打 鲁 班 ，不 加 班

第二届“开悟”大赛技术文档

队　　员：邵键准、张宏昌、蒋雨航、曲云、王博源

作　　者：曲云、王博源

单　　位：清华大学自动化系

指导教师：季向阳教授

日　　期：2022 年 5 月

第一节 网络结构设计

1.1 总体架构

在网络设计方法上，首先，关于网络的总体架构，我们参考一些论文的方法，进行了三种设计，如图 1.1 所示：

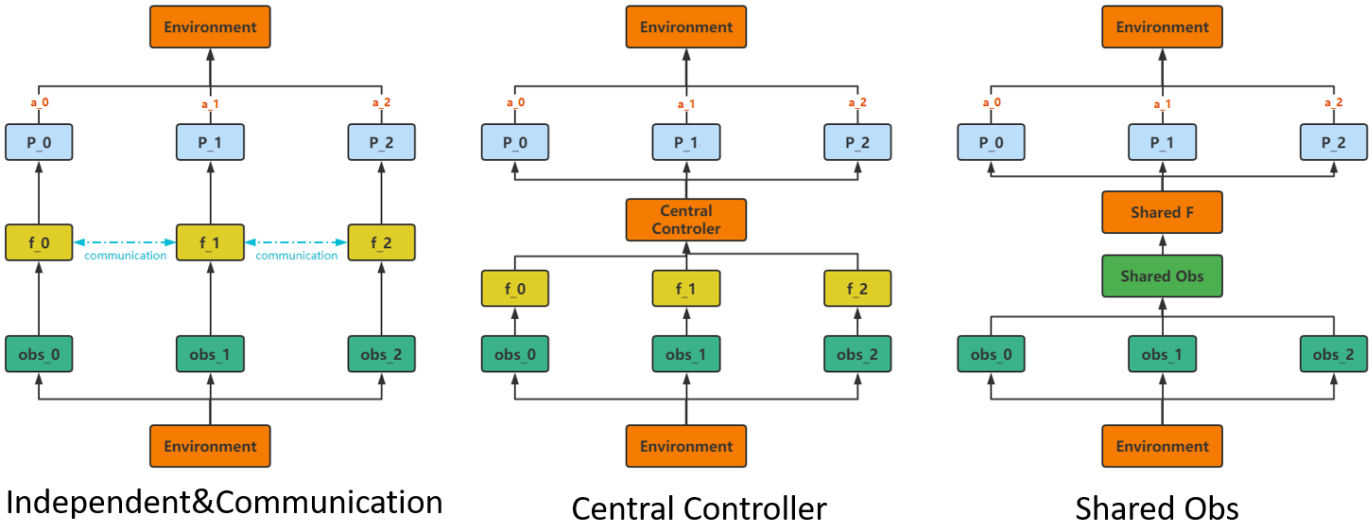

图 1.1 三种网络设计

- **独立网络加交流**

如图 1.1 左侧网络示意图所示，此种设计对于三位英雄采用相对独立的状态编码网络、策略网络等。这样做的好处是，使得每位英雄各自的网络处理之间互不影响，这样就可以针对每位英雄各自的特点设计出最符合其作战特性的网络处理结构。同时，为保证英雄之间的紧密合作，在各自英雄网络保持独立的同时增加交流机制，这样既给予了独立的空间，又进行了有效的信息传递。

- **中央控制器网络**

如图 1.1 中部网络示意图所示，每位英雄首先各自独立处理自己的状态特征，处理完毕后采用一个中央控制器来汇总当前团队的所有英雄编码特征，最后使用这个中央控制器来为每个英雄产生自己的动

作策略。这样做的好处是将中央控制器视作一个全局指挥官，其能看到全局团队所有人的信息，以一个宏观的视角来做出决策，这样的决策会更具系统性与完备性。

• 单智能体控制网络

如图 1.1 右部网络示意图所示，这种设计初衷是，鉴于每位英雄均有一个状态$[s_1, s_2, s_3]$，其首先通过特征的整合，将团队所有成员的特征整合为一个统一的s_{team}，之后每位英雄之间采用共享参数的处理方式对s_{team}进行处理，最后输出每位成员对应的动作策略，在某种程度上将多智能体的决策转化为了单智能体控制。这样的优势在于，团队每位成员的$[s_1, s_2, s_3]$之中存在大量的冗余信息，进行特征合并可以对信息进行精简，从而加速训练；同时由于一开始就合并了整个团队的信息，更贴近一个上帝视角来控制比赛，从而使模型的决策更全面。

