Chapter 1

General MARL/PPO

1.1 The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games

https://www.researchgate.net/publication/349727671_The_Surprising_Effectiveness_of_MAPPO_in_Cooperative_Multi-Agent_Games

PPO sample efficiency

1 GPU desktop, 1 Multicore CPU for training

centralized value function - global state s insted of local o

environments: Particle world environment

PPO used be seen as sample less efficient, hence for academic purposes MADDPG anv value-decomposed Q-learning

Minimal hyperparameter tuning and no domain specification

Decentralized learning each agent its own policy, suffer from non-stationary transitions

two lines of research - CTDE (this) and value decomposition

in single agent PG advantage normalization is crucial

considered implementation details - input norm, value clipping, orthogonal init, gradient clip - all helpful and included

another - discretization action space for PPO to avoid bad local minima in continuous, layer normalization

MLP vs Recurrent

5 implementation details:
Value norm: running average over value estimates, value network regress to normalized target values (Pop art technique)
Agent-specific global state: concate of all o_i as input to critic
(agent specific global cannot be used in QMix, which uses single mixer network common to all agents)
Training Data Usage: importance sampling to perform off-policy correction ??
multiple epochs may suffer from non stationarity -¿ 15 to 5 epochs (easy to hard)
No mini-batches -¿ more data to estimate gradients -¿ imporved practical performance
Action masking: unavailable actions when computing action probabilities - both forward and backward
Death masking: zero states with agent ID as value input for dead agents

Chapter 2

Overcooked related

https://github.com/HumanCompatibleAI/overcooked_ai

2.1 On the Utility of Learning about Humans for Human-AI Coordination

https://arxiv.org/abs/1910.05789

agents assume their partner to be optimal or similar fail to be understood by humans

gains come from having agent adapt to human's gameplay

effective way to tackle two-player games is train agent with set of other AI agents, often past versions

collaboration is fundamentally different from competition (we need to go beyond self-play to account for human behavior)

incorporating human data or models into training leads to significant improvements (behavior cloning model)

Population Based Training is online evolutionary alg, adapts training hyperparameters and perform model selection agents, whose policies are parametrized by NN and trained with DRL alg. each PBT iteration pair of agents are drawn, trained for number of steps and have performance recorded at end of PBT iteration, worst performing agents are replaced with copies of best and parameters mutated

human behavior cloning peformed better than with Generative Aversarial Imitation Learning (GAIL)

PBT better than PPO self-play because they are trained to be good with population coordination

Agents designed for humans. Start with one learned human BC as part of environment dynamic and train single agent PPO.

start with ppo self-play and continue with training with human model

planning alg A*

two human behavior cloning models Hproxy used for evaluation and PPOBC learned against learned human models

best result self-play with self-play

for human interaction was best PPOBC with HProxy...PPOBC is overall preferable

PPOBC outperformes human-human performance

SP agents became very specialized and so suffered from distributional shift when paired with human models

future work - better human models, biasing population based training towards humans

READ AGAIN if interested

2.2 PantheonRL: A MARL Library for Dynamic Training Interactions

`https://iliad.stanford.edu/pdfs/publications/sarkar2022pantheonrl.pdf`

PantheonRL new software package for marl dynamic

Combination of PettingZoo and RLLIB - customziation of agents

prioritizes modularity of agent objects - each has separate replay buffer, own learning alg, role

(other powerfull DRL library - StableBaselines3)

The modularity of the agent policies com- bined with the inheritance of StableBaselines3 capabilities together give users a flexible and powerful library for ex- perimenting with complex multiagent interactions

2.3 Investigating Partner Diversification Methods in Cooperative Multi-agent Deep Reinforcement Learning

https://www.rujikorn.com/files/papers/diversity_ICONIP2020.pdf

PBT have diversity problem -¿ PBT agents are not more robust than self-play agents and aren't better with humans

creating diversity by generating pre-trained partners is simple but effective

(partner sampling - playing with uniformly sampled past versions of partner - lacks diversity, past versions have similar behavior)

(population-base training, pre-trained partners)

testing self-play and cross-play with these agent types (SP, SPpast, PBT, PTseeds, PTdiverse)

PTdiverse(hyperparameters) and PTseeds come from self-play

LOVED THIS ARTICLE for it's simplicity

2.4 Evaluating the Robustness of Collaborative Agents

https://arxiv.org/abs/2101.05507

how test robustnes if cannot rely on validation reward metric

unit testing (from software engineering) - edge cases, eg. where soup was cooked but not delivered

incorporating Theory of mind to human model

human modal diversity by using population of human models

state diversity - init from states visited in human-human gameplay

test suite provides significant insight into robustness that is not correlated with average validation reward

"improved robustness as measured by test suite, but decrease in average validation reward"

simple ML metrics are insufficient to capture performance and we must evaluate results base on human judgement

2.5