# LEED：一个多智能体强化学习中高效且可扩展的 LLM 增强专家演示框架

*Tianyang Duan*[1], *Zongyuan Zhang*[1], *Songxiao Guo*[1], *Dong Huang*[2], *Yuanye Zhao*[3], *Zheng Lin*[4], *Zihan Fang*[5], *Dianxin Luan*[6], *Heming Cui*[1], *Yong Cui*[7]

[1] Department of Computer Science, The University of Hong Kong, Hong Kong, China.

[2] Institute of Data Science, National University of Singapore, Singapore.

[3] College of International Education, Hebei University of Economics and Business, China.

[4] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China.

[5] Department of Computer Science, City University of Hong Kong, Hong Kong, China.

[6] Institute for Imaging, Data and Communications, University of Edinburgh, UK.

[7] Department of Computer Science and Technology, Tsinghua University, China.



**摘要—多智能体强化学习（MARL）在复杂环境中的智能决策方面具有巨大的潜力。然而，随着代理数量的增加，它面临着协调和扩展性的瓶颈问题。为了解决这些问题，我们提出了用于多智能体强化学习的大型语言模型赋能专家演示框架（LEED）。LEED 包含两个组成部分：演示生成（DG）模块和策略优化（PO）模块。具体来说，DG 模块利用大型语言模型生成与环境交互的指令，从而产生高质量的演示。PO 模块采用分散式训练范式，每个代理都使用生成的演示来构建专家政策损失，并将其与其自身的政策损失相结合。这使每个代理都能够根据专家知识和个人经验有效地个性化和优化其本地策略。实验结果表明，LEED 在样本效率、时间效率以及鲁棒可扩展性方面优于最先进的基线方法。**

***Index Terms*—多智能体强化学习，大型语言模型，去中心化训练**

## I. 介绍

多智能体强化学习（MARL）已成为解决多智能体系统中复杂决策问题的强大范式，具有广泛的应用，如物联网 [1]–[4]、机器人协作 [5]–[8] 和交通信号控制 [9], [10]。现有的 MARL 方法分为两大范式。分散式方法 [11], [12] 独立训练智能体，强调可扩展性和适应性。相比之下，集中训练与分散执行（CTDE）[13], [14] 在训练过程中使用集中策略评估来促进协调，并在执行时提高整个团队的表现。

尽管在多智能体强化学习中取得了近期进展，实现可扩展且可靠的协调仍然是一个重大挑战 [15], [16]。虽然去中心化方法能够很好地随着代理数量的增加而扩展，但它们限制了每个代理仅使用局部观察和个体奖励，这妨碍了对其他代理策略的准确建模，通常会产生策略冲突，特别是在具有全局奖励的设置中。CTDE 方法减轻了策略冲突。然而，当代理数量增加时，多智能体强化学习中的联合状态-动作空间呈指数级增长，导致优化成本大幅增加，并在使用函数近似时放大价值估计误差。这些误差可能导致各代理的动作值同质化，减少行为多样性并损害找到最优策略的能力。

大型语言模型（LLMs）[17], [18] 表现出强大的能力，可以抽象高维状态-动作空间并解决复杂决策问题 [19], [20] 。最近的 LLM 驱动的进步，如状态表示提取 [21]，子任务组合 [22] 和奖励设计 [23] 已显著提高了单个代理的适应性和泛化能力。然而，如何将从 LLM 中获得的知识与策略优化相结合以实现高度高效和可扩展的多智能体强化学习（MARL）仍很大程度上未被探索。

为此，我们提出 LLM-empowered expert demonstrations 框架（LEED），这是一个分散的多智能体强化学习框架，它将大语言模型的广泛领域知识与自主策略学习相结合。LEED 包含两个关键组件：演示生成（DG）模块和策略优化（PO）模块。DG 模块使用大语言模型来生成特定环境的指令，提供关于每个代理任务的演示。这些演示通过使用环境反馈进行迭代改进以确

保相关性和有效性。PO 模块采用分散式训练范式，在该范式中，每个代理通过构建混合策略损失将生成的演示纳入其自己的学习过程中。这种混合损失使代理能够定制和优化它们的地方政策，实现大语言模型生成指导与自主探索之间的最优平衡。我们在两个具有挑战性的现实世界多智能体城市交通规划任务上评估了 LEED。结果显示，LEED 相对于最先进的基线在训练性能方面表现出色。