上述三种设计方案均有各自的逻辑性，最终，经实验验证与比对，选择第一种独立网络加交流的机制作为后续 AI 系统的基础架构。

1.2 状态编码网络

由于每位英雄输入 4586 维度，过于庞大不能直接处理，需要进行更精细的编码设计。每位英雄的状态特征s包含：图像、英雄共有、英雄私有、士兵、防御塔、全局这 7 大特征部分。对于图像特征，采用卷积神经网络进行处理，对于剩余的特征每部分进行专有的特征处理网络，如图 1.2 所示。

同时，我们针对于不同的英雄进行了略有区别的处理。其中，考

虑到赵云和李元芳主要针对单一英雄进行输出，因此，更关注于提取更可能击杀的目标的特征；而貂蝉则倾向于在范围内进行攻击，则不着眼于单一敌方英雄特征。此外，因为赵云的主要职能是打野，因此特征提取方面，更加关注野怪特征，而选择性忽略士兵和塔等无关特征，由于这样的设计，在实验中发现，赵云的确更加关注野怪而忽略士兵，同时能够越塔杀人并全身而退；而李元芳和貂蝉的设计则相反。

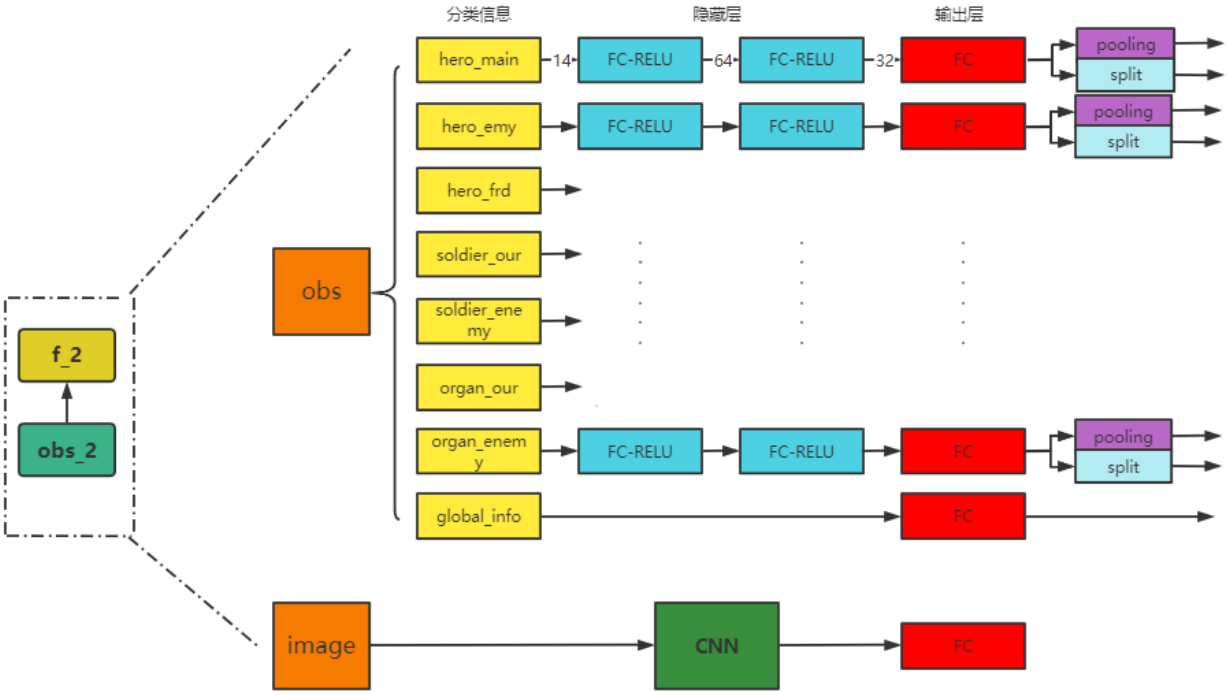

图 1.2 状态编码网络设计

1.3 policy 网络

我们的交流机制设计参考了 VBC 方法。如图 1.3 所示，该方法是针对 QMIX 算法的改进。其在每个 agent 网络中首先对局部观测进行编码。利用编码后的特征一方面估计 local Q，另一方面经过进一步处理得到与 local Q 维度相同的交流信息并发送，同时接收其他智能体的交流信息。之后将自身估计得到的 local Q 与接收的交流信息相加，即得到最终的 local Q。

