丁彪. 融合 LLaMA 模型与强化学习的智能家居环境任务规划系统[J]. 智能计算机与应用,2026,16(6):220-225. DOI:10.20169/j.issn.2095-2163.24101902

融合 LLaMA 模型与强化学习的智能家居环境任务规划系统

丁 彪
(重庆理工大学 理学院, 重庆 400054)

摘 要: 大语言模型使用大量的文本数据进行训练,从而具有丰富的世界知识。然而,使用大语言模型处理现实世界的问题时常常表现不佳,因为大语言模型的知识与实际环境并不一致。相比之下,强化学习是通过与环境的交互来逐步学习策略的,因此始终能与环境保持一致,但在缺乏一定先验知识的情况下强化学习的样本效率是较低的。针对上述问题,本文提出了一种结合大语言模型和强化学习来进行任务规划的系统,其中将大语言模型作为决策者,并由强化学习算法来指导模型做出决策。仿真实验结果表明,与原始模型相比,微调后的模型性能明显提升,任务执行过程中的平均奖励值从 0.139 提升至 0.760,任务执行效率和稳定性均优于原始模型。

关键词: 大语言模型; 强化学习; 任务规划
中图分类号: TP312 **文献标志码**: A **文章编号**: 2095-2163(2026)06-0220-06

Task planning system for intelligent home environment integrating LLaMA model and reinforcement learning

DING Biao
(College of Science, Chongqing University of Technology, Chongqing 400054, China)

Abstract: Large language models are trained on extensive text data, which gives them a wealth of world knowledge. However, when applied to real-world problems, they often underperform because their knowledge does not align with the actual environment. In contrast, reinforcement learning learns strategies through interactions with the environment, thus always maintaining consistency with the environment. However, reinforcement learning tends to have lower sample efficiency without prior knowledge. To address these challenges, this paper proposes a system that integrates large language models with reinforcement learning for task planning. In this approach, the large language model serves as the decision-maker while the reinforcement learning algorithm guides the decision-making process. Experimental results demonstrate that the performance of the fine-tuned model has significantly improved compared to the original model. The average reward during task execution increased from 0.139 to 0.760, and the task execution efficiency and stability are better than the original model.

Key words: large language model; reinforcement learning; task planning

0 引 言

近年来,大语言模型[1]的兴起和发展对自然语言处理和人工智能领域带来了革命性的影响[2]。Ahn 等学者[3]将大语言模型的语义知识与机器人实际的物理行动能力相结合,使得机器人可以利用语言知识完成长时间的复杂任务。Cai 等学者[4]利用大语言模型生成并利用工具来解决复杂问题。然而,大语言模型通常依赖预训练的静态数据,并不具备对当前环境的实时感知能力。这意味着模型对现实世界的即时变化和动态信息缺乏认知,无法做出准确的决策[5]。例如在进行具体任务编排时,大语言模型可能会生成当前环境中无法完成的操作,例如拿起环境中不存在的物体。相反,强化学习[6]通过持续与环境交互,从头开始学习策略,确保智能体能够与环境保持一致。Mutti 等学者[7]提出的 αMEPOL 算法通过无监督的多环境交互训练显著提升了样本效率与适应性,这表明多环境交互有助

基金项目: 重庆市自然科学基金(CSTB2022NSCQ-MSX0829)。
作者简介: 丁 彪(1998—),男,硕士研究生。主要研究方向:人工智能,自然语言处理。Email:1328425917@stu.cqut.edu.cn。
收稿日期: 2024-10-19

于强化学习代理的有效对齐。然而，传统强化学习方法往往从随机策略开始学习，缺乏先验知识的支持，从而导致样本效率低下，Kumar 等学者[8]通过将先验知识与策略的初始化和训练期间的探索相结合的方法来提高样本效率。Ouyang 等学者[9]提出的 RLHF 方法说明来自人类反馈的强化学习方法对大语言模型对齐和内容质量提升的有效性。因此，本次研究大语言模型作为决策支持的工具，负责接收观察到的信息并生成在环境中执行的动作指令，实验中使用大语言模型进行具体任务的决策，通过检测环境内的物品并发出指令，对给定的任务进行规划。同时为了克服大语言模型的局限性，在实验中引入强化学习，通过强化学习代理在执行过程中与环境的不断交互与试错获得的反馈来调整大语言模型的参数，从而实现大语言模型生成策略的修正，提高任务完成的效率和准确率。