## II. 相关工作

多智能体强化学习（MARL）因其能够处理涉及多个智能体的复杂合作和竞争任务而吸引了大量关注 [15]。去中心化方法，如 MAPPO，通过共享策略提高了样本效率 [11]。IPPO [12] 将 PPO [24] 扩展到了多智能体环境，通过对每个智能体训练独立的策略而不进行参数共享。通信增强的方法通过信息交换实现更好的协调 [25], [26]。CTDE 方法通过 Q 函数估计联合行动的预期回报，并据此推导出最优的联合策略 [27]–[29]。QMIX 通过单调价值函数分解实现高效的联合 Q 函数估计 [13]。基于 QMIX，HMDQN 整合了层级结构以减轻多机器人任务中的稀疏奖励问题 [30]。HATRPO 通过优势分解在合作场景中增强稳定性 [31]。MACPO 将安全约束纳入 MARL，在策略更新过程中利用信任区域限制确保代理始终满足这些约束 [32]。确定性策略方法直接输出动作而不是概率分布，并通过利用全局信息进行集中训练来提高策略的鲁棒性 [14], [33]。

## III. 方法论

### A. 问题形式化

多智能体强化学习被公式化为一个去中心化的部分可观察马尔科夫过程（Dec-POMDP）[34]，由元组 $\langle \mathcal{A}, \mathcal{S}, \boldsymbol{\mathcal{U}}, P, \{R^i\}_{i\in\mathcal{A}}, \Omega, \{O^i\}_{i\in\mathcal{A}}, \gamma \rangle$ 定义。$\mathcal{A} = \{1, \ldots, n\}$ 表示 $n$ 个代理的集合，$\mathcal{S}$ 表示状态空间，而 $\boldsymbol{\mathcal{U}} = \prod_{i=1}^{n} \mathcal{U}^i$ 表示联合动作空间。在每个时间步骤 $t$，每个代理 $i$ 接收来自观测空间 $\Omega$ 的局部观测 $o_t^i$，根据观测函数 $O^i(s_t)$。代理选择动作 $u_t^i \sim \pi^i(\cdot \mid o_t^i)$，形成联合行动 $\boldsymbol{u}_t = (u_t^1, \ldots, u_t^n)$。环境转换到下一个状态 $s_{t+1} \sim P(\cdot \mid s_t, \boldsymbol{u}_t)$，并为每个智能体提供奖励 $r_t^i = R^i(s_t, \boldsymbol{u}_t, s_{t+1})$。该过程继续，每个智能体接收一个观察-动作-奖励元组 $(o_t^i, u_t^i, r_t^i)$，形式化如下：

$$\boldsymbol{\tau}^a \sim \mathrm{Env}(\boldsymbol{\pi} \mid s_0, P), \tag{1}$$

其中 $\boldsymbol{\tau}^a = \{\tau^{a,i} \mid i \in \mathcal{A}\}$ 表示所有智能体的观察轨迹集，$\boldsymbol{\pi} = \prod_{i=1}^{n} \pi^i$ 表示联合策略，$\tau^{a,i} = (o_0^i, u_0^i, r_0^i, o_1^i, u_1^i, r_1^i, \ldots)$ 表示智能体 $i$ 的观察轨迹，$\mathrm{Env}(\cdot)$ 表示智能体交互的环境。所有代理的目标是学习一个联合策略 $\boldsymbol{\pi}$，以最大化期望折扣回报 $\mathbb{E}\left[\sum_{t=0}^{\infty} \sum_{i=1}^{n} \gamma^t R^i(s_t, \boldsymbol{u}_t, s_{t+1})\right]$，其中 $\gamma \in [0, 1]$ 表示折扣因子。

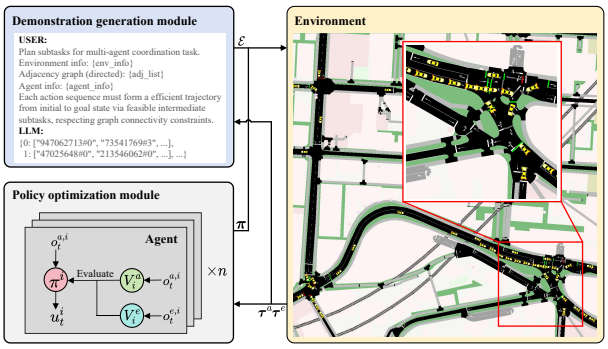



图 1. LEED 的工作流程。

### B. 去中心化的 LLM 赋能多智能体强化学习框架

如图 1所示，LEED 框架由两个组件组成：演示生成（DG）模块和政策优化（PO）模块。DG 模块使用 LLM 生成可执行序列集 $\mathcal{E}$，该序列集在环境中执行以获得专家演示轨迹 $\boldsymbol{\tau}^e$。在 PO 模块中，这些轨迹与联合策略 $\boldsymbol{\pi}$ 在环境交互期间收集的自主探索轨迹 $\boldsymbol{\tau}^a$ 一起被用于优化每个代理的政策。