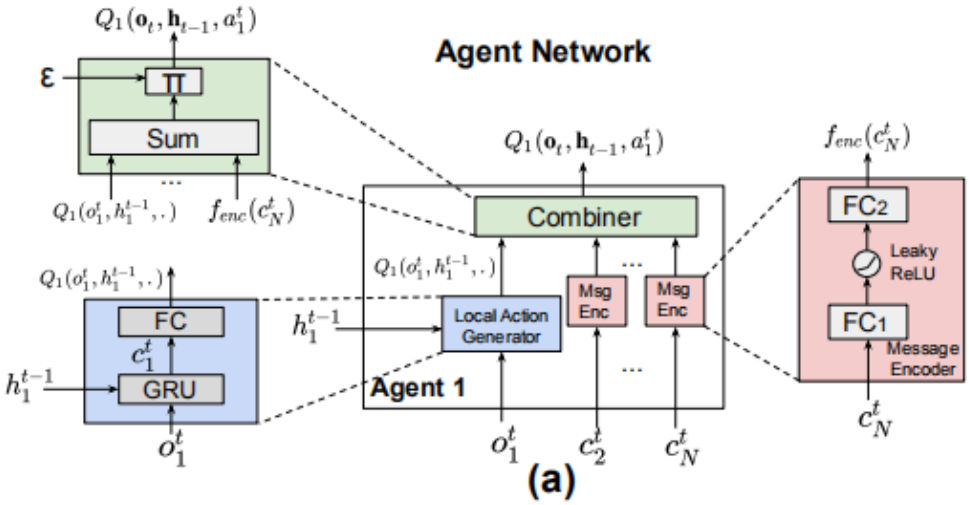

图 1.3 VBC[18]

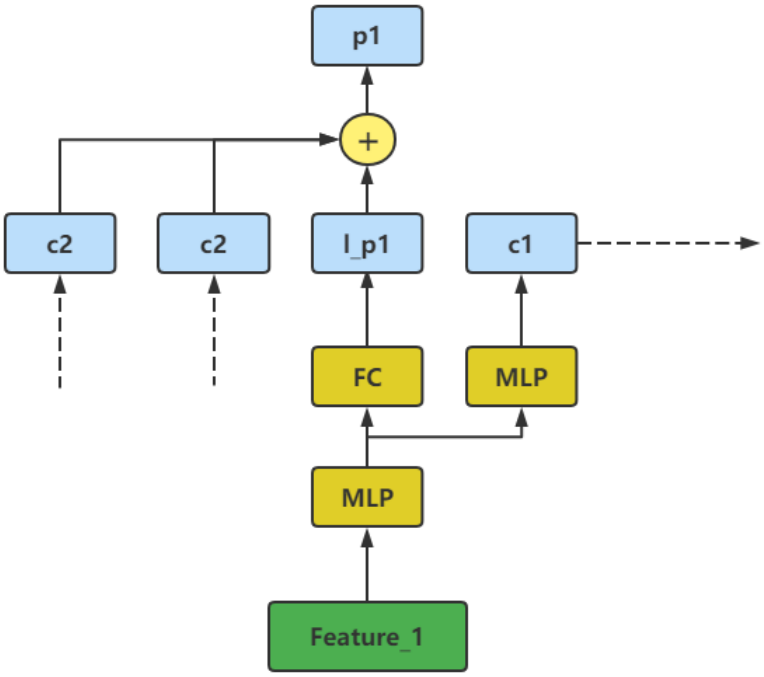

图 1.4 policy with communication

因此，参考 Variance Based Control[18]论文中的 communication 思想，在自身生成 local policy 的同时，生成维度相同的交流信息并发给其他英雄，而自身接收其他英雄的交流信息，与自己的 local policy 相融合，产生自己最终的 policy。这样做的目的，是将自己的 policy 分为自身意愿和队友建议两部分，即既考虑自身的动作意图，又考虑队

