Overview of Google Football Environment

Linghui Meng

2021.03.05

Google Football Environment is a tough and dynamic environment to learn in, presented by Google Research and Manchester City F.C，aiming to explore the potential power of artificial intelligence on football sports, guide the coach based on these data and promote the development of AI. *https://github.com/google-research/football*

Observation and Action Set

- Observation Wrappers
  - simple115_v2: 115 floats
    - 22 - (x, y) coordinates of left team players
    - 22 - (x, y) direction of left team players
    - 22 - (x, y) coordinates of right team players
    - 22 - (x, y) direction of right team players
    - 3 - (x, y and z) ⚽ ball position

    3 - ball direction

    3 - one hot encoding of ball ownership (noone, left, right)

    11 - one hot encoding of which player is active

    7 - one hot encoding of game_mode

Observation and Action Set

- Observation Wrappers
  - extracted: It consists of several 72 * 96 planes of bytes, filled in with 0s except for:
    - 1st plane: 255s represent positions of players on the left team
    - 2nd plane: 255s represent positions of players on the right team
    - 3rd plane: 255s represent position of a ball ⚽
    - 4th plane: 255s represent position of an active player

Table 1: Action Set

Top	Bottom	Left	Right
Top-Left	Top-Right	Bottom-Left	Bottom-Right
Short Pass	High Pass	Long Pass	Shot
Do-Nothing	Sliding	Dribble	Stop-Dribble
Sprint	Stop-Moving	Stop-Sprint	—

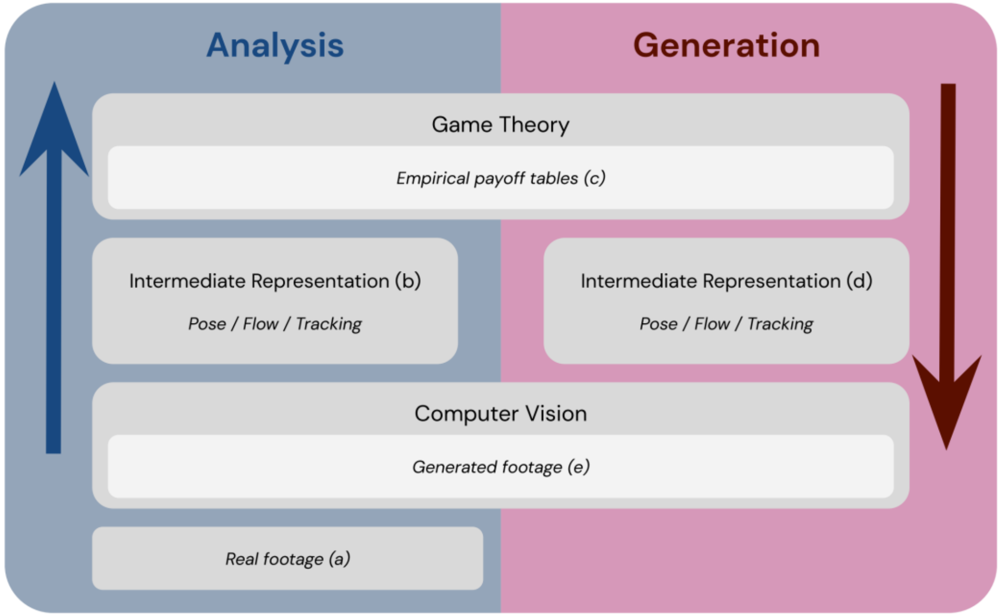
Scenarios in Google Football Environment

Name	Description
Empty Goal Close	*Our player starts inside the box with the ball, and needs to score against an empty goal*
Empty Goal	*Our player starts in the middle of the field with the ball, and needs to score against an empty goal*
Run to Score	*Our player starts in the middle of the field with the ball, and needs to score against an empty goal. Five opponent players chase ours from behind*
Run to Score with Keeper	*Our player starts in the middle of the field with the ball, and needs to score against a keeper. Five opponent players chase ours from behind*
Pass and Shoot with Keeper	*Two of our players try to score from the edge of the box, one is on the side with the ball, and next to a defender. The other is at the center, unmarked, and facing the opponent keeper*
Run, Pass and Shoot with Keeper	*Two of our players try to score from the edge of the box, one is on the side with the ball, and unmarked. The other is at the center, next to a defender, and facing the opponent keeper*