1　相关工作

1.1　大语言模型工作原理

本文使用 LLaMA[10] 大模型进行任务规划，LLaMA 模型是基于原始 Transformers[11] 架构的生成式预训练模型，能够生成、理解和处理自然语言。大语言模型通常通过自回归语言建模[12]的方式进行训练，即通过给定上下文预测下一个词的概率。其训练目标可以表示为最大化条件概率 $P(w_t \mid w_{1:t-1})$，公式为：

$$L = -\sum_{t=1}^{T} \log P(w_t \mid w_{1:t-1};\theta) \tag{1}$$

其中，L 表示损失函数的值；w_t 表示在时间步 t 的目标词；θ 表示模型参数；T 表示序列长度；$P(w_t \mid w_{1:t-1};\theta)$ 表示模型在给定上下文 $w_{1:t-1}$ 下预测目标词 w_t 的概率。

1.2　近端策略优化算法

强化学习中有多种经典的算法，其中具有代表性的包括 Q-learning[13-14]、DQN[15] 和近端策略优化[16-17]（Proximal Policy Optimization，PPO），其中 PPO 算法是基于策略梯度方法[18]的。策略梯度方法是一类直接对策略进行优化的强化学习方法，通过最大化累积奖励的期望来更新策略参数。在这种方法中，优势函数 $A(s_t,a_t)$ 用于衡量一个动作相对于平均策略的优越性，并引导策略的更新，使得策略学习更加高效和稳定。优势函数的定义为：

$$A(s_t,a_t) = Q(s_t,a_t) - V(s_t) \tag{2}$$

其中，s_t 表示在时间步 t 时的状态，a_t 表示该时间步选择的动作；$Q(s_t,a_t)$ 表示动作值函数；$V(s_t)$ 表示状态值函数。同时，PPO 算法使用 clip 操作限制新旧策略的更新幅度，从而防止策略在一次更新中变化过大。

LLaMA 模型作为大语言模型，其参数规模巨大、且微调过程中需要保持原有模型的能力[19]，而 PPO 的稳定性和渐进式策略更新特性使其能够在不破坏语言模型原有能力的情况下，逐步优化策略。

2　实验方法

2.1　初始策略生成

通过大语言模型直接生成的操作可能在环境中是无效的，例如大语言模型给出的操作是“拿起苹果”，但是环境中可能并不存在苹果，从而导致动作执行失败，这就是前面提到的大语言模型的知识与实际环境不一致的问题。本文采用的方法是根据当前的环境信息人工整理出所有可执行的动作生成一个动作表 $A = \{a_1,a_2,\cdots,a_k\}$，其中 k 是指动作表的长度，在环境中执行的动作都来源于这个动作合集 A，然后将观察环境得到的信息 S 一起合并后作为大语言模型的输入，大语言模型通过计算动作表 A 中的各个动作的 token 序列的联合概率来确定下一步的最优动作，任务规划的具体过程如图 1 所示。

对于每个动作 a_i，模型会计算其 token 序列的联合概率 $P(a_i \mid S)$，其中 S 为环境提示，即模型接收到的观察信息。联合概率可以通过以下公式来表示：

$$P(a_i \mid S) = \prod_{t=1}^{k} P(w_t^i \mid w_1^i, w_2^i, \ldots, w_{t-1}^i, S) \tag{3}$$

其中，w_t^i 表示动作 a_i 中的第 t 个 token，k 表示该动作所包含的 token 的总数。式(3)表示模型根据前面的 token 和环境提示 S 来预测当前 token 的概率。

模型会对动作表中的每个动作 a_i 的联合概率进行计算，并选择联合概率 $P(a_i \mid S)$ 最大的动作作为下一步要执行的动作，对此可表示为：

$$a^* = \underset{a_i \in A}{\operatorname{argmax}} P(a_i \mid S) \tag{4}$$

其中，a^* 表示最优动作，即模型最终选择的动作。$\underset{a_i \in A}{\operatorname{argmax}}$ 表示从动作集合 A 中找到能够使联合概率 $P(a_i \mid S)$ 最大的动作 a_i。

