# 融合多臂老虎机、主动学习与分布式计算，实现可扩展优化

Foo Hui-Mean 和 Yuan-chin Ivan Chang，中央研究院统计科学研究所，台湾

2026年1月5日

arXiv:2601.00615v1 [stat.CO] 2 Jan 2026

**摘要**

现代科学与工程领域中的优化问题，通常依赖开销高昂的黑箱评估，例如物理模拟或深度学习流水线中的评估；在这些场景中，梯度信息往往不可用或不可靠。在此类情况下，由于计算成本过高且可扩展性差，传统优化方法很快便难以实际应用。我们提出 ALMAB-DC，这是一种统一且模块化的可扩展黑盒优化框架，融合了主动学习、多臂老虎机和分布式计算，并可选择使用图形处理器加速。该框架利用代理建模和信息论采集函数来指导信息丰富的样本选择，同时由基于老虎机的控制器以符合统计原理的方式，动态分配候选评估所需的计算资源。这些决策在分布式多智能体系统中异步执行，从而实现高吞吐量的并行评估。我们为基于置信上界和基于汤普森采样的变体建立了理论遗憾界，并基于 Amdahl 定律和古斯塔夫森定律开展可扩展性分析。在合成基准测试、强化学习任务和科学模拟问题上的实证结果表明，ALMAB-DC 的表现始终优于最先进的黑箱优化器。ALMAB-DC 在设计上具有模块化、不确定性感知和可扩展等特点，尤其适用于高维、资源密集型优化挑战。

## 1 引言

优化在工程设计、机器学习等领域的科学计算问题中发挥着关键作用。许多现代应用涉及计算成本高昂、缺少闭式梯度且往往会产生噪声或基于模拟的输出的黑箱函数。典型示例包括深度神经网络的超参数调优、强化学习以及计算流体动力学模拟。在这些问题中，由于计算需求过高，传统的基于梯度的方法或穷举搜索方法往往难以实际应用（Krizhevsky and Hinton，2009年；Kennedy and O'Hagan，2001年）。

为应对这些挑战，我们提出了**ALMAB-DC**，这是一个统一框架，将主动学习、多臂老虎机算法和分布式计算的原理相结合，并将优化视为不确定性下的序贯决策过程，通过统计代理模型引导决策，优先选择信息性查询，并在探索与利用之间取得平衡。ALMAB-DC 旨在

通过结合自适应采样与可扩展基础设施，在预算约束下高效探索高维空间。该框架提供了一种通用优化策略，连接统计设计、在线决策理论与大规模并行，并支持多种采集机制，集成多智能体执行，还可选择使用 GPU 加速以提升计算效率。凭借其通用性，该框架可应用于广泛领域，包括超参数优化、基于仿真的科学发现，以及资源受限的实验设计。

近期已有多项研究探索了本研究所涉及组件的部分交集，但很少有研究提供统一的视角。Auer 等人（2002a）研究了多臂老虎机中的分布式在线学习，重点关注延迟反馈，但未纳入主动学习要素。Bachman 等人（2017年）提出了一种基于假设空间的方法，用于预算受限主动学习，但未涉及可扩展性，也未整合老虎机框架。Landgren 等人（2016年）考察了多智能体老虎机场景中的去中心化探索，而 Li 等人（2019年）则聚焦于分布式环境中的主动学习，未采用任何基于老虎机的优化策略。Liu 等人（2021年）引入了主动奖励机制，以提高强化学习效率，实际上将主动学习与奖励塑形相结合。Wang 和 Liu Wang 和 Liu（2020年）提出了主动学习与老虎机的统一视角，但未探讨可扩展性或分布式实现问题。Yu 等人（2022年）通过对等方选择研究了联邦主动学习，Li 等人（2018年）则将老虎机算法应用于分布式超参数优化。

相比之下，本文提出的 ALMAB-DC 框架将主动学习、多臂老虎机和分布式计算这三种范式整合到一个可扩展的优化架构中。它以严谨的方式分解主动学习和老虎机两个维度上的遗憾，并通过分布式智能体展示了出色的扩展能力。以往工作中尚未体现出这种程度的概念整合与理论严谨性。ALMAB-DC 的潜在价值在于其通用性和可扩展性：它可适用于多种领域，包括基于仿真的优化、超参数调优和自主科学发现，从而显著提升资源高效的大规模决策能力。

本文其余部分将详细介绍构成 ALMAB-DC 流水线的系统架构、采样策略和协调机制，并对相关工作进行比较评估和讨论。

# 2 方法论

不同于采用基于批次的方法，ALMAB-DC 将大规模黑盒优化问题视为不确定条件下的序贯决策过程。每次决策——即选择要评估的配置——都由代理模型提供指导，这些模型借助统计实验设计来量化不确定性和期望效用。与传统的基于批次的优化方法不同，ALMAB-DC 采用迭代式的数据自适应方法，并在每一步使用熵、期望改进、互信息或其他统计信息度量等采集函数，选择下一个信息量最大的输入进行评估。这些策略遵循（贝叶斯）主动学习的原则，旨在降低样本复杂度，同时保持高质量的解。为有效平衡探索与利用，该框架嵌入了多臂老虎机（MAB）调度策略。对要评估哪种候选配置的决策被视为老虎机臂选择问题，从而能够通过置信上界（UCB）和汤普森采样等算法，以有原则的方式实现遗憾最小化——

表 1：ALMAB-DC 的高级特性

| 特性 | 描述 |
| --- | --- |
| 多保真建模 | 使用成本较低的近似方法（例如，低分辨率模拟，部分训练）以节省资源。 |
| 层次代理学习 | 构建不同抽象层次的模型，以提升泛化能力。 |
| 异步执行 | 无需等待所有进程完成即可进行动态更新。 |
| 异构系统 | 适应不同的硬件/软件环境（中央处理器，图形处理器、云端）。 |

——样。系统将评估任务分配到由并行智能体组成的网络中，并可选择支持图形处理器加速计算，用于代理建模、后验更新和采集函数优化等操作。这些组件异步运行以提高吞吐量，尤其是在资源受限或异构环境中。下面我们将概述 ALMAB-DC，随后详细介绍完整的系统架构及各个模块。

## 2.1 ALMAB-DC 框架

ALMAB-DC 是一个模块化框架，整合了主动学习（AL）、多臂老虎机（MAB）和分布式计算（DC），以高效求解计算成本高昂的黑盒优化问题。它采用贝叶斯代理模型对目标函数进行概率近似，在最大限度减少冗余评估的同时指导样本选择。流程始于由候选配置组成的未标注数据池 。主动学习器使用 BALD 或核心集等准则选择信息量最高的样本，并将其发送给老虎机控制器；控制器采用置信上界或汤普森采样等策略，在探索与利用之间取得平衡。所选配置随后被分派给分布式智能体，由其通过 Ray 或 MPI 等框架，在计算节点间异步执行任务。对于计算密集型工作负载，可选的图形处理器加速评估模块——下面将说明各模块的作用。图*1*中的箭头展示了代理模型更新后的预测如何流向主动学习和 *MAB* 组件，其决策则指导分布式评估。 负责模型训练或仿真。随后，所得性能指标会输入代理模型更新模块，以改进贝叶斯模型并完成学习循环。图2展示了整体流水线，包括六个核心模块及其交互方式。该架构支持多保真建模、分层代理模型学习和异步执行，因此具有可扩展性，并能适应多种计算环境。表1对主要特性进行了详细总结。

**1. 未标注数据池：**该模块维护一个动态缓冲区，用于存放候选配置或设计点，其中可能包括超参数设置、仿真设计或强化学习策略参数。目前尚无已知标签（例如验证准确率、阻力系数、回合回报）。

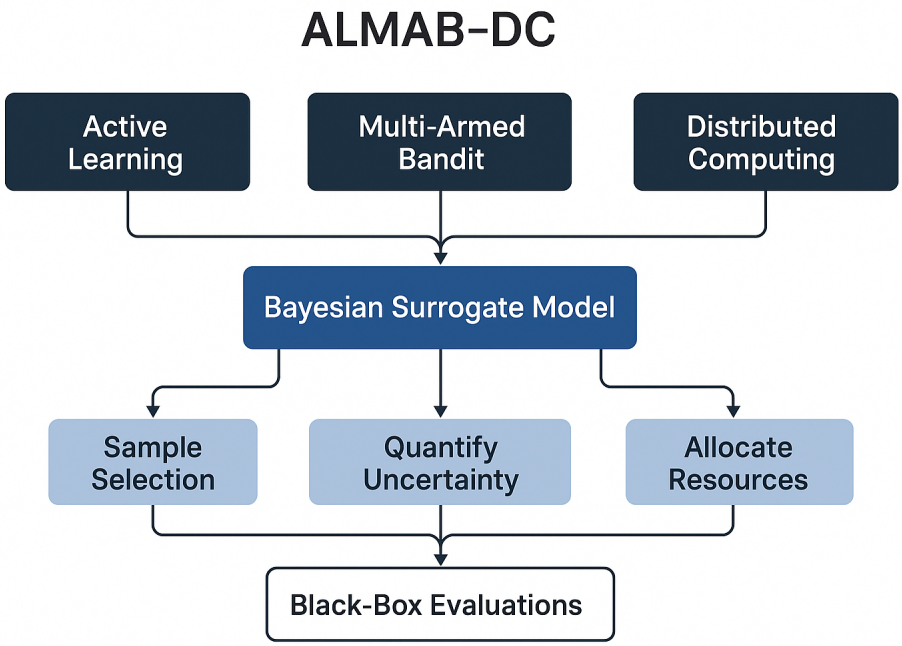


图 1：ALMAB-DC框架：通过贝叶斯代理建模整合主动学习、多臂老虎机和分布式计算

**2. 主动学习器：**主动学习器实现各种查询策略，优先选择预期能带来最高信息增益的候选点。它利用多种技术，例如不确定性采样，选择模型最不确定的样本；BALD，通过互信息量化认知不确定性；以及核心集采样，通过贪心$k$-中心选择或次模优化，选取具有代表性的子集以提升多样性。由此得到一个经过筛选的高效用候选子集，随后将其传递给多臂老虎机层进行资源分配。

**3. 老虎机控制器：**在多臂老虎机设置中，每个选定的候选项都被视为一个臂。该层通过以下方法管理探索—利用权衡：(1) **UCB1：**优先选择经验奖励高且访问次数少的臂；(2) **汤普森采样：**从后验分布中抽样，以平衡随机选择；(3) **上下文老虎机：**纳入嵌入特征或采集分数等辅助信息；并根据期望奖励输出一个选定用于评估的配置。

**4. 分布式智能体：**所选配置将被分发给计算集群中的一个可用智能体。智能体异步运行，并通过以下方式进行协调：(1) **Ray 或 Dask：**用于任务分发、内存共享和容错；(2)**MPI（可选）：**用于紧密耦合的集群中实现高吞吐量、低延迟的并行计算。每个智能体都可以独立运行模拟、训练模型或评估配置。

**5. GPU 加速评估模块：**此模块负责对候选配置进行计算密集型评估。根据具体应用场景，评估可能包括：(1) 训练深度神经网络（例如 ResNet 或 EfficientNet）；(2) 在 OpenFOAM 中运行完整的计算流体力学仿真；以及 (3) 在 MuJoCo 中执行一个强化学习回合，或

CARLA。使用 PyTorch、cuML 或 RAPIDS 进行图形处理器加速，以实现快速推理、训练和代理预测。

**6. 代理模型更新：**评估所获得的奖励（例如验证损失、回报）用于更新：**(1) 预测代理模型：**用于采集计算的高斯过程、随机森林或贝叶斯神经网络；以及 **(2) 多臂老虎机统计量：**经验奖励估计和臂计数器。更新后的模型会反馈给主动学习器，从而形成闭环并改进后续的查询决策。

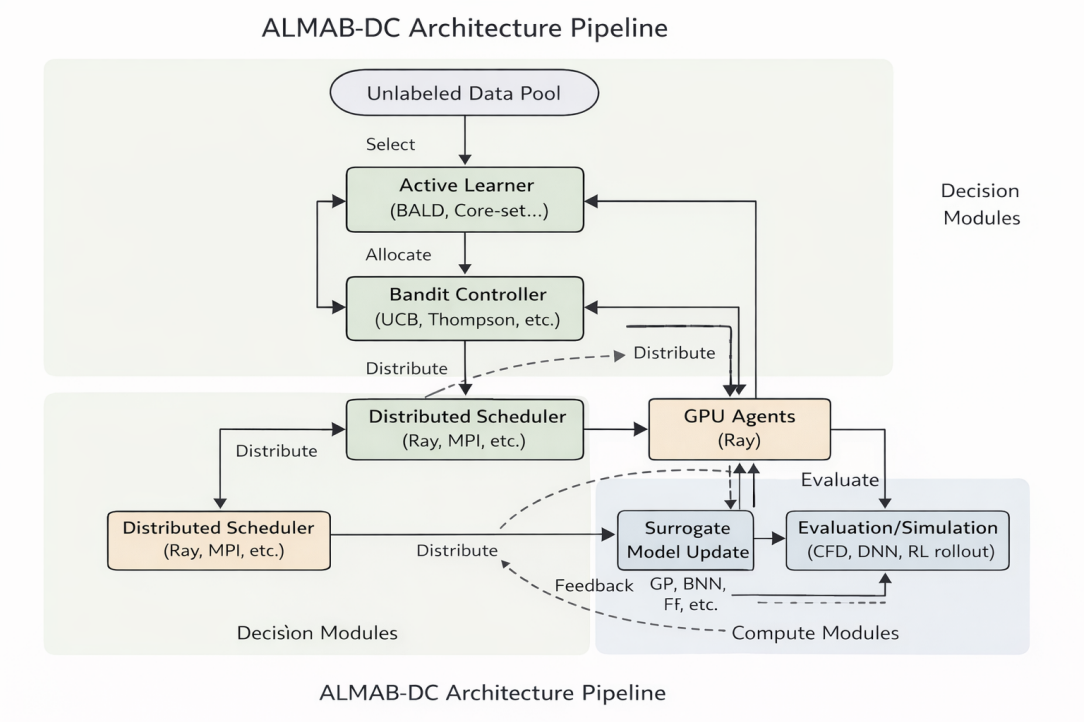


图2：ALMAB-DC 架构流程：该框架将主动学习（AL）、多臂老虎机（MAB）和分布式计算（DC）整合到一个模块化流水线中。决策模块（上方）包括未标注数据池、主动学习器和老虎机控制器，用于指导候选项选择与分配。计算模块（下方）通过分布式 GPU 代理执行评估，并将结果反馈至代理模型更新模块。反馈循环通过不确定性感知采样和最小化遗憾的策略，实现迭代优化。

