arXiv:2601.02649v1 [cs.RO] 6 Jan 2026

使用蒙特卡洛树搜索和前瞻包裹的有效在线三维装箱

Jiangyi Fang[1,4], Bowen Zhou[2], Haotian Wang[3], Xin Zhu[3], Leye Wang[1]*, 4*[1] 北京大学高可信软件技术重点实验室（教育部），中国[2] 哈尔滨工业大学计算学院，哈尔滨，中国[3] 京东物流，北京，中国[4] 北京大学计算机科学学院，北京，中国
fangjiangyi2001@gmail.com,leyewang@pku.edu.cn

摘要
在现代物流中，使用机械臂进行在线三维装箱（三维装箱）对于降低运输和劳动力成本至关重要。尽管深度强化学习（DRL）表现出色，但它往往无法适应真实世界中的短期分布偏移；当不同批次的货物依次到达时，就会出现这种偏移，并导致性能下降。我们认为，现代物流系统中可用的短期前瞻信息是缓解这一问题的关键，尤其是在发生分布偏移时。我们将带有前瞻包裹的在线三维装箱建模为模型预测控制（MPC）问题，并对蒙特卡洛树搜索（MCTS）框架进行调整以求解该问题。我们的框架采用动态探索先验，根据前瞻特征自动平衡已学习的强化学习策略与稳健的随机策略。此外，我们还设计了一种辅助奖励，用于惩罚单次放置造成的长期空间浪费。在真实世界数据集上进行的大量实验表明，我们的方法始终优于最先进的基线方法：在分布偏移情况下性能提升超过 10%，在线部署中的平均性能提升 4%，最佳情况下提升超过 8%，充分证明了该框架的有效性。

关键词
三维装箱、强化学习、模型预测控制

1 引言
根据市场报告 [31]，全球物流市场预计将从2024年的6.68万亿美元增长至2033年的11.27万亿美元，增长率为5.98%。对于物流公司（例如，京东物流 [2] 和亚马逊 [1]）而言，装箱是供应链中的一项关键质量控制措施。例如，提高装箱的空间利用率可大幅降低运输成本 [4]。然而，由于劳动力成本高昂以及实际操作中的顺序约束（例如，放置

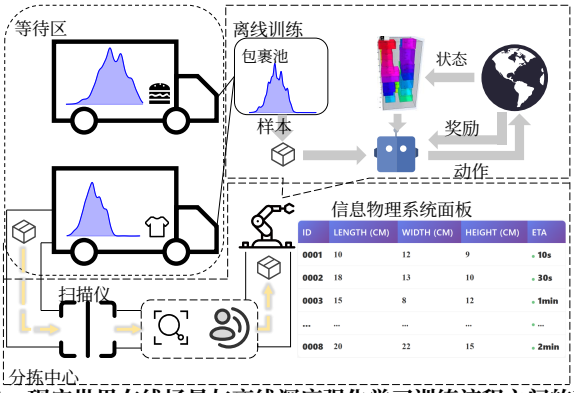


图1：现实世界在线场景与离线深度强化学习训练流程之间的不匹配。

对于在传送带上移动的包裹，必须逐个即时做出决策，以防止包裹堆积），研究人员正日益将研究重点转向使用机械臂进行在线三维装箱 [42, 45]

现有研究主要分为启发式方法 [7, 8, 11, 13, 17, 25, 39, 41, 45, 47] 和深度强化学习（DRL）方法 [39, 44, 45, 47–49]。启发式规则通常根据工人的经验总结而来，例如先放置底层 [40], 深处优先、左侧优先 [18], 角落优先 [20] 等。尽管经过大量训练后，深度强化学习近期已超越启发式方法，但这两种方法都无法应对短期分布偏移这一现实场景。如图1所示，空间有限的分拣中心必须按照卡车的到达顺序依次卸货。由于卡车装载着来自不同仓库的包裹（例如，尺寸各异的食品与尺寸统一的衣物），这一过程会形成非平稳包裹流。然而，当前的在线深度强化学习算法是在从静态包裹池 [47, 48], 中采样的包裹上训练的，该包裹池汇集了来自所有仓库的包裹，并按固定周期（例如，每周）更新。因此，这些算法在实际应用中无法适应此类偏移。

随着信息物理系统和传感技术的发展， [26, 32, 52], 在线智能体可以在包裹经过扫描仪时获取相关信息。如图1所示，面板上显示的信息构成一个前瞻队列：新扫描的包裹添加到队列末尾，而成功装入箱子的包裹则从队首移除。该队列部分揭示了即将到来的包裹尺寸分布，使智能体能够检测偏移并提前调整策略。

然而，如何高效且有效地利用前瞻信息来增强在线策略仍颇具挑战。一种方法是使用基于规划的方法 [3, 15, 51] ，并结合序列

*通讯作者。

允许为个人用途制作本作品全部或部分内容的数字副本或纸质副本，仅在不以营利或商业利益为目的复制或分发副本，且副本上保留本声明及首页完整引文的情况下，方可免费用于课堂教学。必须尊重非作者所拥有的本作品组件的版权。注明出处后可以摘录。如需以其他方式复制、再版、发布到服务器或分发至列表，须事先取得特定许可和/或支付费用。请向permissions@acm.org 申请许可。会议缩写 'XX，纽约州伍德斯托克© 2018 年版权归所有者/作者所有。出版权授权给 ACM。ACM ISBN 978-1-4503-XXXX-X/2018/06 https://doi.org/XXXXXXX.XXXXXXX

决策约束。然而，由于其固有的计算复杂度（表 2 显示，广度优先搜索 的推理开销超过 100 秒），其效率通常不足以满足在线应用的需求。另一种直接的方法是在深度强化学习场景中，将前瞻包裹编码为状态的一部分，但我们的实验（表1中的 *PCT* 前瞻 ）和已有研究 [47]表明，这种方法并不充分。此外，Zhao 等人提出了一种重排序算法，使当前包裹的放置取决于后续包裹 [47]。然而，这些方法忽略了根本问题。鉴于强化学习本质上已经具有前瞻性 [21]，，这些方法没有进一步追问：为什么这种固有的前瞻能力在在线三维装箱中仍然不足？事实上，在三维装箱场景中，前瞻包裹并不总能带来同等程度的改进。如图 14 和第 3 节所示，短期分布偏离原始分布的程度越大，前瞻信息带来的潜在性能增益就越大。

面对不熟悉的包裹时，人工工作者可能会拆开最近放置的项目 [35, 45]，进行多次放置试验，并根据以往经验评估最佳方案。受这种人类智能的启发，我们将带有前瞻包裹的在线三维装箱问题构建为一个模型预测控制（MPC）问题 [12, 19]。在这一构建中，“放置试验”对应一组有限时域的放置轨迹，而“基于以往经验进行评估”则通过使用训练良好的在线强化学习评论家来选择最优路径。该数学构建使我们能够运用先进技术解决这一问题。本文的主要贡献如下：

- 据我们所知，我们首次证实，前瞻信息在短期分布偏移期间最有价值，并将带有前瞻包裹的在线三维装箱问题构建为一个模型预测控制问题。此外，我们认为，我们的方法可以为其他涉及有限前瞻时域的类似马尔可夫决策过程问题提供启发。

- 我们将蒙特卡洛树搜索（MCTS） [37]框架适配到三维装箱场景，以求解上述模型预测控制问题。为使MCTS探索策略适应短期分布偏移，我们设计了一种动态机制，根据训练数据集中出现前瞻包裹的可能性，在稳健的随机策略与离线学习的强化学习策略之间调整探索先验，实现二者的平衡。此外，为使MCTS的模拟和回传阶段适用于三维装箱，我们引入了一个辅助奖励项：考虑到每次放置对未来包裹的长期影响，该项会对放置造成的浪费体积进行惩罚。
- 我们在两个现实场景中开展了大量实验，并进行了在线评估。结果表明，在分布偏移下，我们的方法相较于最先进的深度强化学习基线提升超过10%；在线部署时，平均提升约4%，验证了所提框架的有效性。

2 问题表述

我们将**状态空间**下的决策过程正式定义为一个元组 (B_t, d_t)，其中 B_t可由二维高度图 [47] 或装箱配置树 [48] 表示，而 $d_t = (l_t, w_t, h_t)$表示第 t步包裹的长度 l_t、宽度 w_t和高度 h_t。

动作空间。对于状态 s_t，动作空间 A_{s_t}是包裹 d_t所有有效放置方式的集合。放置 $a_t = (x, y, z) \in A_{s_t}$是一个映射到特定位置和朝向的决策。任何有效动作都必须满足所有物理约束（例如边界、碰撞和稳定性）。

转移函数。转移函数 $P(s_{t+1}|s_t, a_t)$ 具有混合特性。下一个状态 $s_{t+1} = (B_{t+1}, d_{t+1})$ 包含两部分。对于下一个箱体状态 B_{t+1}，状态转移函数是**确定性的**，记为 $B_{t+1} = f(B_t, d_t, a_t)$。对于下一个包裹状态 d_{t+1}，状态转移函数是**随机的**： $d_{t+1} \sim \mathrm{D}$。实践中，我们收集离线数据集 D 来估计理想分布 D。

奖励函数。即时奖励 $r_t = R(s_t, a_t)$ 可以定义为已装箱物品的体积（即 $l_t * w_t * h_t$）除以箱子的总体积（即 $L * W * H$），以鼓励提高空间利用率。

带前瞻包裹的在线三维装箱。我们的问题设定包含几个已知组成部分：长度为 N 个包裹的前瞻序列（即 $d_t, d_{t+1}, \ldots, d_{t+N-1}$ 在步骤 t 时已知）、离线训练的深度强化学习评论家 V_θ 和策略 π_θ、离线数据集 D，以及确定性的箱体状态转移函数 f。给定这些组成部分，我们的目标是优化以下函数：

$$\max_{a_t, \ldots, a_{t+N-1}} \mathbb{E}\left[\sum_{k=0}^{N-1} \left(R(s_{t+k}, a_{t+k}) + V(s_{t+N})\right)\right] \tag{1}$$