通过这种方式，在当前环境约束下，模型能够从动作表 A 中选出最符合上下文的动作，从而生成良好的初始策略。

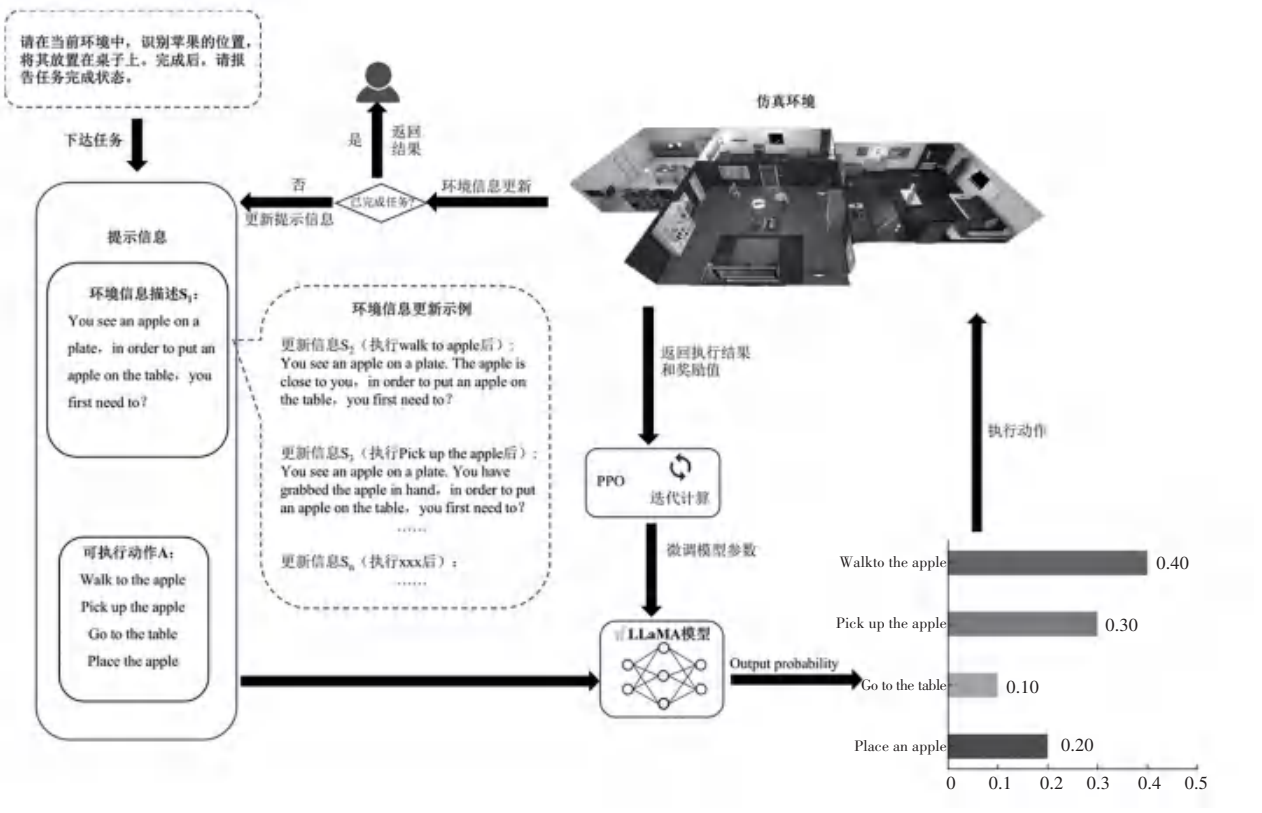


图 1 任务规划流程示意图

Fig. 1 The diagram of the task planning process

2.2 模型微调

在对 LLaMA 模型进行微调时，LLaMA 模型生成的文本或动作序列可以看作是策略 π_θ，每次生成的动作序列是根据当前策略从动作空间中采样得到的。在每一步生成中，通过 PPO 来优化策略，使其生成的序列更符合任务目标。