友目前对我的需求，从而实现更好地团队协作。

1.4 价值 value 估计网络

- **multi-head**

为了更准确地估计不断变化的状态价值，通过分解奖励，引入多头奖励机制，根据先验知识设计了不同 head，具体来说，分为四个头，分别代表发育、杀人、血量、胜利,这样可以使得 value 网络更好的专注拟合这四个方面。

- **global value function**

使用一个全局价值估计函数来估计每方阵营的零和团队奖励，从而将输出结果与每位英雄的多头奖励进行融合。这样的好处是更好地拟合团队奖励，从而促进交流合作。

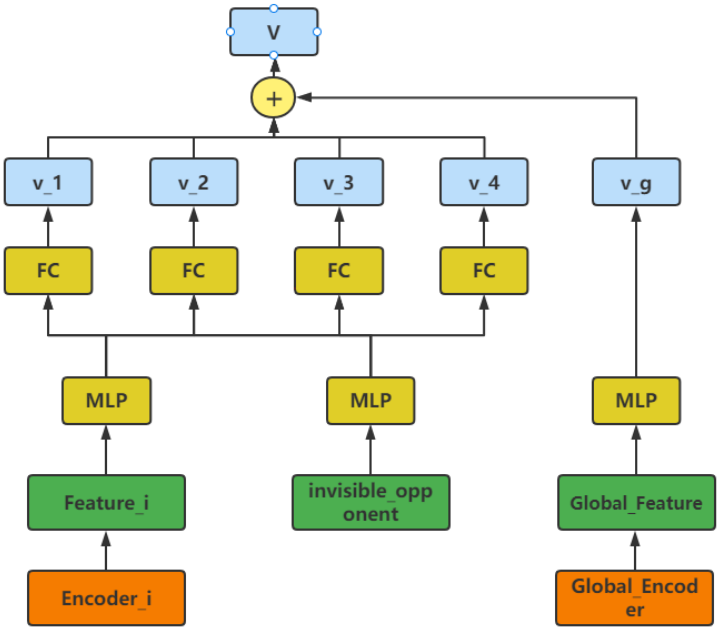

图 1.5 value function

- **centralized training**

如图 1.5 所示，借鉴 centralized training 的思想，在训练时引入不

可见的敌方信息来辅助训练。这样的好处是游戏环境对我方是部分可见的，使用全局可见的信息可以辅助价值估计网络的拟合。

第二节 奖励设计

2.1 个人奖励调试经验

最终的模型中，训练前期的奖励设计相对简单，仅包含较少的项以加速训练，在后期通过奖励调整来提升效果。

具体调整思路是，当发现团队某个英雄作战时对某方面表现不足或是关注不够时，就加大其相关奖励权重，比如发现貂蝉和李元芳不关注自己的血量，经常进行送人头行为，就加大奖励中的 hp_rate_sqrt_sqrt、deadCnt 等来使其意识到送人头行为的弊端，从而更好地进行策略调整，经研究后较为有效的奖励如表 2.1 所示。

表 2.1 三位英雄有效训练奖励权重

奖励名称	赵云	貂蝉	李元芳
hp_rate_sqrt_sqrt	5.0	0	1.0
money	0.005	0.020	0.020
exp	0	0	0
tower	0.05	0	0
killCnt	0	5.0	0
deadCnt	0	-5.0	-3.0
assistCnt	0	2.0	0
total_hurt_to_hero	0.30	0.05	1.0
atk_monster	0.50	0	0
win_crystal	6.0	6.0	6.0
atk_crystal	0	0	0

2.1 团队奖励调试经验

同时，为了加强团队协作，在后期引入团队奖励，之所以不在前期引入是因为实验发现，前期引入团队奖励会减慢训练速度。总体的设计倾向为，以金钱和经验为主要奖励，推塔和伤害为次要奖励，其他作为辅助项，以此思想进行调整。

第三节 训练调优

3.1 加强探索

注意到游戏环境较为复杂，游戏内空间可探索性巨大，使用跳帧(frame-skip)的方式来加强探索。具体做法是当检测到当前状态下的动作是移动动作(move)时，就将该移动动作保存下来，在此之后跳过t个决策帧，在这t个决策帧之中使用相同的动作，即之前保存下来的移动动作。这样可以增大智能体对环境的探索性，从而更好地适应环境。

