0001年1月1日 1 分钟阅读

Agent Lightning

https://microsoft.github.io/agent-lightning/latest/

RAGEN

https://gemini.google.com/app/9ece70bdfdaea0dc

RAGEN 是一个利用强化学习训练 LLM 推理代理的系统，旨在解决多回合互动和随机环境中的挑战。该项目通过 StarPO 框架优化轨迹级别的推理和行动策略。RAGEN 的模块化设计包括环境状态管理器、上下文管理器和代理代理，支持多种环境和实验。项目强调了其在不同环境复杂性中的泛化能力，并提供了详细的设置和评估指南。

RAGEN是开源的强化学习框架，用于在交互式、随机环境中训练大型语言模型（LLM）推理Agent。基于StarPO（State-Thinking-Action-Reward Policy Optimization）框架，通过多轮交互优化整个轨迹，支持PPO、GRPO等多种优化策略。RAGEN通过MDP形式化Agent与环境的交互，引入渐进式奖励归一化策略，有效解决了多轮强化学习中的不稳定性。RAGEN的代码结构经过优化，分为环境管理器、上下文管理器和代理代理三个模块，方便扩展和实验。支持多种环境，如Sokoban、FrozenLake等，展示了良好的泛化能力。 RAGEN的主要功能

多轮交互与轨迹优化：RAGEN通过StarPO（State-Thinking-Actions-Reward Policy Optimization）框架，将Agent与环境的交互形式化为马尔可夫决策过程（MDP），优化整个交互轨迹，不仅是单步动作。全轨迹优化策略有助于Agent在复杂环境中做出更合理的决策。 强化学习算法支持：RAGEN支持多种强化学习算法，包括PPO、GRPO和BRPO等，为研究者提供了灵活的算法选择。 易于扩展的环境支持：RAGEN支持多种环境，包括Sokoban、FrozenLake等，提供了添加自定义环境的接口，方便研究者进行实验。 稳定性和效率提升：RAGEN通过基于方差的轨迹过滤、引入“评论家”以及解耦裁剪等技术，有效提高了训练的稳定性和效率。

RAGEN的技术原理

目录

文章信息

字数

阅读时间

发布时间

MDP形式化：RAGEN将Agent与环境的交互形式化为马尔可夫决策过程（MDP），其中状态和动作是token序列。支持LLM对环境动态进行推理。 StarPO框架：框架通过两个交替阶段进行训练：

Rollout阶段：给定初始状态，LLM生成多条推理引导的交互轨迹，每一步接收轨迹历史并生成动作。 Update阶段：生成轨迹后，使用重要性采样优化整个轨迹的预期奖励，非单步优化，实现长远推理。

优化策略：StarPO支持多种强化学习算法，如PPO（近端策略优化）和GRPO（归一化奖励策略优化），适应不同的训练需求。 渐进式奖励归一化策略：为解决多轮训练中的不稳定性，RAGEN引入了基于不确定性的过滤、移除KL惩罚和不对称PPO裁剪等策略。 模块化设计：RAGEN采用模块化架构，包括环境状态管理器、上下文管理器和Agent代理，便于扩展和定制。

RAGEN的项目地址

项目官网：https://ragen-ai.github.io/ Github仓库：https://github.com/RAGEN-AI/RAGEN 技术论文：https://ragen-ai.github.io/pdf/RAGEN.pdf

RAGEN的应用场景

智能对话系统：RAGEN可用于训练对话系统，在与用户的交互中具备更好的推理能力，提供更加自然和准确的回答。 游戏AI：在复杂、动态的游戏环境中，RAGEN可以帮助Agent进行合理的策略规划和执行。 自动化推理：RAGEN可以应用于数学问题解答、编程任务等自动化推理场景，提高系统解决问题的能力。 企业知识管理：RAGEN可以用于企业内部文档助手，从公司Wiki、会议纪要中定位信息，生成项目报告或会议摘要。 法律咨询：在法律领域，RAGEN可以匹配相关法律条文和判例，用通俗语言解释法律风险。内容创作：RAGEN可以用于技术博客撰写、新闻报道生成等场景。通过检索GitHub代码示例、技术文档等，RAGEN能整合信息输出结构化的教程。