2.2.1 模型输出与动作生成

在 LLaMA 模型中，生成的每个动作或 token 都对应一个 logits 值，这些 logits 可以通过 Softmax 转换为概率分布。给定当前的输入序列 s_t，LLaMA 模型生成的动作 a_t 是从这些概率分布中采样得到的：

$$\pi_\theta(a_t \mid s_t) = \mathrm{Softmax}(f_\theta(s_t)) \tag{5}$$

其中，$f_\theta(s_t)$ 表示 LLaMA 模型输出的 logits。

2.2.2 优势函数的计算

PPO 算法依赖于优势函数 A_t 来衡量在某一状态下选择特定动作的优越性。优势函数通常通过值函数 $V(s_t)$ 进行估计：

$$A_t = R_t - V(s_t) \tag{6}$$

其中，R_t 表示从时间步 t 开始的累积奖励，$V(s_t)$ 表示状态 s_t 的估值，通常通过一个评论家网络来预测。评论家网络 $V_\varphi(s_t)$ 被用来估计当前状态下的预期奖励，参数 φ 通过最小化以下损失来更新：

$$L_V(\varphi) = \mathbb{E}_t[(V_\varphi(s_t) - R_t)^2] \tag{7}$$

2.2.3 策略更新与优化

在每次生成动作后，通过 PPO 算法对 LLaMA 模型的参数进行更新。具体地，模型的生成策略 π_θ 会通过最小化 PPO 损失函数 $L^{\mathrm{PPO}}(\theta)$ 来更新。每次更新过程中的梯度可以表示为：

$$\nabla_\theta L^{\mathrm{PPO}}(\theta) = \mathbb{E}_t[\nabla_\theta \min(r_t(\theta)A_t,\ \mathrm{clip}(r_t(\theta),1\pm\epsilon)A_t)] \tag{8}$$

其中，A_t 表示策略更新前的优势函数；ϵ 表示用于限制策略更新的超参数；$r_t(\theta)$ 表示新旧策略之间的比率，计算公式为：

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)} \tag{9}$$

其中，$\pi_\theta(a_t \mid s_t)$ 表示新策略下生成动作 a_t 的概率；$\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ 表示旧策略下生成动作 a_t 的概率。

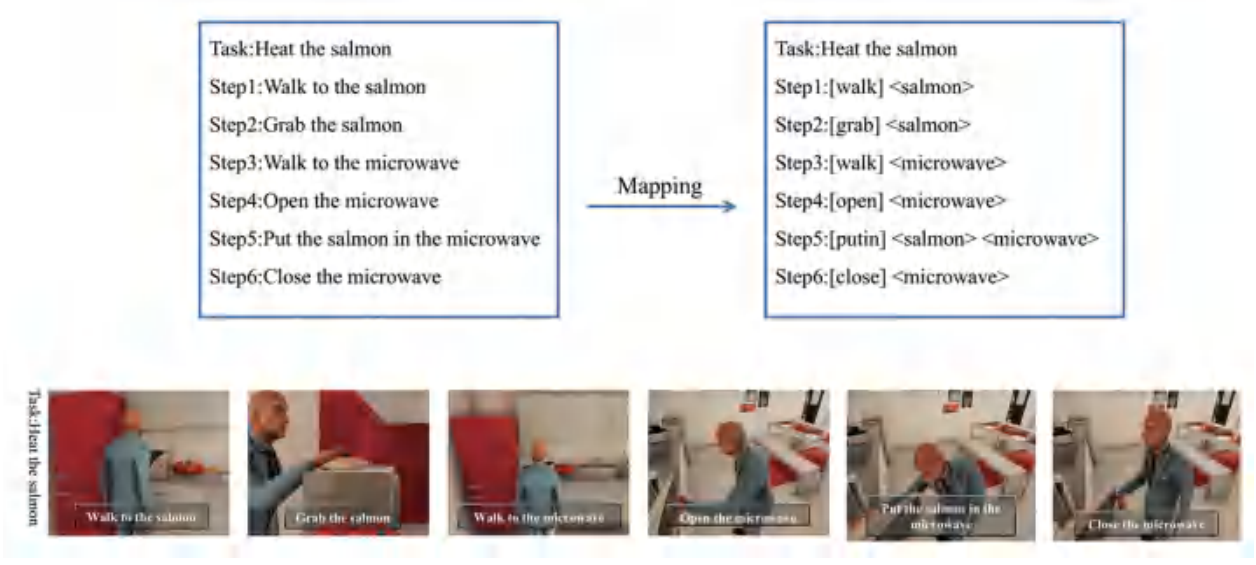
通过引入 clip 操作，PPO 限制了策略的变化幅度，避免策略更新过大，从而保持训练的稳定性。