*1) 演示生成模块：* 大型语言模型在 DG 模块中生成一个可执行序列集 $\mathcal{E} = \{\boldsymbol{e}^i \mid i \in \mathcal{A}\}$，其中每个序列 $\boldsymbol{e}^i = [e_1^i, e_2^i, \ldots]$ 包含一系列结构化的文本指令。每条指令 $\boldsymbol{e}_k^i$ 的格式为 `[命令，参数，代理 ID]`，其中命令指定要执行的操作，参数提供相关参数，代理 `ID` 标识目标代理。例如，在车辆路径规划中，`[移动到，相交点\_ 5，代理\_ 3]` 指示代理 3 导航到交叉路口 5。代理在环境中执行 $\mathcal{E}$ 以获取专家演示轨迹：

$$\boldsymbol{\tau}^e \sim \mathrm{Env}(\mathcal{E} \mid s_0, P), \tag{2}$$

其中 $\boldsymbol{\tau}^e = \{\tau^{e,i} \mid i \in \mathcal{A}\}$ 表示所有代理的专家演示轨迹，每个 $\tau^{e,i}$ 的观察-动作-奖励序列格式与 $\tau^{a,i}$ 相同。值得注意的是，上标 $a$ 和 $e$ 表示观察轨迹的来源：$a$ 指的是从智能体-环境交互中获得的，而 $e$ 表示通过在环境中执行序列集 $\mathcal{E}$ 收集的。

*2) 策略优化模块：* 在 PO 模块中，我们将 PPO [24] 扩展到多智能体场景，其中每个智能体的训练和执行都是去中心化的，以确保可扩展性。具体来说，我们为 $i$ 的智能体和专家轨迹定义了价值函数：

$$V_i^x(o_t^{x,i}) = \mathbb{E}_{\tau^{x,i}}\left[\sum_{k=t}^{\infty} \gamma^k r_k^{x,i}\right], \tag{3}$$

其中 $x \in \{a, e\}$ 和 $o_t^{x,i}, r_k^{x,i} \in \tau^{x,i}$。随后，我们计算代理和专家轨迹的有限时域引导回报：

$$R_t^{x,i} = \sum_{k=0}^{T^x - t - 1} \gamma^k r_{t+k}^{x,i} + \gamma^{T^x - t} V_i^x\left(o_{T^x}^{x,i}\right) \tag{4}$$

其中 $r_{t+k}^{x,i}, o_{T^x}^{x,i} \in \tau^{x,i}$，和 $T^x$ 表示轨迹 $\tau^{x,i}$ 的长度。代理价值函数 $V_i^a$ 和专家价值函数 $V_i^e$ 通过最小化均方误差进行优化：

$$\mathcal{L}_{\mathrm{value}}(V_i^x) = \mathbb{E}_{\tau^{x,i}}\left[\left(V_i^x(o_t^{x,i}) - R_t^{x,i}\right)^2\right]. \tag{5}$$

**Algorithm 1** LEED 培训程序

**Require:** 大型语言模型 $\Pi$，提示 d，采样间隔 $q$。

**初始化** $V_i^a$ ，$V_i^e$ 和 $\pi^i$ 对于每个代理。
**for** each episode **do**
  样本 $\boldsymbol{\tau}^a \sim \mathrm{Env}(\boldsymbol{\pi} \mid s_0, P)$
  **if** $i \bmod q = 0$ **then**
    样本 $\mathcal{E} \sim \Pi(\mathrm{d})$
    样本 $\boldsymbol{\tau}^e \sim \mathrm{Env}(\mathcal{E} \mid s_0, P)$
  **end if**
  **for** each agent **do**
    计算 $R_t^{x,i}$ 和 $A_t^{x,i}$ 使用方程 4和 6。
    更新 $V_i^a$ 和 $V_i^e$ 使用方程 5。
    更新 $\pi^i$ 使用方程 9。
    $\mathrm{d} \leftarrow \mathrm{d} \cup \{\boldsymbol{\tau}^a, \boldsymbol{\tau}^e\}$
  **end for**
**end for**

