# 开悟比赛-哈斯特尔队技术整理与分享

刘一真 华中科技大学计算机系

张伟明 华中科技大学计算机系

谭頔凡 华中科技大学计算机系

指导老师: 刘渝、周可

## 一、 简介

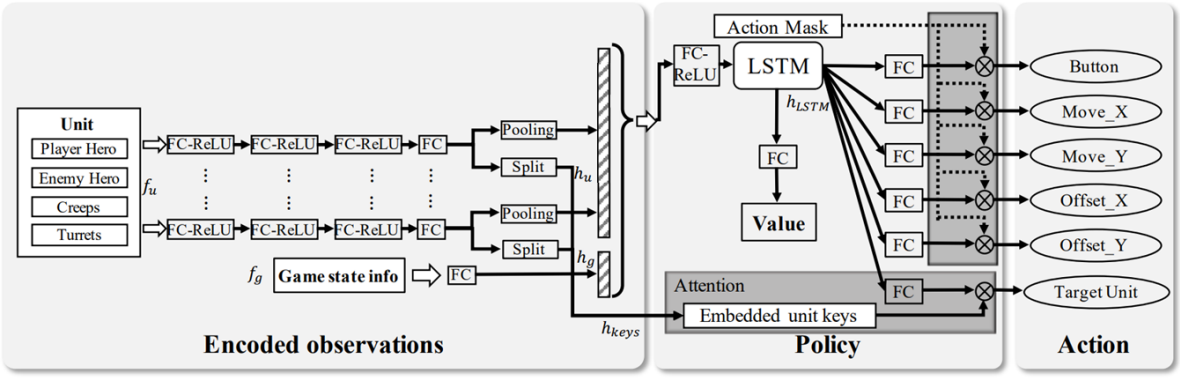
在 2022 年 9 月-2023 年 4 月举办的腾讯第三届开悟 MOBA 多智能体强化学习大赛中，我们队伍(哈斯特尔)在初赛中有幸获得了第 9 名的成绩。这得益于实验室老师与腾讯官方的大力支持，也得益于我们队伍本身的技术探索与积累。本文首先会简单叙述比赛的基本情况，随后从各关键模块出发，简要介绍本队伍在开悟比赛中的探索历程与心得体会。

## 二、 参赛概况

由刘渝、周可老师带队，刘一真为队长，张伟明、谭頔凡为主力的 HUSTER 战队受邀参与第三届腾讯开悟比赛。队员们不仅对王者荣耀游戏充满兴趣，更痴迷于游戏背后的技术，在比赛期间全队平均每天投入五至六个小时。

## 三、网络设计

1v1 模型设计参考了论文 Mastering Complex Control in MOBA Games with Deep Reinforcement Learning 中给出的神经网络架构：




权衡模型的表现和复杂度后，我们将模型规模控制到 23M。强化了英雄单位的特征提取网络、特征融合网络、价值网络和部分决策网络（Button、Target Unit），简化了非英雄单位的特

征提取网络和部分决策网络（Move、Offset），并将 LSTM 替换成 GRU。

## 四、奖励体系

先由初始值训练出模型，再依据对战效果调整奖励设置。

如果击杀和死亡影响了推塔，则会调整击杀和死亡的权重。若在比赛中发现英雄由于击杀后血量低而不敢推塔推线等现象时，则会提高英雄的血量奖励比重并且降低击杀奖励。若发现英雄只发育获取经济而不推塔时，则会提高塔的奖励占比。

基于以上设计思路，最终的奖励版本为：

```
{
    "reward_money": "0.007",
    "reward_exp": "0.007",
    "reward_hp_point": "2.0",
    "reward_ep_rate": "0.65",
    "reward_kill": "-0.4",
    "reward_dead": "-1.5",
    "reward_tower_hp_point": "5.0",
    "reward_last_hit": "0.3",
}
```

## 五、特征与规则

在初赛阶段，我们并未针引入新的特征，也未引入后置规则。

## 六、 强化学习算法

在算法方面我们队伍主要是基于 Mastering Complex Control in MOBA Games with Deep Reinforcement Learning 中提到的 GAE 和 dual clip PPO 方法，其中 dual clip PPO 是针对 PPO 算法的一种 off-policy 优化版本。