Scenarios in Google Football Environment

Name	Description
3 versus 1 with Keeper	*Three of our players try to score from the edge of the box, one on each side, and the other at the center. Initially, the player at the center has the ball, and is facing the defender. There is an opponent keeper.*
Corner	*Standard corner-kick situation, except that the corner taker can run with the ball from the corner. The episode does not end if possession is lost.*
Easy Counter-Attack	*4 versus 1 counter-attack with keeper; all the remaining players of both teams run back towards the ball.*
Hard Counter-Attack	*4 versus 2 counter-attack with keeper; all the remaining players of both teams run back towards the ball.*
11 versus 11 with Lazy Opponents	*Full 11 versus 11 game, where the opponents cannot move but they can only intercept the ball if it is close enough to them. Our center-back defender has the ball at first. The maximum duration of the episode is 3000 frames instead of 400 frames.*

Properties of Google Research Football



Opening three topics

- Self-play training for Multi-player setting
- Multi-Agent Settings
- Representation learning based on the observation and even knowledge

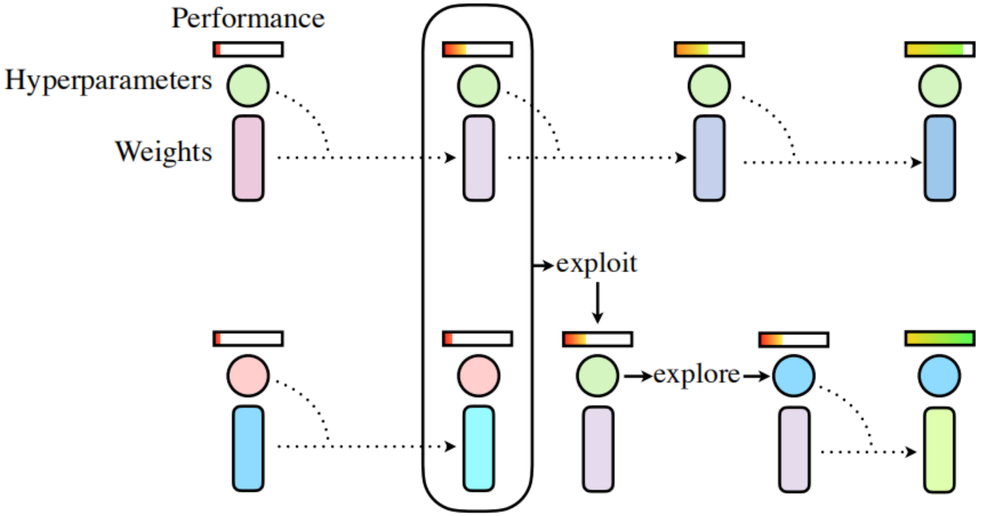
Opening Topics: 1. Population-Based Training

Motivation

- Using a process of population based training (PBT) to optimize internal rewards and hyperparameters to maximize performance.

Method

- Periodically sampled another agent from the League, and estimated the win probability of a team using Elo scores.
- If under-performing, copying the policy, the internal reward transformation, and hyperparameters of the better agent, and explored new internal rewards and hyperparameters.



Future Work

- To further improve the reward shaping
- To increase the diversity of the opponent's strategy pool
- To design a more reasonable evaluation mechanism

Reference : Jaderberg M, Czarnecki W M, Dunning I, et al. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. arXiv[J]. arXiv preprint arXiv:1807.01281, 2018.
Jaderberg M, Dalibard V, Osindero S, et al. Population based training of neural networks[J]. arXiv preprint arXiv:1711.09846, 2017.

Opening Topics: 2. Multi-Agent Settings

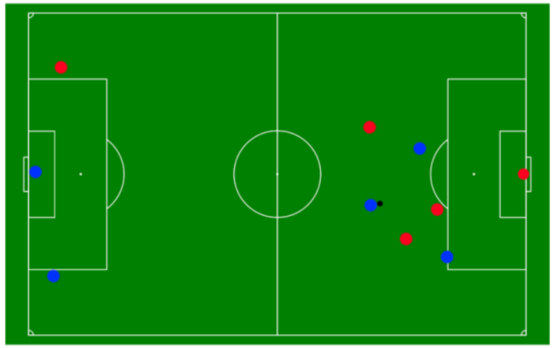
Implementation details for multi-agent reinforcement learning:

- Environment creation. You need to pass number_of_players_agent_controls argument to gym.make called here: [*seed_rl/football/env.py*]
- The network. If you wish to modify the network used by default, it is defined here: *seed_rl/football/networks.py* and used here: *seed_rl/football/vtrace_main.py#L39*
- Packet-bit observation wrapper. It is responsible for compressing data before sending them over the network. It's good to be aware that it expects specific format of data and adds additional padding to observations: *seed_rl/football/observation.py*

The goal is to train a set of individual agents starting from random policy to a cooperation team.

