

大型语言模型（LLM）在自主执行复杂任务方面展现出巨大潜力，例如网页导航、设备控制、代码编写和维护等。然而，要在涉及多轮决策的任务中实现最佳性能，智能体需要直接优化多轮目标，如成功率，这比仅仅模仿每个回合中最可能的动作更具挑战性。

现有的单轮强化学习算法（如RLHF）在多轮任务中表现不佳，主要原因在于它们无法在多个回合之间进行有效的信用分配。此外，尽管一些研究尝试应用值函数学习方法（如TD-learning），但这些方法在有限的微调数据下泛化能力较差。因此，如何开发一种能够充分利用LLM推理能力的多轮强化学习算法，仍然是一个未解之谜。

**2. 相关工作**

**2.1 LLM智能体的基准测试**

近年来，许多基准测试被提出用于评估LLM智能体在不同场景下的能力，如软件工程、网页导航、设备控制和旅行规划等。然而，大多数基准测试主要关注评估最先进的通用LLM，而缺乏一个研究友好的交互环境和训练任务集来研究多轮强化学习算法。ColBench是第一个专门为多轮强化学习算法设计的基准测试，专注于具有强推理能力的任务，并提供了可靠的功能验证器。

**2.2 多轮强化学习算法**

与单轮场景不同，多轮强化学习算法需要智能体在完成任务时进行一系列动作。尽管一些早期研究直接应用了单轮强化学习方法（如REINFORCE、[DPO](https://zhida.zhihu.com/search?content_id=255344436&content_type=Article&match_order=1&q=DPO&zhida_source=entity" \t "_blank)和PPO），但这些方法在任务时间跨度较长时往往表现不佳。最近的研究尝试应用更先进的深度强化学习技术（如Bellman bootstrapping和Path Consistency）来减少长期方差，但这些方法在多轮任务中的表现仍然有限。

**3. 协作智能体基准测试（ColBench）**

**3.1 设计原则**

ColBench的设计遵循三个核心原则：

1. **任务复杂性**：基准测试应反映现实世界中的复杂推理和泛化挑战。
2. **最小工程开销**：为了快速研究原型设计，基准测试应尽量减少工程开销。
3. **任务多样性**：基准测试应包含足够的任务，以确保不同多轮强化学习算法的可扩展性。

**3.2 后端编程协作**

在**后端编程协作**任务中，智能体需要与人类模拟器协作编写一个自定义的Python函数。任务开始时，智能体获得函数的高级描述和签名，但许多具体细节（如条件和边缘情况）并未提供，智能体需要通过提问来获取更多信息。人类模拟器会根据参考代码提供简要的自然语言解释，但不会直接编写代码。交互最多进行10轮，最终通过10个隐藏的单元测试来评估智能体的成功与否。

**3.3 前端设计协作**

在**前端设计协作**任务中，智能体需要与人类模拟器协作设计一个网页。任务开始时，智能体获得网页的高级描述，但许多具体细节（如布局和颜色方案）并未提供。智能体在每个回合中编写HTML代码，并通过浏览器渲染网页。人类模拟器会检查渲染的网页与参考网页的差异，并向智能体描述这些差异。最终通过CLIP嵌入的余弦相似度来评估智能体的表现。

**4.1 问题设置**

在多轮协作任务中，智能体需要与人类模拟器进行多次交互，逐步完成任务。研究团队将这一问题建模为一个**有限时域的部分可观测马尔可夫决策过程（POMDP）**，记为 M={O,C,A,T,*μ*1​,R,*N*}。其中：

* O 表示智能体可以观察到的状态空间部分。
* C 表示隐藏的状态空间部分，通常包括任务的关键信息（如参考解决方案）。
* A 表示智能体的动作空间，即智能体在每个回合中生成的响应。
* T 表示状态转移函数，智能体在采取动作后，环境会根据转移函数更新状态。
* *μ*1​ 是初始状态分布，智能体在任务开始时从该分布中采样初始指令和隐藏信息。
* R 是奖励函数，智能体在每个回合中根据其动作和隐藏信息获得奖励。
* *N* 是任务的最大回合数。