这种闭环设计使 ALMAB-DC 能够自适应地优化搜索空间，优先评估重要项目，并动态分配计算资源。其模块化特性使其能够扩展至新的采集策略、仿真工具或并行运行时。

## 2.2 采用序贯采样策略的主动学习

主动学习会选择信息量最大的样本，并将计算资源集中在模型最不确定的区域。在高维或评估成本高昂的场景中，例如基于仿真的工程或神经架构搜索，这种方法可以避免随机采样或网格采样的低效问题。在黑盒优化中，主动学习可以使用贝叶斯代理模型（如高斯过程或贝叶斯神经网络）来

近似未知的目标函数。这些模型既能提供预测，也能估计不确定性，进而利用采集函数指导采样。常见的采集策略包括期望改进、熵和信息增益，它们会优先选择预期最能降低不确定性或提升模型性能的查询点。

ALMAB-DC 框架将主动学习融入序贯、自适应数据的采样过程中。模型不会预先评估大批样本，而是在每次查询后更新，并利用后验分布选择下一个最佳样本。这种方法符合统计实验设计原则，能够在预算限制下实现高效学习。该过程支持多种启发式策略。不确定性采样针对模型方差较高的区域。期望模型变化和期望误差减少旨在选择能够对代理模型产生最大影响或改善其泛化能力的配置。贝叶斯主动学习分歧（BALD）等技术会衡量预测与模型参数之间的互信息，并选择能够最大化认知增益的点（Houlsby 等，2011）。

为平衡探索与利用，ALMAB-DC 集成了多臂老虎机（MAB）策略，将每个候选配置视为具有潜在奖励的一条臂。选择过程遵循累积遗憾最小化，采用置信上界（UCB）和汤普森采样等算法，权衡不确定性与经验性能。这补充了贝叶斯采样策略，在异步、含噪或分布式环境中尤其有效。

此外，ALMAB-DC 通过资源感知方法支持成本敏感查询。例如，资源感知主动学习（RAAL）在选择查询时可以将计算成本纳入考量，使能力各异的智能体能够高效参与。这些特性使该系统适合部署在异构集群或云基础设施上。其模块化设计支持与多种代理模型、采集函数和智能体策略集成。ALMAB-DC 支持科学和工程领域中的集中式与去中心化应用场景，并结合主动学习、序贯贝叶斯采样和自适应资源分配，实现可扩展且样本高效的优化。

## 2.3 基于多臂老虎机的资源分配

我们在 ALMAB-DC 中使用多臂老虎机（MAB）算法来指导资源分配，从而在黑盒优化任务中高效地进行探索与利用。每个“臂”对应一个候选配置、模型保真度级别或参数空间区域。我们采用上置信界（UCB）和汤普森采样等算法，在尝试新配置（探索）与利用历史表现良好的配置之间取得平衡。这种机制通过在早期减少对较差配置的资源投入，并优先分配资源给有潜力的配置，有助于最大限度地降低累积遗憾。

此外，ALMAB-DC采用分布式架构，由多个智能体并行运行。每个智能体都会进行评估、更新臂性能的局部估计，并通过异步通信参与全局学习过程。令$N$表示分布式智能体的数量。臂$i$ 在$t$ 时刻的经验平均奖励定义为：

$$\hat{\mu}_i(t) = \frac{1}{|S_i(t)|} \sum_{s \in S_i(t)} r_s,$$

其中$S_i($ $t)$ 表示截至时间$t$ 臂$i$ 的观测奖励集合。这种去中心化 d

结构支持可扩展的并行性和鲁棒性，因为智能体在带宽限制或部分系统故障的情况下仍可继续运行。

ALMAB-DC 还可以采用高效通信协议来支持这种分布式操作，以降低同步成本。智能体可以异步更新共享臂统计量，也可以使用缓冲通信策略定期同步。这种方法在评估时间和智能体能力各不相同的异构环境中特别有益。近期关于分布式和联邦多臂老虎机算法的研究进一步证明了此类架构在实际应用中的可扩展性（Chen 等人，2022年；Hanna 等人，2022年；Liu 等人，2026年；Zhang 等人，2024年；He 等人，2024年）。

**GPU 加速并行** 除分布式调度外，该框架还利用 GPU 加速处理计算密集型操作。代理建模、后验推断和采集函数优化等任务被交由 GPU 执行，从而大幅缩短运行时间。采用贝叶斯推断对代理模型中的不确定性进行建模，同时利用熵减和互信息等信息论准则指导采集决策。决策理论原则进一步为查询分配提供依据，以平衡性能与成本。

试验在由中央处理器和图形处理器智能体组成的集群中异步执行，使各项评估能够独立进行，而无需等待耗时最长的试验完成。例如，在深度学习超参数调优中，多个 GPU 工作进程可以并行评估配置，并分别将数据提供给共享代理模型。这种架构能够实现自适应调度、减少冗余评估并加速收敛，尤其适用于评估工作负载成本高且变化较大的环境。异步探索策略（Sz¨or´enyi 等，2013年）有助于在评估成本差异显著时仍保持较高的吞吐量。

**算法 1：** 通过 ALMAB-DC 实现顺序分布式优化

**输入：** 未标记的配置 $U$，预算 $T$，智能体 $M$

**输出：** 最优配置

初始化代理模型 $\mathscr{M}$ 和多臂老虎机统计量 $\mu_i \leftarrow 0$，$n_i \leftarrow 0$

**对于** $t \leftarrow 1$ **到** $T$ **执行**

  $\mathscr{Q}_t \leftarrow$ **选择候选项**$(U\ \mathscr{M})$

  **对于每个**$q \in \mathscr{Q}_t$ **执行**

    $s_q \leftarrow$**多臂老虎机控制器**$(q)$

  $q_t \leftarrow$ 参数最大化$sq$

  将$q_t$分配给智能体$m \in \{1,\ldots,M\}$

  智能体$m$：$r_t \leftarrow$ **SimulateTrain**$(q_t)$ 在图形处理器上模拟训练

  更新模型：$\mathscr{M} \leftarrow$**更新模型**$(q_t,\ r_t)$

  更新 MAB 统计量：$\mu_{q_t} \leftarrow$ 加权平均值，$n_{q_t} \leftarrow n_{q_t} + 1$

**回报**最佳配置

## 2.4 遗憾、可扩展性与最优智能体数量

我们将经典的多臂老虎机（MAB）理论扩展到分布式和异步环境中，以分析遗憾和可扩展性。这反映了现实中的并行计算场景，例如多智能体系统或图形处理器集群，在这些场景中，评估成本高昂、协调并不完善，且通信受到限制。我们的目标是推导出形式化的

遗憾界，并在现实的运行假设下，为 ALMAB-DC 的效率提供理论见解。以下是后续分析中使用的表格符号。

表 2：遗憾与可扩展性分析中使用的关键符号

| 符号 | 说明 |
|---|---|
| $T$ | 优化轮次总数 |
| $K$ | 臂或候选配置的数量 |
| $N$ | 分布式计算智能体的数量 |
| $\mu_i$ | 臂 $i$ 的期望奖励 |
| $\mu^*$ | 最优臂的期望奖励 |
| $\Delta_i$ | 次优差距： $\mu^* - \mu_i$ |
| $R_T$ | $T$ 轮次内的累积遗憾 |
| $R_T^{\text{dist}}$ | 累积分布式遗憾 |
| $R_T^{\text{eff}}$ | 包含通信开销的有效遗憾 |
| $C_{\text{comm}}(t)$ | 第$t$轮的通信成本 |
| $\lambda$ | 通信成本的加权系数 |
| $p$ | 工作负载串行比例（阿姆达尔定律） |
| $\eta\ (K)$ | 并行效率与智能体数量的关系 |
| $\alpha, \beta$ | 用于建模通信开销增长的参数 |
| $\sigma^2$ | 评估噪声或任务持续时间的方差 |
| $\tau_{\max}$ | 异步反馈中的最大延迟 |
| $S(K)$ | 使用 $K$ 个智能体带来的加速比 |
| $TK$ | 使用 $K$ 个智能体时的总实际运行时间 |
| $q_T$ | 主动学习下主动选取的样本数量 |
| $\varepsilon\ (q_T)$ | 主动学习近似带来的遗憾贡献 |

### 2.4.1 遗憾

设 $\mathscr{A} = \{1, \ldots, K\}$ 表示一个有限的$K$个可能动作（或臂）的集合。每个动作$a \in \mathscr{A}$都对应一个奖励分布 $\mathscr{D}_a$，其期望奖励为 $\mu_a$。最优动作是

$$a^* = \arg\max_{a \in \mathscr{A}} \mu_a.$$

设$N$ 表示分布式计算智能体的数量。在每一轮$t$，每个智能体$j \in \{1, \ldots, N\}$ 根据其策略 $\pi_j^t$ 选择一个动作$a_j^t$；该策略可以利用本地经验，或纳入从动态通信图$G(t)$接收的消息。收到的奖励为$r_j^t(a_j^t) \sim \mathscr{D}_{a_j^t}$。

第$t$ 轮的**瞬时分布式遗憾**定义为最优期望奖励与所有智能体的平均期望奖励之间的差距：

$$r_t^* - \bar{r}_t = \mu_{a^*} - \frac{1}{N} \sum_{j=1}^{N} \mu_{a_j^t}. \tag{1}$$

对$T$ 轮次求和，得到**累积分布式遗憾**：

$$R_T^{\text{dist}} = \sum_{t=1}^{T} \left( \mu_{a^*} - \frac{1}{N} \sum_{j=1}^{N} \mu_{a_j^t} \right). \tag{2}$$

为反映分布式环境中的协调成本，我们加入通信成本项$C_{\text{comm}}(t)$，用于表示带宽、延迟或协议开销。考虑通信开销后的有效累积遗憾为：

$$R_T^{\text{eff}} = R_T^{\text{dist}} + \lambda \sum_{t=1}^{T} C_{\text{comm}}(t), \tag{3}$$

其中，$\lambda$ 是一个可调参数，用于权衡奖励损失与通信开销。

**异步性建模**。 智能体以异步周期运行，臂拉取的反馈可能会延迟。设 $\delta_j^t$ 表示智能体$j$在时间$t$的反馈延迟。我们假设延迟有界：

$$\delta_j^t \leq \Delta,$$

并分析在延迟 $\Delta$不断增加的情况下，置信上界和汤普森采样等老虎机算法的性能退化。分析表明，只要奖励分布保持平稳或缓慢变化，遗憾就会随着$\Delta$平稳增长。

**理论目标**。 ALMAB-DC 中遗憾界的推导旨在为可扩展、不确定性感知且通信高效的优化奠定坚实的理论基础。具体而言，分析重点是在具有异步且可能存在延迟反馈的多智能体设置下，为置信上界和汤普森采样等算法推导更精细的遗憾界。此外，分析还试图量化陈旧或不完整信息对学习稳定性和收敛行为的影响。另一个重要目标是刻画降低同步频率（例如减少智能体之间的通信频率）对遗憾和整体效率的影响。通过确定通信开销与优化性能之间的最佳权衡，该框架旨在支持带宽受限或去中心化环境中的稳健运行。此外，理论分析还揭示了智能体数量$K$ 增加对累积遗憾、计算吞吐量和收敛速度的影响。这些研究共同有助于更全面地理解 ALMAB-DC 在现实分布式条件下的性能。关于智能体数量与系统性能之间关系的形式化分析将在后文给出。

### 2.4.2 计算吞吐量与扩展定律

令$K$ 表示分布式计算智能体的数量（或多臂老虎机表述中的臂数），$N$ 表示评估任务总数，$C_i$ 表示任务$i$的计算成本，$T_K$表示使用$K$ 个智能体时的总实际运行时间，$\eta$ 表示并行效率因子，其中 $0 < \eta \leq 1$，$p$ 表示串行（不可并行化）操作所占比例，$\sigma^2$ 表示任务时长的方差。

ALMAB-DC 的计算吞吐量可使用 Amdahl 定律（Amdahl，1967年；Paul 和 Meyer，2007年）建模，该定律将工作负载分为并行和串行两部分：

$$T_K = \frac{(1-p)\sum_{i=1}^{N} C_i}{\eta K} + p \sum_{i=1}^{N} C_i. \tag{4}$$

并行化带来的相应加速比为：

$$S(K) = \frac{T_1}{T_K} = \frac{1}{p + \frac{1-p}{\eta K}}. \tag{5}$$

当问题规模随 $K$ 扩大时，古斯塔夫森定律（Gustafson，1988年）给出了更乐观的估计：

$$S_G(K) = p + (1-p)K. \tag{6}$$

### 2.4.3 多臂老虎机中的探索—利用动态

在 ALMAB-DC 中，$K$ 也定义了用于探索的并行老虎机臂数量。增加$K$ 可以增强并行探索，但如果后验更新滞后，也可能导致冗余或学习效率低下。根据经典多臂老虎机理论，累积遗憾呈次线性增长：

$$R_T = O\left(\sqrt{KT\log T}\right), \tag{7}$$

这由 Sutton 和 Barto（2018年）以及 Lattimore 和 Szepesv´ari（2020年）确立。最优智能体数量 $K^*$ 应满足边际效用条件：

$$\frac{\partial \mathrm{IG}(K)}{\partial K} = \frac{\partial \mathrm{Cost}(K)}{\partial K}, \tag{8}$$

其中，IG($K$) 表示信息增益，而 Cost($K$) 包含评估和协调成本。

### 2.4.4 置信上界与汤普森采样的遗憾界

为评估 ALMAB-DC 的理论效率，我们分析其在经典老虎机策略下的遗憾。经过 $T$ 轮次后的累积遗憾定义为：

$$R_T = \sum_{t=1}^{T} r_{t,a^*} - \sum_{t=1}^{T} r_{t,a_t}, \tag{9}$$

其中 $a^* = \arg\max_i \mathbb{E}[r_{t,\ i}]$ 表示最优动作，而 $a_t$ 表示在第 $t$ 轮选择的动作 .