其中 $s_t = (B_t, d_t)$、 $B_{t+1} = f(B_t, d_t, a_t)$ 和 $d_{t+N} \sim D$。我们将折扣因子设为 $\gamma = 1$，因为每个回合都会在有限步数内终止（即箱子被装满）。这是一个典型的 MPC 问题，通过在线求解该优化问题，使智能体能够抵抗短期分布偏移。根据滚动时域原理，找到最优动作轨迹后，我们仅执行第一个动作 a_t^*，然后在后续步骤 $t+1$ 中，将新的前瞻包裹追加进来并重新进行规划。

3 理论分析

本节首先给出传统在线三维装箱问题的优化目标，并介绍适用于该问题的基本马尔可夫决策过程（MDP）形式。随后，我们从理论上推导分布偏移的幅度与先前最优策略所对应的性能下降之间的关系。

3.1 在线三维装箱问题的预备知识

根据上述定义，传统在线三维装箱问题的目标是获得满足以下方程的最优策略 $\pi(a|s)$：

$$\bar{\pi} = \arg\max_{\pi} \sum_{t=1}^{\infty} r_t \tag{2}$$

其对应的最优价值函数 $V_D^*(s)$ 唯一满足这一稳态环境下的贝尔曼最优性方程：

$$V_D^*(s_t) = \max_{a \in A}(R(s_t, a_t) + \gamma \sum_{s_{t+1} \in S} P(s_{t+1}|s_t, a) V_D^*(s_{t+1})) \tag{3}$$

由于转移函数的随机部分仅包含 $d_{t+1} \sim D$，该方程可改写为：

$$V_D^*(s_t) = \max_{a \in A}(R(s_t, a_t) + \gamma \mathbb{E}_{d_{t+1} \sim D}\left[V_D^*(s_{t+1})\right] \quad (4)$$

3.2 非平稳场景中的失败。

考虑一个现实的非平稳分布，总体长期分布 D 是这些短期分布D_x的混合：

$$D(d) = \sum_x D_x(d) \cdot P(X = x) \quad (5)$$

其中， X 表示包裹批次类型的离散随机变量。

对于一个确定的包裹批次 x，在策略 π 下新环境中的价值函数 $V_{D_x}^{\pi}$ 定义如下：

$$V_{D_x}^{\pi}(s_t) = \sum_{a_t \in A} \pi(a_t|s_t)\left(R(s_t, a_t) + \gamma \mathbb{E}_{d_{t+1} \sim D_x}\left[V_{D_x}^{\pi}(s_{t+1})\right]\right) \quad (6)$$

分布 D_x 下的最优策略 π^* 定义如下：

$$V_{D_x}^*(s_t) = \max_{a_t \in A}\left(R(s_t, a_t) + \gamma \mathbb{E}_{d_{t+1} \sim D_x}\left[V_{D_x}^*(s_{t+1})\right]\right) \quad (7)$$

分布 D_x 下的最优策略 π^* 与 π 之间的性能差距定义为：

$$\mathrm{Gap}(s) = V_{D_x}^*(s) - V_{D_x}^{\pi}(s) \quad (8)$$

实际上，我们可以发现，性能差距可以由分布之间的全变差距离界定。

$$|\mathrm{Gap}(s)| \le |V_{D_x}^*(s) - V_D^*(s)| + |V_{D_x}^*(s) - V_D^{\pi}(s)| \le C \cdot D_{TV}(D_x, D) \quad (9)$$

其中， $D_{TV}(D_x,\ D)$ 是分布之间的全变差距离 [6, 28]，而 C 是一个常数（证明细节请参见附录 A）。因此，短期分布 D_x 与长期分布 D 之间的距离越大，性能下降幅度就越大，这也意味着潜在的改进空间更大（即考虑前瞻包裹）。

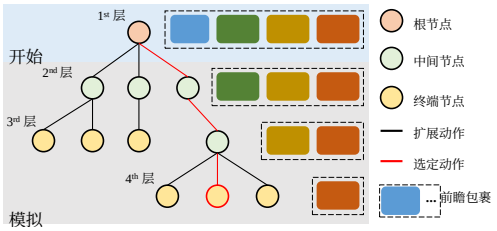


图2：使用暴力树搜索求解包含前瞻包裹的在线三维装箱模型预测控制问题的示意图。

4 方法

4.1 树搜索问题建模

上述模型预测控制问题可以视为一个树搜索问题。图2展示了如何在单个决策步 t 中使用暴力搜索来解决该问题。根节点表示初始状态，由当前箱体状态 B_t 和前瞻队列中的第一个包裹 d_t 组成。从根节点（即1st 层）开始，我们模拟蓝色包裹的所有可能放置动作 $a \in A_{(B_t, d_t)}$。通过状态转移函数 f 执行每个动作后，都会在下一层生成一个新的箱体状态 B'。

图2中的边表示这些状态转移。该过程递归重复；例如，从第二层的每个状态出发，我们模拟绿色包裹的放置，从而生成第三层。搜索会在以下两种情况下于某个节点（终端节点）处终止：箱子已装满，或前瞻序列中只剩下一个包裹。对于后一种情况，我们使用强化学习评论家 $V(s)$来估计终端状态的价值。遍历完整棵树后，我们找出通往价值最高的终端节点的轨迹（以红色突出显示），并将该路径上的第一个动作选为最优选择。因此，求解 MPC 问题等同于在这棵搜索树中寻找最优的根到叶路径。尽管这种方法有效，但由于其复杂度为 $O(|A|^N)$ 的指数级，暴力搜索在计算上难以实现。为解决暴力搜索效率低下的问题，我们将蒙特卡洛树搜索（MCTS）框架应用于3D-BP场景，以有效平衡探索与利用。我们提出的*MPC-3D-BP*框架保留了 MCTS 的标准阶段：选择、扩展、模拟和回传。我们的主要技术贡献集中在两个阶段：首先，为选择阶段设计了一种经过调整的探索策略；其次，为模拟与回传阶段所用的价值估计提出了一种新颖的惩罚项。MCTS 和我们提出的*MPC-3D- BP*均使用试验总次数 n 来控制搜索时间预算。

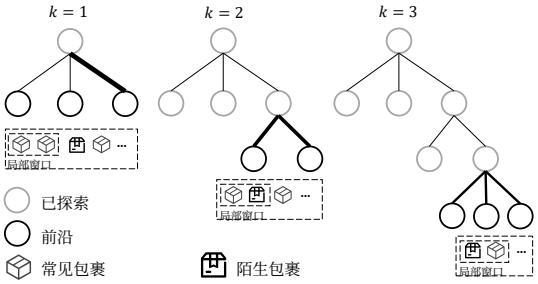


图3：逐步展示移位感知多项式上置信树如何引导选择。边的粗细表示动作概率先验。

4.2 MPC-3D-BP

4.2.1 使用移位感知多项式上置信树进行选择。MCTS 中选择阶段的目标是从根节点沿现有树遍历至一个有望扩展的叶节点。此过程必须在利用已知高价值节点与探索访问次数较少但可能具有较大潜力的节点之间取得平衡。

最先进的 MCTS 变体（如 AlphaZero [33], 中使用的变体）采用多项式上置信树（PUCT）算法进行节点选择。通过最大化以下指标来选择动作：

$$a^* = \arg\max_a(Q(s, a) + c \cdot P_\pi(s, a) \cdot \frac{\sqrt{N(s)}}{1 + N(s, a)}) \quad (10)$$

其中， $Q(s, a)$ 是利用项（动作价值均值），而第二项则鼓励探索，其依据是学习得到的策略网络给出的先验概率$P_\pi(s, a)$ 以及固定的探索系数 c。 $N(s)$ 表示节点 s的总访问次数，而 $N(s, a)$ 表示通过动作 a生成的节点 s的子节点的访问次数。

标准 PUCT 公式的一个主要局限在于，它依赖学习得到的策略先验 $P_\pi(s,a)$ 在所有情况下都同样可靠。当发生短期分布偏移时，这一假设在我们的场景中便不再成立。对于罕见且未曾见过的前瞻序列 I_{sub}，学习得到的先验 $P_\pi(s,a)$ 可能会产生误导，但 PUCT 算法仍会继续信任它。这可能会使搜索陷入由不可靠的"直觉"所指向的次优区域。

为了解决这一问题，我们提出了一种新颖的机制，根据当前前瞻序列的稀有程度动态调整探索先验。我们不使用原始策略先验 $P_\pi(s,\ a)$，而是引入动态偏移感知先验 $P_{SA}(a|s)$，以有原则的方式在信任学习到的策略与回归稳健的均匀探索策略之间进行插值。

动态先验定义如下：

$$P_{SA}(a|s) = \alpha(s) \cdot P_\pi(s,a) + (1-\alpha(s)) \cdot \frac{1}{|A_s|} \tag{11}$$

插值权重 $\alpha(s) \in [0,1]$ 是状态 s下局部前瞻子序列 I_k的"熟悉度分数"。$\alpha(s)$ 的计算方式力求简单而有效。该过程分为两个阶段。

离线预计算。 我们处理整个训练数据集 D，为物品类型构建概率查找表。首先，我们通过可视化启发式方法，将连续的物品尺寸（l，w，h）离散化为由 M 种独特物品类型组成的有限词表。然后，通过统计训练数据中每种物品类型出现的频次，计算其经验概率 $P(type_j)$。

