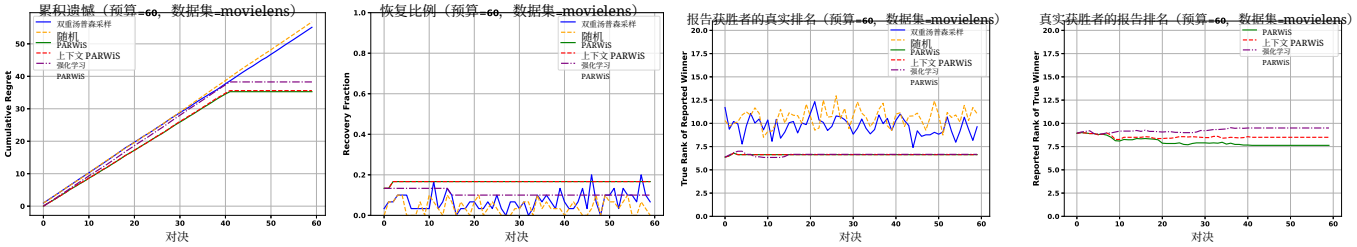


图 9. $B = 60$ 时 MovieLens 数据集上的性能。从左到右：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

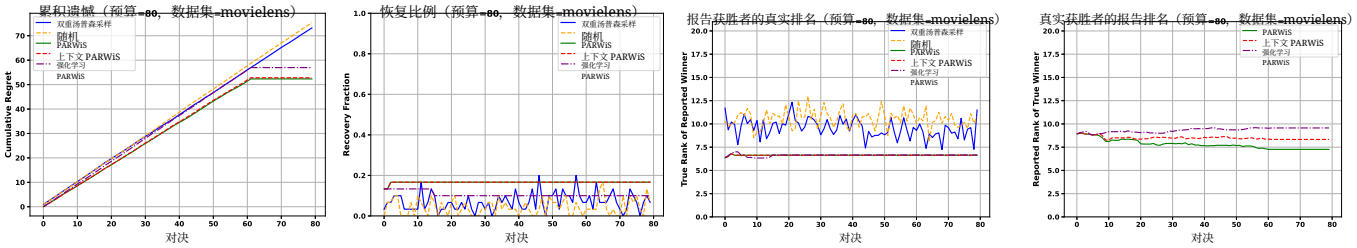


图 10. $B = 80$ 时 MovieLens 数据集上的性能。从左到右：累积遗憾、恢复比例、报告获胜者的真实排名、真实获胜者的报告排名。

较小的 $\Delta_{1,2} = 0.0008$ 带来了重大挑战，所有算法的恢复比例都较低（0.100–0.167）。PARWiS 的累积遗憾最低，但随着预算增加，各算法之间的差距逐渐缩小，反映出在该数据集中区分排名靠前的项目有多困难。这些图共同展现了 PARWiS 和强化学习 PARWiS 在不同问题难度和预算下的稳健性，同时也凸显了所有算法在 MovieLens 等具有挑战性的数据集上的局限性。

B. 最终指标的分布

图 11、12 和 13 中的箱线图展示了每个数据集和预算下 30 次运行中最终指标值（累积遗憾、恢复比例、报告获胜者的真实排名以及真实获胜者的报告排名）的分布。这些图详细呈现了每次模拟结束时性能的变化情况，与性能图中显示的平均趋势相辅相成。对于合成数据集（图 11），与双重汤普森采样和随机算法相比，PARWiS 和强化学习 PARWiS 在恢复比例和真实排名上的分布更集中，表明其性能更为稳定；而双重汤普森采样和随机算法的分布范围更广、离群值更多，尤其是在累积遗憾方面。对于 Jester 数据集（图 12），PARWiS 和强化学习 PARWiS 的分布明显更集中，恢复比例集中在 0.467 左右，真实排名的变异也很小（约为 2.067），这反映出该问题较为简单（$\Delta_{1,2} = 0.0946$）。双重汤普森采样在 $B = 80$ 时表现得更加稳定，这与其恢复比例的提升相符。对于 MovieLens 数据集（图 13），由于问题难度较高（$\Delta_{1,2} = 0.0008$），所有算法的分布都更分散；PARWiS 和上下文 PARWiS 在真实排名上的变异略小（约为 6.633），而强化学习 PARWiS 为 6.667。这些箱线图凸显了 PARWiS 和强化学习 PARWiS 在多次运行中的稳健性，同时也强调了问题难度对性能稳定性的影响，尤其是在 MovieLens 数据集上，所有算法的变异都更大。