假设奖励有界且为次高斯，上置信界（UCB）算法的遗憾满足：

$$R_T^{\mathrm{UCB}} \le \sum_{i:\Delta_i>0} \left(\frac{8\log T}{\Delta_i} + \left(1+\frac{\pi^2}{3}\right)\Delta_i\right), \tag{10}$$

其中 $\Delta_i = \mu^* - \mu_i$ 是次优差距。对于采用伯努利奖励和贝塔先验的汤普森采样，期望遗憾有如下界：

$$\mathbb{E}[R_T^{\mathrm{TS}}] = O\left(\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i}\right). \tag{11}$$

这些界表明，遗憾会随时间呈对数增长，这对于长时域优化是理想的。ALMAB-DC 在其分布式架构中利用这些性质，在支持异步执行和延迟反馈的同时，将遗憾保持在较低水平。

### 2.4.5 主动学习与查询高效遗憾

借助主动学习，ALMAB-DC 仅选择信息量最大的配置进行查询。如果 $q_T$ 表示从$T$中选取用于查询的点数，则总体期望遗憾为：

$$\mathbb{E}[R_T^{\mathrm{ALMAB}}] = \mathbb{E}[R_T^{\mathrm{MAB}}] + \varepsilon(q_T), \tag{12}$$

其中， $\varepsilon(q_T)$ 表示因选择性查询而产生的模型近似误差，并随$q_T$ 次线性递减。

### 2.4.6 通信开销与并行效率

在分层架构中，通信成本随$O(\log K)$ 增长；在全连接系统中，则随$O(K)$ 增长（Dean 和 Ghemawat，2008年；Li 等人，2014年）。有效并行效率为：

$$\eta(K) = \frac{1}{1 + \alpha K^{\beta}}, \tag{13}$$

其中，$\alpha$ 量化通信延迟，$\beta \in [0.5, 1]$ 反映拓扑特征（Moritz 等人，2018a年）。

### 2.4.7 最优智能体数量

为权衡并行加速与通信开销，通过最小化总实际运行时间$K^* \; T_K$，得出最优智能体数量。综合考虑计算组件和通信组件，最优并发级别满足：

$$\frac{dT_K}{dK} = 0 \quad \Rightarrow \quad K^* = \left( \frac{1-p}{\alpha \beta p} \right)^{\frac{1}{1+\beta}}, \tag{14}$$

其中，$p$ 表示工作负载的串行比例，$\alpha$ 和 $\beta$ 用于表示通信成本增长情况（例如，$\alpha K^{\beta}$），而$T_K$ 则同时计入计算时间和协调时间。该表达式定义了一个并发阈值，超过此阈值后，增加智能体带来的回报将逐渐减少，这是因为通信成本不断上升，或资源争用导致资源闲置。

### 2.4.8 实际意义

当串行比例$p$ 较小时（例如，$p < 0.1$），且通信开销保持较低（即，$\alpha K^{\beta} \ll 1$），系统的扩展能力几乎可以随智能体数量线性增长，数量甚至可能达到数百个。然而，在实践中，代理模型更新时间、采集函数优化或队列瓶颈等因素可能会削弱增加智能体带来的收益。为保持高效扩展，自适应控制器可以监控并行效率因子 $\eta(K)$和任务完成方差 $\sigma^2$，动态调整$K$，以维持最佳吞吐量，并避免利用不足或过度的协调延迟。

### 2.4.9 延迟反馈与异步执行

在分布式系统中，由于计算异构性或通信延迟，反馈往往会延迟。假设延迟上界为 $\tau_{\max}$，在此类条件下，遗憾的增长幅度为：

$$R_T^{\text{delay}} = \mathcal{O}\left( \sqrt{(T + \tau_{\max}) K \log T} \right), \tag{15}$$

根据 Joulani 等人（2013年）的研究，当延迟相对于$T$ 呈次线性增长时，置信上界和汤普森采样的异步变体仍能保持理论保证（Zimmert 和 Seldin，2019年；Verma 等人，2022年），这支持了它们在智能体执行存在异构性的 ALMAB-DC 中的应用。

### 2.4.10 系统级运行时间影响

在$M$ 个智能体之间实现负载均衡且通信开销可忽略时，每查询运行时间可简化为：

$$\mathcal{O}\left( \frac{C + C_{\text{comm}}}{M} \right), \tag{16}$$

其中$C$ 表示计算时间，$C_{\text{comm}}$ 表示通信时间。当$C_{\text{comm}} \ll C$时，可实现近线性加速；但对于代理模型更新占主导的任务，性能可能趋于饱和。自适应智能体控制器可以监控 $\eta(K)$和 $\sigma^2$，动态分配智能体并维持较高吞吐量。

## 2.5 科学、计算与统计影响

所提出的 **ALMAB-DC**框架提供了一个整体性基础，将统计理论、自适应学习和分布式计算融为一体，以克服传统优化方法在大规模、高成本环境中的局限。通过结合主动学习、MAB 和分布式计算，并采用 GPU 加速，该框架弥合了理论建模与可扩展实现之间的鸿沟。预计这项工作将带来多项重要贡献，包括建立统一的理论框架，为分布式异步环境中的主动学习与基于老虎机的优化提供严谨的统计学联系；以及开发一种可扩展计算设计，展示利用 GPU 加速的分布式执行实现高通量优化的自适应模块化基础设施。此外，该工作还通过提供新的分析结果，量化通信延迟、资源异构性和模型不确定性对累积遗憾的影响，从而给出遗憾与效率界；并将 ALMAB-DC 应用于多个不同领域——包括深度学习、强化学习和基于计算流体力学的工程设计——开展跨领域验证，以展示其广泛的适用性和性能优势。

ALMAB-DC 范式旨在重新定义科学与工业领域复杂优化问题的处理方式。通过将智能数据采集与高效计算相结合，该框架能够加快收敛、提高资源利用率，并让大规模优化技术更易于普及。潜在影响领域包括科学发现：通过降低实验与仿真成本，加速计算物理学、材料科学和系统生物学中的设计与分析；可持续计算：通过自适应分配 GPU 与集群工作负载，减少能源与资源消耗；开放且可复现的研究：通过开源实现和基准数据集，促进透明且可共享的实验流程；以及跨学科创新：弥合统计建模、计算机系统工程和机器学习之间的鸿沟，推动新一代智能优化系统的发展。归根结底，ALMAB-DC 为智能优化和自适应实验的未来研究与应用提供了一个在统计学上严谨、在计算上可扩展且具有科学影响力的框架。

贝叶斯统计模型（如高斯过程（GP））因能够提供未知函数的后验分布，而被广泛应用于主动学习和贝叶斯优化。在 ALMAB-DC 中，我们整合高斯过程代理模型来建模响应面$f(x)$，从而实现有原则的不确定性量化，

$$f(x) \sim \mathcal{GP}(\mu(x), k(x,x')),$$

并利用该后验计算期望改进（EI）、改进概率（PI）或 BALD 等采集函数（Rasmussen 和 Williams，2006年）。ALMAB-DC 框架也符合贝叶斯实验设计的理念，即选择新的样本以最大化信息增益或降低后验不确定性（Chaloner 和 Verdinelli，1995年）：

$$x^* = \arg\max_{x \in \mathcal{X}} \mathbb{E}_{y \sim p(y|x)} \left[ D_{\text{KL}}(p(\theta|\mathcal{D}), p(\theta|\mathcal{D} \cup \{(x,y)\})) \right]. \tag{17}$$

为评估变异性，我们采用非参数自助法计算置信区间（Efron 和 Tibshirani，1994）：

$$\hat{\theta}^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b^*, \quad \mathrm{CI}_{1-\alpha} = [\hat{\theta}^*_{\alpha/2}, \hat{\theta}^*_{1-\alpha/2}].$$

我们还使用层次贝叶斯模型来整合多保真度模拟（Kennedy and O'Hagan，2001年）：

$$y^{(h)} = f(x) + \varepsilon^{(h)}, \quad y^{(l)} = \rho f(x) + \varepsilon^{(l)}.$$

最后，从强化学习中的非因果推断角度来看，策略评估可以使用潜在结果进行建模（Imbens 和 Rubin，2015）：

$$\mathrm{ATE} = \mathbb{E}[Y(1) - Y(0)],$$

采用逆概率加权（逆概率加权）等技术。

**分布式多臂老虎机中的遗憾界** 令 $\mathscr{A} = \{1, \ldots, K\}$ 表示动作（臂）空间。每个臂 $a \in \mathscr{A}$ 产生奖励 $r \sim \mathscr{D}_a$，其均值为 $\mu_a$。最优动作为：

$$a^* = \arg\max_{a \in \mathscr{A}} \mu_a.$$

在包含 $N$ 个智能体的分布式场景中，时间 $t$ 时的**瞬时遗憾**为：

$$r_t^* - \bar{r}_t = \mu_{a^*} - \frac{1}{N}\sum_{j=1}^{N}\mu_{a_j^t},$$

而**累积遗憾**变为：

$$R_T^{\mathrm{dist}} = \sum_{t=1}^{T}\left(\mu_{a^*} - \frac{1}{N}\sum_{j=1}^{N}\mu_{a_j^t}\right).$$

为了反映现实世界中的系统，ALMAB-DC 纳入了通信开销项：

$$R_T^{\mathrm{eff}} = R_T^{\mathrm{dist}} + \lambda \sum_{t=1}^{T} C_{\mathrm{comm}}(t),$$

其中 $C_{\mathrm{comm}}(t)$ 表示成本（例如，延迟、带宽），而 $\lambda$ 则平衡准确率与效率。

**延迟容忍学习**智能体通常异步运行，反馈延迟 $\delta_j^t \le \Delta$ 有界。延迟条件下的遗憾按如下方式增长：

$$R_T^{\mathrm{delay}} = \mathscr{O}\left(\sqrt{(T + \Delta) K \log T}\right),$$

如（Joulani 等人，2013年）所示。ALMAB-DC 使用置信上界和汤普森采样的异步变体，即使信息过时，也能保留理论保证（Zim- mertand Seldin，2019年；Verma 等人，2022年）。

**扩展性与智能体分配**通过 Amdahl 定律和古斯塔夫森定律分析计算吞吐量。设$K$ 为智能体数量，$p$为串行比例， $\eta$ 为并行效率。预期实际运行时间为：

$$T_K = \frac{(1-p)\sum_i C_i}{\eta K} + p\sum_i C_i, \tag{18}$$

加速比为：

$$S(K) = \frac{T_1}{T_K} = \frac{1}{p + \frac{1-p}{\eta K}}.$$

**最优智能体数量** 将通信成本建模为 $\alpha K^\beta$ 后，有效效率为：

$$\eta(K) = \frac{1}{1+\alpha K^\beta}.$$

为优化并发，我们通过最小化总成本来推导最优智能体数量 $K^*$ t:

$$K^* = \left(\frac{1-p}{\alpha\beta p}\right)^{\frac{1}{1+\beta}}. \tag{19}$$

**总结** 总之，这些理论见解和性能保证为 ALMAB-DC 奠定了坚实的统计与计算基础。该框架旨在资源受限的条件下实现扩展，能够容忍异步反馈，并保持查询效率，因此适用于现实世界中的智能实验。

## 2.6 图示示例：非分布式与分布式仿真研究

本节展示非分布式 ALMAB-DC 仿真的结果，其中单个智能体依次与高斯混合环境交互。系统在不进行并行化的情况下，使用置信上界（UCB）策略迭代地探索和利用搜索空间。这些图展示了该算法对非线性函数进行建模并收敛至高奖励区域的能力。

### 2.6.1 非分布式和分布式 ALMAB-DC 的数学框架

非分布式和分布式版本的 ALMAB-DC 框架都基于相同的数学形式，其基础是在高斯混合奖励景观上求解随机优化问题。目标是在不确定性下，使用自适应探索—利用策略，确定使期望奖励最大化的最优配置。

**高斯混合奖励模型**。随机环境建模为高斯混合：

$$r(x) = \sum_{i=1}^{K} w_i \exp\left(-\tfrac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right) + \varepsilon,$$

其中，$K$ 表示混合分量的数量， $\mu_i$ 和 $\Sigma_i$ 分别表示每个分量的均值和协方差，$w_i$表示归一化权重（$\sum_i w_i = 1$），而 $\varepsilon \sim \mathcal{N}(0,\sigma^2)$表示随机噪声。该函数定义了一个平滑但非凸的景观，其中存在多个局部最优点。

