开悟比赛 X-CLAW 队技术整理与分享

何金岷 中科院自动化所模式识别与智能系统专业

徐航 中科院自动化所模式识别与智能系统专业

臧一凡 中科院自动化所模式识别与智能系统专业

景煜恒 中科院自动化所模式识别与智能系统专业

魏彤 清华大学计算机科学与技术专业

指导老师：李凯

一、 简介

在 2022 年 9 月-2023 年 4 月举办的腾讯第三届开悟 MOBA 多智能体强化学习大赛中，我们队伍（X-CLAW）在初赛中有幸获得了第 7 名的成绩。这得益于实验室老师与腾讯官方的大力支持， 也得益于我们队伍本身的技术探索与积累。本文首先会简单叙述比赛的基本情况，随后从各关键模块出发，简要介绍本队伍在开悟比赛中的探索历程与心得体会。

二、 参赛概况

成员信息：

姓名	在读信息	王者荣耀段位
何金岷	中国科学院自动化研究所博士二年级	钻石
徐航	中国科学院自动化研究所博士三年级	星耀
臧一凡	中国科学院自动化研究所博士四年级	/
景煜恒	中国科学院自动化研究所博士一年级	/
魏彤	清华大学大学四年级	/

参赛动机：曾参与开悟前两届比赛，借此机会可以进行一个强化学习的落地实验，以及积累一些大规模强化学习模型训练的工程经验。

时间投入：人均大概每周在比赛上花费 30 小时。

三、网络设计

本节主要描述我队在神经网络架构上的一些优化和尝试。

1. 独立网络设计：最终模型每个英雄对应独立网络，使不同英雄网络更具专一性，由于英雄之间的技能差异性较大，如果使用单一网络进行拟合难度较大，所以在训练后期将共用网络拆分为 5 个独立网络分别学习；

2. 循环神经网络逻辑单元替换：使用 GRU 替换 LSTM。该优化取决于经验性结论，即 GRU 参数更少，收敛更快；GRU 更适用于小模型，LSTM 更适用于大模型；

3. 循环神经网络和全连接网络相结合：考虑 GRU 主要处理时序状态信息；MLP 主要处理当前时刻状态信息。结合往届参赛队伍经验，将特征处理网络后的输出分别输入到一个 GRU 和一个 MLP 网络层中，并将二者的输出拼接在一起作为后续 Actor 和 Critic 网络的输入。

4. Value Dropout：Critic 网络中价值估计通常存在过估计问题，所以添加 Dropout 层缓解过估计问题。

5. 多头价值估计：不同 head 估计不同类别奖励，使奖励估计更精准，同时方便奖励权重调整，减少奖励调整后带来的价值网络需重新学习问题。

6. 识别英雄辅助任务：我们认为特征处理部分可以视为智能体对当前环境进行的态势感知，所以设计感知类的辅助任务，帮助网络进行拟合，即根据当前状态判断敌我双方分别是什么英雄的分类任务。但由于该任务过于简单，很快收敛，效果不明显，故后期弃用。

四、奖励体系

本节主要概括性描述我队在奖励设计中的一些想法。

总体奖励调整思路：训练前期主要使用稠密奖励（例如金币、经验奖励等）引导；训练后期主要使用稀疏奖励（例如 KDA）引导；

阶段性奖励设计：将单局游戏划分为不同阶段（前/中/后期），不同阶段使用不同的奖励权重，例如前期注重发育类奖励、后期注重推塔类奖励；

奖励衰减：单局游戏中，由于前期英雄能力较弱，获取的奖励较少，后期英雄能力较强，获取的奖励较多，所以为防止比赛无限制进行，随游戏时间进行奖励衰减；

人为奖励设计：人为添加一些引导性奖励，并通过录像进行奖励的增删和权重调整。具体奖励例如释放技能奖励、无伤点塔奖励等；

奖励零和：根据奖励属性考虑单项奖励是否进行零和操作，例如击杀和死亡二者本身自带对称属性，如果进行零和操作仅需设计一个奖励即可，为方便对 KD 奖励进行独立调整，故二者奖励无需进行零和操作。

五、特征与规则

由于原始特征信息为黑盒，所以未对特征信息进行增删修改。本节主要简单介绍我队一些规则后处理工作。

合法动作规则处理：通过观看录像发现，智能体在学习前期容易进行技能空放、无单位平 A 等操作，为减少探索空间，可以将对应情境下的相应动作进行人为 Mask 操作。

固定连招规则写入：基于游戏理解，固定技能的连续释放能使英雄能力最大化，所以可以人为写入一些固定技能释放连招。例如马可波罗在释放大招后一段时间释放眩晕技能。

六、 强化学习算法

算法层面的优化并不太多，主体框架仍使用 PPO 算法框架，我队主要对采样逻辑和对手池构建部分进行了一些优化，在本节进行简单介绍。

多英雄样本量均衡：不同对战的训练数据由独立的经验池存储，即按照敌我双方分别 5 个英雄，共 25 种组合划分为 25 个经验池，具体采样时，每个 Batch 均匀、随机或按照一定比例从不同的经验池中进行采样；