代理优势 $A_t^{a,i}$ 和专家优势 $A_t^{e,i}$ 对于轨迹 $\tau^{a,i}$ 和 $\tau^{e,i}$ 使用相应的价值函数和回报计算：

$$A_t^{x,i} = R_t^{x,i} - V_i^x(o_t^{x,i}). \tag{6}$$

我们采用截断代理目标来定义智能体策略损失 $\mathcal{L}^a$ 和专家策略损失 $\mathcal{L}^e$ 如下：

$$\mathcal{L}^x(\pi^i) = \mathbb{E}_{\tau^{x,i}}\left[\min(\omega_t^{x,i}, \mathrm{clip}(\omega_t^{x,i}, 1-\epsilon, 1+\epsilon))A_t^{x,i}\right], \tag{7}$$

其中 $\omega_t^{x,i} = \pi^i(u_t^{x,i}|o_t^{x,i})/\pi^{\mathrm{old},i}(u_t^{x,i}|o_t^{x,i})$ 表示重要性采样比，$\pi^{\mathrm{old},i}$ 表示上一次迭代中智能体 $i$ 的策略，而 $\epsilon \in (0,1)$ 代表截断系数。我们然后定义一个混合策略损失函数 $\mathcal{L}_{\mathrm{mix}}$ 如下：

$$\mathcal{L}_{\mathrm{mix}}(\pi^i) = \alpha\mathcal{L}^a(\pi^i) + (1-\alpha)\mathcal{L}^e(\pi^i), \tag{8}$$

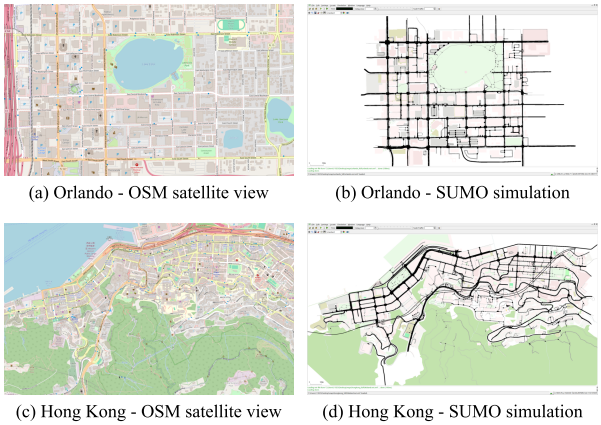
其中 $\alpha = \exp\left(-k/K \cdot \mathrm{D}_{\mathrm{DTW}}(\tau^{a,i}, \tau^{e,i})\right)$ 动态调整代理和专家的策略损失之间的权重，$k$ 和 $K$ 分别表示当前训练周期数和总训练周期数，$\mathrm{D}_{\mathrm{DTW}}(\tau^{a,i}, \tau^{e,i})$ 表示动态时间规整（DTW）距离，用于通过非线性时间对齐来衡量代理和专家轨迹之间的相似度。训练初期，较大的 DTW 距离会导致 $\alpha$ 变小，从而使模仿学习更加重视专家轨迹。随着训练的进行和轨迹对齐，$\alpha$ 增加，逐渐将训练重点转向自主探索。

最后，PO 模块为每个代理的策略损失引入了最大熵正则化项以促进探索。因此，每个代理的策略损失 $\mathcal{L}(\pi^i)$ 为：

$$\mathcal{L}(\pi^i) = \mathcal{L}_{\mathrm{mix}}(\pi^i) + \beta\mathbb{E}_{\tau^{a,i}}[\mathcal{H}(\pi^i(\cdot|o_t^{a,i}))], \tag{9}$$

其中 $o_t^{a,i} \in \tau^{a,i}$，和 $\beta \in (0,1]$ 表示熵正则化系数，而 $\mathcal{H}(\pi^i(\cdot|o_t^{a,i}))$ 表示策略 $\pi^i$ 的熵。

算法 1概述了 LEED 的训练过程，包括使用代理和专家轨迹（第 12 行）对提示 d 进行周期性优化。为了稳定更新并减少推理开销，我们采用了一个采样间隔 $q$，在多个回合中重复使用相同的专家轨迹（第 4 – 7 行）。