Open research fields in the right 3v3 example:

- Skill learning: Like offside strategy for violation of opponents
- Topology of Network structure for represent the team: based on locations
- Distributed learning like SEED RL
- Curriculum learning for transferable model and solve the complex task
- Self-training for population based learning with Elo value



Opening Topics: 3. Representation learning

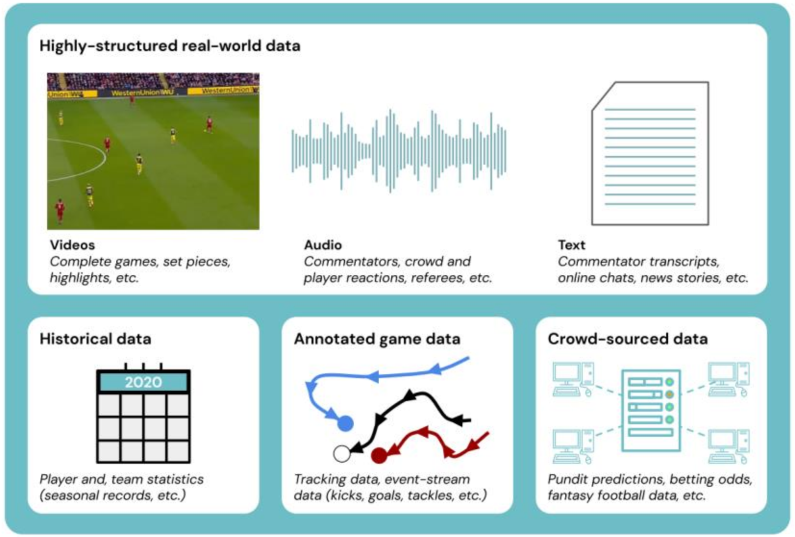
The goal of offline representation learning from structured multi-modal observation with different dimensions is for the better perceptron ability of decision making process.

Method used for representation learning:

- Multiple different hierarchical module for transformation of data with different dimensions and modals
- Pretraining by advanced backbone to explore the semantic information of the raw observation
- Self-supervised learning especially with contrastive method for a better representation
- Extracting knowledge from the history data and integrate it into representation

Open research fields for representation learning in football game

- Joint training with the latter policy models by integrate multi-modal data at different levels
- Offline RL for learning the common knowledge from the input data
- Symbolic learning with inductive bias for the invariant representation

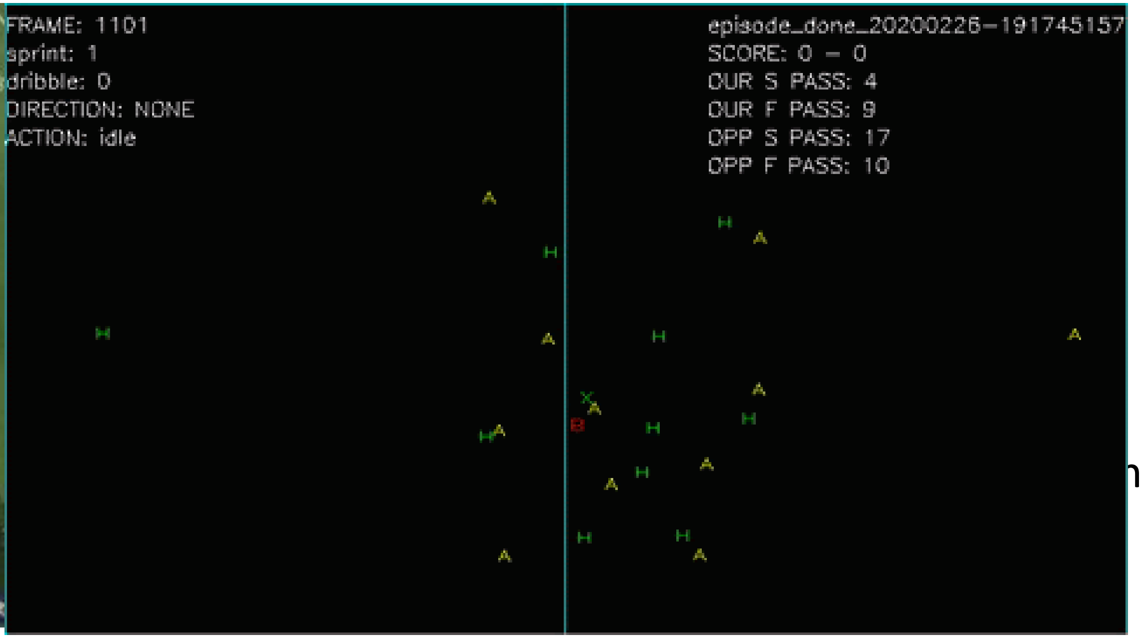


Brief Introduction of the Football Competition

Challenge implemented on Kaggle platform including single-player and multi-player competition

Rules

In this competition you control a single player in 11-player teams. Rules are similar to the official football (soccer) rules, including offsides, yellow and red cards.