红蓝双方样本均衡：可以按照上述方法进行相同划分，但将导致经验池划分过于细致，所以仅在生成数据的 Actor 端认为调整初始化在红蓝双方的比例；

关键样本重采样：类似于 PER 优先经验采样，PER 是根据 TD 误差进行样本排序，但基于现有框架不方便实现，所以将其简化为人为关键样本判别，例如将技能释放等重要且稀疏的样本建立独立 Skill 经验池，每次采样从 Skill 经验池中进行一定样本比例采样；

历史对手池构建：选取部分历史训练保存的具有风格化或能力较强的模型，作为后续训练的固定对手，数据生成时以一定概率从固定对手池中进行选取。

七、 系统工程架构

本节主要描述我队实现了一些自动化脚本可以大大提升工作效率。

本地自动模型对战：自动下载模型，并在本地进行相互对战测试，维护 Top K 的训练模型池，方便进行模型筛选。后期未大量使用，因为游戏 GameCore 要求使用 Windows 系统，无法在服务器上运行，本地单机测试效率较低；

批量录像下载：脚本实现多个批量录像下载，并将对战及胜负信息写入录像名称，方便进行录像可视化观察；

帧状态数据输出：将游戏帧状态数据转化为 Json 格式数据，本地测试时可以输出测试对战的完整数据，方便统计和 Debug；

初始化模型脚本：方便进行模型的解压、移动等操作；

自动获取对战信息：从平台爬取所有测试对战信息，方便进行胜率统计和模型挑选。

八、 模型迭代过程

本节主要描述模型迭代的几个阶段。

混合单网络训练：参考 Baseline，使用一个模型对所有英雄进行共同学习，忽略英雄间的差异性，英雄之间随机进行相互对战，该方案的数据利用率较高，能快速获得一个基础能力不错的初始模型；

混合多分支网络训练：将 Policy 和 Value 的网络区分为多个分支头，每个英雄对应独立分支头，即不同英雄共享特征处理部分网络，而决策网络进行区分，但由于未能灵活平衡不同分支的训练数据量，训练效果不佳；

混合多模型网络训练：每个英雄对应一个完整的独立网络，使不同英雄网络更具专一性，仍然采用英雄之间随机进行相互对战，Trainer 端每次训练随机选取单一英雄或 Batch 内平均样本分别训练对应的独立网络；

独立多模型网络训练：考虑到不同英雄之间的克制关系，以及不同英雄模型能力之间存在的区别，将模型能力较强的英雄作为固定对手，仅训练模型能力较弱的单一英雄，此过程可以循环提升各个英雄的能力。

上述各个阶段间均需要使用“手术操作”对模型进行初始化操作。

九、 训练效果分析

本节主要对第八节的各迭代过程进行分析，并简要分析最终提交模型的能力水平。

混合单网络训练阶段的模型能力提升较快，但也很快达到能力瓶颈，尽管通过奖励调节可以人为引导智能体进行进一步的能力提升，但比较消耗人力资源进行不断的调参工作；

混合多分支网络训练阶段的模型能力出现了一些问题，分析可能原因是未能灵活平衡不同分支的训练数据量，导致不同分支的梯度很难进行平衡，未做进一步探索；

混合多分支网络训练的模型能力提升相对较快，但是由于英雄特性，导致弱势英雄和强势英雄很难得到针对性的训练，因为基本为负样本或正样本，而且容易使单个英雄模型陷入单一策略；

独立多模型网络训练的模型能力提升相对较慢，需要人为控制当前的训练英雄，且由于其他英雄为固定对手，不能进行长时间训练，需要人为较频繁切换训练英雄，但模型能力基本在稳步缓慢提升，用于后期的微量调整。

最终提交的模型由于采用多模型独立网络，仍存在英雄能力失衡的情况，与其他战队的模型相比，有几个固定英雄明显能力较弱。

十、 总结与展望

这里对上述工作进行总结：

1. 整体训练分为不同的阶段，基本符合比赛初期确定的大体技术路线和优化方向；
2. 网络架构上进行了少量尝试，尽量在比赛初期将网络架构确定下来；
3. 为配合不同的训练阶段，在样本采样部分进行了一些优化尝试，有一定效果；
4. 为引导智能体的探索方向，也在规则和奖励部分进行了较多调试；
5. 完成了一些自动化脚本编写，极大提高了生产效率，对模型测试及选择提供了帮助。

以及提供一些可尝试的优化方向：

Value Normalization：通过引入 Value 的归一化和去归一化提高样本效率；

Phasic Policy Gradient (PPG)：Policy 使用 on policy 学习，Value 使用 off policy 学习。可以充分利用 GPU 的 slow time 时间进行 Value 的更新，提高样本效率；

Attention Unit：用 Attention 替换 MaxPooling，因为我们的网络涉及到的 Unit 较少，MaxPooling 可能损失大量信息；

自动参数调节：主要针对 RL 较敏感参数的自动调整，学习率、Entropy 等参数；

模型蒸馏：蒸馏操作在本次比赛中并未尝试，取巧的将模型压缩至符合要求大小故没进行实现，但未来可能涉及到。