在线计算。 在蒙特卡洛树搜索过程中，对于树的 k^{th} 层中的给定节点s，我们按如下方式计算其熟悉度分数 $\alpha(s)$。我们考虑一个固定大小的 w 项目局部窗口（对于图8中由 2 个普通包裹和 1 个大包裹组成的模式，根据经验将其设为 3），该窗口从节点前瞻队列中当前要放置的项目开始，记为 $I_k = (d_{t+k-1}$、d_{t+k}，$\ldots\ldots d_{t+k+w-1})$。假设该窗口中每个项目的出现彼此独立，则联合概率等于它们各自概率的乘积，这些概率从我们的预计算表中获取。熟悉度分数就是该联合概率：

$$\alpha(s) = P(I_k) = \prod_{i=1}^{w} P(Disc(d_{t+i+k-1}) \tag{12}$$

其中，$Disc(d_{t+i+k-1})$ 是将项目 $d_{t+i+k-1}$ 映射到其离散类型的函数。

将这一动态先验代入 PUCT 公式后，我们得到最终的**偏移感知 PUCT**选择规则：

$$a^* = \arg\max_a (Q(s,a) + c \cdot P_{SA}(a|s) \cdot \frac{\sqrt{N(s)}}{1+N(s,a)}) \tag{13}$$

该公式展现出智能、自适应的行为，能够感知局部前瞻序列的稀有程度。

- **当局部前瞻序列较为常见时**：先验项相对近似于 $P_\pi(a|s)$。如图3中的 $k=1$ 步所示，算法相信其已学习的直觉，并高效利用已有知识，其行为类似于标准 PUCT（即动作分布的不确定性较低）。
- **当发生显著的分布偏移时**：先验项近似于均匀分布 $\frac{1}{|A_s|}$。如图3 $k=2$所示，$k=3$ 这一步中，算法识别到其习得的直觉并不可靠，并平稳地退化为稳健、非信息性的探索策略，从而避免搜索被误导性的先验带偏。

该机制使搜索能够动态分配其探索偏好：面对新颖挑战时投入更多资源进行广泛探索，同时在熟悉的情境下高效利用已学习的知识。

4.2.2 扩展。当选择阶段到达一个此前尚未扩展的叶节点 s_L时，便启动扩展阶段以扩展搜索树。为避免为所有可能的动作创建节点所带来的难以承受的成本，我们遵循最先进的方法，并使用学习得到的策略网络 $P_\pi(a|s)$ 来引导这一过程。

对于每个有效动作 $a \in A_{s_L}$，我们在树中添加一个对应的新子节点。连接父节点 s_L 与这个新子节点的边将使用以下统计信息进行初始化：

- Visit Count: $N(s_L,a) \leftarrow 0$
- Action Value: $Q(s_L,a) \leftarrow 0$
- Prior Probability: $P_{SA}(a|s) \leftarrow \alpha(s) \cdot P_\pi(s,a) + (1-\alpha(s)) \cdot \frac{1}{|A_s|}$

执行此步骤后，节点 s_L 现在成为内部节点，搜索树也已成功扩展出新的叶节点，供后续选择和评估。

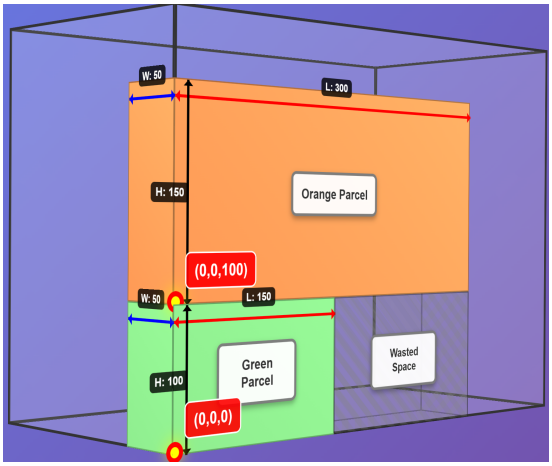


图4：浪费空间计算示意图。

4.2.3 模拟与价值回传。传统蒙特卡洛树搜索中的模拟阶段，会从新扩展的叶节点开始，采用随机或基于启发式的方法进行模拟展开，直至博弈结束，以估算该节点的价值。然而，为了提高效率，AlphaZero [33] 等现代实现通常会用学习得到的评论器直接进行价值估计 V_θ，取代这种成本高昂的模拟展开。在我们的基于模型预测控制的框架中，整个 N 步计划（即搜索树中的一条根到叶路径）的价值也会以类似方式回传。标准做法是将沿路径收集的即时奖励相加，并使用评论器对终端价值进行自举估计：

$$Q(a_t, a_{t+1}, \ldots a_{t+N-1}) = \sum_{k=0}^{N-1} r_{t+k} + V_\theta(s_{t+N}) \tag{14}$$

之所以明确累加短期奖励，而不是仅依赖 $V_\theta(s_t)$，是为了将价值估计建立在

已知前瞻序列的实际结果之上，从而减轻评论器在面对短期分布偏移时可能存在的不可靠性。

然而，在三维装箱中，这种标准回传机制存在一个显著缺点。标准奖励信号 r_{t+k}（通常基于已放置物品的体积）在装箱过程的早期和中期往往不敏感。大多数有效放置都会产生相似的正奖励，无法区分结构上高效的放置与会导致后续物品空间被严重浪费的放置。这使得终端价值估计 V_θ（s_{t+N}）承担了巨大的压力，而它可能不足以稳健地区分结构差异细微但至关重要的状态。

因此，受机器人装箱物理约束的启发，我们提出一种启发式奖励塑形方案，以生成更具辨别力的逐步奖励信号。我们不再仅奖励已放置物品的体积，而是直接惩罚短期轨迹中浪费空间的产生。在真实场景中，由于重力作用，机械臂只能从上往下放置物品。因此，新放置物品下方留下的任何空间，实际上都无法用于后续放置，因而等同于被浪费。如图4所示，将橙色包裹放在绿色包裹上方，会产生后续任何包裹都无法使用的浪费空间。

为衡量这一点，我们引入**浪费空间惩罚**。该惩罚项 $R_p(s_k, a_k)$ 计算为由动作 a_k 放置的物品占地面积正下方空间的归一化体积。例如，在图4中，浪费比例为

$$\begin{aligned} R_p(s_k, a_k) &= (l_o * w_o - l_g * w_g) * (z_o - z_g)/(L * W * H) \\ &= (150 * 50 - 300 * 50) * (100 - 0)/(400 * 200 * 300) \\ &= 3.125\% \end{aligned} \tag{15}$$

其中，l_o，w_o，h_o 是橙色包裹的尺寸，l_g，w_g，h_g 是绿色包裹的尺寸，而 z_o，z_g 分别是橙色包裹和绿色包裹的放置坐标。该体积表示由于重力放置约束而无法再利用的空间。我们新的塑形奖励函数 r' 纳入了这一惩罚项：

$$r'_{t+k} = R_{\text{vol}}(s_k, a_k) - \lambda \cdot R_p(s_k, a_k) \tag{16}$$

其中，R_{vol} 是标准的基于体积的奖励（例如，图4 $\frac{l_o * w_o * h_o}{L * W * H}$ = 中橙色包裹放置的 R_{vol} 计算为 $\frac{300*50*150}{400*200*300} = 9.375$ %），而 λ 是用于调整惩罚项权重的超参数（由于图12，我们将 $\lambda = 0.5$ 设为）。

通过蒙特卡洛树搜索树回传的最终值（记为 Q'）使用了这一增强的奖励信号：

$$Q'(a_t, a_{t+1}, ...a_{t+N-1}) = \sum_{k=0}^{N-1} r'_{t+k} + V_\theta(s_{t+N}) \tag{17}$$

计算出值 Q'后，算法会沿着所选轨迹回传该值，并更新每个访问节点的 $Q(s, a)$值。

5 离线评估

5.1 实验设置

5.1.1 数据集、基线和实验设置。 我们通过挖掘历史日志获取真实包裹流，并将前瞻数量设置为 $N = 4$，因为这涵盖了大多数在线场景（见附录B中的图13）。为评估我们框架的有效性，我们将其与四种启发式方法（*LSAH* [13],*OnlineBPH* [11], *MACS* [14] , *DBL* [18]）和两种深度强化学习方法（*PCT* [48], *GOPT* [44]）进行比较，并将其作为基线。此外，我们实现了两个融入前瞻信息的变体：***PCT*** 前瞻和*PCT-reorder* [47]。为验证我们具体的技术贡献，我们还使用若干替代技术（广度优先搜索 [50], *Random* [29], 实时动态规划 [5], 蒙特卡洛树搜索 [33]）开展了消融研究。此外，我们在**Realistic**和**Virtual**两种设置下测试了框架的有效性。我们按照*PCT* [48], 中的设置建立**Virtual**设置，而**Realistic**设置在动作空间方面包含更多现实世界的约束，例如碰撞避免。数据集、基线和设置的详细信息见附录C。

5.1.2 评估指标。根据 Zhao 等人的研究，我们使用两个核心指标 [47]评估三维装箱算法的性能：空间利用率（“空间利用率”）和装载物品数量（“项目数”）（见附录 C.3）。为评估效率，我们还记录每次放置的平均推理时间，记为“推理时间”。此外，我们定义了两种评估场景：**总体**和**偏移**。**总体**场景在所有测试箱上计算这些指标，而**偏移**场景仅针对到达包裹流呈现显著分布偏移的箱子计算这些指标。

5.2 结果与分析

5.2.1 主要结果。表1报告了两种设置下**总体**和**偏移**场景中的空间利用率、项目数和推理时间。表1显示，恰当考虑前瞻包裹的方法（*PCT-reorder* 和*MPC-PCT*）比基线取得了更好的结果，证明了前瞻信息的有效性。具体而言，在**偏移**场景下，*MPC-PCT* 在两项指标上均比*PCT*提升超过15%，这表明前瞻包裹在分布偏移情况下尤其有用。此外，我们的*MPC-PCT* 取得了显著优于*PCT-reorder* 和*PCT-lookahead* 的结果（在**偏移**场景下，两项指标均提升超过10%），表明我们考虑前瞻包裹的方法在实践中达到了最先进水平。此外，我们还在另一个深度强化学习基线*GOPT* 上测试了框架，并得到了类似的观察结果（*MPC-GOPT* 和*MPC-PCT* 明显优于对应的基线，绿色背景色所示），验证了框架的泛化能力。此外，在**Virtual**设置下也能观察到上述所有结果，表明我们的框架能够泛化到不同的动作空间（具有不同的实际约束）。尽管我们的框架取得了令人期待的结果，但其推理开销约为*PCT-reorder* 的3倍，且超过*PCT* 的10倍。不过，根据第6节中的详细分析，该方法能够满足在线要求。