对庞大的动作空间进行解耦，实现简化策略网络设计的同时，还能在训练阶段保证动作的多样性。基于子动作间的重要性和层次赋予不同的损失权重（例如，$w_{offset} \leq w_{target} \leq 1$）：

$$\mathcal{L}_{policy}(\theta) = -\sum_{i=0}^{N_a-1} w_i \mathbb{E}_t \left[ \max \left( \min \left( r_t^{(i)}(\theta) A_t, clip \left( r_t^{(i)}(\theta), 1-\epsilon, 1+\epsilon \right) A_t \right), cA_t \right) \right]$$

引入 policy entropy，防止策略过早固化。根据我们的经验，在 fine-tuning 阶段，调整 entropy term 也能起到比较正面的作用。

$$\mathcal{L} = \mathcal{L}_{policy} + \alpha\mathcal{L}_{value} + \beta\mathcal{L}_{entropy}$$

## 七、 系统工程架构

我们使用了一些常见的调参手段，比如随着训练时间推移逐步减少 entropy 的权重，对学习率进行退火，调整 gamma 和 slow_time。

我们还使用了 Gradient Clipping 技巧来增强数值稳定性，防止训练过程中梯度爆炸，让参数更新维持在相对稳定的水平。

召唤师技能的默认设置是狂暴，考虑到召唤师技能这块会是一个特别容易过拟合的地方，我们为不同的英雄配备不同的召唤师技能，希望在训练过程中能带来差异性和意外性。

```
{
    "houyi": "sprint",
    "gongsunli": "frenzy",
    "makeboluo": "execute",
    "direnjie": "heal",
    "luban": "flash"
}
```

## 八、 模型迭代过程

比赛的英雄池：鲁班、后羿、狄仁杰、公孙离和马可波罗。

论文 Towards Playing Full MOBA Games with Deep Reinforcement Learning 强调了课程学习对泛化性任务的好处。我们结合实验和观察，也探索出了一套针对五个射手的模型迭代方案：

① 第一阶段，只训练后羿和公孙离，后羿的操作比较单调，公孙离的技能组合丰富，都是有代表性的英雄。较原先完整的多任务学习，该模式的训练难度直线下降。虽然公孙离的强度低后羿一档，但也能勉强跟上后羿的学习。

② 第二阶段，从两个英雄泛化拓展到四个英雄，不包括马可波罗（相较其余四个英雄，马可波罗的机制比较特殊）。在这个阶段，通过放弃直接学习马可波罗，来提升另外四个英雄的能力上限。

③ 第三阶段，从四个英雄泛化拓展到五个英雄。最终，以轻微损害其余四个英雄的表现为代价，让马可波罗的能力有了提升。

④ 第四阶段，进一步 fine-tune。

## 九、 训练效果分析

在模型迭代过程中，我们注意到了英雄间的强度差异，鲁班>后羿>狄仁杰>公孙离>马可波罗。总体上英雄可以分为两类，一类包含鲁班、后羿、狄仁杰，另一类包含公孙离和马可波罗。前者操作简单（学习相对容易），后者操作复杂（学习难度大），最终在模型表现上前者会远强于后者。狄仁杰介于暴力流和技巧流之间，他的经验对公孙离、鲁班和后羿有帮助。而马可波罗因为其特殊的机制，会拖累模型的整体表现。基于以上观察，我们采取了 “弃车保帅”的策略，通过放弃马可波罗换来其余英雄对线时的统治力。

## 十、 总结与展望

泛化性这个课题的难度明显高于我们的预期，我们最终模型的效果也并不算突出。在比赛资源受限的情况下，我们做出了取舍，通过放弃马可波罗换取了其他英雄的优势。在未来，我们可能会尝试知识蒸馏实现模型多合一。

## 参考文献

[1] Mastering complex control in moba games with deep reinforcement learning

[2] Towards Playing Full MOBA Games with Deep Reinforcement Learning