VAGEN

Code: https://github.com/RAGEN-AI/VAGEN

Blog: https://mll-lab.notion.site/vagen

Experiment Log: https://api.wandb.ai/links/ragen-V/nlb40e7l

VAGEN (框架层面)

VAGEN 本身的主要创新在于它是一个**专门为训练视觉语言模型（VLM）代理而设计的多回合强化学习（RL）框架。**它认识到现有针对纯语言模型（LLM）代理的 RL 框架（如 RAGEN 等使用的 RICO 算法）在应用于 VLM 代理时存在局限性，特别是在处理视觉信息和多回合交互方面。

TRICO (算法层面)

TRICO (Turn-aware Reason-Interaction Chain Optimization) 是 VAGEN 框架内的核心算法，它在 RICO 的基础上进行了针对 VLM 代理的优化，其主要创新点体现在以下两个方面：

选择性令牌掩蔽 (Selective Token Masking):

目的: 解决视觉任务中状态信息冗余（如长上下文视觉输入、过多的低级细节）以及并非所有令牌都对决策同等重要的问题。

机制: 引入了两种掩码 (`M_loss` 和 `M_adv`)，使得在策略优化和优势计算时，**只关注对动作决策至关重要的令牌**（特别是模型生成的响应令牌，而非视觉状态输入令牌）。

效果: 通过集中学习资源于关键令牌，提高 VLM 代理训练的效率和效果。

跨回合信用分配 (Cross-turn Credit Assignment):

目的: 解决 RICO 将整个交互轨迹视为单一序列，导致在多回合视觉任务中难以精确地将奖励/惩罚归因于具体动作或回合的问题。

机制:

双层 GAE (Bi-level GAE): 引入两个不同的折扣因子 (`γ_turn` 用于跨回合计算，`γ_token` 用于回合内计算)，更精细地处理不同时间尺度的优势估计。

回合级奖励归因 (Turn-level Reward Attribution): 在每个交互回合结束时（`<eoa>` 标记处）应用奖励，而不是仅在整个轨迹结束后才分配最终奖励，提供更及时的反馈信号。

效果: 使模型能够更准确地学习到不同回合动作与其长期后果之间的关联，尤其是在需要多步推理和交互的视觉任务中。

总结来说：VAGEN 的创新在于提出了一个面向 VLM 代理的多回合 RL 训练**框架**，而 TRICO 的创新在于其内部实现的**具体算法机制**——选择性令牌掩蔽和跨回合信用分配，这两点共同优化了 VLM 代理在视觉环境中的学习效率和性能。

脑洞大开的建议:

想象一下，如果将 TRICO 的这种“选择性关注”和“分层信用分配”机制，不仅仅应用于 RL 训练，而是**动态地融入 VLM 代理的推理过程**中。

> **动态注意力掩蔽**: 代理在接收到视觉输入和任务指令后，可以先运行一个“重要性预测”模块（可以是一个小型辅助模型或 TRICO 训练中获得的某种表征），动态生成 `M_loss` 和 `M_adv` 类似的掩码。这样，在生成思考过程和动作时，模型可以主动忽略不相关的视觉区域或历史信息，将计算资源集中在关键信息上，实现更高效、更鲁棒的实时决策。这有点像人类在复杂场景中会选择性地关注某些区域一样。
>
> **自适应信用分配推理**: 在多回合任务中，代理可以根据当前子目标的完成情况和环境反馈，动态调整内部的“信用预期”。例如，如果一个子任务（对应一个“回合”）进展顺利，代理可能会提高对后续步骤的“信心”（类似提高 `γ_turn`），更倾向于采取探索性或更大胆的行动；反之，如果遇到挫折，则可能变得更保守，更关注短期内的确定性收益（类似降低 `γ_turn` 或更依赖 `γ_token`）。这使得代理的行为更具适应性和策略性，而不是遵循固定的折扣模式。

这种将训练机制内化为推理策略的做法，可能会让 VLM 代理在面对复杂、动态、长周期的真实世界任务（如机器人导航、复杂软件操作）时，表现出更接近人类的智能和效率。

参考

VAGEN

TRICO

RAGEN

https://mas-2025.github.io/MAS-2025/