Overcooked – Multi Agent Reinforcement Learning

Jasper Robbins

https://tenor.com/view/overcooked-chef-overcooked-chef-overcooked-melon-overcooked-whistle-gif-22607790

Gameplan

1

Problem Description

2

Algorithms

3

Data

4

Plan

5

Evaluation

6

End

Problem Description

With RL woven into much of today’s tech, learning it gives you a leg up - and lets you build some cool projects along the way

Robotics : OpenAI Dactyl, DeepMind RGB-Stacking

- Policies trained in simulations transfer to real robotics via domain randomization

Datacenter Cooling : Google/Deepmind autonomous cooling

- An agent tweaks HVAC setpoints from feedback to cut energy under safety constraints

LLM Alignment : InstructGPT/ChatGPT (RLHF)

- Human preference rewards + PPO steer a language model towards more helpful responses

Internet Congestion Control : Aurora

- Chooses sending rates to balance throughput, latency, and loss across diverse links

Discovery & Design : AlphaDev, REINVENT, MolDQN

- Feedback-guided search over large data/design spaces to surface high value candidates (medical discoveries, algorithm discoveries, & more)

What is the goal?

To train cooperative policies that can

(a) Complete game tasks efficiently

(b) Coordinate with each other & real people to perform in team-based tasks

Why do this?

MARL tests systems in the messy, multi-actor world. It learns the environment and adapts to other agents (and human mistakes), exposing real-life inconsistencies

TLDR

RL is everywhere, even if we prototype in games, the same decision logic transfers to the real world…
Train safely in simulations, deploy on hardware.

Algorithms

Which Algorithm do we choose?

Due to its proven success in OpenAI Five (Dota 2), the plan is to focus on PPO, but experiment with other algorithms

	PPO Proximal Policy Optimization	SAC Soft Actor Critic	MADDPG Multi-Agent Deep Deterministic Policy Gradient	QMIX Value Decomposition for MARL
What is it?	• **On-policy** actor-critic with a **clipped surrogate** loss to keep updates small and stable	• **Off-policy** actor-critic that maximizes **expected return + entropy** (temperature α controls exploration)	• Deterministic decentralized actors with centralized critics during training to reduce non-stationarity	• Value Decomposition Method For Cooperative Agents
STRENGTHS	• Stable updates • Simple, robust baseline • Good for continuous control	• Off-policy: sample efficient • Entropy leads to robust exploration • Strong for continuous actions	• Centralized critic stabilizes • Continuous multi-agent control • Better credit assignment	• Cooperative, discrete tasks • Value decomposition crediting • Stable and scalable
WEAKNESSES	• On-policy: data hungry • Weak on sparse rewards • Needs careful LR/clip	• More hyperparameters • Temp tuning matters • Struggles in non-stationary MARL	• Exploration-fragile • Poor scaling with MANY agents • Needs global info in training	• Monotonicity limits synergy • Not for competitive settings • Not for continuous actions

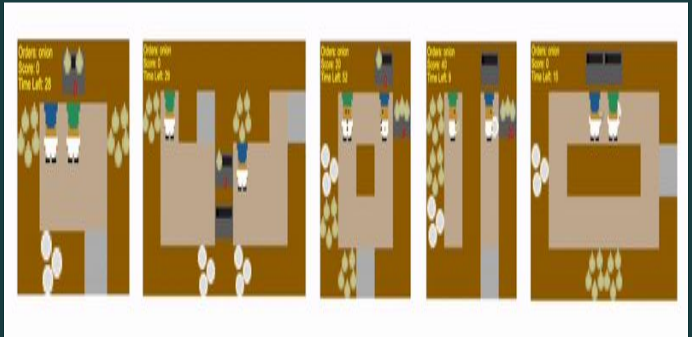
Data

How do we collect data?

Data is gathered through **simulated gameplay**

States: Through the game wrapper
Actions: Through game wrapper

Berkeley Environment:
https://github.com/HumanCompatibleAI/overcooked_ai



Left GIF - https://github.com/HumanCompatibleAI/overcooked_ai/blob/master/images/layouts.gif

Right Gif - https://tenor.com/view/overcooked-chef-overcooked-chef-overcooked-melon-overcooked-whistle-gif-22607790

States: Full screen screenshots
Actions: OS-level key injections (PyDirectInput)

Actual Game:
https://steamcommunity.com/app/728880

Plan

How do we accomplish the goal?

Step 1

Get a single agent to play the Berkeley environment alone

Step 2

Get two agents to play the Berkeley environment together

Step 3

Build a single agent to play the actual game alone

Step 4

Build two agents to play the actual game together

Evaluation

Evaluation during training will be done through a custom reward system we build

Evaluation post training will be done through reward counting, as well as eye testing our models

(a) Are they performing?

(b) Do they understand the game?

Note: If using libraries, can also gain additional insight into training w/ Tensorboard and custom implementations such as heat maps and short video rollouts during training

Related Work

Overcooked AI benchmark & human AI coordination : (https://arxiv.org/pdf/1910.05789)

Generalization & Zero-Shot Coordination : (https://arxiv.org/pdf/2406.17949)

PPO in MARL : (https://arxiv.org/html/2409.03052v1#S1)

Thank you

Looking for 2-4 people, can expand according to interest

Should be

- Knowledgeable about python
- Interested in RL

Would like help in any of the following:

- Better understanding the process of Starting & Resetting training in the actual game (building a game wrapper for the actual game?)
- Diving into any specific algorithm you would like to understand & implement
- Help in actually training models if willing
  - Note: I will be doing all of training if needed as it can be compute heavy and I don't want to mess up any of yalls computers

Interested?
Reach out on discord at Jasper or @swf.

Or just talk to me, im likely just the tall one

TY again