(a) Orlando - OSM satellite view　(b) Orlando - SUMO simulation

(c) Hong Kong - OSM satellite view　(d) Hong Kong - SUMO simulation

图 2. 实验场景。

| **动作空间（离散，形状：$m$）** | |
|---|---|
| **索引** | **注意** |
| $0..m-1$ | Select the $i$-th available edge |
| **观测空间（形状：$2+2m$）** | |
| **索引** | **注意** |
| [0] | Current junction id |
| [1] | Destination junction id |
| $[2+2i]$ | Score of the $i$-th available edge ($i = 0..m-1$) |
| $[2+2i+1]$ | 第 $i$ 条边的末端接点 ID（$i = 0..m-1$） |

表 I
动作空间和观测空间

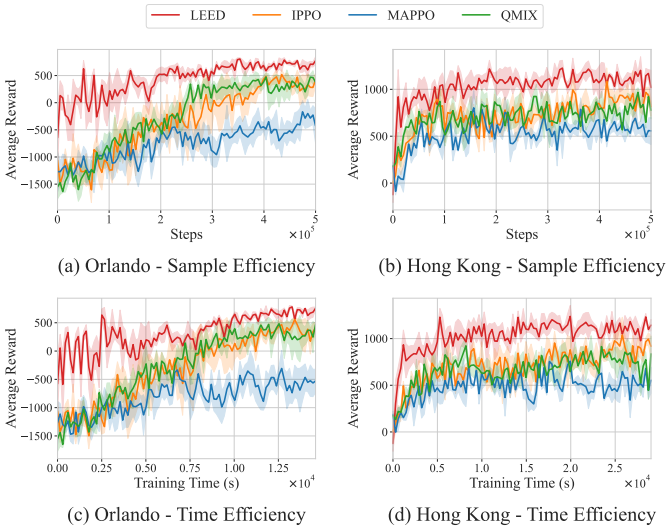
## IV. 评估

### A. 实验设置

*1) 环境：* 我们在基于真实世界的 OpenStreetMap (OSM) 数据 [35] 的大规模交通环境中，对 LEED 进行了评估，通过 SUMO [36] 实现模拟。实验涵盖了两个具有代表性的城市网络：奥兰多，一个具有均匀交叉口的标准网格（图 2(a,b)），以及 *(2)* 香港，一个复杂、多山的网络，拥有异质的道路几何结构和非标准交叉口（图 2(c,d)）。奖励函数包括时间惩罚项，距离目的地的距离整形项，以及到达后的完成奖励。详细的环境规范总结在表 I中。

*2) 提示设计：* 我们使用 GPT-3.5 生成基于大语言模型的专家演示，用于 LEED。我们将任务描述、节点坐标、有向图邻接表和代理信息（起始/结束点及出发时间）提供给大语言模型，以建立交通环境的背景。指示大语言模型为每个代理输出可解析的路标序列作为 Python 字典，在仿真中可执行。每 $q = 5$ 个周期生成细化提示，使用来自代理探索 $\tau^a$ 和专家演示 $\tau^e$ 的轨迹数据。计算 DTW 距离以确定方程 8中的混合权重 $\alpha$，并分析这些轨迹之间的差异。

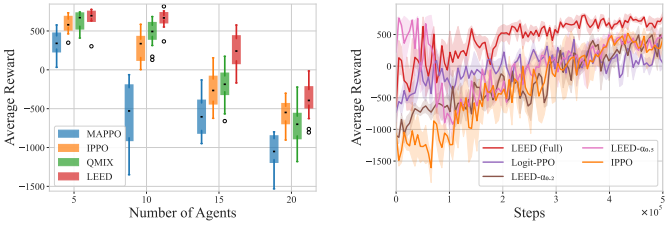
*3) 基线和超参数：* 我们将 LEED 与几个最先进的 MARL 基线（IPPO [12]，MAPPO [11]，QMIX [13]）在两个场景中进行了比较。所有实验均通过 5 次独立运行进行。每次运行包含 500 个训练周期，每个周期包括 200 步，总计 $1 \times 10^5$ 步。




(a) Orlando - Sample Efficiency (b) Hong Kong - Sample Efficiency

(c) Orlando - Time Efficiency (d) Hong Kong - Time Efficiency

图 3. 样本和时间效率的比较结果。




(a) Scalability Analysis (b) Ablation Study

图 4. 可扩展性分析和消融研究。

我们使用了 10 个智能体和一个学习率 $3 \times 10^{-4}$。策略网络和评价网络都采用了每层 128 个单元的两层架构。

### B. 实验结果与分析

*1) 样本效率：* 图 3(a,b) 显示 LEED 在样本效率和最终性能方面始终超越所有基线。在奥兰多中，LEED 达到 $\sim 500+$ 奖励，而表现最佳的基线仅达到 $\sim 400$。在更具挑战性的香港中，LEED 维持 $\sim 1000+$ 奖励，相比之下其他方法为 $\sim 750-800$。