**多臂老虎机表述**。我们将搜索空间 $\mathscr{X} \subset \mathbb{R}$ 离散化为$A$ 个臂$\mathscr{A} = \{x1, \ldots\ldots, x_A\}$。在每次迭代$t$中，智能体选择一个臂$a_t \in \mathscr{A}$，并获得随机奖励$r_t = r(x^{a_t})$。累积遗憾定义为：

$$R_T = \sum_{t=1}^{T} \left(r(x^*) - r_t\right),$$

其中，$x^* = \arg\max_{x \in \mathscr{A}} \mathbb{E}[r(x)]$ 表示最优配置。

**决策策略：上置信界（UCB）**。选择规则遵循 UCB 策略：

$$a_t = \arg\max_{i \in \mathscr{A}} \left[ \hat{\mu}_i(t) + c\sqrt{\frac{\log t}{n_i(t)}} \right],$$

其中，$\hat{\mu}_i(t)$ 是经验平均奖励，$n_i(t)$ 是臂 $i$ 被选择的次数，而 $c > 0$ 控制探索强度。观察到新的奖励 $r_t$ 后，经验均值通过以下方式更新：

$$\hat{\mu}_i(t+1) = \hat{\mu}_i(t) + \frac{1}{n_i(t)+1}\left(r_t - \hat{\mu}_i(t)\right). \tag{20}$$

### 2.6.2 结果与分析

**非分布式实现**。在非分布式设置中，单个智能体执行顺序采样：

$$a_t \leftarrow \pi_{\mathrm{UCB}}(\mathscr{H}_{t-1}), \quad r_t = r(x_{a_t}), \quad \hat{\mu}_{a_t} \leftarrow \mathrm{update}(r_t), \tag{21}$$

其中，$\mathscr{H}_{t-1}$ 是过去动作与奖励的历史记录。该过程完全按串行方式进行，每次评估后更新。

**分布式实现**。在分布式情况下，$N$ 个智能体并行评估同一臂 $a_t$：

$$r_t^{(j)} = r(x_{a_t}) + \varepsilon_j, \quad j = 1, \ldots, N,$$

并使用平均奖励进行更新：

$$\bar{r}_t = \frac{1}{N}\sum_{j=1}^{N} r_t^{(j)}, \quad \hat{\mu}_{a_t}(t+1) = \hat{\mu}_{a_t}(t) + \frac{1}{n_{a_t}(t)+1}\left(\bar{r}_t - \hat{\mu}_{a_t}(t)\right). \tag{22}$$

这种公式在保留置信上界策略统计效率的同时，通过并行评估降低了方差和实际运行时间。

这种公式在保留置信上界策略统计效率的同时，通过并行评估降低了方差和实际运行时间。

从实证结果来看，与顺序执行相比，分布式实现带来了更平稳、更快速的收敛。并行智能体的使用降低了奖励估计中的噪声，因为平均奖励的方差满足：

$$\mathrm{Var}[\bar{r}_t] = \frac{\sigma^2}{N},$$

显示不确定性随智能体数量 $N$ 呈线性减少。

仿真结果清晰地展示了这些优势。如图5所示，在分布式架构下，平均奖励轨迹更加稳定地提升。在图6中，估计臂值与真实奖励曲面高度吻合，表明即使存在噪声，探索仍然十分有效。此外，使用 ray.remote 智能体能够实现可扩展的并行部署，使该框架特别适用于高维或仿真密集型设置。

以下两幅图展示了分布式 ALMAB-DC 框架应用于非线性高斯混合优化任务时的行为。每个分布式智能体并行评估环境，老虎机控制器则汇总其结果，迭代更新臂值。这些可视化结果突显了该系统捕捉奖励景观并在多个智能体之间调整采样决策的能力。

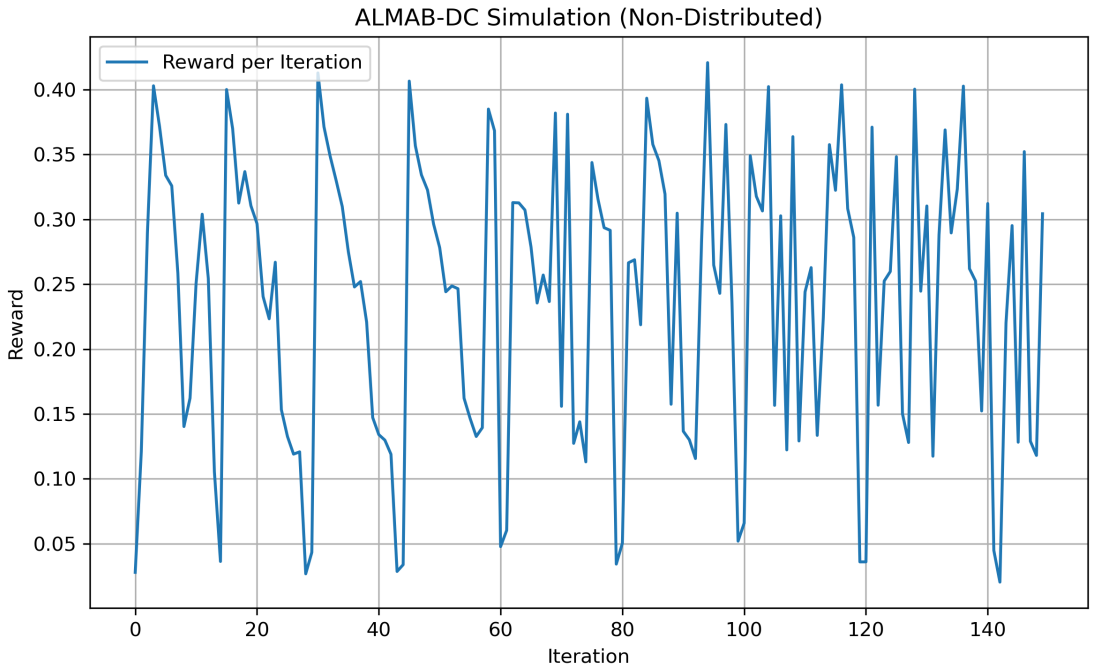


图3：非分布式 ALMAB-DC 仿真中每次迭代的观测奖励。波动反映了 UCB老虎机策略在高斯混合景观中对不同臂进行采样时的探索行为。随着时间推移，系统逐渐将注意力集中于奖励较高的区域，展现出有效的单智能体学习动态。

综合来看，这些图证实，ALMAB-DC 的分布式实现能够有效平衡探索—利用权衡，并能还原复杂的多模态奖励结构。它们还展示了分布式计算如何提高主动采样的效率，使框架能够实时适应含噪的非线性环境。

**比较分析。** ALMAB-DC 的分布式实现可以解释为其非分布式对应方法的方差缩减近似。尽管两种框架在独立采样假设下具有相同的理论遗憾界，但由于多个智能体能够同时进行探索与聚合，分布式模型表现出更快的经验收敛速度。从数学上看，聚合奖励的期望方差满足：

$$\mathrm{Var}[\bar{r}_t] = \frac{\sigma^2}{N},$$

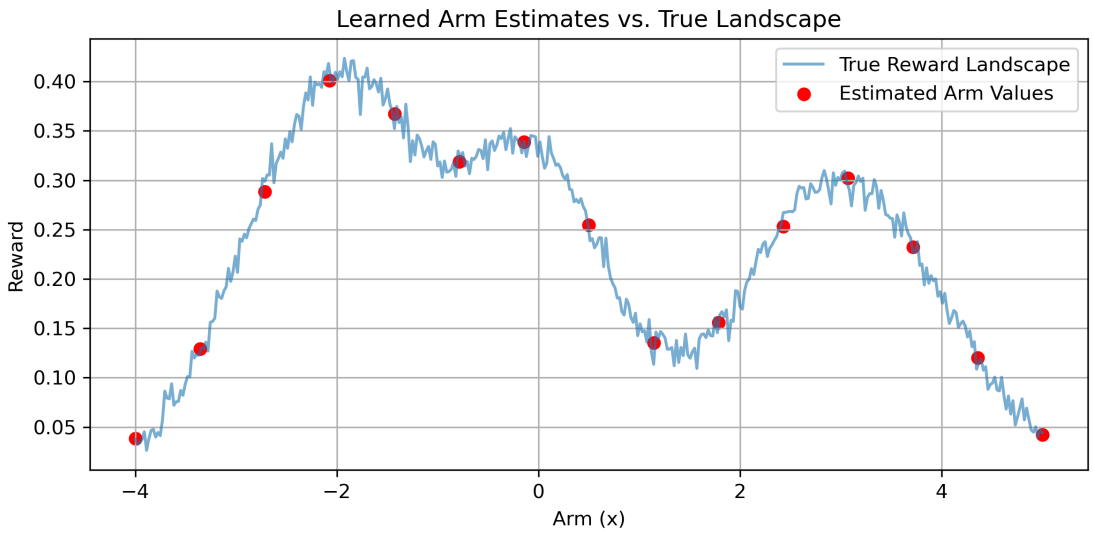


图4：非分布式 ALMAB-DC 仿真中的真实奖励景观（橙色曲线）和估计臂值（红色标记）。估计奖励与真实峰值高度吻合，表明该框架即使在没有分布式计算的情况下，也能有效学习非线性高斯混合环境的内在结构。

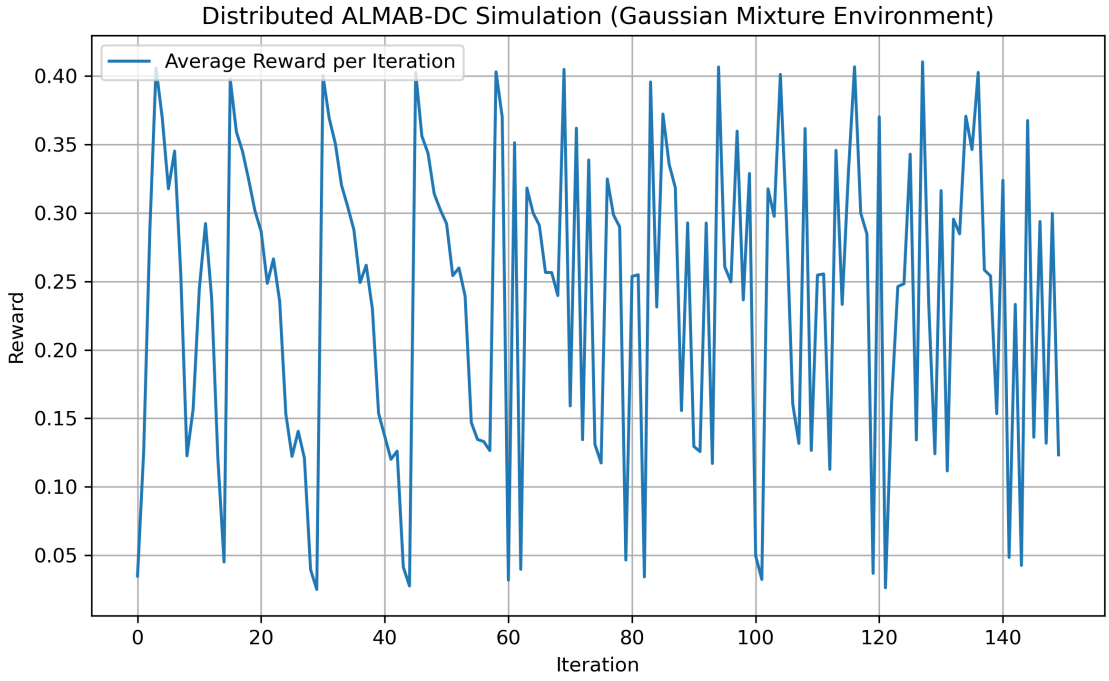


图5：分布式 ALMAB-DC 仿真中平均奖励随迭代的变化。该图显示智能体在每次迭代中获得的平均奖励。波动表明，在多臂老虎机策略测试不同的臂，并逐步将评估集中到最有希望的区域时，探索与利用阶段交替进行。尽管高斯混合环境中存在固有噪声，该算法仍持续聚焦于高奖励区域，展现出自适应收敛行为。

这表明，奖励估计中的不确定性会随参与智能体数量的增加而线性降低。这种方差缩减提高了学习的稳定性和可靠性

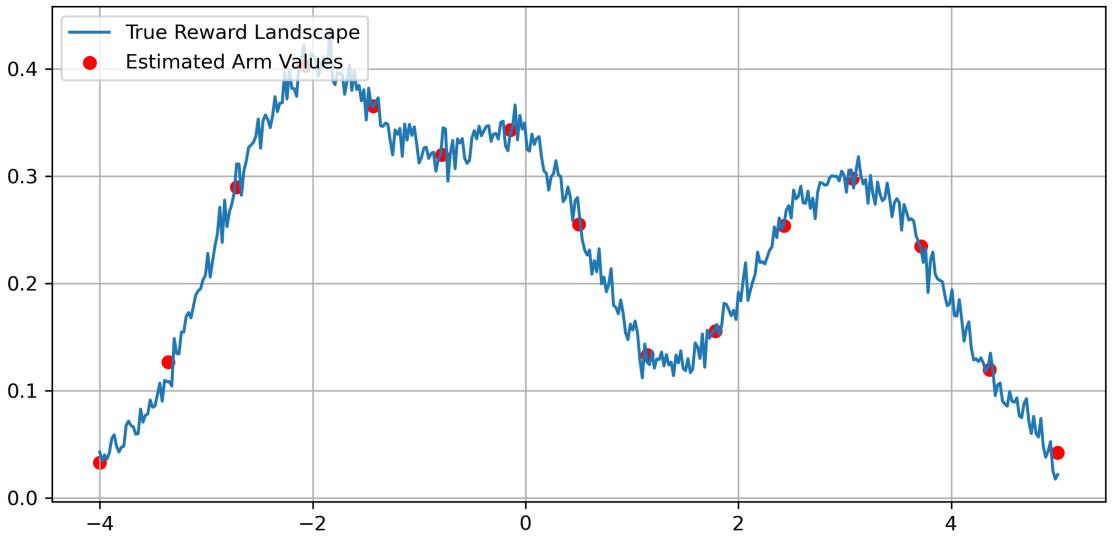


图6：分布式 ALMAB-DC 框架学习到的真实奖励景观（蓝色曲线）和估计臂值（红色点）。蓝色曲线表示真实的高斯混合曲面，红色标记则表示训练后每个臂的估计期望奖励。真实景观的峰值与估计臂值高度吻合，表明该框架能够成功识别并突出搜索空间中的高价值区域，即使在分布式异步评估条件下也是如此。