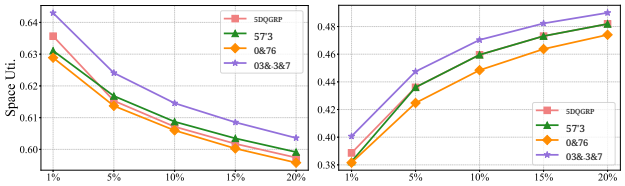
5.2.2 消融研究。 为更好地评估我们的技术贡献，我们通过实现多种替代树搜索技术来解决模型预测控制问题，从而开展消融研究。表 2 展示了现实场景下的结果。首先，在两种情形中，所有树搜索技术的表现均优于基线方法 *PCT*，这验证了将问题建模为模型预测控制问题的有效性

表1：在相同数据集上针对 Realistic 和 Virtual 设置的结果。深度强化学习基线以浅绿色标示。最优的两个结果已突出显示（最佳 结果以粗体和斜体显示，次优结果以粗体显示）。

	Realistic					Virtual				
	总体		偏移		推理时间	总体		偏移		推理时间
	空间利用率	项目数	空间利用率	项目数		空间利用率	项目数	空间利用率	项目数	
基线										
LSAH	0.2396	15.29	0.2570	21.20	$< 1s$	0.2928	19.47	0.3218	23.64	$< 1s$
OnlineBPH	0.2903	18.50	0.3345	27.07	$< 1s$	0.3568	23.28	0.3700	26.83	$< 1s$
MACS	0.3826	24.41	0.3899	31.44	$< 1s$	0.4783	30.68	0.4668	32.86	$< 1s$
DBL	0.3677	23.56	0.4234	33.90	$\sim 5s$	0.4379	28.37	0.4584	32.46	$\sim 5s$
PCT	0.5395	34.14	0.4600	36.37	$< 1s$	0.5519	35.30	0.5227	36.47	$< 1s$
GOPT	0.5211	33.14	0.4461	36.06	$< 1s$	0.5331	34.27	0.5234	36.60	$< 1s$
前瞻信息增强										
PCT 前瞻	0.5286	33.41	0.4788	37.59	$< 1s$	0.5457	34.90	0.5428	37.63	$< 1s$
PCT-reorder	**0.5435**	**34.39**	0.4903	38.36	$\sim 3s$	**0.5638**	**36.01**	0.5651	38.98	$\sim 3s$
Ours										
MPC-PCT	***0.5498***	***34.80***	***0.5258***	**40.61**	$\sim 10s$	***0.5765***	***36.83***	***0.6034***	***41.45***	$\sim 10s$
MPC-GOPT	0.5403	34.31	**0.5165**	**40.10**	$\sim 20s$	0.5527	35.48	**0.5927**	**40.70**	$\sim 20s$

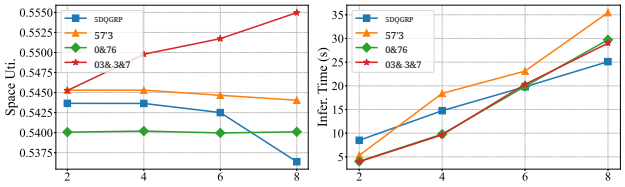
表 2：现实场景下的消融研究结果。除广度优先搜索 外，排名前两位的结果以最佳 突出显示（最佳结果以粗体和斜体显示，次佳结果以粗体显示）。

	Realistic				
	总体		偏移		推理时间
	空间利用率	项目数	空间利用率	项目数	
替代技术					
BFS	0.5513	34.91	0.5403	41.71	$\sim 100s$
Random	0.5436	34.42	0.4955	38.75	$\sim 15s$
RTDP	0.5453	34.50	0.4956	38.54	$\sim 18s$
MCTS	0.5402	34.19	0.4679	37.03	$\sim 10s$
Ours					
~ 不使用 *WSP*	**0.5494**	**34.77**	**0.5218**	**40.32**	$\sim 10s$
~ 不使用 *SAPUCT*	0.5410	34.25	0.4723	37.41	$\sim 10s$
MPC-PCT	***0.5498***	***34.80***	***0.5258***	***40.61***	$\sim 10s$



(a) 顶层 (b) 底层

图5：不同层级下的性能比较。



(a) 前瞻数量 N (b) 前瞻数量 N

图6：可扩展性

框架。其次，尽管暴力搜索（广度优先搜索）取得了最佳性能，但其推理开销约为其他技术的 10 倍，因此并不实用。此外，采用 ϵ-贪心策略的 实时动态规划 在**总体** 场景中的表现比 *Random* 基线方法的纯探索更稳健。然而，采用多项式上置信树的标准蒙特卡洛树搜索过度依赖先验深度强化学习策略，因此性能提升有限。值得注意的是，移除我们提出的探索策略（~ 不使用 *SAPUCT*）会导致性能显著下降，在**偏移** 场景中尤为明显（空间利用率降低 11.3%）。这一结果凸显了以下因素的重要性：

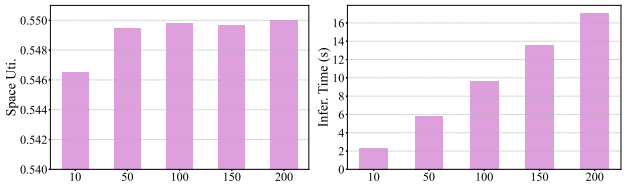
精心设计的策略，在深度强化学习先验与随机探索之间取得平衡。同样，不包含我们空间浪费惩罚的变体（~ 不使用 *WSP*）表现不佳，这证实了所提出项的有效性。

5.2.3 不同水平下的鲁棒性分析。为评估*MPC-PCT*的鲁棒性，我们分析其在以下子集上的性能

这些箱子按最终空间利用率排序，具体包括排名前列和末尾的 {1%, 5%, 10%, 15%, 20%}。图 5 展示了各性能排名对应的空间利用率。我们观察到，*MPC-PCT*（紫色曲线）在所有水平下始终实现最高的空间利用率，展现出其鲁棒性。在各基线中，实时动态规划 （绿色曲线）在中位百分位上的表现优于 *Random* 策略，但在极端情况下则表现较差。这表明，尽管 ϵ-贪心探索策略可能适用于常见情形，但纯粹的探索方法在最极端（最容易或最困难）的场景中可能更有优势。

5.2.4 可扩展性分析。为了分析*MPC-PCT*和其他搜索技术的可扩展性，我们通过改变前瞻时域 N进行实验。结果如图6所示。我们发现，随着 N 增大，只有*MPC-PCT* 的性能持续提升。值得注意的是，纯随机探索策略（*Random*，蓝线）的性能急剧下降

当 N增大时，最终甚至不如基线*PCT*（不使用前瞻）。此外，随着 N增大，蒙特卡洛树搜索 和实时动态规划 的性能基本保持不变。这表明，只有在探索与利用之间取得适当平衡，才能有效利用更大的前瞻时域。此外，*MPC-PCT* 的推理时间随 N 近似线性增长，这与其理论计算复杂度 $O(nN)$一致。



(a) 性能 (b) 效率

图7：试验次数 n 敏感性。

5.2.5 试验次数的影响 n。 图7展示了在前瞻时域 N保持不变时，*MPC-PCT* 的空间利用和推理时间随 n （试验总次数）变化的情况。我们观察到两个主要趋势：（i）随着 n 增加，空间利用率得到提升，并最终趋于稳定；（ii）推理时间随 n线性增加。因此，在实践中，可以在有限的时间预算内进行权衡，以获得足够有效的策略。例如，当 $N = 4$时，设置 $n = 100$ 可在约10秒内取得接近最优的结果。

5.3 案例研究

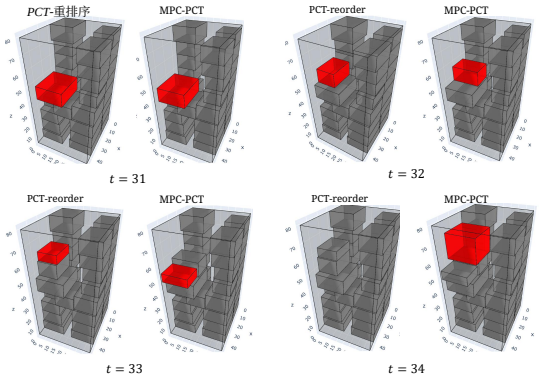


图8：*PCT-reorder* 与*MPC-PCT*的案例研究。红色包裹是步骤 t 中当前待装箱的包裹。如果没有红色包裹，则表示箱子已装满。

我们展示了*PCT-reorder* 和*MPC- PCT* 在相同包裹序列上的装箱结果。为确保比较清晰且公平，我们使用基线*PCT* 策略在步骤 $t = 31$之前装入所有包裹，以确保两种算法从相同的箱体状态开始。如图8所示，从 $t = 31$ 到 $t = 34$的输入序列由两个超大包裹和两个常规尺寸包裹组成。*PCT-reorder* 未能在 $t = 34$ 装入最后一个包裹，因为它本质上仍依赖于学习得到的*PCT* 策略，无法针对这一困难序列充分调整行为。相比之下，*MPC-PCT* 通过提前调整策略，成功装入了最后一个包裹；它旋转了 $t = 32$ 和 $t = 33$ 处的包裹，为 $t = 34$处的物品腾出所需空间。这证明了我们的框架能够解决更简单的前瞻方法（如*PCT-reorder*）无法处理的复杂情况。