*2) 时间效率：* 图 3(c,d) 显示，尽管在训练过程中引入了大语言模型推理，LEED 在墙钟训练时间方面仍具有竞争力。它在这两种场景中都高效地实现了最佳性能。

*3) 可扩展性分析：* 图 4(a) 展示了在不同代理人口规模（5、10、15、20）下奥兰多的比较性能，报告了最终模型的 20 次测试运行的结果。LEED 随着代理数量的增加保持更好的性能。虽然所有方法在从 5 到 20 个代理进行扩展时都显示出奖励下降的情况，但 LEED 经历了最小的减少，并且始终实现了最高的平均奖励。随着系统规模的增大，LEED 能够有效缓解代理之间的政策冲突，在大规模系统中具有更好的可扩展性。

*4) 消融研究：* 图 4(b) 展示了我们的消融研究，比较了固定专家权重的 LEED 变体在奥兰多:*(1) LEED*（完整版）;*(2) LEED-*$\alpha_{0.2}$ 和 *(3) LEED-*$\alpha_{0.5}$; *(4)* 对数几率 *PPO*，该变体通过基于最短路径成本的 Logit 模型的概率抽样路线生成演示；以及 *(5) IPPO*。结果显示，LEED（完整版）的表现优于所有变体。LEED-$\alpha_{0.5}$ 在早期表现出色，但由于代理探索不足，显示出较差的收敛性。LEED-$\alpha_{0.2}$ 表现出较慢的学习速度。这表明动态损失加权机制相较于静态加权具有更优的表现。Logit-PPO 的表现优于 IPPO，但劣于 LEED。这可以归因于随机方法生成样本的质量较低以及缺乏反馈机制。

*5) LLM 生成的演示质量：* 表 II展示了我们在三个精炼阶段（初始、经过 5 次精炼以及经过 10 次精炼）对大语言模型生成的演示进行评估的结果，这些结果在奥兰多中呈现。对于每个阶段，收集了 100 条专家轨迹（10 个提示词 × 10 个代理程序）。结果显示随着精炼次数增加质量逐步提高：由于上下文的扩展，标记数量从 3680 增加到 7576，路径有效性比率从 74% 提高到 100%，奖励分数从 478.42 上升至 503.26。此外，动态时间弯曲距离的减少表明随着精炼次数的增加代理程序与专家行为之间的一致性提高。

| **度量** | **初始** | **5 细化** | **10 种细化** |
|---|---|---|---|
| Token | 3680 | 4645 | 7576 |
| Validity Rate (%) | 74 | 82 | 100 |
| Reward | 478.42 | 495.05 | 503.26 |
| DTW Distance | 189.91 | 50.53 | 41.25 |

表 II
LLM 生成演示的分析。

## V. 结论

我们提出了 LEED，一个可扩展的去中心化多智能体强化学习框架，它将 LLM 生成的专家演示与自主智能体探索相结合。通过利用混合损失函数，LEED 自适应地平衡专家知识和个体智能体经验，从而实现对每个智能体的高效和鲁棒的策略学习。实验结果表明，LEED 不仅提高了样本和时间效率，而且与最先进的基线相比，实现了卓越的可扩展性。作为未来的潜在方向，我们期待将我们的方法扩展到改善各种应用，例如大型语言模型 [37], [38]，多模态训练 [18], [39]，分布式机器学习 [40]–[43]，以及自动驾驶 [10], [44], [45]。

## 参考文献

[1] Zheng Lin, Guanqiao Qu, Wei Wei, Xianhao Chen, and Kin K Leung, "Adaptsfl: Adaptive Split Federated Learning in Resource-Constrained Edge Networks," *IEEE Trans. Netw.*, 2024.

[2] Mohamad A Hady, Siyi Hu, Mahardhika Pratama, Zehong Cao, and Ryszard Kowalczyk, "Multi-agent reinforcement learning for resources allocation optimization: a survey," *Artif. Intell. Rev.*, vol. 58, no. 11, pp. 354, 2025.