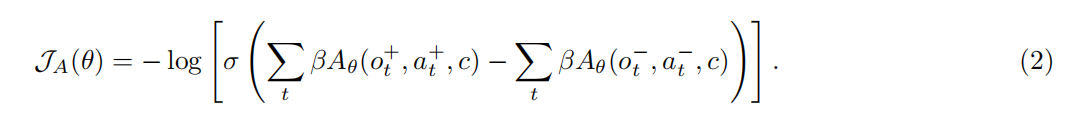
在每个回合 t*t*，智能体观察到历史交互信息 ot*ot*​，并生成一个动作 at*at*​（即一个由多个令牌组成的响应）。用户会根据智能体的动作给出反馈，智能体的状态更新为新的交互历史。任务结束时，智能体根据累积奖励进行评估。

**4.2 学习回合级优势函数**

在多轮任务中，智能体需要在每个回合中做出决策，而这些决策的长期影响往往难以直接评估。为了有效分配信用，研究团队提出直接学习每个回合动作的**优势函数**，而不是传统的值函数。

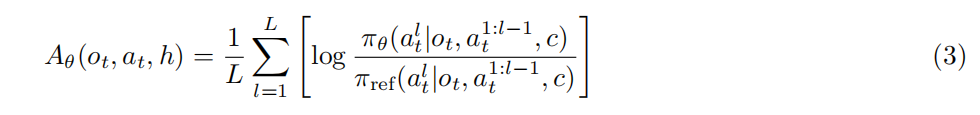
优势函数 *Aπ*(*ot*​,*at*​,*c*) 表示在当前状态(*ot*​,*c*) 下采取动作 *at*​ 的相对收益。与值函数不同，优势函数直接衡量每个动作的效用，而不需要先估计当前状态的值。

研究团队使用**[Bradley-Terry目标函数](https://zhida.zhihu.com/search?content_id=255344436&content_type=Article&match_order=1&q=Bradley-Terry%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0&zhida_source=entity" \t "_blank)**来训练优势函数。具体来说，给定两个轨迹 *τ*+ 和 *τ*−，其中 *τ*+ 是成功轨迹，*τ*− 是失败轨迹，优势函数的目标是最大化成功轨迹中每个动作的优势，同时最小化失败轨迹中每个动作的优势。优势函数的学习目标可以表示为：



其中，*σ* 是sigmoid函数，*β* 是一个超参数，用于控制优势函数的缩放。

为了与LLM的预训练目标保持一致，研究团队将优势函数参数化为LLM的均值对数概率：



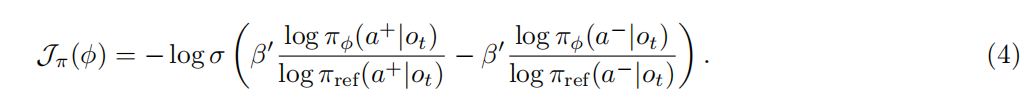
其中，*πθ*​ 是训练中的LLM模型，  ​ 是一个冻结的初始模型，*L* 是响应的长度。通过这种方式，优势函数能够充分利用LLM的推理能力，并在多轮任务中进行有效的信用分配。

**4.3 通过回合级优势优化智能体**

在训练过程中，智能体的策略 *πϕ*​ 无法依赖于隐藏信息 *c*，但这些信息在训练时是可用的。因此，研究团队提出了一种**不对称的演员-评论家结构(**actor-critic structure**)**，其中评论家（优势函数）可以访问隐藏信息，而演员（策略）只能访问交互历史。

具体来说，评论家通过训练时信息（如参考解决方案）来评估每个动作的优势，而策略则根据交互历史生成动作。这种不对称结构使得评论家能够更好地评估策略的动作，从而帮助策略在多轮任务中进行优化。

研究团队使用\*\*DPO（Direct Preference Optimization）\*\*算法来优化策略。在每个回合 t*t*，策略生成多个候选动作，并根据评论家评估的优势值对这些动作进行排序。然后，策略通过DPO损失函数进行优化：



其中，*a*+ 和 *a*− 分别是根据优势值排序后的优选动作和劣选动作，*β*′ 是一个超参数。

通过这种方式，策略能够在每个回合中生成更优的动作，从而在多轮任务中取得更好的表现。

**6. 结论**

本文提出了一个全新的多轮强化学习算法SWEET-RL，并在ColBench基准测试上验证了其有效性。实验结果表明，SWEET-RL在多轮协作任务中显著优于现有的最先进算法，为LLM智能体的训练提供了新的思路。未来的研究可以进一步探索如何更好地利用训练时信息，以开发更有效的多轮强化学习算法。