6 在线部署

为验证我们框架的实际效用和现实环境中的性能，我们与京东物流合作，将我们的框架部署到其智能装箱系统中（附录C.5）。

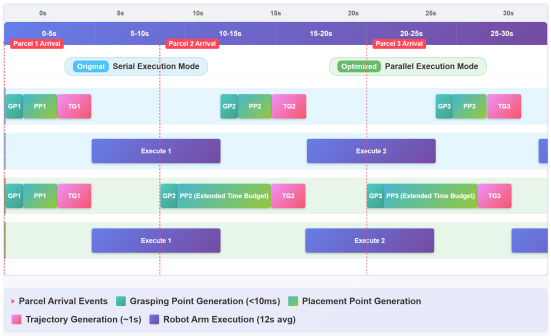


图9：为提高系统效率而进行的工程调整。可将连续两个包裹到达的时间间隔控制在15秒。

6.1 工程调整

为了将*MPC-3D-BP* 框架部署到在线机械臂装箱系统中，我们首先将训练充分的生产基线模型整合到框架中。此外，我们还尝试从系统层面和算法层面提高效率。在系统层面，如图9原始部分所示，现有在线系统将计算（即抓取点生成、放置点生成和轨迹生成）与机械臂执行视为两个独立的顺序过程，因此计算的时间预算十分紧张，以免包裹堆积。然而，如图9优化部分所示，最后一个包裹的机械臂执行与当前包裹的计算可以并行进行。这项优化最多可为放置点生成（即在线3D-BP算法）节省约10秒（即图9中延长的时间预算），其中$T_p = \min(T_a, T_e) - T_t - T_g \approx 12s - 1s - 10ms \approx 11s$表示连续包裹到达的平均时间间隔，$T_a$ 表示机械臂执行的平均耗时，T_e 表示轨迹生成的平均耗时，T_t表示抓取点生成的平均耗时。在算法层面，我们将算法的终止条件从试验次数改为运行时间。此外，由于当前前瞻队列与下一个前瞻队列有 $N - 1$个重叠包裹，因此我们缓存根节点的最佳子节点及其对应子树，以便在下一个包裹到达时初始化蒙特卡洛树搜索树。

6.2 A/B 测试结果

我们在四条并行机械臂生产线上开展在线 A/B 测试，其中两条配备在线三维装箱 3D-BP 深度强化学习模型（生产基线），另外两条则配备我们提出的



图10：采用机械臂的智能三维装箱系统在真实场景中的案例。

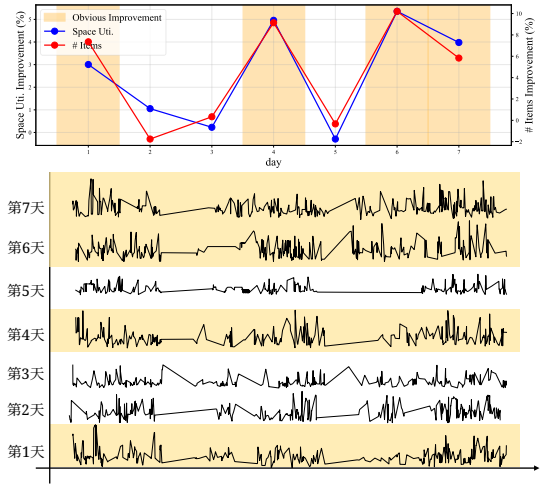


图11：为期7天的在线部署结果。方法。为期一周的在线A/B 测试结果如图11所示。与生产基线相比，我们的方法平均可将空间利用率提高2.6%，并使装箱包裹数量提高4.4%。

为了解我们的模型为何在某些日子（1、4、6 和 7 日，以橙色突出显示）表现更好，我们分析了对应的包裹体积流。图11下方绘出了这一周内包裹的体积（即 $l*w*h$），显示每天都有不同批次的包裹陆续到达（例如上午、下午）。关键观察结果是，性能提升最大的日子恰好也是包裹流波动最剧烈的日子（即标准差约为 966 cm^{3}alue of standard deviation[1]）；而改进幅度最小的日子，包裹体积则更加稳定、一致（即标准差约为 705 cm^3）。总体而言，我们在 3 个分拣中心使用了 20 台机械臂。截至 2025-07 至今，我们已处理超过 100000 个包裹，并装满超过 2000 个箱子。图10展示了真实场景中的装箱案例。

7 相关工作

关于在线三维装箱问题的研究通常分为两类：基于启发式方法和基于深度强化学习的方法。启发式方法将人工设计的策略编纂为规则，例如最大化可用空间 [3, 43]。一些离线启发式方法可以直接应用于在线场景 [3]。代表性方法包括 OnlineBPH [11], , 该方法引入了最大空闲空间（EMS）启发式方法，优先利用较大的空闲区域；以及 WallE [39], , 该方法结合了构建墙面和构建层的策略。相比之下，基于深度强化学习的方法通过与模拟环境交互来学习放置策略。PackMan [39]是最早将深度强化学习应用于在线三维装箱的工作之一。Zhao 等人 [47] 引入了可行性掩码，将任务建模为受约束的马尔可夫决策过程；Zhao 等人[49] 则提出了装箱配置树（PCT），以指导深度强化学习在连续动作空间中的应用。AR2L [27] 设计了一种基于排列的攻击器，以在正常和对抗场景下维持深度强化学习的性能。GOPT [44] 采用 Transformer，以更好地捕捉包裹与箱子之间的空间关系。此外，一些研究通过引入受人类策略启发的重新装箱操作来提升性能 [30, 35, 38, 45]。前瞻信息在许多序贯决策问题中至关重要，例如模型预测控制（MPC）和马尔可夫决策过程（MDP） [19, 22, 23]。在马尔可夫决策过程中，许多算法结合规划与强化学习来利用前瞻信息，通常在训练过程中通过搜索或优化来改进基础强化学习策略 [9, 10, 24, 34]。另一种做法是，基于模型预测控制的方法在每个步骤在线求解规划问题，以提升决策质量 [12]。

8 结论

本文提出了*MPC-3D-BP*，这是一种基于蒙特卡洛树搜索的在线优化框架，用于解决带有前瞻包裹的在线三维装箱问题，旨在应对现实场景中常见的短期分布偏移。为有效平衡探索与利用，我们提出了一种新颖的探索策略，能够识别前瞻队列中可能不熟悉的包裹。此外，我们在奖励函数中引入逐步惩罚，以惩罚浪费空间，从而将每次放置的长期影响纳入考量。我们在来自现实和虚拟环境的数据集上进行了大量实验。结果表明，*MPC-3D-BP* 优于不具备前瞻能力的基线，并且与不同的底层深度强化学习模型集成时，始终表现出色。此外，我们将该框架与其他问题表述和模型预测控制求解技术进行了比较，结果显示，我们的方法性能更优，且推理开销仍处于在线应用可接受的范围内。此外，在京东物流的实际机器人装箱系统上开展为期一周的在线 A/B 测试，证实*MPC-3D-BP* 优于生产基线，验证了其在现实场景中的有效性。作为首批研究带有前瞻包裹的在线三维装箱问题的工作之一，我们希望本研究能够助力智能装箱系统的发展，并为其他涉及有限前瞻时域的类似问题提供启示。

[1]我们以 30 个包裹为一个窗口计算标准差，再对所有窗口的标准差取平均，得到最终结果

参考文献

[1] 2025. 亚马逊。https://www.amazon.com 访问日期：2025年7月18日。[2] 2025. 京东物流。https://www.jdl.com 访问日期：2025年7月18日。[3] Sara Ali、António Galrão Ramos、Maria Antónia Carravilla 和 José Fernando Oliveira。2022年。在线三维装箱问题：离线