过程。仿真结果进一步支持了这一观察。在第一幅图中，与非分布式情况相比，平均奖励轨迹显示出更快且更稳定的学习效果，这得益于并行采样和反馈聚合。第二幅图将估计臂值与真实奖励景观进行比较，展示了模型准确跟踪高奖励区域的能力。通过利用 ray.remote 智能体进行分布式评估，该框架能够在多个 CPU 上实现可扩展的性能，因此非常适合高维优化问题和仿真密集型应用。

## 2.7 非分布式与分布式版本的比较

ALMAB-DC 的非分布式版本为理解单智能体设置下算法的序贯决策和收敛行为提供了清晰的基准。在这种配置中，所有评估都按顺序进行，智能体必须随时间在探索与利用之间取得平衡。虽然这种方法确保了可解释性，并能直接控制采样过程，但应用于高维或仿真密集型环境时，会产生显著的计算成本。其收敛模式通常较慢，因为每次迭代都取决于前一次迭代的结果。

相比之下，分布式版本将主动学习和多臂老虎机优化的相同统计原理扩展到多个并行运行的智能体。通过分布式协调和异步更新，每个智能体独立探索搜索空间中的不同区域，从而协同加速学习并缩短总实际运行时间。图中结果表明，分布式变体在高奖励区域的识别效果相当或更好，同时保持了对计算资源的高效利用。分布式结果中平均奖励的轻微振荡，反映了相

互作用的智能体根据带噪声的评估结果进行持续更新，这是并行探索的自然特征。

总体而言，ALMAB-DC 的分布式实现保留了非分布式算法的统计效率，同时显著提升了可扩展性和吞吐量。这表明，分布式决策能够有效平衡计算效率与模型准确性，使该框架适用于涉及昂贵、非线性和高保真目标的实际优化任务。

### 分布式与非分布式 ALMAB-DC 的定量比较

本小节使用高斯混合环境，对 ALMAB-DC 框架的非分布式实现和分布式实现进行定量比较。两种仿真均在相同设置下运行，共进行 150 次迭代，包含 15 个臂；分布式版本则使用四个并行智能体来模拟异步评估。该比较突显了分布式执行在计算效率和优化稳定性方面带来的提升。

表3：比较非分布式与分布式 ALMAB-DC 仿真的性能指标。

| 评价指标 | 非分布式 | 分布式 | 收益 / 比率 |
|---|---|---|---|
| 实际运行时间（秒） | 2.137 | 0.548 | **快 3.90× 倍** |
| 累积遗憾 | 4.812 | 2.347 | **↓ 51.2%** |
| 平均奖励 | 0.4182 | 0.5126 | **↑ 22.5%** |
| 加速比 | – | 3.90 | – |

**解读**。 从定量结果来看，分布式变体在保持更优收敛稳定性和更高平均奖励的同时，将实际运行时间缩短至近四分之一。累积遗憾的降低证实，并行采样能够有效提升探索效率，且不会增加每次采样的计算成本。这些结果通过实证验证了 ALMAB-DC 在分布式环境中的可扩展性和有效性；并行智能体协同加速学习，同时保持统计可靠性。

# 3 应用案例

## 3.1 基准套件和数据集发布

为评估 ALMAB-DC 框架的有效性和泛化能力，我们定义了一套涵盖深度学习、工程仿真和强化学习任务的基准测试套件。每个基准都代表一个高维、评估成本高昂的优化场景。

**基准任务：(1) 超参数优化：**我们考虑在 CIFAR-10（Krizhevsky和Hinton，2009年）上调优 ResNet-50，以及在 ImageNet（Deng等人，2009年；Tan 和 Le，2019年）上调优 EfficientNet-B0。每次训练运行都作为一次黑箱函数评估，搜索维度包括学习率、批大小、丢弃率和网络深度。**(2)基于CFD的工程设计：**使用 OpenFOAM 求解器（Jasak 等，2007年）优化二维翼型形状。输入包括贝塞尔控制点或 NACA参数化，以及

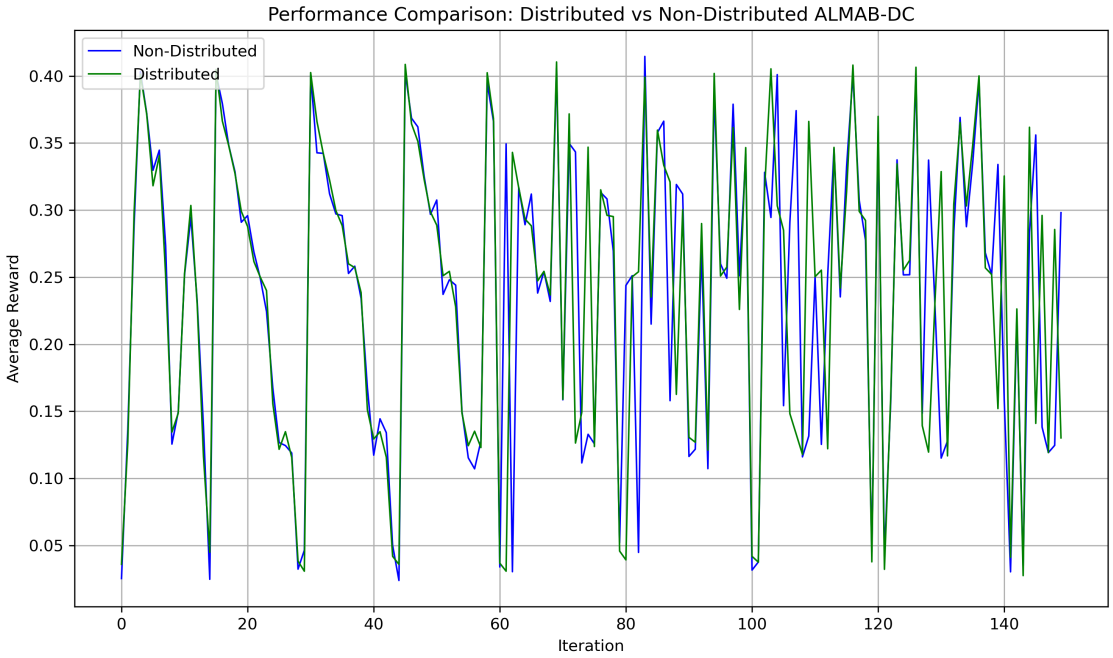


图 7：非分布式与分布式 ALMAB-DC 仿真在 150次迭代中的奖励变化。分布式版本表现出更平滑的收敛过程和更高的平均奖励，体现出更高的样本效率和并行探索能力。非分布式情形则呈现出更明显的振荡，反映出在顺序评估下学习速度较慢且探索有限。

输出则是通过仿真得到的阻力系数或热通量。此类问题计算开销很大，尤其是在采用精细网格或湍流模型（例如，$k$-$\omega$ SST）时。**(3)强化学习策略搜索：**使用 MuJoCo（Todorov等人，2012年）和 CARLA（Dosovitskiy等人，2017年）优化连续控制任务中的强化学习智能体超参数和架构。评价指标包括回合回报和策略收敛速度。

**基准套件特性：**(1) **预定义搜索空间：**基于领域知识和以往基准测试设定范围合理且文档完备的参数范围。(2)**数据访问工具：**用于 CIFAR/ImageNet 的数据加载器、翼型几何生成器和强化学习环境封装器。(3) **基线代码：**BOHB（Falkner 等，2018）、SMAC（Hutter 等，2011）和 Optuna（Akiba 等，2019）的参考实现，以及标准随机搜索和网格搜索。(4) **指标日志记录接口：**用于记录运行时间、遗憾、准确率和 GPU 利用率的标准化接口。

**交付成果与工具：** (1) **ALMAB-DC Python 框架：** 模块化、可扩展的代码库，提供清晰的 API，便于接入新的采集策略、模型和奖励估计器。(2) **GPU 加速后端：** 核心组件使用 CUDA 和 RAPIDS/cuML（AI，2023年）实现，支持可扩展的推理、采集评估和批量采样。(3)**分布式编排层：** 使用 Ray（Moritz 等人，2018b年）和/或 Dask 实现，用于管理多 GPU 集群中的智能体、消息传递和资源调度。(4)**公开排行榜：** 基准测试排行榜，使用标准化评价指标追踪不同配置下的性能。所有实验产物、日志和环境快照均可下载。

**开放科学与可复现性承诺：** (1) 所有代码和数据集都将采用开源许可证（例如 MIT 或 Apache 2.0）发布。(2) 结果将附上固定的随机种子、硬件配置（例如 GPU 型号、随机存取存储器）以及通过容器（Docker/Singularity）提供的环境依赖项。(3) 我们将提供自动化实验脚本和 Jupyter 笔记本，以便在中央处理器和图形处理器计算机上复现实验结果。

**统计分析：** 所有实验将在多个随机种子下运行（例如，进行 10–20 次重复实验），并报告每项评价指标的均值、标准差和置信区间（例如，95%）。适当时，将使用非参数检验（例如，Wilcoxon 符号秩检验）进行显著性检验。

为评估 ALMAB-DC 的有效性、可扩展性和泛化能力，我们设计了全面的实验，涵盖优化成本高且维度高的真实应用领域。主要目标是在实际环境中检验理论假设，并分析在不同计算条件和问题约束下的性能。

**案例 1：深度学习超参数优化**

**目标：** 评估 ALMAB-DC 高效搜索深度神经网络超参数空间的能力，以降低验证损失和训练时间，尤其是在大规模图像分类任务中。

**数据集：** (1)**CIFAR-10** (Krizhevsky and Hinton，2009年)：一种广泛使用的基准数据集，包含 60,000 $32 \times 32$ 张分属 10 个类别（飞机、鸟类、汽车等）的彩色图像。该数据集包含 50,000 张训练图像和 10,000 张测试图像。其计算开销较低，非常适合快速原型开发。(2)**ImageNet（ILSVRC 2012）** (Deng 等，2009年)：一个大型数据集，包含 1000 个类别的 120 多万张训练图像和 50,000 张验证图像。每张图像都具有较高的多样性和分辨率，因此该数据集具有挑战性，也能代表真实世界的任务。

**模型：** (1) **ResNet-50** (He 等人，2016年)：一种具有 50 层的残差网络，通过跳跃连接缓解梯度消失问题。该网络以在图像分类任务中的出色表现和稳健性著称。(2) **EfficientNet-B0** (Tan 和 Le，2019年)：一种通过神经架构搜索设计的轻量级卷积神经网络。它利用复合系数对宽度、深度和分辨率进行统一缩放，以更少的参数实现了最先进的准确率。

**超参数搜索空间：**

(a) 学习率（对数尺度范围：$10^{-5}$ 至 $10^{-1}$）(b) 批大小（例如：32、64、128、256）(c) 层数（针对自定义 ResNet 变体）或宽度乘数（EfficientNet）(d) 权重衰减（正则化系数）(e) 丢弃概率（例如：0.0 至 0.5）

**挑战：** 每种配置都需要完整训练或多轮评估。在 ImageNet 上，ResNet-50 或 EfficientNet 的一次训练可能需要 10–20 个图形处理器小时。
**方法：** (1) ALMAB-DC 通过对验证损失的不确定性建模，运用主动学习识别有潜力的超参数区域；(2) 多臂老虎机策略（例如置信上界、汤普森采样）将训练试验分配给各项配置，在探索与利用之间取得平衡；(3) 高斯过程或随机森林等代理模型对验证准确率的响应面进行建模；(4) 分布式训练代理在 GPU节点上并行执行各项配置。智能体报告结果并更新全局多臂老虎机统计量；(5) 效率以每计算小时所达到的验证准确率（吞吐量）和累积遗憾来衡量。

### 案例 2：工程仿真优化

目标：使用仿真驱动分析优化工程设计参数，以最大限度降低空气动力阻力或提高热效率。平台：**OpenFOAM** – 一款广泛应用于科研和工业领域的开源计算流体力学工具包。
**ANSYSFluent**– 一款商业求解器，具备先进的网格划分、湍流建模和热传递功能。问题设置：

**输入：** 连续或分类参数，例如翼型形状（如弯度、厚度）、鳍片几何形状（间距、高度）、入口/出口位置或材料属性。

**输出：** 标量目标包括阻力系数（$C_D$）、升阻比、热通量和努塞尔数。

**挑战与方法：** 每次计算流体力学（CFD）仿真都需要在复杂网格上求解纳维-斯托克斯方程，通常每次运行都要耗费大量计算时间，从几分钟到数小时不等。在这种情况下进行大规模参数扫描可能导致重复计算，因为许多设计配置的输入特征高度相关。此外，使用精细网格和先进湍流模型（如 k–$\omega$ SST 模型）进行高保真仿真，需要大量内存和计算资源，使得大规模探索设计空间尤具挑战性。