C. 补充表格

本节提供补充表格，以便更深入地了解各算法的性能，并补充正文中的结果与讨论。这些表格涵盖合成数据、Jester 和 MovieLens 数据集在预算 $B = 40, 60, 80$ 下真实获胜者的报告排名、统计显著性检验以及误差分析。这些表格从定量角度展示了各算法的排名准确率、性能差异的显著性，以及胜者识别的一致性，尤其是在失败情形中。综合来看，它们有助于加深对算法在不同问题难度和预算约束下行为的理解，从而对其有效性和稳健性进行更全面的评估。

表 V 报告了在所有数据集和预算下，根据 PARWiS、上下文 PARWiS 和强化学习 PARWiS 的内部排名确定的真实获胜者平均报告排名。由于这些算法采用谱排序，该指标仅适用于它们，用以衡量真实获胜者在智能体估计排序中的位置（数值越低越好）。对于合成数据集（$\Delta_{1,2} = 0.0152 \pm 0.0190$），PARWiS 始终取得最低的报告排名（例如，在 $B = 40$ 时为 4.400，在 $B = 60, 80$ 时降至 4.267），体现了其对真实获胜者位置的准确估计。上下文 PARWiS 的表现相近（在 $B = 40$时为 4.433，在 $B = 60, 80$ 时为 4.233），而强化学习 PARWiS 的报告排名较高（例如，在 $B = 40$ 时为 5.333），表明其排序机制仍有改进空间，正如正文中所述

表 IV 合成数据和 Jester 数据集的误差分析。

	合成数据 ($\Delta_{1,\ 2} = 0.0152 \pm 0.0190$)						Jester ($\Delta_{1,2} = 0.0946 \pm 0.0000$)					
	$B = 40$		$B = 60$		$B = 80$		$B = 40$		$B = 60$		$B = 80$	
智能体	Fail	平均 TR	Fail	平均 TR	Fail	平均 TR	Fail	平均 TR	Fail	平均 TR	Fail	平均 TR
双重汤普森采样	0.800	10.042	0.933	6.929	0.733	6.136	0.833	7.840	0.767	6.304	0.533	5.250
随机	0.967	11.103	0.933	10.357	1.000	10.733	0.967	11.069	1.000	9.367	0.933	11.071
PARWiS	**0.533**	5.188	**0.533**	5.188	**0.533**	5.188	**0.533**	**3.000**	**0.533**	**3.000**	**0.533**	**3.000**
上下文 PARWiS	0.633	5.842	0.633	5.842	0.633	5.842	0.567	3.176	0.567	3.176	0.567	3.176
强化学习 PARWiS	0.633	**5.000**	0.633	**5.000**	0.633	**5.000**	**0.533**	**3.000**	**0.533**	**3.000**	**0.533**	**3.000**

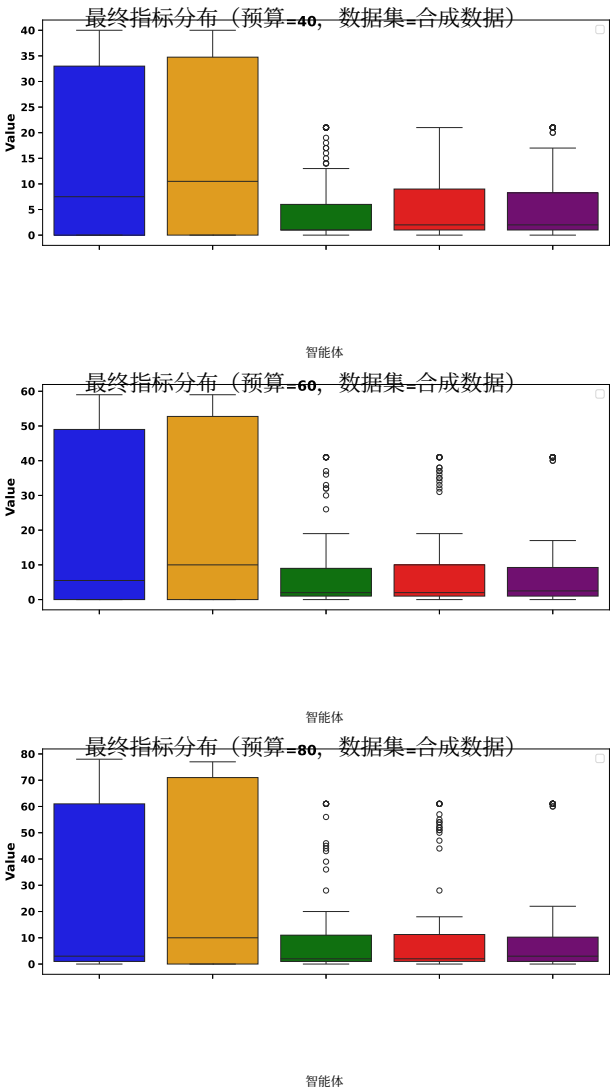


图 11. 合成数据集在预算为 $B = 40$、60、80 时，各次运行最终指标值的分布。指标包括累积遗憾、恢复比例、报告获胜者的真实排名和真实获胜者的报告排名。