和在线求解方法综述。《计算机与工业工程》168（2022年），108122。[4] Mauro Maria Baldi、Daniele Manerba、Guido Perboli 和 Roberto Tadei。2019年。面向最后一公里物流包裹配送的广义装箱问题。《欧洲运筹学杂志》274，3（2019年），990–999。[5] Andrew G Barto、Steven J Bradtke 和 Satinder P Singh。1995年。运用实时动态规划学习行动。《人工智能》72，1-2（1995年），81–138。[6] Arnab Bhattacharyya、Sutanu Gayen、Kuldeep S Meel、Dimitrios Myrisiotis、Aduri Pavan 和 NV Vinodchandran。2022年。全变差距离的近似。《arXiv》预印本 arXiv:2206.07209（2022年）。[7] Bernard Chazelle。1983年。底左装箱启发式算法：一种高效实现。《IEEE计算机汇刊》（1983年）。[8] Teodor Gabriel Crainic、Guido Perboli 和 Roberto Tadei。2008年。三维装箱的极点启发式方法。《INFORMS计算杂志》（2008年）。[9] Yonathan Efroni、Gal Dalal、Bruno Scherrer 和 Shie Mannor。2019年。如何在强化学习中结合树搜索方法。载于《AAAI人工智能大会论文集》，第33卷。3494–3501。[10] Yonathan Efroni、Mohammad Ghavamzadeh 和 Shie Mannor。2020年。利用前瞻策略进行在线规划。《神经信息处理系统进展》33（2020年），14024–14033。[11] Chi Trung Ha、Trung Thanh Nguyen、La m Thu Bui 和 Ran Wang。2017年。动态环境和物理互联网中的三维集装箱装载问题在线装箱启发式方法。载于《进化计算应用》。[12] Nicklas Hansen、Xiaolong Wang 和 Hao Su。2022年。用于模型预测控制的时序差分学习。arXiv预印本 arXiv:2203.04955（2022年）。[13] Haoyuan Hu、Xiaodong Zhang、Xiaowei Yan、Longfei Wang 和 Yinghui Xu。2017年。采用深度强化学习方法解决一种新的三维装箱问题。arXiv预印本 arXiv:1708.05930（2017年）。[14] Ruizhen Hu、Juzhan Xu、Bin Chen、Minglun Gong、Hao Zhang 和 Hui Huang。2020年。TAP-Net：利用强化学习实现运输与装箱。《ACM图形学汇刊》（2020年）。[15] Meng-Yu Huang、Ling-Ying Huang、Yu-Xing Zhong、Hui-Wen Yang、Xiao-Meng Chen、Wei Huo、Jia-Zheng Wang、Fan Zhang、Bo Bai 和 Ling Shi。2025年。混合整数线性规划加速：从单纯形法初始化和基于学习的分支定界角度出发的综述。《中国运筹学会学报》13，1（2025年），1–55。[16] Peiyuan Jiang、Daji Ergu、Fangyao Liu、Ying Cai 和 Bo Ma。2022年。YOLO算法发展的综述。《计算机科学 Procedia》199（2022年），1066–1073。[17] Korhan Karabulut 和 Mustafa Murat İnceoğlu。2004年。采用最深底左填充法的三维装箱混合遗传算法。载于《信息系统进展国际会议》。施普林格，441–450。[18] Korhan Karabulut 和 Mustafa Murat Inceoglu。2004年。采用最深底左填充法的三维装箱混合遗传算法。载于《信息系统进展》。[19] Basil Kouvaritakis 和 Mark Cannon。2016年。模型预测控制。瑞士：施普林格国际出版社 38，13-56（2016年），7。[20] Silvano Martello、David Pisinger 和 Daniele Vigo。2000年。三维装箱问题。《运筹学》48，2（2000年），256–267。[21] Yutaka Matsuo、Yann LeCun、Maneesh Sahani、Doina Precup、David Silver、Masashi Sugiyama、Eiji Uchibe 和 Jun Morimoto。2022年。深度学习、强化学习与世界模型。《神经网络》152（2022年），267–275。[22] Nadav Merlis。2024年。利用前瞻信息进行强化学习。arXiv预印本 arXiv:2406.02258（2024年）。[23] Nadav Merlis、Dorian Baudry 和 Vianney Perchet。2024年。奖励前瞻在强化学习中的价值。《神经信息处理系统进展》37（2024年），83627–83664。[24] Thomas M Moerland、Anna Deichler、Simone Baldi、Joost Broekens 和 Catholijn M Jonker。2020年。思考不可过快，也不可过慢：规划与强化学习之间的计算权衡。arXiv预印本 arXiv:2005.07404（2020年）。[25] Ankush Ojha、Marichi Agarwal、Aniruddha Singhal、Chayan Sarkar、Supratim Ghosh 和 Rajesh Sinha。2021年。自动化分拣中心在线三维装箱的通用算法与框架。载于2021年第七届印度控制会议（ICC）。IEEE，135–140。[26] Sascha Julian Oks、Max Jalowski、Michael Lechner、Stefan Mirschberger、Marion Merklein、Birgit Vogel-Heuser 和 Kathrin M Möslein。2024年。工业4.0背景下的信息物理系统：综述、分类与展望。《信息系统前沿》26，5（2024年），1731–1772。[27] Yuxin Pan、Yize Chen 和 Fangzhen Lin。2023年。面向在线三维装箱的可调鲁棒强化学习。《神经信息处理系统进展》（2023年）。[28] Kishan Panaganti、Zaiyan Xu、Dileep Kalathil 和 Mohammad Ghavamzadeh。2022年。利用离线数据进行鲁棒强化学习。《神经信息处理系统进展》35（2022年），32211–32224。

[29] Robin Pemantle 和 Yuval Peres。1995年。树上随机环境中的临界随机游走。《概率年刊》（1995年），105–140。[30] Aaron Valero Puche 和 Sukhan Lee。2022年。带缓冲区的在线三维装箱强化学习解决方案。载于 IEEE/RSJ智能机器人与系统国际会议。[31] Research and Markets。2025年。物流市场——趋势、创新与2025-2033年预测。https://www.researchandmarkets.com/reports/6101938 访问于2025年6月。[32] Hong SHAO、Daxiong XIE 和 Yihua HUANG。2021年。自动驾驶智能传感技术综述。《中兴通讯》19卷3期（2021年），56。[33] David Silver、Thomas Hubert、Julian Schrittwieser、Ioannis Antonoglou、Matthew Lai、Arthur Guez、Marc Lanctot、Laurent Sifre、Dharshan Kumaran、Thore Graepel 等。2018年。通过自我对弈掌握国际象棋、将棋和围棋的通用强化学习算法。《科学》362卷6419期（2018年），1140–1144。[34] David Silver、Julian Schrittwieser、Karen Simonyan、Ioannis Antonoglou、Aja Huang、Arthur Guez、Thomas Hubert、Lucas Baker、Matthew Lai、Adrian Bolton、Yutian Chen、Timothy P. Lillicrap、Fan Hui、Laurent Sifre、George van den Driess- che、Thore Graepel 和 Demis Hassabis。2017年。无需人类知识掌握围棋。《自然》（2017年）。[35] Shuai Song、Shuo Yang、Ran Song、Shilei Chu、Wei Zhang 等。2023年。迈向在线三维装箱：通过深度强化学习学习装箱与拆箱之间的协同作用。载于机器人学习会议。机器学习研究论文集，1136–1145。[36] Balakumar Sundaralingam、Siva Kumar Sastry Hari、Adam Fishman、Caelan Garrett、Karl Van Wyk、Valts Blukis、Alexander Millane、Helen Oleynikova、Ankur Handa、Fabio Ramos 等。2023年。CuRobo：并行化的无碰撞机器人运动生成。载于2023年IEEE机器人与自动化国际会议（ICRA）。IEEE，8112–8119。[37] Maciej Świechowski、Konrad Godlewski、Bartosz Sawicki 和 Jacek Mańdziuk。2023年。蒙特卡洛树搜索：近期改进与应用综述。《人工智能评论》56卷3期（2023年），2497–2562。[38] YP Tsang、DY Mo、KT Chung 和 CKM Lee。2025年。一种采用箱体替换策略的在线并发三维装箱优化深度强化学习方法。《工业中的计算机》164（2025年），104202。[39] Richa Verma、Aniruddha Singhal、Harshad Khadilkar、Ansuma Basumatary、Siddharth Nayak、Harsh Vardhan Singh、Swagat Kumar 和 Rajesh Sinha。2020年。一种用于在线三维装箱的广义强化学习算法。arXiv预印本 arXiv:2007.00463（2020年）。[40] Richa Verma、Aniruddha Singhal、Harshad Khadilkar、Ansuma Basumatary、Siddharth Nayak、Harsh Vardhan Singh、Swagat Kumar 和 Rajesh Sinha。2020年。一种用于在线三维装箱的广义强化学习算法。arXiv预印本 arXiv:2007.00463（2020年）。[41] Fan Wang 和 Kris Hauser。2019年。使用机器人操作臂稳定装箱非凸三维物体。载于机器人与自动化国际会议。[42] Fan Wang 和 Kris Hauser。2021年。不规则及新型三维物体的机器人密集装箱。《IEEE机器人学汇刊》（2021年）。[43] Wenjie Wu、Changjun Fan、Jincai Huang、Zhong Liu 和 Junchi Yan。2023年。多维装箱问题的机器学习：文献综述与实证评估。arXiv预印本 arXiv:2312.08103（2023年）。[44] Heng Xiong、Changrong Guo、Jian Peng、Kai Ding、Wenjie Chen、Xuchong Qiu、Long Bai 和 Jianfeng Xu。2024年。GOPT：通过基于Transformer的深度强化学习实现可泛化的在线三维装箱。《IEEE机器人学与自动化快报》（2024年）。[45] Shuo Yang、Shuai Song、Shilei Chu、Ran Song、Jiyu Cheng、Yibin Li 和 Wei Zhang。2023年。用于在线三维装箱的启发式集成深度强化学习。《IEEE自动化科学与工程汇刊》（2023年）。[46] Junpeng YU 和 Yiyu CHEN。2023年。一种用于自动雷达检测的实用强化学习框架。《中兴通讯》21卷3期（2023年），22。[47] Hang Zhao、Qijin She、Chenyang Zhu、Yin Yang 和 Kai Xu。2021年。采用约束深度强化学习的在线三维装箱。载于AAAI人工智能大会论文集，第35卷。741–749。[48] Hang Zhao、Yang Yu 和 Kai Xu。2022年。在装箱配置树上学习高效的在线三维装箱。载于国际学习表征会议。[49] Hang Zhao、Chenyang Zhu、Xin Xu、Hui Huang 和 Kai Xu。2022年。为在线三维装箱学习实际可行的策略。《中国科学：信息科学》（2022年）。[50] Rong Zhou 和 Eric A Hansen。2006年。广度优先启发式搜索。《人工智能》170卷4-5期（2006年），385–408。[51] Qianwen Zhu、Xihan Li、Zihan Zhang、Zhixing Luo、Xialiang Tong、Mingxuan Yuan 和 Jia Zeng。2021年。学习装箱：一种用于大规模三维装箱问题的数据驱动树搜索算法。载于第30届ACM信息与知识管理国际会议论文集。4393–4402。[52] Zhuo Zou、Qing Chen、Ismail Uysal 和 Lirong Zheng。2014年。射频识别赋能的无线传感在智能食品物流中的应用。《英国皇家学会哲学汇刊A辑：数学、物理与工程科学》372

A 证明细节

为了量化传统策略 π 在非平稳环境中的次优性，我们将推导性能差距的上界。该上界将表明，性能损失与长期平均分布 D 和当前短期分布 D_x 之间的统计距离直接相关。我们先介绍一个关键引理，它给出了当底层环境分布从 D 偏移至 D_x 时，任意固定策略价值变化的上界。

引理 1。设 π 为任意固定策略。设 $V_D^\pi(s)$ 和 $V_{D_x}^\pi(s)$ 分别为在由分布 D 和 D_x 支配的环境中执行 π 时的状态价值函数。假设奖励在 $[0, R_{max}]$ 中有界（在三维装箱场景中，与空间利用率相对应的奖励属于 $[0, 1)$），则这些价值函数之间的最大差异有如下界：

$$\max_s |V_D^\pi(s) - V_{D_x}^\pi(s)| \leq \frac{\gamma R_{max}}{(1-\gamma)^2} \cdot D_{TV}(D, D_x) \tag{18}$$