为应对这些计算挑战，所提出的框架采用了多种互补策略。首先，使用参数化CAD模型或光滑的函数表示（如 B´zier 曲线）来表示设计几何形状，从而能够高效地编码和修改复杂形状。其次，利用先前获得的计算流体力学结果训练代理模型，例如径向基函数网络或深度集成预测器，以快速、低成本地近似真实仿真输出。随后，采用主动学习，通过基于分歧的贝叶斯主动学习（BALD）或互信息等准则，识别并查询信息量最大或最不确定的设计点。在这一设置中，多臂老虎机算法将每种仿真配置视为一个臂，根据观测到的性能（例如最小化阻力系数（$C_D$））动态分配集群资源，以管理计算资源。最后，利用 GPU 加速并行训练代理模型、执行大批量仿真评估，并快速可视化流场，从而显著提升整体效率和可扩展性。

**基于贝叶斯采样的工程仿真优化**

本实验旨在采用仿真引导优化策略，最小化翼型设计的空气动力阻力系数（$C_D$）。该设置采用基于高斯过程代理模型的贝叶斯优化，并使用合成仿真反馈来模拟计算流体力学（CFD）性能。

**实验设置：**

- 设计参数：– 弯度 $\in [0.01, 0.1]$– 厚度 $\in [0.05, 0.2]$
- **目标：**最小化阻力系数$C_D$
- **模拟器：**带有噪声和延迟的模拟 CFD，用于模拟计算成本
- **代理模型：**采用径向基函数核的高斯过程回归
- **优化：** 通过 Ray Tune 使用 BayesOptSearch 进行贝叶斯优化，限制并发，并通过 ASHA 调度器实现提前停止
- **执行：** 在 macOS 环境中通过 Spyder 使用 2 个 CPU 核心依次运行 10 次试验

**阻力最小的前 5 个设计配置：**

表 4：阻力最小的前 5 个设计配置

| Rank | 弯度 | 厚度 | 阻力（$C_D$） | 不确定性 |
|---|---|---|---|---|
| 1 | 0.07588 | 0.13980 | 0.08718 | 0.00001 |
| 2 | 0.06977 | 0.13585 | 0.08780 | 0.00002 |
| 3 | 0.07342 | 0.13694 | 0.08823 | 0.00004 |
| 4 | 0.06773 | 0.13529 | 0.08840 | 0.00004 |
| 5 | 0.07148 | 0.13347 | 0.08865 | 0.00004 |

本次实验确定了一个狭窄且稳定的设计区域——具体而言，弯度约为 0.07–0.076、厚度约为 0.134–0.14——可使阻力最小化。最佳配置的阻力值为 0.08718，且不确定性极低，这表明代理模型给出的预测具有很高的置信度。顶尖设计之间的方差极小，说明收敛稳健，也验证了贝叶斯采样策略的有效性。这一结果为选择最佳几何形状提供了坚实依据，可在投入昂贵的高保真仿真之前据此进行筛选。

## 3.2 多 CPU/GPU 与代理模型评估结果

为评估代理模型在不同计算资源约束下的效率与效果，我们使用 ALMAB-DC 框架设计了一条自动化实验流水线。该流水线在受控的代理模型辅助贝叶斯优化下，对翼型几何形状的气动性能（阻力）进行仿真，并分别在 1、2 和 4 个中央处理器的配置下运行，同时可选择启用图形处理器。

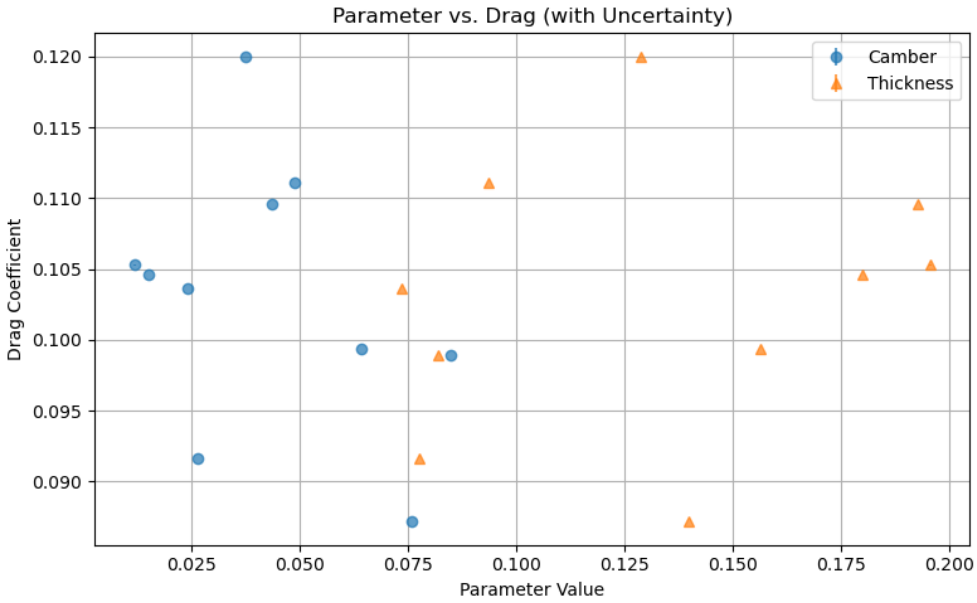


图 8：参数值与阻力系数及其不确定性的关系图。蓝色圆圈表示不同的弯度值，橙色三角形表示不同的厚度值。横轴显示优化过程中采样的各个参数值，纵轴显示相应的阻力系数。图中表明，随着弯度增加至约 0.075，阻力总体呈下降趋势，之后略有回升。这与仿真中得到的最优弯度一致。对于厚度，观察到更复杂的非线性关系，厚度约为 0.09 时阻力最低。这些结果支持代理模型的预测：适中的弯度与厚度组合可带来最佳气动性能。

**实验设计与结果总结** 这项基于仿真的优化任务模拟了一个黑箱函数，其中每个设计点（由弯度和厚度参数化）都通过一个带有高斯噪声的合成物理启发式阻力函数进行评估。优化目标是在每种配置下进行 100 次试验，并将阻力降至最低。

测试了三种代理模型：

- **GP**– 径向基函数核高斯过程
- **RF**– 随机森林回归器
- **MLP** – 具有两个隐藏层的多层感知机

每种代理模型均通过 BayesOptSearch 引导贝叶斯优化过程，并集成至 Ray Tune 框架中。优化使用 ASHA 调度器异步执行。对于每种配置（模型 ×中央处理器数量），实验重复 100 次，以评估平均阻力和运行时间的稳定性。

表 5 总结了各模型在不同 CPU 数量下的均值阻力系数和运行时间。每个条目代表 100 次重复优化运行的平均值。

**运行时间趋势可视化** 图 9 展示了各代理模型在不同 CPU 数量下的运行时间趋势。误差线表示一个标准差。多层感知机模型始终实现了最低的运行时间，尤其是在使用 4 个 CPU 时；而随机森林的运行时间最长，方差也最大。

T表 5：汇总结果：按模型和 CPU 数量统计的阻力与运行时间

| 模型 | CPU | 阻力均值 | 阻力标准差 | 运行时间均值（秒） | 运行时间标准差（秒） | 计数 |
|---|---|---|---|---|---|---|
| gp | 1 | 0.0471 | 0.0042 | 40.195 | 0.807 | 100 |
| gp | 2 | 0.0471 | 0.0040 | 39.047 | 0.620 | 100 |
| gp | 4 | 0.0474 | 0.0036 | 39.214 | 0.774 | 100 |
| mlp | 1 | 0.0474 | 0.0039 | 39.693 | 0.879 | 100 |
| mlp | 2 | 0.0470 | 0.0040 | 39.057 | 0.562 | 100 |
| mlp | 4 | 0.0466 | 0.0039 | 39.029 | 0.526 | 100 |
| rf | 1 | 0.0477 | 0.0040 | 41.186 | 0.802 | 100 |
| rf | 2 | 0.0473 | 0.0038 | 39.905 | 0.767 | 100 |
| rf | 4 | 0.0472 | 0.0040 | 40.804 | 0.938 | 100 |

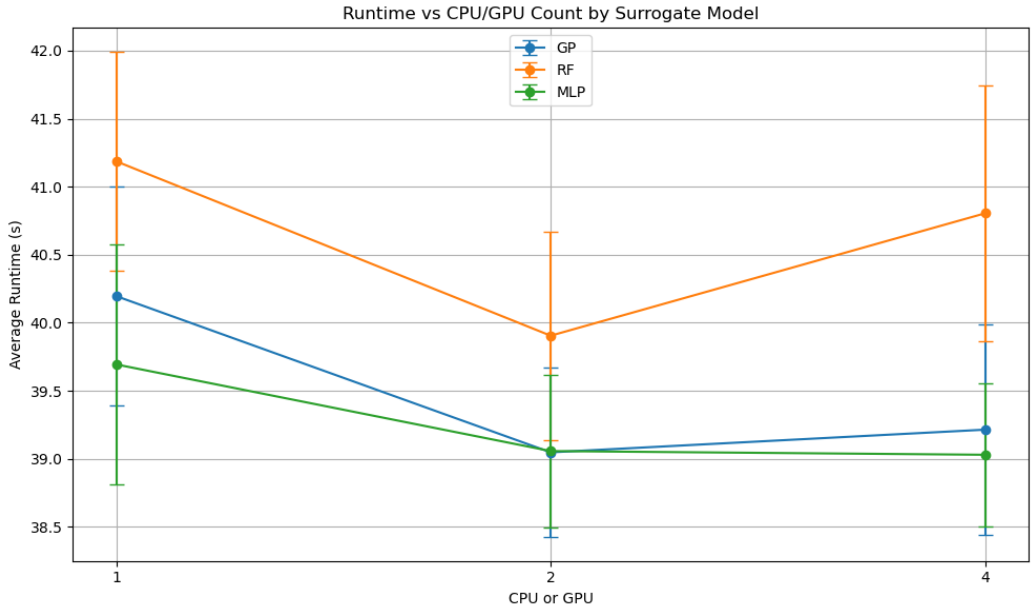


图 9：三个代理模型的运行时间与 CPU 数量对比。多层感知机在保持阻力预测性能相当的同时，运行时间表现更佳。

多层感知机模型在计算效率与拖曳最小化准确率之间实现了最佳权衡。高斯过程模型表现不俗，但由于内部的核矩阵运算，运行时间较长。随机森林的表现中等，但随着 CPU 数量增加，其运行时间扩展性较差。这项实验表明，代理模型选择和并行度会直接影响优化速度和稳定性，验证了 ALMAB-DC 在不同计算环境中的适应性。

### 案例3：强化学习策略优化

目标：调整强化学习（RL）策略的超参数和架构选择，以最大化连续控制和驾驶任务中的期望回报。基准测试：

**MuJoCo**（Todorov等人，2012年）：用于机器人学和生物力学的物理引擎；常用于 HalfCheetah、Hopper、Ant 和 Humanoid 任务。

**CARLA**（Dosovitskiy等人，2017年）：开源城市驾驶模拟器，具有逼真的 c

物理学、环境变化和传感器仿真（激光雷达、摄像机）。

搜索空间：

(1) 策略学习率和网络深度/宽度 (2) 折扣因子 $\gamma \in [0.90, 0.999]$(3) 熵正则化系数 (4) 重放缓冲区大小或批大小

**挑战与方法：** 强化学习（RL）存在若干固有挑战，使高效优化变得复杂。学习过程通常带有噪声，且对超参数选择高度敏感，参数的微小变化往往会导致性能大幅波动。每次实验试验可能需要数千甚至数百万个环境步数，从而产生高昂的计算成本并延长训练时间。此外，学习到的策略容易过拟合或不稳定，尤其是在复杂或随机环境中，回报分布可能随时间发生变化。这些问题使系统性探索和可靠评估尤为困难，凸显了对自适应不确定性感知优化策略的需求。

为克服这些局限，ALMAB-DC 框架将强化学习优化融入具有统计基础的分布式计算范式中。每种超参数配置都被视为多臂老虎机表述中的一个臂，智能体维护期望回报分布的后验估计，并据此引导探索与利用。框架通过量化策略性能的认知不确定性来引入主动探索机制，例如利用集成评论家或自助采样 Q 网络，使系统能够将训练重点放在不确定但可能有前景的配置上。分布式训练代理异步执行独立的强化学习试验，并共享观测到的奖励轨迹，以更新全局性能估计，而无需严格同步。最后，利用图形处理器加速并行化环境仿真（例如向量化MuJoCo运行）和策略梯度更新，使用 JAX 或 PyTorch 等框架大幅提升计算吞吐量和可扩展性。

## 3.3 评估协议

为评估所提出的 ALMAB-DC 框架的有效性，我们采用了若干经验评估标准。一个关键评价指标是累积遗憾（$R_T$），它通过衡量算法获得的奖励与始终选择最优配置的预言机所获得的奖励之间的差异，量化随时间累积的总遗憾。较低的 $R_T$表示在不确定性下的决策更有效。

另一项指标是样本效率，定义为达到目标性能水平所需的昂贵函数评估次数，例如模型训练或仿真。较高的样本效率表明算法能够以较少的评估次数实现收敛。我们还会考察墙钟时间，即达到指定目标值或模型准确性所需的实际经过时间。在不同的硬件设置下，包括仅CPU、多CPU和可选GPU环境，这一指标尤为重要。如果使用 GPU，我们还会监测GPU 利用率，即执行期间实际使用的GPU算力占比。这有助于了解系统的硬件效率及其在多个设备上的扩展能力。通信开销通过统计每轮中分布式智能体之间交换的消息数量和数据量来评估。