。在 Jester 数据集（$\Delta_{1,\ 2} = 0.0946$）上，由于问题较简单，所有算法对真实获胜者的排名都更靠前，其中上下文 PARWiS 的表现最佳（例如，在 $B = 40$ 时为 2.000，在 $B = 60$ 时为 1.967）。对于 MovieLens 数据集（$\Delta_{1,\ 2} = 0.0008$）中，由于区分难度较高，报告排名更高；其中 PARWiS 表现最佳（例如，在 $B = 40$ 时为 7.833，在 $B = 80$ 时提升至 7.267），而强化学习 PARWiS 表现不佳（例如，在 $B = 80$ 时为 9.567）。这些结果凸显了 PARWiS 在不同数据集上的卓越排名准确率，尤其是在 MovieLens 上，由于 $\Delta_{1,\ 2}$ 较小，识别获胜者十分困难。

表 VII 展示了成对 t 检验的结果，用于比较双重汤普森采样与 PARWiS 在所有数据集和预算下的恢复比例，并评估二者性能差异的统计显著性。恢复比例定义为真实获胜者被推荐的运行次数占总运行次数的比例，是本研究衡量成功的关键指标。在合成数据集上，PARWiS 在 $B = 40$ (t 统计量 = -2.246，p 值 = 0.029) 和 $B = 60$(t 统计量 = -3.862，p 值 = 0.000) 下显著优于双重汤普森采样；随着比较次数增加，双重汤普森采样的表现有所提升，二者差距在 $B = 80$ (t 统计量 =-1.616，p 值 = 0.112) 时缩小。对于 Jester 数据集，二者差异在 $B = 40$ (t 统计量 = -2.594，p 值 = 0.012) 时显著，在 $B = 60$ (t 统计量=-1.921，p 值 = 0.060) 时则达到边缘显著；但在 $B = 80$ (t 统计量 = 0.000，p 值 = 1.000) 时，差异有所减小，此时双重汤普森采样迎头赶上，恢复比例达到 0.467，与 PARWiS 持平。在 MovieLens 上，差异并不显著（例如，$B = 40$ 时 p 值为 = 0.723，$B = 60$，80 时为 0.235），这反映了该数据集的难度（$\Delta_{1,\ 2} = 0.0008$），并对所有算法产生了类似影响。这些检验证实，在大多数情况下，PARWiS 相较于双重汤普森采样的改进具有统计显著性，尤其是在中等至较大 $\Delta_{1,\ 2}$ 下的合成数据集和 Jester 数据集上；而在 MovieLens 上，这种显著性较弱，因为二者差距较小，性能差异也因此减弱。

表 IV 和表 VI 通过报告各算法在不同数据集和预算下的失败率（真实获胜者未被推荐的运行所占比例）以及失败情形下所报告获胜者的平均真实排名（平均 TR），对误差进行了分析。这项分析有助于了解算法的一致性，以及它们未能识别真实获胜者时错误的严重程度。对于合成数据集（表 IV），PARWiS 的失败率最低（所有预算下为 0.533）；而且失败时，所报告获胜者的真实排名相对靠前（5.188），说明它选择的项目接近真实获胜者。强化学习 PARWiS 的失败率略高（0.633），但失败时选择的项目更接近真实获胜者（5.000）；双重汤普森采样和随机算法的失败频率则更高（例如，在