其中，$D_{TV}(D, D_x)$ 是两个分布之间的全变差距离。

证明。令 $\delta(s) = V_D^\pi(s) - V_{D_x}^\pi(s)$。根据固定策略的贝尔曼评估方程 π：

$$\begin{aligned}\delta(s) = &\left(R(s,\pi(s)) + \gamma \mathbb{E}_{d'\sim D}\left[V_D^\pi(s')\right]\right) - \\ &\left(R(s,\pi(s)) + \gamma \mathbb{E}_{d'\sim D}\left[V_{D_x}^\pi(s')\right]\right)\end{aligned} \tag{19}$$

$$\delta(s) = \gamma\left(\mathbb{E}_{d'\sim D}\left[V_D^\pi(s')\right] - \mathbb{E}_{d'\sim D}\left[V_{D_x}^\pi(s')\right]\right) \tag{20}$$

我们在绝对值内加上并减去项 $\gamma \mathbb{E}_{d'\sim D}\left[V_D^\pi(s')\right]$：

$$\begin{aligned}|\delta(s)| = \gamma\Big|&\left(\mathbb{E}_{d'\sim D}\left[V_D^\pi(s')\right] - \mathbb{E}_{d'\sim D_x}\left[V_D^\pi(s')\right]\right) \\ &+ \left(\mathbb{E}_{d'\sim D_x}\left[V_D^\pi(s')\right] - \mathbb{E}_{d'\sim D_x}\left[V_{D_x}^\pi(s')\right]\right)\Big|\end{aligned} \tag{21}$$

$$\begin{aligned}|\delta(s)| \leq \gamma &\left|\mathbb{E}_{d'\sim D}\left[V_D^\pi(s')\right] - \mathbb{E}_{d'\sim D_x}\left[V_D^\pi(s')\right]\right| + \\ &\gamma\left|\mathbb{E}_{d'\sim D_x}\left[\delta(s')\right]\right|\end{aligned} \tag{22}$$

任意策略的价值都由 $V_{max} = \frac{R_{max}}{1-\gamma}$ 界定。有界函数 f 在两个分布下的期望之差由 $2||f||_\infty D_{TV}$ 界定。将此结论应用于第一项：

$$|\delta(s)| \leq \gamma\left(2 \cdot \frac{R_{max}}{1-\gamma} \cdot D_{TV}(D, D_x)\right) + \gamma \mathbb{E}_{d'\sim D_x}\left[|\delta(s')|\right] \tag{23}$$

令 $||\delta||_\infty = \max_s |\delta(s)|$。对所有状态 s 取最大值：

$$\begin{aligned}||\delta||_\infty &\leq \frac{2\gamma R_{max}}{1-\gamma} D_{TV}(D, D_x) + \gamma ||\delta||_\infty \\ (1-\gamma)||\delta||_\infty &\leq \frac{2\gamma R_{max}}{1-\gamma} D_{TV}(D, D_x)\end{aligned} \tag{24}$$

由此得到以下界（由于界定技术的不同，常数略有差异，但原理仍然成立）：

$$||\delta||_\infty \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} D_{TV}(D, D_x) \tag{25}$$

该引理形式化地说明，分布发生微小变化时，任意给定策略的价值也会产生相应幅度的微小变化。借助该引理，我们现在可以证明主定理。

定理 2（性能差距界）。在由 D_x 支配的环境中，传统策略 π 的性能差距由统计距离 $D_{TV}(D, D_x)$ 的函数界定。

证明。状态 s 下的性能差距，定义为 D_x 环境中的真实最优值与我们在同一环境中采用传统策略 π 所达到的价值之差：

$$\text{Gap}(s) = V_{D_x}^*(s) - V_{D_x}^\pi(s) \tag{26}$$

我们知道，策略 π 对平稳分布 Q 是最优的。因此，它的价值函数在该环境中是最优的：$V_D^\pi(s) = V_D^*(s)$。

我们可以通过加上再减去平稳世界中最优策略的价值 $V_D^*(s)$，来改写这一差距：

$$\text{Gap}(s) = \left(V_{D_x}^*(s) - V_D^*(s)\right) + \left(V_D^*(s) - V_{D_x}^\pi(s)\right) \tag{27}$$

代入 $V_D^*(s) = V_D^\pi(s)$：

$$\begin{aligned}\text{Gap}(s) = &\underbrace{\left(V_{D_x}^*(s) - V_D^*(s)\right)}_{\text{Term 1: Inherent Optimality Difference}} + \\ &\underbrace{\left(V_D^\pi(s) - V_{D_x}^\pi(s)\right)}_{\text{T性能2 i策略性能退化 D d i}}\end{aligned} \tag{28}$$

这一分解很有启发性。它表明，总体性能差距来自两个方面：

(1) 术语 1：新环境 $V_{D_x}^*$ 与旧环境 V_D^* 中可能达到的最佳性能之间的差异。这反映了环境变化本身如何影响最优可实现价值。

（2）**术语 2**：我们的特定固定策略 π 的性能下降

利用**引理 1**，我们可以直接界定术语 2：

$$\left|V_D^\pi(s) - V_{D_x}^\pi(s)\right| \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} D_{TV}(D, D_x) \tag{29}$$

基于贝尔曼最优算子作为压缩映射的性质，采用类似但更为复杂的证明可以表明，术语 1 同样受到分布距离的界定：

$$\left|V_{D_x}^*(s) - V_D^*(s)\right| \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} D_{TV}(D, D_x) \tag{30}$$

应用三角不等式可得，总体性能差距因此受到如下界定：

$$|\text{Gap}(s)| \leq |V_{D_x}^*(s) - V_D^*(s)| + |V_D^\pi(s) - V_{D_x}^\pi(s)| \leq C \cdot D_{TV}(D, D_x) \tag{31}$$

其中 $C = \frac{4\gamma R_{max}}{(1-\gamma)^2}$ 是一个常数。

此外，该证明同样适用于可在有限时域内结束的马尔可夫决策过程，其中 γ 通常设为 1。尽管看起来当 $\gamma = 1$ 时，常数 C 会趋于 ∞，但实际上，如果博弈最多可在 T 个有限步内结束，则任意策略的价值仍有界于 $T \cdot R_{\max}$。其他证明细节保持不变。

B 数据证据与其他实验结果

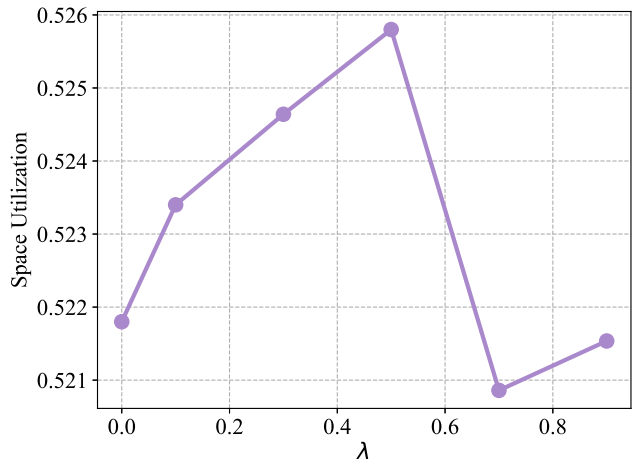


图12：Shift 场景中空间浪费惩罚 λ 参数的敏感性研究。

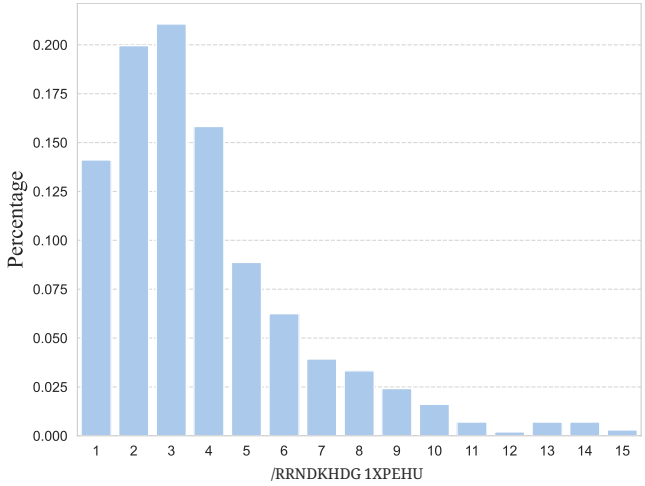


图13：信息物理系统中的常见前瞻数量 N。

C 实现细节

C.1 数据集预处理与统计

该**离线训练数据集**是从京东物流的机械臂历史日志中挖掘得到的。我们收集包裹的长、宽、高（详细统计见表 3）以及其抓取点，作为训练输入。所有数据均在一致的真实生产条件下获取。训练期间是否将抓取点纳入输入特征，取决于具体的实验设置。

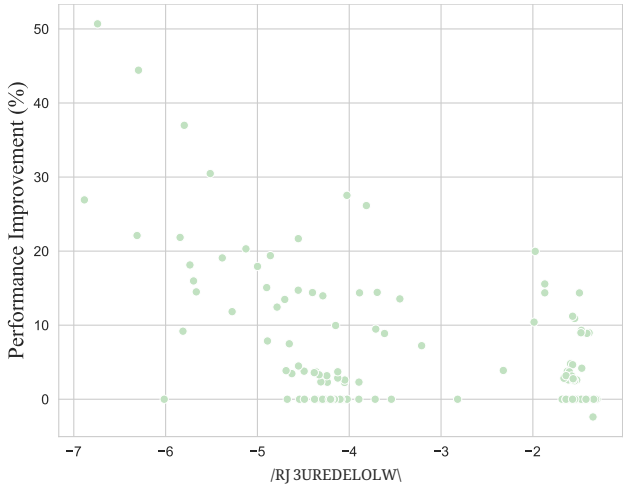


图14：在线包裹批次相对于离线训练数据集计算的对数概率，与广度优先搜索和 PCT 方法之间相应的性能差距的关系。

京东物流通过挖掘历史机械臂日志创建了**前瞻包裹数据集**。原始日志包含每个包裹的详细记录，包括其尺寸（长、宽和高）、到达时间以及相应的机械臂动作（例如抓取点和放置坐标）。我们提取这些关键信息，然后按到达时间对记录排序，以重建原始包裹流。为了生成每条记录的前瞻信息，我们首先设置前瞻时域 N。然后，对于每个放置决策，从排序后的数据流中选取接下来的 $N-1$ 个包裹，组成其前瞻队列。前瞻包裹数据集共包含 882 条记录。