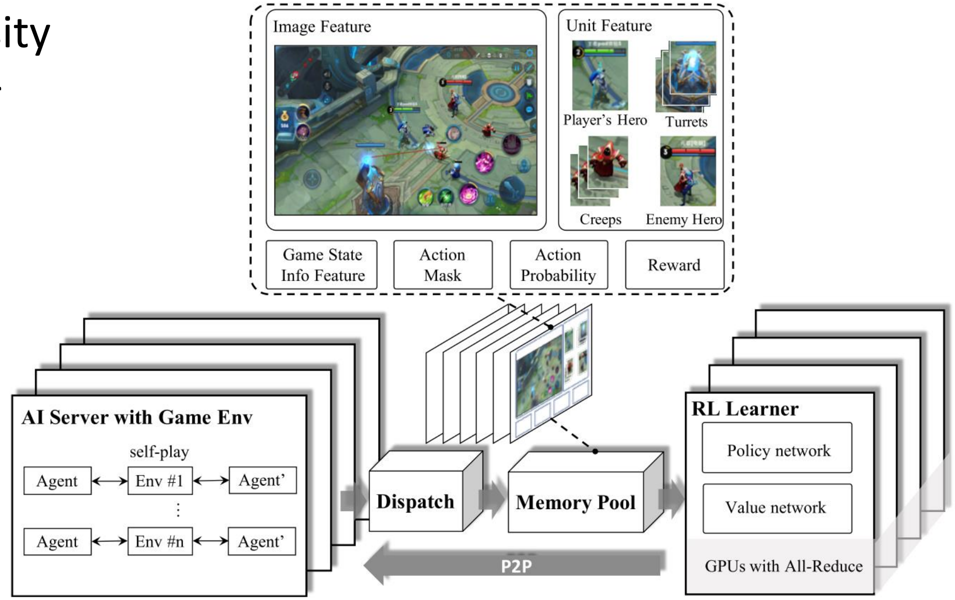


Solutions for the Competition: 1st WeKick

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

The key techniques in WeKick (based on distributed PPO training)

- Self-play: Maintain an opponent pool based on the LB and add diversity
- GAIL: Generative adversarial imitation learning from the trajectory of opponent team from the 2nd and 3st for a better reward shaping
- Multi-skill training: Training series of skilled agents as multiple opponents to augment the opponent pool and generalize the main model



Feature Engineering (simple115 based)

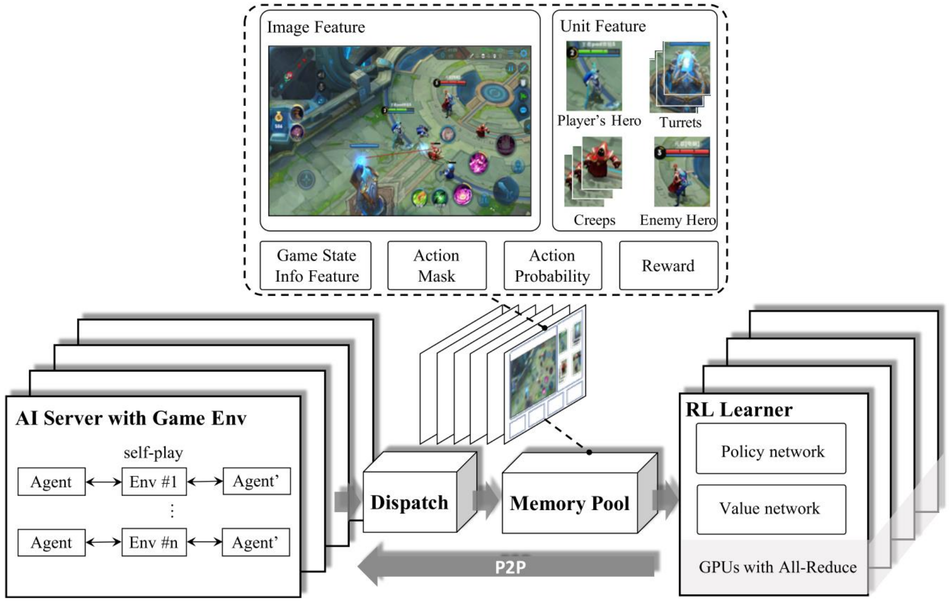
- Relative pose(position, direction) among teammates and opponents
- Relative pose between active player and ball
- Offside flag to mark potential offside teammates
- Sticky actions/card/tired factor status