3.2 退火

注意到学习率的参数α与控制熵值大小的参数β具有相同性，对于学习率α来说，当α较大时有助于加速模型的训练，不过不容易进行更精细的调优，当α较小时容易使模型陷入局部最优解，同时使模型更确定性。对于熵值参数β来说，β较大时熵值的惩罚较大，有助于避免英雄的动作策略过于确定，可以增大模型对环境的探索，而当β较小时熵值惩罚较为微弱，有利于动作策略的稳定。

因此，实验中采用退火技术，每隔一定时间将学习率α和熵值参数β增大到一定的值，之后逐步随时间下降。这样的目的是首先通过较大的学习率与熵值来敦促模型跳出局部最优，之后再不断降低两个参数，逐步进行训练巩固，最终使模型寻找到更合适的最优解。

3.3 环境不对称

游戏作战环境是红蓝双方博弈形式，注意到红蓝双方的状态特征存在偏差，并不是完全对称的。实验验证蓝色方的平均胜率显著高于红色方。为解决对应的问题，尝试将红蓝色方的模型进行单独训练，以及更改两侧的样本比例数来平衡弱势方的劣势，如设定蓝色方训练样本比例为 40%，对应的红色方样本比例为 60%。

参考文献

[1] Ye D, Liu Z, Sun M, et al. Mastering complex control in moba games with deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 6672-6679.

[2] Ye D, Chen G, Zhang W, et al. Towards playing full moba games with deep reinforcement learning[J]. arXiv preprint arXiv:2011.12692, 2020.

[3] V. d. N. Silva and L. Chaimowicz. Moba: a new arena for game ai. arXiv preprint arXiv:1705.10443, 2017.

[4] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou. Playing Atari with Deep Reinforcement Learning. NIPS 2013.

[6] Rashid T, Samvelyan M, Schroeder C, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2018: 4295-4304.

[7] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. arXiv preprint arXiv:1706.02275, 2017.

[8] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[9] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, 2018: 1861-1870.

[10] Yu C, Velu A, Vinitsky E, et al. The surprising effectiveness of mappo in cooperative, multi-agent games[J]. arXiv preprint arXiv:2103.01955, 2021.

[11] Lin T, Huh J, Stauffer C, et al. Learning to Ground Multi-Agent Communication with Autoencoders[J]. Advances in Neural Information Processing Systems, 2021, 34.

[12] Panna Felsen et al. “Where Will They Go? Predicting Fine-Grained Adversarial Multi-agent Motion Using Conditional Variational Autoencoders” European Conference on Computer Vision (2018).

[13] Choi, E., Lazaridou, A., & de Freitas, N. (2018). *COMPOSITIONAL OBVERTER COMMUNICATION LEARNING FROM RAW VISUAL INPUT*. 18.

[14] Eccles, T., Bachrach, Y., Lever, G., Lazaridou, A., & Graepel, T. (2019). Biases for Emergent Communication in Multi-agent Reinforcement Learning. *arXiv:1912.05676 [cs]*. http://arxiv.org/abs/1912.05676

[15] Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *arXiv:1605.06676 [cs]*. http://arxiv.org/abs/1605.06676

[16] Iqbal, S., & Sha, F. (2019). Actor-Attention-Critic for Multi-Agent Reinforcement Learning. *Proceedings of the 36th International Conference on Machine Learning*, 2961–2970. https://proceedings.mlr.press/v97/iqbal19a.html

[17] Jiang, J., & Lu, Z. (2018). Learning Attentional Communication for Multi-Agent Cooperation. *Advances in Neural Information Processing Systems*, *31*. https://proceedings.neurips.cc/paper/2018/hash/6a8018b3a00b69c008601b8becae392b-Abstract.html

[18] Zhang, S. Q., Zhang, Q., & Lin, J. (2019). Efficient Communication in Multi-Agent Reinforcement Learning via Variance Based Control. *Advances in Neural Information Processing Systems*, *32*. https://proceedings.neurips.cc/paper/2019/hash/14cfdb59b5bda1fc245aadae15b1984a-Abstract.html