表 3：离线训练数据集的数据集统计。

时间跨度	记录数	长度范围	宽度范围	高度范围	箱体尺寸
2025.05-2025.06	5652	16 ~ 58	12 ~ 40	4 ~ 48	(94,80,160)

C.2 基线方法

启发式方法是在线三维装箱问题中的重要基线。

• *LSAH* [13]：根据 Hu 等人的说法，这是一种启发式算法，会遍历当前物品的所有最大空闲空间和 2 种朝向，并选择放入物品后表面积最小的最大空闲空间和朝向。• *OnlineBPH* [11]：根据 Ha 等人的说法，这是一种启发式算法，会在允许的最大空闲空间和朝向中找到第一个合适的放置位置。所有最大空闲空间都按深度-底部-左侧顺序排序。• *MACS* [14]：根据 Hu 等人的说法，这是一种启发式算法，按照最大化剩余空闲空间策略计算得分最高的位置。候选位置由

每个最大空闲空间在两种可能朝向下的四个角生成。

• *DBL* [17]：根据 Karabulut 和 İnceoğlu 的说法，这是一种启发式算法，它依据最深底部左侧填充（DBLF）策略选择位置。该策略优先选择最深的可用位置，其次是最低的位置，最后是最靠左的位置。

此外，我们还实现了两种考虑前瞻信息的方法：

• *PCT-lookahead*：一种将前瞻包裹编码为状态组成部分的深度强化学习方法。• *PCT-reorder*：遵循 Zhao 等人的工作，我们实现了一种重排序算法，根据即将到来的包裹来调整当前的放置决策。该方法通过虚拟重排序列，优先放置前瞻项目，同时遵守一项关键的顺序依赖约束：较早到达的包裹不能占用只有在放置较晚到达的包裹后才腾出的空间。

此外，我们选择了两种深度强化学习方法作为骨干，其中*PCT*是我们的主要骨干，我们还测试了我们的方法在*GOPT*上的有效性。

• *PCT* 是用于在线三维装箱问题的最先进深度强化学习模型。它通过新颖的基于树的表示，巧妙地利用图注意力网络（GAT）[46] 将启发式规则与深度强化学习相结合，以解决在线三维装箱问题中动作空间过于复杂这一核心挑战。

• *GOPT* 是用于在线三维装箱问题的最先进深度强化学习模型。它通过独特的“生成器-Transformer”架构，在三维装箱任务中实现了出色的性能和强大的泛化能力，解决了传统深度强化学习方法往往难以适应不同环境配置的问题。

为验证我们所做适配的有效性，我们还实现了许多其他通用技术，以解决相同的模型预测控制问题。

• 广度优先搜索 [50]：我们实现了一种暴力搜索，遍历所有可能的根到叶路径，以识别并选出最优路径。
• *Random* [29]：我们在树上执行多次随机游走；每次遍历路径时，算法都从根节点开始，并在每一步随机选择一个子节点进行访问，直至到达叶节点。完成预先设定次数的搜索后，我们从所有探索过的轨迹中选出找到的最优路径。• *RTDP* [5]：根据 Barto 等人的方法，我们在在线

三维装箱问题。• 蒙特卡洛树搜索 [33]：根据Silver等人，我们将AlphaZero中使用的蒙特卡洛树搜索框架直接应用于三维装箱环境。

C.3 指标

空间利用率（空间利用率）按如下方式计算：

$$\text{space_utilization} = \sum_{i=1}^{M} \frac{l_i * w_i * h_i}{L * W * H} \tag{32}$$

其中，M表示箱子中的包裹数量，L、W H分别表示箱子的长、宽和高，l_i、w_i、h_i分别表示 i^{th} 包裹的长、宽和高。

包裹数量（项目数）的含义由其字面意思直接定义。

C.4 实验设置

训练。在离线训练阶段，我们训练两个模型：*PCT*和*GOPT*。对于*PCT*训练，我们使用空最大空间（EMS）方法生成动作空间，并使用单个图注意力层对状态进行编码。两个深度模型的超参数均通过网格搜索进行微调。关键架构参数包括输入嵌入维度为64、隐藏层维度为128、内部节点长度为120以及叶节点长度为80。此外，学习过程的超参数如下：学习率设为 1×10^{-6}，actor损失和评论家损失的系数均设为1.0。对于*GOPT*训练，我们使用EMS作为动作空间，并使用三个Transformer层对状态进行编码。当模型选择一个EMS作为输出时，会将包裹放置在所选EMS内最深、最低且最靠左的可行位置。关键架构参数包括输入嵌入维度为128、EMS长度为80。此外，学习过程的超参数如下：学习率初始设为 7×10^{-5}，并线性递减；策略更新和评论家损失的系数分别设为0.96和0.5。

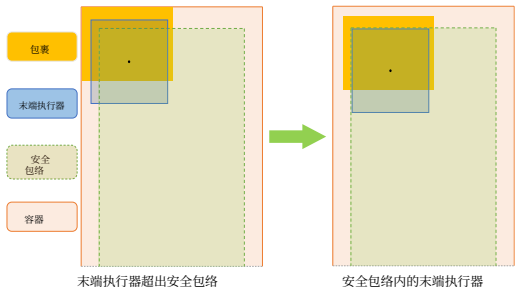


图15：位置安全偏移方法的俯视图。

测试。我们在同一数据集的两种设置下进行测试：**Realistic** 和 **Virtual**。

• **Virtual**。为全面评估所有方法在包裹流数据集上的性能，我们采用滑动窗口技术生成短期测试批次，以模拟在线场景。具体而言，我们将窗口大小设为 100，即单个箱子历史装载项目的最大数量，并以步长 1 生成这些批次。为评估某种方法，我们在每个批次上运行该方法，然后对所有生成的装载包裹的箱子的指标取平均值，计算最终性能。此外，我们遵循设置 1（见*PCT* [48]）。仅允许两种水平方向，并在放置当前包裹时检查箱体状态的稳定性。

• **Realistic**。采用与**Virtual** 设置相同的评估方案的基础上，我们进一步考虑了机械臂作业中的实际约束。在**Realistic** 设置中，数据包含机械臂抓取包裹时的抓取点信息，用于确保现实场景中的操作安全。鉴于现实生产

环境的特点与限制，机械臂的末端执行器在进行物品抓取与放置操作时，可能会与笼壁或先前放置的包裹发生碰撞。如图15所示，为降低这一风险，我们在笼内划定了一个称为安全包络的安全区域，末端执行器只能在绿色区域内作业。根据上游提供的抓取点，我们计算当前放置位置是否会导致末端执行器超出允许范围。值得注意的是，当包裹尺寸很小时，规划的位置可能导致末端执行器超出安全区域，从而使许多放置位置无法使用。为解决这一问题，我们设计了一种位置安全偏移方法，在满足安全要求的同时，计算偏移量最小的放置位置（图15）。这种偏移方法适用于所有动作生成方案以及启发式方法。我们在所有现实场景设置中均采用该方法，以确保所用方法在现实场景中可靠运行。

为确保比较公平，所有深度强化学习模型均在一台 Linux 服务器上训练，该服务器配备 92 个 CPU 核心（Intel® Xeon® Platinum 8468V）、922 GB 内存和一块 NVIDIA GeForce RTX 4090D GPU。随后，所有方法均在一台配备 32 个 CPU 核心和 120 GB 内存的 Linux 服务器上进行测试。

C.5 部署系统



图16：用于将包裹信息上传至信息物理系统的扫描仪。

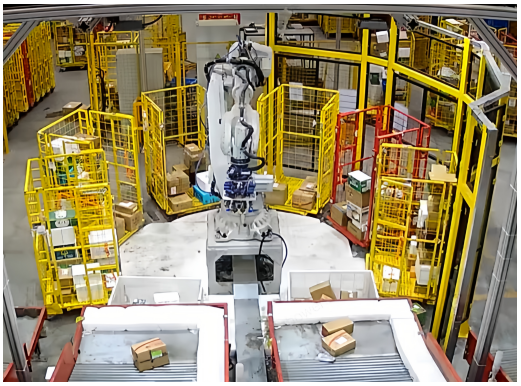


图17：使用机械臂进行在线三维装箱的概览。

京东物流的智能在线三维装箱系统每天至少可由每条机械臂处理 1500 个包裹，劳动力成本的节省相当于每条机械臂减少约一名全职员工。该系统由用于获取包裹信息的扫描仪和用于分拣的输送系统组成。如图16所示，包裹经过扫描仪时，扫描仪可读取其任意一面的条形码。随后，这些信息会上传至信息物理系统，由其确定包裹的目的地路线。当包裹到达传送带末端时（图17），俯视摄像机拍摄图像并将其传输至系统。我们的系统使用基于 YOLO 的模型 [16] 检测图像中的包裹，并计算合适的抓取点。此外，服务器将当前箱体状态输入基于在线深度强化学习的三维装箱模型，以确定最佳放置位置。最后，中央服务器使用 CuRobo [36], 计算从包裹初始位置到目标放置位置的无碰撞轨迹，同时避开箱壁和其他已装箱包裹。

收稿日期：2007年2月20日；修订日期：2009年3月12日；接受日期：2009年6月5日