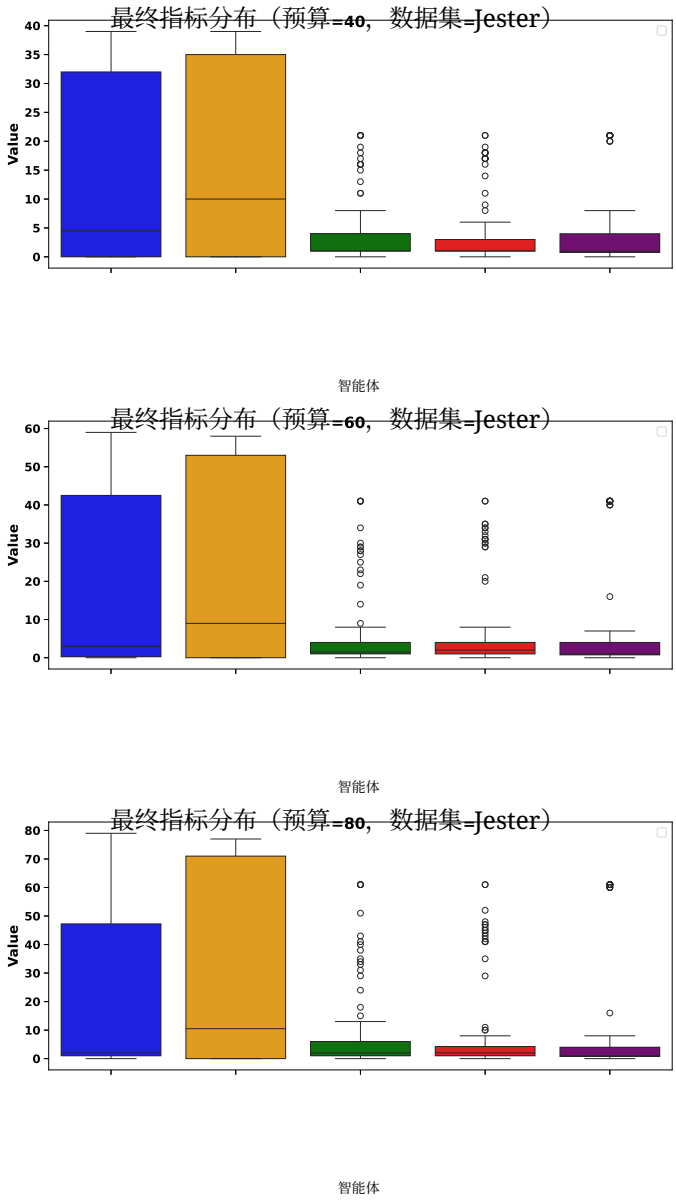


Fig. 12. Jester 数据集在预算为 $B = 40$、60、80 时，各次运行的最终指标值分布。指标包括累积遗憾、恢复比例、报告获胜者的真实排名以及真实获胜者的报告排名。

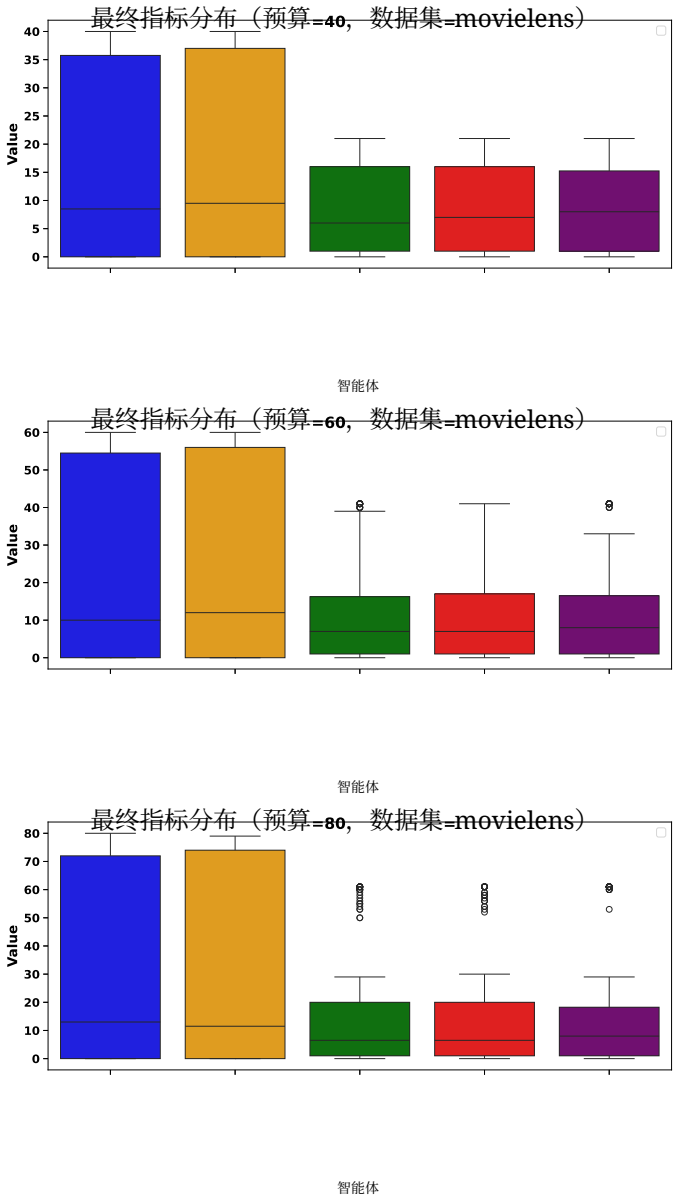


Fig. 13. MovieLens 数据集在预算为 $B = 40$、60、80 时，各次运行的最终指标值分布。指标包括累积遗憾、恢复比例、报告获胜者的真实排名以及真实获胜者的报告排名。