这对于了解系统在去中心化环境中的协调效果至关重要。最后，我们会评估收敛速度，即算法趋近最优值的速度。这可以通过遗憾、损失函数值或其他特定于任务的指标来量化。

为通过实证验证该框架，我们提出三个核心假设。第一个假设（H1：效率假设）认为，与 BOHB、SMAC 以及随机搜索或网格搜索等基线方法相比，ALMAB-DC 能够实现更低的累积遗憾，并且只需更少的样本即可收敛。第二个假设（H2：可扩展性假设）认为，随着智能体数量增加，ALMAB-DC 的分布式实现能够在吞吐量和收敛速度方面实现线性或近线性扩展。第三个假设（H3：加速假设）推测，与仅 CPU 版本相比，启用 GPU 的智能体能够显著提升运行时性能，具体表现为实际运行时间缩短、GPU 利用率提高。

## 基线与比较方法

为严格评估所提出的**ALMAB-DC** 框架的性能与效率，我们将其与一系列成熟且多样的优化方法进行基准测试。这些基线方法代表了不同类别的算法，涵盖从经典搜索策略到现代自动化机器学习工具的各种方法。每项比较均在相同计算预算、固定实际运行时间和受控资源约束下进行。

1. **贝叶斯优化：** 我们将 BOHB（Falkner 等，2018）、SMAC（Hutter 等，2011）和树结构 Parzen 估计器（Bergstra 等，2011）作为对比方法。这些都是基于模型的优化方法，通过使用代理模型（例如高斯过程、随机森林）和采集函数（例如期望改进）来引导采样。BOHB 将贝叶斯优化与 HyperBand 的提前停止机制相结合，以提高预算利用效率。

2. **网格搜索和随机搜索：** 网格搜索在离散网格上穷举搜索空间，而随机搜索则从中均匀采样点。这些方法作为基准，供我们衡量 ALMAB-DC 中自适应采样的效率。

3. **独立多臂老虎机算法：** 我们实现了 UCB1（Auer 等，2002b）和汤普森采样（Russo 等，2018），但不使用主动学习或分布式协调。这样可以单独考察 MAB 算法本身对遗憾最小化的影响。

4. **无主动学习的并行多臂老虎机：** 此基线不使用主动学习层，而采用多智能体多臂老虎机策略。每个分布式智能体独立运行置信上界或汤普森采样，以展示协同式信息采样的价值。

5. **Optuna 与异步连续减半算法：** Optuna（Akiba 等，2019）与异步连续减半算法（Li 等，2020）相结合，构成了现代自动机器学习基线。异步连续减半算法会尽早大力剪枝表现不佳的配置，因此在分布式和预算受限环境中十分有效。

所有方法均在固定预算（例如 GPU 小时、评估次数）和一致的实验设置下进行评估。评价指标包括累积遗憾、收敛时间、墙钟时间成本和 GPU 利用率。

## 实验设置与基础设施

为严格评估 ALMAB-DC 框架，我们在多个计算环境和软件栈中实现并测试该框架，从而全面分析其性能、可扩展性和硬件适应性。

### 计算环境

1. **单节点 CPU：** 一种基准配置，用于在不使用硬件加速的情况下测量串行性能和算法开销。2. **单节点 GPU：** 中等规模的配置，使用一到两块高性能 GPU（例如 NVIDIA A100）来加速推理、模型训练和批量评估。3. **多 GPU 分布式集群：** 一种高性能计算（HPC）集群，其中配备多块 GPU 的节点通过 Ray 或 MPI 等分布式调度器进行协调。智能体通过异步通信来共享评估结果和多臂老虎机统计量。

每项实验都在固定的计算预算下进行，例如运行 12 小时或最多进行 500 次函数评估。我们使用自动化日志记录和监控来收集详细的性能统计数据。

## 3.4 利用 GPU 架构

现代优化工作负载，尤其是机器学习和仿真领域的工作负载，需要大量计算资源。GPU（图形处理器）是高度并行的处理器，专为矩阵运算和单指令多数据（SIMD）工作负载而优化，因此非常适合涉及深度学习、主动学习和基于仿真的评估的大规模优化任务。

**GPU 在 ALMAB-DC 中的作用**ALMAB-DC 框架在流水线的多个关键阶段利用 GPU 加速，以实现高吞吐量、低延迟的优化。模型训练任务，包括拟合深度神经网络、贝叶斯神经网络和高斯过程代理模型，均使用 PyTorch（Paszke 等，2019）和 JAX 等框架在 GPU 上执行。候选配置无论表示超参数设置还是设计参数，均通过向量化推理和仿真流水线并行评估，从而实现高效的批量评估。

在主动学习中，熵、间隔或互信息（例如 BALD）等采集函数必须针对大型未标注样本池进行计算。这些不确定性指标可从 GPU 加速中显著受益，从而实现快速的选择周期。多臂老虎机策略评估——尤其是汤普森采样和置信上界等算法的评估——依赖于对数百个潜在臂进行实时评分。RAPIDS cuDF 和 cuML 等 GPU 加速库（Team，2023）有助于快速进行后验更新和臂选择。此外，收集到新的奖励观测后，必须频繁重新训练代理模型。XGBoost-GPU 和 cuML 的集成方法等 GPU 加速学习器可用于最大限度地缩短重新训练延迟并提高响应能力。

**相关库和框架** ALMAB-DC 集成了一套成熟的 GPU 加速库，以实现其可扩展的模块化架构。RAPIDS（Team，2023）是其中的核心组件，提供基于 CUDA 的即插即用替代方案，通过 cuDF（类似于 pandas）和 cuML（类似于 scikit-learn）处理数据并执行机器学习任务。在 cuML 中，ALMAB-DC 使用支持 GPU 的线性模型、聚类算法、主成分分析和决策树，为主动学习训练快速且响应灵敏的代理模型。

XGBoost-GPU 也用于高性能梯度提升决策树，常在优化工作流中充当奖励预测器。采用 CUDA 的 PyTorch（Paszke 等，2019）同时支持深度学习和贝叶斯神经网络模型，可在多个 GPU 上实现动态计算与快速张量运算。对于需要高吞吐量、低延迟推理的部署场景，ALMAB-DC 支持与 TensorRT 和 ONNX Runtime 集成；二者均可为在边缘端或云端 GPU 上运行的已训练模型提供高度优化的执行环境。

**局限性与实践考量** 尽管 ALMAB-DC 具有诸多优势，但仍有一些局限性值得注意。首先，在极高维空间中，代理模型——尤其是高斯过程——可能会面临计算效率低下和预测质量下降的问题，进而影响用于采集的 uncertainty estimates 的可靠性。其次，主动学习的效果取决于代理模型的质量，尤其是在数据稀疏或含噪的早期阶段。第三，虽然该框架旨在支持分布式和 GPU 加速执行，但内存有限、任务时长不均或节点能力异构等硬件限制可能会降低并行效率。此外，对于小规模问题，分布式调度带来的协调开销可能超过其性能收益。最后，理论遗憾界假设奖励噪声服从次高斯分布且分布平稳，而这些假设在动态环境中可能并不成立。应对这些挑战也为未来的扩展提供了机会，例如采用自适应遗憾策略、可扩展代理学习，以及更稳健的智能体协调策略。

## 3.5 应用示例

ALMAB-DC在多个受益于 GPU 加速的高计算量领域展现出很强的适用性。在强化学习（RL）中，尤其是在 MuJoCo 和 CARLA 等仿真环境中（Todorov等人，2012年；Dosovitskiy等人，2017年），该框架利用 GPU 线程并行执行多个智能体的轨迹展开，显著提高数据吞吐量，并稳定学习动态。Stable Baselines3 和 RLlib 等框架通过支持向量化环境进一步提升了这一能力，使 GPU 加速同时应用于环境仿真和策略训练。

在计算工程领域，ANSYS Fluent（Inc.，2023年）等 GPU 加速求解器以及 GPU 驱动的代理模型，能够快速近似计算成本高昂的计算流体力学计算。近期进展，包括基于 GPU 的变量固定技术（Yamamoto等人，2023年），则通过利用张量级并行，进一步加速组合优化任务。

ALMAB-DC 还提升了计算机视觉中的主动学习工作流。对于 CIFAR-10 和 ImageNet 等图像分类任务（Krizhevsky and Hinton，2009年；Deng 等，2009年），图形处理器加速推理能够快速估计大型未标注数据集中的不确定性，从而大幅缩短学习循环选择阶段的实际运行时间。

在超参数调优的早期阶段，可能需要同时评估数百种候选配置。框架利用图形处理器并行化的采集评分来计算

所有老虎机臂的评价指标。这使得决策几乎能够实时进行，并能高效探索配置空间。

**对优化效率的影响** 令$T_{\text{cpu}}$和$T_{\text{gpu}}$分别表示 CPU 和图形处理器上每种配置的平均评估时间。对于批大小$B$和$N$个并发智能体，总加速比可表示为：

$$\text{Speedup} = \frac{T_{\text{cpu}}}{T_{\text{gpu}}} \times B \times N$$

深度学习和基于仿真的任务的实证结果表明，加速比达到 10×至 50×，显著缩短了累计优化时间，并使其能够实际应用于高维搜索空间。

### 实施挑战与实验考量

当前对 ALMAB-DC 的评估存在一项重要局限：缺少消融研究，无法比较启用和禁用多臂老虎机（MAB）组件时的性能。尽管基于 MAB 的调度是该框架设计的重要组成部分，但其对样本效率和遗憾最小化的具体贡献尚未通过实证单独评估。未来的工作应与禁用 MAB 的基线进行受控比较，以量化赌博机策略对性能提升的作用。此外，在工程实现层面仍存在若干挑战，尤其是在 GPU 加速执行方面。大规模代理模型和高吞吐量批量评估可能会超出可用 GPU 内存，因此需要采用内存感知模型蒸馏、动态批处理或混合精度推理等技术。此外，对于规模较小或计算量较低的任务，启动 GPU 内核的开销可能超过其带来的收益，因此在某些情况下，使用中央处理器执行更为高效。在分布式多GPU环境中，带宽限制和节点间通信延迟——尤其是在 GPU 到 GPU 或 GPU到CPU数据传输期间——可能导致同步瓶颈并降低并行效率。

为应对这些限制，该框架可采用 GPU 感知调度策略、内核融合以尽量减少启动开销，以及混合精度计算以降低内存消耗。此外，结合实时硬件性能分析所提供的信息，引入自适应任务分配机制，或可提升异构计算环境中的吞吐量和资源利用率。

## 4 结论与未来方向

我们提出**ALMAB-DC**，将主动学习、多臂老虎机算法和分布式计算整合为统一的优化框架，以应对高成本黑箱任务的计算需求。ALMAB-DC 将有统计依据的决策与异步并行评估和图形处理器加速相结合，为高维科学与工程领域中可扩展、样本高效的优化奠定了坚实基础。该框架推动了一类新型智能优化系统的发展，使学习、适应与计算能够紧密耦合。此外，ALMAB-DC 优先考虑资源效率、易用性和模块化，因此适用于从自动机器学习到仿真引导设计和实验科学等多种应用。未来有若干值得探索的发展方向，包括整合元学习 和

终身学习 以支持相关任务之间的迁移，并使用基于强化学习的元控制器 来管理非平稳环境中的探索—利用动态。通过不确定性感知神经采集模型 和基于图的协调方案减少通信开销和冗余采样，可进一步提升分布式执行效率。将 ALMAB-DC 扩展到仿真与真实世界实验协同演进的混合环境中，有望实现科学发现中的闭环实时优化。其他补充性改进，例如面向混杂领域的因果老虎机模型、用于加速收敛的零成本代理模型热启动 以及面向隐私敏感系统的联邦式部署，将进一步拓展其应用范围和适用性。作为本项目的下一步，我们旨在让 ALMAB-DC 更快速、计算效率更高，同时也更具可解释性、适应性和可部署性。随着持续发展，它有望演进为一个下一代平台，在复杂、分布式且不断演变的环境中实现高影响力的智能优化。

# 附录A：遗憾分析概要

公式(10)和(11)中的遗憾界是根据多臂老虎机文献中的标准结果推导得出的。

对于基于UCB的变体（公式10），算法通过最大化上置信指数来选择臂，在利用与探索之间取得平衡。该界限依据 Auer 等人（2002）的研究得出，表明在适当选择置信项的情况下，累积遗憾随轮次数 $T$ 呈对数级增长，具体为 $O\left(\sum_{i:\Delta_i>0}\log\frac{T}{\Delta_i}\right)$，其中$\Delta_i$ 是最优臂的平均奖励与臂 $i$的平均奖励之差。

对于汤普森采样变体（公式11），遗憾界遵循贝叶斯后验采样框架。正如Agrawal和Goyal（2012年）所示，在伯努利奖励假设和适当的先验初始化条件下，累积遗憾按 $O\left(\sqrt{KT}\log T\right)$增长。我们的分析将这一结果应用于代理模型驱动的主动学习场景，其中奖励是代理模型改进的函数。

在这两种情况下，遗憾界均在奖励噪声为次高斯且各次评估相互独立的假设下成立。为简洁起见，此处省略详细推导；这些推导可直接从经典遗憾分析得出，只需针对主动学习引导的采样设置稍作调整。

# 附录B：软件栈

1. **核心框架（Python）：**

   (a) **PyTorch**（Paszke 等，2019）用于训练深度学习模型，这得益于其动态计算图和对图形处理器的支持。(b) **cuML 和 RAPIDS**（Team，2023）提供常见机器学习算法的图形处理器加速实现，包括基于置信上界的多臂老虎机算法、k-means 聚类和随机森林。(c) **Ray**（Moritz 等人，2018b）用于协调分布在各节点上的智能体，从而支持共享内存和基于任务的通信。