[3] Zheng Lin, Guangyu Zhu, Yiqin Deng, Xianhao Chen, Yue Gao, Kaibin Huang, and Yuguang Fang, "Efficient Parallel Split Learning over Resource-Constrained Wireless Edge Networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9224–9239, 2024.

[4] Zheng Lin, Wei Wei, Zhe Chen, Chan-Tong Lam, Xianhao Chen, Yue Gao, and Jun Luo, "Hierarchical Split Federated Learning: Convergence Analysis and System Optimization," *IEEE Trans. Mobile Comput.*, 2025.

[5] Zekai Sun, Xiuxian Guan, Zheng Lin, Zihan Fang, Xiangming Cai, Zhe Chen, Fangming Liu, Heming Cui, Jie Xiong, Wei Ni, et al., "Intra-dp: A high performance collaborative inference system for mobile edge computing," *arXiv preprint arXiv:2507.05829*, 2025.

[6] James Orr and Ayan Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: A survey," *Sensors*, vol. 23, no. 7, pp. 3625, 2023.

[7] Zheng Lin, Guanqiao Qu, Qiyuan Chen, Xianhao Chen, Zhe Chen, and Kaibin Huang, "Pushing Large Language Models to the 6G Edge: Vision, Challenges, and Opportunities," *arXiv preprint arXiv:2309.16739*, 2023.

[8] Zekai Sun, Xiuxian Guan, Zheng Lin, Yuhao Qing, Haoze Song, Zihan Fang, Zhe Chen, Fangming Liu, Heming Cui, Wei Ni, et al., "Rrto: A high-performance transparent offloading system for model inference in mobile edge computing," *arXiv preprint arXiv:2507.21739*, 2025.

[9] Haiyan Zhao, Chengcheng Dong, Jian Cao, and Qingkui Chen, "A survey on deep reinforcement learning approaches for traffic signal control," *Eng. Appl. Artif. Intell.*, vol. 133, pp. 108100, 2024.

[10] Zihan Fang, Zheng Lin, Senkang Hu, Hangcheng Cao, Yiqin Deng, Xianhao Chen, and Yuguang Fang, "IC3M: In-Car Multimodal Multi-Object Monitoring for Abnormal Status of Both Driver and Passengers," *arXiv preprint arXiv:2410.02592*, 2024.

[11] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24611–24624, 2022.

[12] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?," *arXiv preprint arXiv:2011.09533*, 2020.

[13] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 178, pp. 1–51, 2020.

[14] Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," *arXiv preprint arXiv:1910.01465*, 2019.

[15] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handb. Reinforcement Learn. Control*, pp. 321–384, 2021.

[16] Zongyuan Zhang, Tianyang Duan, Zheng Lin, Dong Huang, Zihan Fang, Zekai Sun, Ling Xiong, Hongbin Liang, Heming Cui, Yong Cui, et al., "Robust deep reinforcement learning in robotics via adaptive gradient-masked adversarial attacks," *arXiv preprint arXiv:2503.20844*, 2025.

[17] Zheng Lin, Yuxin Zhang, Zhe Chen, Zihan Fang, Xianhao Chen, Praneeth Vepakomma, Wei Ni, Jun Luo, and Yue Gao, "HSplitLoRA: A Heterogeneous Split Parameter-Efficient Fine-Tuning Framework for Large Language Models," *arXiv preprint arXiv:2505.02795*, 2025.

[18] Zihan Fang, Zheng Lin, Senkang Hu, Yihang Tao, Yiqin Deng, Xianhao Chen, and Yuguang Fang, "Dynamic uncertainty-aware multimodal fusion for outdoor health monitoring," *arXiv preprint arXiv:2508.09085*, 2025.

[19] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che, "Towards reasoning era: A survey of long chain-of-thought for reasoning large language models," *arXiv preprint arXiv:2503.09567*, 2025.

[20] Zheng Lin, Zhe Chen, Zihan Fang, Xianhao Chen, Xiong Wang, and Yue Gao, "Fedsn: A federated learning framework over heterogeneous leo satellite networks," *IEEE Transactions on Mobile Computing*, 2024.

[21] Boyuan Wang, Yun Qu, Yuhang Jiang, Jianzhun Shao, Chang Liu, Wenming Yang, and Xiangyang Ji, "LLM-empowered state representation for reinforcement learning," *arXiv preprint arXiv:2407.13237*, 2024.

[22] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," *arXiv preprint arXiv:2302.01560*, 2023.