3 实验

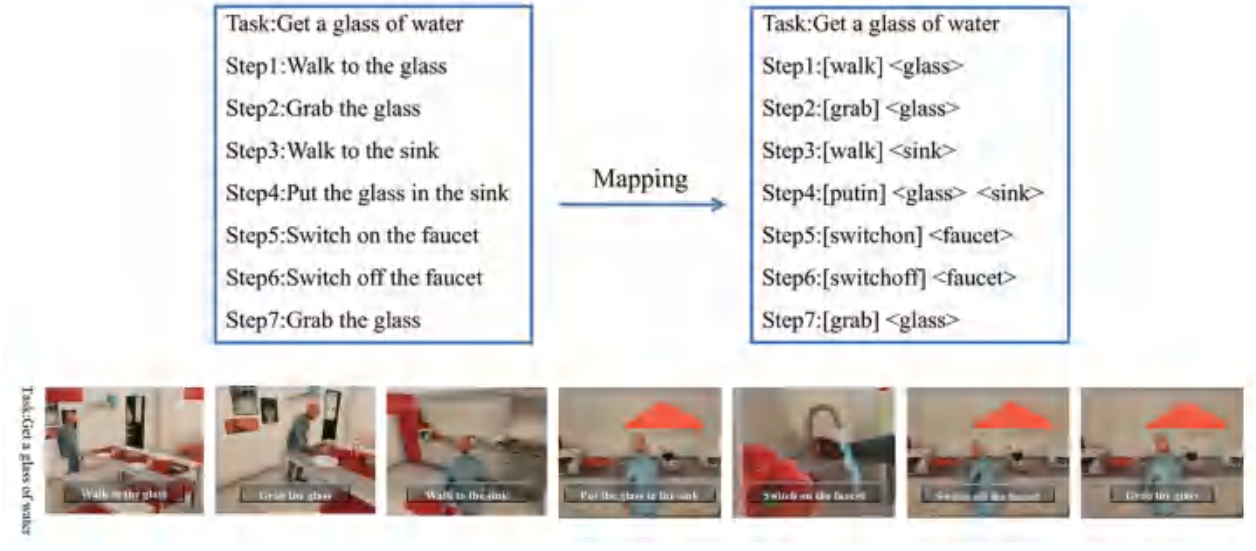
3.1 实验部署

实验过程中，评估了对使用 PPO 算法进行微调

后的模型性能，具体的评估环境来源于 Xavier。Puig 等学者[20]开发的 VirtualHome 仿真环境，在该环境的基础上进行了适当的修改，以适应本实验的需求。在 VirtualHome 环境中，智能体身处一个物品齐全的家庭环境中。图 2 展示了 2 个具体的任务示例，旨在阐明智能体在执行过程中的具体操作。



（a）任务示例 1



（b）任务示例 2

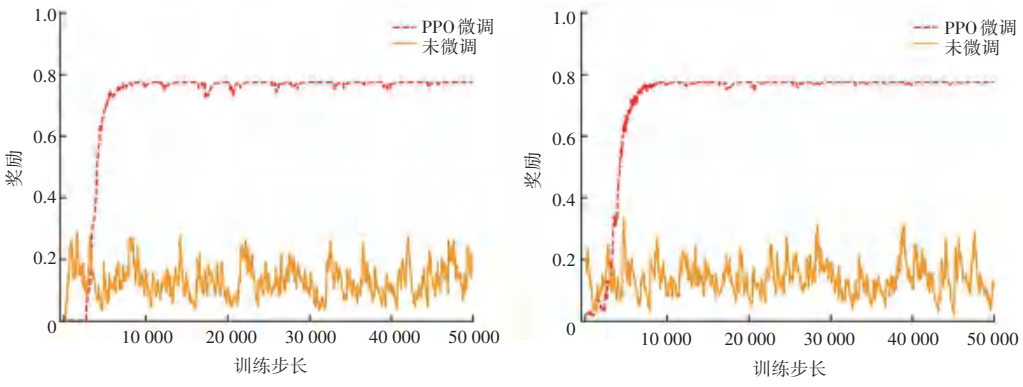
图 2　模型生成的示例计划（左）、映射到 VirtualHome 的执行脚本（右）以及 VirtualHome 中的执行实例（下）

Fig. 2　LLaMA model generated plan example (left), execution script mapped to VirtualHome (right), and execution instance in VirtualHome (bottom)