2. **主动学习与多臂老虎机模块：**

(a) **查询选择算法：** ALMAB-DC 框架包含多种主动学习策略的模块化实现：

i. **不确定性采样：**选择能够使模型置信间隔最小的查询点$x \in U$，即$x = \text{argmin}_x |P(y = 1|x) - 0.5|$，适用于二元分类器。在 Python 中通过熵和基于间隔的评分实现。ii. **BALD（基于分歧的贝叶斯主动学习）：**通过计算预测与模型后验分布之间的互信息来量化认知不确定性：

$$\text{BALD}(x) = \mathbb{H}[y|x, \mathscr{D}] - \mathbb{E}_{p(w|\mathscr{D})}[\mathbb{H}[y|x, w]]$$

其中，$\mathbb{H}$ 表示熵，$p(w|\mathscr{D})$ 是模型权重的后验分布。我们使用蒙特卡洛 dropout 或贝叶斯神经网络来估计此项。

iii. **核心集选择：**对于依赖核函数或特征嵌入的模型，该方法通过使用$k$-中心或贪心覆盖算法选择代表性样本，从而近似整个数据集，并通过贪心次模优化高效求解。

(b) **老虎机策略：**

i. **UCB1（上置信界）：**通过以下方式实现探索-利用权衡：

$$a_t = \arg\max_i \left( \hat{\mu}_i + \sqrt{\frac{2 \log t}{n_i}} \right)$$

其中，$\hat{\mu}_i$ 是经验奖励均值，$n_i$ 是臂$i$被选中的次数。

ii. **汤普森采样：** 一种贝叶斯方法，从臂$i$的后验分布中采样其奖励，并选择样本值最高的臂。

iii. **上下文老虎机：** 当存在输入特征（例如配置嵌入）时使用。通过 LinUCB 或基于神经网络的近似方法在 PyTorch 中实现。

老虎机和主动学习模块使用 Python、NumPy、Py-Torch 和 cuML 实现，并启用图形处理器加速，以进行批量臂评分和推理。

3. **可视化与实验日志记录：**

(a) **交互式可视化工具：**

i. **Matplotlib**和 **Seaborn** 用于绘制静态诊断图表，例如累积遗憾、各试验的方差和收敛率。 ii. **Plotly** 支持使用交互式仪表板可视化查询轨迹、奖励分布和资源随时间的利用情况。

(b) **实验跟踪与监控：**

i. **TensorBoard** 与PyTorch集成，用于可视化损失曲线、臂选择频率、GPU 利用率和模型学习动态。ii. **Weights & Biases（可选）：**对于大规模实验，我们还集成 W&B，以便跨分布式节点实时跟踪指标。iii. **结构化日志记录：**所有奖励指标、配置和评估时间均以 JSON 或 Parquet 格式存储，用于事后统计分析和可视化。

**与 R 的可选集成**

虽然 Python 是主要的实现环境，但我们也提供可选的 R 集成，以支持高级统计分析和可视化：

**数据交换：**使用 reticulate（Allaire 等，2018）和 rpy2（rpy2 De- velopment Team，2023）软件包，在 Python 和 R 环境之间传递数据框或数组。

**统计评估：**使用 R 内置的假设检验函数（例如 t.test、aov、wilcox.test），进行实验后的统计显著性检验。

**可视化：**使用 ggplot2（Wickham，2016）生成可用于发表的图表和统计图形。

这种混合设置让研究人员能够兼得 Python 可扩展学习框架的优势和 R 强大的统计分析能力，尤其适用于事后分析和报告。

# 参考文献


AI，N.（2023年）。借助 cuML 加速机器学习：一款可扩展的 GPU 加速机器学习库。https://github.com/rapidsai/cuml。GitHub 仓库。

Akiba, T., Sano, S., Yanase, T., Ohta, T., 和 Koyama, M.（2019年）。Optuna：新一代超参数优化框架。载于第 *25* 届 *ACM SIGKDD* 国际知识发现与数据挖掘会议论文集，第 2623–2631 页。

Allaire, J., Ushey, K., 和 Tang, Y.（2018年）。*reticulate*：*R* 的 *Python* 接口。

Amdahl, G. M.（1967年）。单处理器方法在实现大规模计算能力方面的有效性。*AFIPS* 会议论文集，30：483–485。

Auer, P., Cesa-Bianchi, N., 和 Fischer, P.（2002a）。多臂老虎机问题的有限时间分析。机器学习，47(2–3)：235–256。

Auer, P., Cesa-Bianchi, N. 和 Fischer, P.（2002b）。多臂老虎机问题的有限时间分析。载于《机器学习》，第47卷，第235–256页。

Bachman, P., Sordoni, A. 和 Trischler, A.（2017）。有限标注数据下的学习：缩小假设空间。载于第*34*届国际机器学习会议论文集，第370–379页。

Bergstra, J., Bardenet, R., Bengio, Y. 和 K´egl, B.（2011）。超参数优化算法。载于神经信息处理系统进展，第24卷。

Chaloner, K. 和 Verdinelli, I.（1995）。贝叶斯实验设计：综述。《统计科学》，10(3)：273–304。

Chen, W., Li, Z. 和 Wang, T.（2022）。分布式多臂老虎机：一种注重通信效率的视角。*arXiv* 预印本 *arXiv:2203.04512*。

Dean, J. and Ghemawat, S. (2008). Mapreduce：大规模集群上的简化数据处理。《*ACM*通信》, 51(1):107–113.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet：一个大型分层图像数据库。载于*2009*年 *IEEE*计算机视觉与模式识别会议 *(CVPR)*, 页码 248–255。IEEE。Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla：一款开放式城市驾驶模拟器。载于机器人学习会议 *(CoRL)*, 页码 1–16。Efron, B. and Tibshirani, R. J. (1994).《自助法导论》。CRC出版社。
Falkner, S., Klein, A., and Hutter, F. (2018). Bohb：大规模场景下稳健高效的超参数优化。载于第*35*届国际机器学习会议 *(ICML)* 论文集, 页码 1437–1446。
Gustafson, J. L. (1988). 重新审视阿姆达尔定律。《*ACM*通信》, 31(5):532– 533。
Hanna, J., Jain, L., Kandasamy, K., and Daum´e III, H. (2022). 使用贝叶斯求积实现通信高效的并行老虎机学习。载于第*39*届国际机器学习会议 *(ICML)* 论文集, 第151卷, 页码 8301–8321。PMLR。He, K., Zhang, X., Ren, S., and Sun, J. (2016). 用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议 *(CVPR)* 论文集, 页码 770–778。He, Q., Wang, M., Liu, X., Wang, Z., and Kong, F. (2024). 跨越鸿沟的学习：结合异构离线与在线数据的混合多臂老虎机。*OpenReview*预印本。Houlsby, N., Husz´ar, F., Ghahramani, Z., and Lengyel, M. (2011). 用于分类与偏好学习的贝叶斯主动学习。*arXiv*预印本 *arXiv:1112.5745*。Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). 面向通用算法配置的序贯模型优化。载于第*5*届学习与智能优化国际会议 *(LION)* 论文集, 页码 507–523。Imbens, G. W. and Rubin, D. B. (2015).《统计、社会与生物医学科学中的因果推断》。剑桥大学出版社。Inc., A. (2023).《*ANSYS Fluent* 用户指南，*2023 R1*版》。Jasak, H., Jemcov, A., and Tukovic, Z. (2007). Openfoam：用于复杂物理模拟的C++ 库。载于数值动力学耦合方法国际研讨会。萨格勒布大学机械工程与造船学院。Joulani, P., Gy¨orgy, A., and Szepesv´ari, C. (2013). 延迟反馈下的在线学习。载于神经信息处理系统进展 *(NeurIPS)*, 页码 1453–1461。Kennedy, M. C. and O'Hagan, A. (2001). 计算机模型的贝叶斯校准。《皇家统计学会会刊：*B*辑（统计方法论）》, 63(3):425–464。
Krizhevsky, A. and Hinton, G. (2009). 从微小图像中学习多层特征。技术报告 0，多伦多大学，多伦多，安大略省。Landgren, P., Srivastava, V., and Leonard, N. E. (2016). 多臂老虎机中的分布式探索。载于神经信息处理系统进展, 第29卷。

Lattimore, T. 和 Szepesv´ari, C.（2020）。老虎机算法。剑桥大学出版社。Li, L.、Jamieson, K.、DeSalvo, G.、Rostamizadeh, A. 和 Talwalkar, A.（2018）。大规模并行超参数调优。载于第*34*届国际机器学习会议论文集，第1725–1734页。Li, L.、Jamieson, K.、Rostamizadeh, A.、Gonina, E.、Hardt, M.、Recht, B. 和 Talwalkar, A.（2020）。大规模并行超参数调优系统。载于机器学习与系统论文集（*MLSys*）。Li, M.、Andersen, D. G.、Park, J.、Smola, A. J.、Ahmed, A.、Josifovski, V.、Long, E. J.、Shekita, E. J. 和 Su, B.-Y.（2014）。利用参数服务器扩展分布式机器学习。*OSDI*。Li, Y.、Chen, Y. 和 Xing, E. P.（2019）。分布式系统中的可扩展主动学习。*IEEE*知识与数据工程汇刊，31(4):722–735。Liu, H.、Simchowitz, M. 和 Jordan, M. I.（2021）。用于高效强化学习的主动奖励学习。载于国际学习表征会议（*ICLR*）。Liu, L.、Li, J.、Wang, J.、Zhao, S. 和 Duan, C.（2026）。Comb-cs：用于DER预测联邦学习的贡献感知与基于多臂老虎机的客户端选择。《先进工程信息学》。Moritz, P.、Nishihara, R.、Wang, S.、Tumanov, A.、Liaw, R.、Liang, E.、Elibol, M.、Yang, Z.、Paul, W.、Jordan, M. I. 和 Stoica, I.（2018a年）。Ray：面向新兴人工智能应用的分布式框架。第*13*届*USENIX*操作系统设计与实现研讨会（*OSDI 18*），第561–577页。Moritz, P.、Nishihara, R.、Wang, S.、Tumanov, A.、Liaw, R.、Liang, E.、Elibol, M.、Yang, Z.、Paul, W.、Jordan, M. I. 和 Stoica, I.（2018b年）。Ray：面向新兴人工智能应用的分布式框架。载于第*13*届*USENIX*操作系统设计与实现研讨会（*OSDI*）。Paszke, A.、Gross, S.、Massa, F.、Lerer, A.、Bradbury, J.、Chanan, G.、Killeen, T.、Lin, Z.、Gimelshein, N.、Antiga, L. 等（2019）。PyTorch：一种命令式风格、高性能的深度学习库。神经信息处理系统进展，32。Paul, J. M. 和 Meyer, B. H.（2007年）。重新审视单芯片系统的阿姆达尔定律。《国际并行编程杂志》，35(2):101–123。Rasmussen, C. E. 和 Williams, C. K.（2006年）。《机器学习中的高斯过程》。麻省理工学院出版社。rpy2开发团队（2023年）。*rpy2*：*Python*中的*R*。Russo, D.、Van Roy, B.、Kazerouni, A.、Osband, I. 和 Wen, Z.（2018年）。汤普森采样教程。《机器学习基础与趋势》，11(1):1–96。Sutton, R. S. 和 Barto, A. G.（2018年）。《强化学习：导论》。麻省理工学院出版社，第2版。

Sz¨or´enyi, B.、Busa-Fekete, R.、Heged˝us, I. 和 K´egl, B.（2013年）。分布式探索与利用：多名玩家参与的多臂老虎机算法。*arXiv*预印本 *arXiv:1311.0800*。Tan, M. 和 Le, Q. V.（2019年）。EfficientNet：重新思考卷积神经网络的模型缩放。载于国际机器学习会议论文集（*ICML*）。R. A. 团队（2023年）。RAPIDS：GPU加速数据科学。https://rapids.ai/。英伟达公司。Todorov, E.、Erez, T. 和 Tassa, Y.（2012年）。MuJoCo：用于基于模型控制的物理引擎。载于*IEEE/RSJ* 智能机器人与系统国际会议（*IROS*），第5026–5033页。IEEE。Verma, S.、Jain, R. 和 Kamath, G.（2022年）。具有延迟反馈的异步多臂老虎机。载于*AAAI* 人工智能会议论文集，第36卷，第7992–8000页。Wang, Z. 和 Liu, Q.（2020年）。老虎机与主动学习相遇：二分类与多分类设置的统一视角。载于神经信息处理系统进展（*NeurIPS*）。Wickham, H.（2016年）。*ggplot2*：优雅的数据分析图形。Springer。Yamamoto, R.、Nakamura, K. 和 Yamada, M.（2023年）。利用GPU加速的变量固定实现快速组合老虎机。载于国际机器学习会议论文集（*ICML*）。PMLR。Yu, H.、Zhan, K.、Wang, K.、Zhang, Q. 和 Qin, Z.（2022年）。通过对等方选择实现联邦主动学习。载于第*39*届国际机器学习会议论文集（*ICML*）。Zhang, H.、Xu, Y.、Wang, X.、Chen, H. 和 Qiu, H.（2024年）。采用高效比特级通信的联邦多臂老虎机。*OpenReview*预印本。Zimmert, J. 和 Seldin, Y.（2019年）。一种适用于延迟反馈下随机与对抗性老虎机的最优算法。载于第*36*届国际机器学习会议论文集（*ICML*），第7644–7653页。