[23] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[25] Robert Müller, Hasan Turalic, Thomy Phan, Michael Kölle, Jonas Nüßlein, and Claudia Linnhoff-Popien, "ClusterComm: Discrete communication in decentralized marl using internal representation clustering," *arXiv preprint arXiv:2401.03504*, 2024.

[26] Justin Lidard, Udari Madhushani, and Naomi Ehrich Leonard, "Provably efficient multi-agent reinforcement learning with fully decentralized communication," in *Proc. Amer. Control Conf.*, 2022, pp. 3311–3316.

[27] Guannan Qu and Na Li, "Exploiting fast decaying and locality in multi-agent mdp with tree dependence structure," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 6479–6486.

[28] Minghai Yuan, Hanyu Huang, Zichen Li, Chenxi Zhang, Fengque Pei, and Wenbin Gu, "A multi-agent double deep-Q-network based on state machine and event stream for flexible job shop scheduling problem," *Adv. Eng. Inform.*, vol. 58, pp. 102230, 2023.

[29] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al., "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[30] Yue Bai, Yaqiong Lv, and Jiatong Zhang, "Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing," *Eng. Appl. Artif. Intell.*, vol. 124, pp. 106534, 2023.

[31] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang, "Trust region policy optimisation in multi-agent reinforcement learning," *arXiv preprint arXiv:2109.11251*, 2021.

[32] Shangding Gu, Jakub Grudzien Kuba, Munning Wen, Ruiqing Chen, Ziyan Wang, Zheng Tian, Jun Wang, Alois Knoll, and Yaodong Yang, "Multi-agent constrained policy optimisation," *arXiv preprint arXiv:2110.02793*, 2021.

[33] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[34] Frans A Oliehoek, Christopher Amato, et al., *A concise introduction to decentralized POMDPs*, vol. 1, Springer, 2016.

[35] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[36] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner, "Microscopic traffic simulation using sumo," in *Proc. Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582.

[37] Zihan Fang, Zheng Lin, Zhe Chen, Xianhao Chen, Yue Gao, and Yuguang Fang, "Automated Federated Pipeline for Parameter-Efficient Fine-Tuning of Large Language Models," *arXiv preprint arXiv:2404.06448*, 2024.

[38] Zheng Lin, Xuanjie Hu, Yuxin Zhang, Zhe Chen, Zihan Fang, Xianhao Chen, Ang Li, Praneeth Vepakomma, and Yue Gao, "Split-LoRA: A Split Parameter-Efficient Fine-Tuning Framework for Large Language Models," *arXiv preprint arXiv:2407.00952*, 2024.

[39] Yongyang Tang, Zhe Chen, Ang Li, Tianyue Zheng, Zheng Lin, Jia Xu, Pin Lv, Zhe Sun, and Yue Gao, "MERIT: Multi-modal Wearable Vital Sign Waveform Monitoring," *arXiv preprint arXiv:2410.00392*, 2024.

[40] Yuxin Zhang, Haoyu Chen, Zheng Lin, Zhe Chen, and Jin Zhao, "Fedac: An adaptive clustered federated learning framework for heterogeneous data," *arXiv preprint arXiv:2403.16460*, 2024.

[41] Zheng Lin, Yuxin Zhang, Zhe Chen, Zihan Fang, Cong Wu, Xianhao Chen, Yue Gao, and Jun Luo, "Leo-split: A semi-supervised split learning framework over leo satellite networks," *arXiv preprint arXiv:2501.01293*, 2025.

[42] Mingda Hu, Jingjing Zhang, Xiong Wang, Shengyun Liu, and Zheng Lin, "Accelerating Federated Learning with Model Segmentation for Edge Networks," *IEEE Trans. Green Commun. Netw.*, 2024.

[43] Yuxin Zhang, Haoyu Chen, Zheng Lin, Zhe Chen, and Jin Zhao, "LCFed: An Efficient Clustered Federated Learning Framework for Heterogeneous Data," *arXiv preprint arXiv:2501.01850*, 2025.

[44] Zheng Lin, Lifeng Wang, Jie Ding, Yuedong Xu, and Bo Tan, "Tracking and transmission design in terahertz v2i networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 6, pp. 3586–3598, 2022.

[45] Zheng Lin, Lifeng Wang, Jie Ding, Bo Tan, and Shi Jin, "Channel Power Gain Estimation for Terahertz Vehicle-to-Infrastructure Networks," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 155–159, 2022.