3.2　PPO 算法微调前后模型性能对比

本实验将训练过程中执行一次任务的累积奖励 return 以及执行一次任务所需动作步数 length 记录下来并进行可视化，可视化结果分别如图 3 和图 4 所示。图 3 分别为使用 PPO 算法进行微调后的模型训练过程中的 return 曲线和未进行微调的模型训练过程中的 return 曲线。图 4 分别为使用 PPO 算法进行微调后的模型训练过程中的 length 曲线和未进行微调的模型训练过程中的 length 曲线。

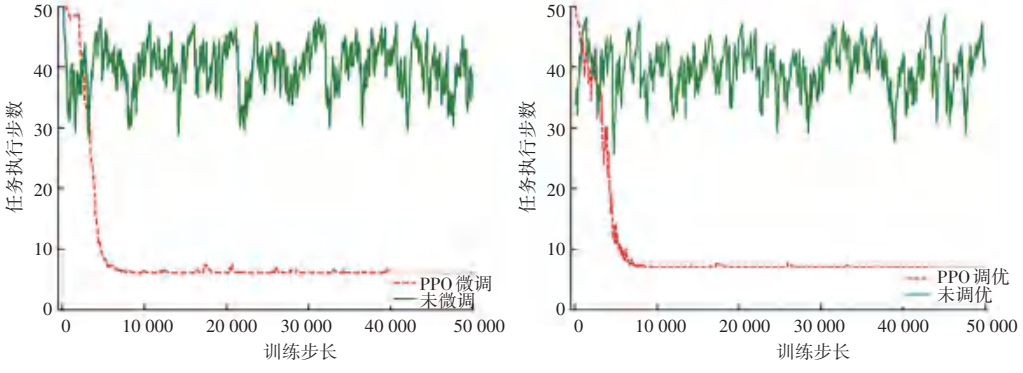
对图 3 的数据进行分析可以看出，使用 PPO 对 LLaMA 模型进行微调后的任务规划表现明显优于直接使用 LLaMA 模型进行任务规划。使用 PPO 算法进行微调后的模型在训练过程中累积奖励能够迅速收敛并稳定在 0.77 左右，显示出模型的高效性和稳定性。而未使用 PPO 算法进行微调的模型则呈现出较大的波动，其累积奖励长期保持在 0~0.2 之间。



（a）加热鲑鱼 （b）倒水

图 3 PPO 算法对任务累积奖励的影响

Fig. 3 The effect of PPO algorithm on task cumulative reward



（a）加热鲑鱼 （b）倒水

图 4 PPO 算法对任务所需步数的影响

Fig. 4 The effect of PPO algorithm on the number of steps required for a task

对图 4 中的数据进行分析可以看出，PPO 算法微调后的模型在训练过程中的表现明显优于未微调模型。使用 PPO 算法进行微调的模型在初期迅速降低了任务执行的步数，最终稳定在约 7 步左右。而未经过 PPO 微调的模型则在训练过程中表现出较大的波动，其任务执行步数长期保持在 30～50 之间。

综上，PPO 微调提升了模型的收敛速度与任务规划效率，确保了模型在执行复杂任务时的稳定性和高效性。

3.3 任务泛化性能测试

得益于大语言模型具备开放词汇的特性，训练后的模型能够将已掌握的技能迁移到各种不同的任务上，而传统的强化学习难以具备这种能力。研究中对微调后的模型以及未进行微调的模型在新的未知任务上的泛化能力进行了评估，评估结果如图 5 所示。研究中重新定义了 5 个任务，对于“倒咖啡”“加热牛奶”“洗衣服”三个任务，将原来的倒水和加热鲑鱼的任务中的水、鲑鱼分别替换为对应的食物和物品；第 4 个和第 5 个任务为交叉实验，即在其中一个任务场景中训练的模型用于推理另一个任务的成功率。微调后的模型在“倒咖啡”“加热牛奶”“洗衣服”这 3 个任务上始终保持了 100%的高成功率。相比之下，未进行微调的模型表现出显著较差的功能效果，在这些任务中的成功率仅为 0.55～0.65 之间。对于进行交叉实验的 2 个任务，模型仍表现出了较高的准确率。在倒水这个任务中，模型的表现虽有所下降，但微调后的模型仍然保持了 86%的成功率，而未进行微调的模型的表现较差，成功率仅为 2%。