$B = 40$时分别为 0.800 和 0.967），且真实排名更高（10.042 和 11.103）。在 Jester 数据集上，较简单的问题（$\Delta_{1,\,2} = 0.0946$）使 PARWiS 和强化学习 PARWiS 的失败率降低至 0.533，且失败时选择的项目非常接近真实获胜者（3.000）。在 $B = 80$ 时，双重汤普森采样的表现有所改善（失败率为 0.533，真实排名为 5.250），与其恢复比例的提升一致。对于 MovieLens 数据集（表 VI），较小的 $\Delta_{1,\,2} = 0.0008$导致所有算法的失败率都很高（在 $B = 40$时为 0.833–0.967），但与强化学习 PARWiS （7.296）和双重汤普森采样（在 $B = 40$时为 9.920，至 $B = 80$时升至 12.250）相比，PARWiS 和上下文 PARWiS 失败时选择的项目更接近真实获胜者（7.760）。这些表格凸显了 PARWiS 和强化学习 PARWiS 的稳健性：它们失败的频率更低，且失败时选择的项目更接近真实获胜者，尤其是在 Jester 这类较容易的数据集上，同时也展现了所有算法在 MovieLens 上面临的挑战。

D. 代码、*Python*包与数据可用性

Dueling Bandit Toolkit 的源代码（包括 PARWiS、上下文 PARWiS、强化学习 PARWiS、双重汤普森采样和随机配对基线算法的实现）已在 GitHub 上公开，网址为 https:// github.com/shailendrabhandari/dueling bandit。该代码库包含完整的代码、文档和示例脚本，可用于复现本文所述的实验。该工具包还以 Python包 的形式发布在 Python

表 V 不同数据集和预算下真实获胜者的报告排名。

	合成数据（$\Delta_{1,2} = 0.0152 \pm 0.0190$）			Jester（$\Delta_{1,2} = 0.0946 \pm 0.0000$）			MovieLens（$\Delta_{1,2} = 0.0008 \pm 0.0000$）		
智能体	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$
PARWiS	**4.400**	**4.267**	**4.267**	2.333	2.333	2.333	**7.833**	**7.833**	**7.267**
上下文 PARWiS	4.433	4.233	4.233	**2.000**	**1.967**	**2.000**	8.400	8.333	8.333
强化学习 PARWiS	5.333	5.433	5.433	2.633	3.133	3.133	9.100	9.100	9.567

表 VI：MovieLens 数据集上的误差分析。

	MovieLens（$\Delta_{1,2} = 0.0008 \pm 0.0000$）					
	$B = 40$		$B = 60$		$B = 80$	
智能体	Fail	平均 TR	Fail	平均 TR	Fail	平均 TR
双重汤普森采样	0.833	9.920	0.933	11.036	0.933	12.250
随机	0.967	9.552	1.000	11.033	0.933	11.500
PARWiS	**0.833**	7.760	**0.833**	7.760	**0.833**	7.760
上下文 PARWiS	**0.833**	7.760	**0.833**	7.760	**0.833**	7.760
强化学习 PARWiS	0.900	**7.296**	0.900	**7.296**	0.900	**7.296**

表 VII 不同数据集和预算下的精选 t 检验结果（双重汤普森采样与 PARWiS、恢复的比较）。

	合成数据			Jester			MovieLens		
预算	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$	$B = 40$	$B = 60$	$B = 80$
t 统计量	-2.246	-3.862	-1.616	-2.594	-1.921	0.000	-0.356	-1.201	-1.201
p 值	0.029	0.000	0.112	0.012	0.060	1.000	0.723	0.235	0.235

软件包索引（PyPI: https://pypi.org/project/dueling-bandit/），可使用 pip 安装:

```
pip install dueling-bandit
```

完整的文档（包括 API 参考、教程和实验详情）托管在 ReadTheDocs 上，网址为 https://dueling-bandit.readthedocs.io/en/latest/。文档提供了设置工具包、运行模拟和可视化结果的分步指南，并详细介绍了本研究中使用的算法和数据集。