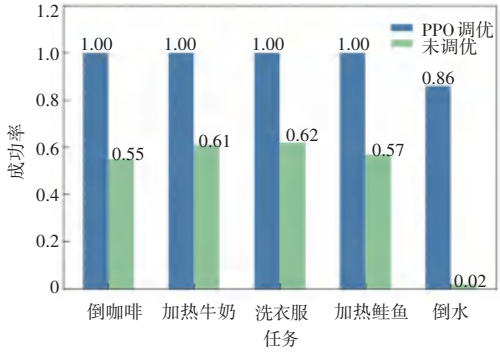


图 5 模型任务泛化能力测试

Fig. 5 Model task generalization ability test

结果表明，使用 PPO 算法进行微调可以显著提高模型对新任务的泛化能力，并在大多数情况下达到接近完美的表现。微调后的模型与未进行微调的模型之间的性能差异凸显了 PPO 算法对于提升模型在未知环境中保持泛化能力的重要性。

4　结束语

本文提出了一种融合 LLaMA 模型与强化学习的自主任务规划系统，通过 PPO 算法对大模型进行策略优化，以实现复杂任务场景中的自主决策能力。系统通过强化学习对 LLaMA 模型进行微调，使其在可交互环境中能够有效地学习并通过调整参数来优化决策过程。PPO 算法通过策略更新和价值函数的结合，使模型在实际任务中的表现逐步收敛，并实现对多样化任务场景的有效适应。实验结果表明，LLaMA 模型在结合强化学习方法后，显著提升了其任务规划的正确率与执行效率，有效解决了大模型在复杂环境中面临的动态决策问题。本研究验证了将大语言模型与强化学习相结合的可行性，为复杂环境下的任务规划提供了解决方法，同时展示了 PPO 算法在微调大语言模型中的优势与潜力。

参考文献

[1] MYERS D, MOHAWESH R, CHELLABOINA V I, et al. Foundation and large language models: Fundamentals, challenges, opportunities, and social impacts [J]. Cluster Computing, 2024, 27: 1-26.

[2] ZHAO W X, ZHOU K, LI J, et al. A survey of large language models [J]. arXiv preprint arXiv, 2303.18223, 2023.

[3] AHN M, BROHAN A, BROWN N, et al. Do as I can, not as I say: Grounding language in robotic affordances [J]. arXiv preprint arXiv, 2204.01691, 2022.

[4] CAI Tianle, WANG Xuezhi, MA Tengyu, et al. Large language models as tool makers[J]. arXiv preprint arXiv, 2305.17126, 2023.

[5] WANG Peiyi, LI Lei, CHEN Liang, et al. Making large language models better reasoners with alignment[J]. arXiv preprint arXiv, 2309.02144, 2023.

[6] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. Cambridge, MA: MIT Press, 2018.

[7] MUTTI M, MANCASSOLA M, RESTELLI M. Unsupervised reinforcement learning in multiple environments[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(7): 7850-7858.

[8] KUMAR A, AGARWAL R, GENG Xinyang, et al. Offline q-learning on diverse multi-task data both scales and generalizes [J]. arXiv preprint arXiv, 2211.15144, 2022.

[9] OUYANG Long, WU J, JIANG Xu, et al. Training language models to follow instructions with human feedback[J]. Advances in Neural Information Processing Systems, 2022, 35: 27730-27744.

[10] TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: Open and efficient foundation language models [J]. arXiv preprint arXiv, 2302.13971, 2023.

[11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[12] ZENG Wei, REN Xiaozhe, SU Teng, et al. Pangu-α: Large-scale autoregressive pretrained Chinese language models with auto-parallel computation [J]. arXiv preprint arXiv, 2104.12369, 2021.

[13] WATKINS C J C H, DAYAN P. Q-learning [J]. Machine Learning, 1992, 8(3-4): 279-292.

[14] DU S S, LUO Yuping, WANG Ruosong, et al. Provably efficient Q-learning with function approximation via distribution shift error checking oracle[J]. Advances in Neural Information Processing Systems, 2019, 32: 8-14.

[15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518: 529-533.

[16] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv, 1707.06347, 2017.

[17] HEESS N, DHRUVA T B, SRIRAM S, et al. Emergence of locomotion behaviours in rich environments[J]. arXiv preprint arXiv, 1707.02286, 2017.

[18] WILLIAMS R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8: 229-256.

[19] 张钦彤, 王昱超, 王鹤羲. 大语言模型微调技术的研究综述[J]. 计算机工程与应用, 2024, 60(17): 17-33.

[20] PUIG X, RA K, BOBEN M, et al. VirtualHome: Simulating household activities via programs[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 8494-8502.