Solutions for the Competition: 1st WeKick

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

General reward shaping because of too sparse score

- Intercept & outside & offside: zero-sum, +0.2/-0.2 possession
- Slide: zero-sum, +0.2/-0.2 our/opponent successful slide, useful for defence
- Hold ball: zero-sum, +0.001/-0.001 if we/opponent hold the ball
- Pass: +0.1 for each pass before a goal

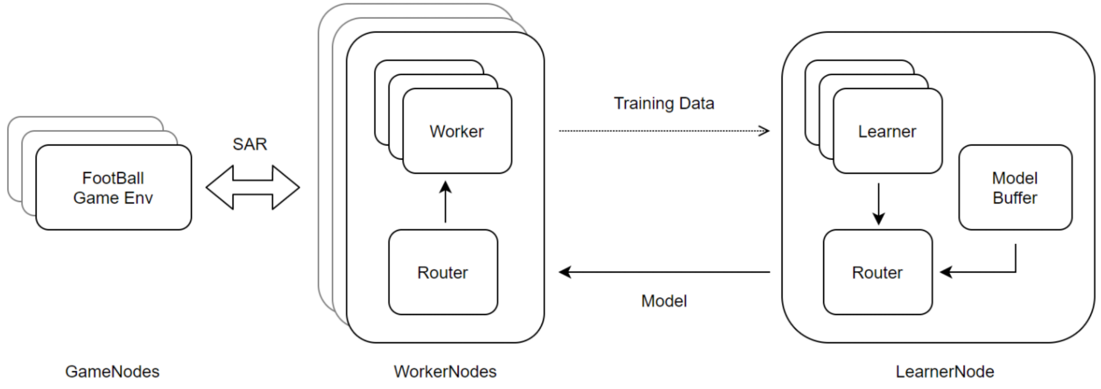


Network Architecture

- A few dense layers and a LSTM block
- Fixed learning rate and Adam optimizer
- CNN is abandoned for the high memory consumption
- Multi-head value (MHV) is used for different skills such as offsides and slides

Solutions for the Competition: 2nd SaltyFish

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

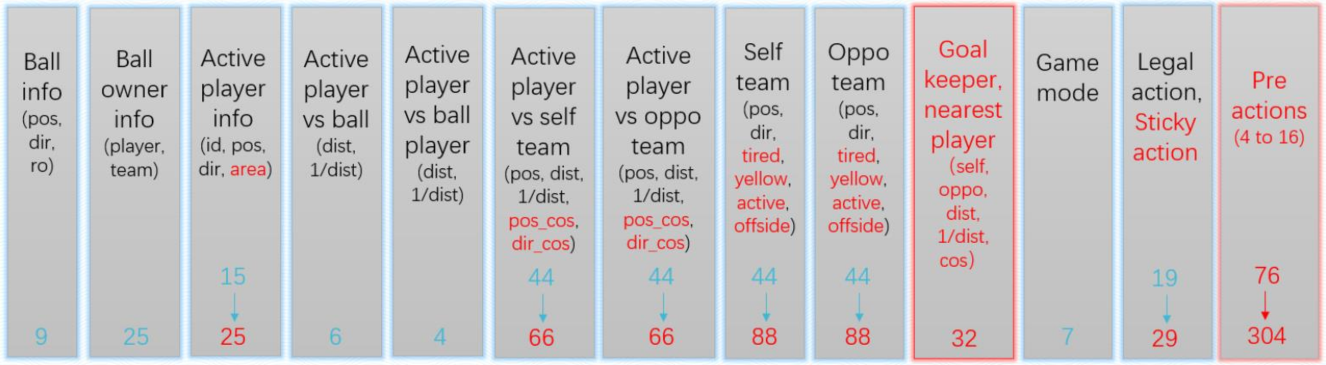


The key techniques in SaltyFish (based on distributed IMPALA training)

- Self-play: Maintain an opponent with the latest or Elo-based (similar)
- Behavior Cloning

Feature Engineering (simple115 based)

- Group related features into one head and make multi-head vector
- Modify the action set by specify the available actions for accelerating learning by avoiding unnecessary actions
- Without legal action mask makes it possible to learn the skill

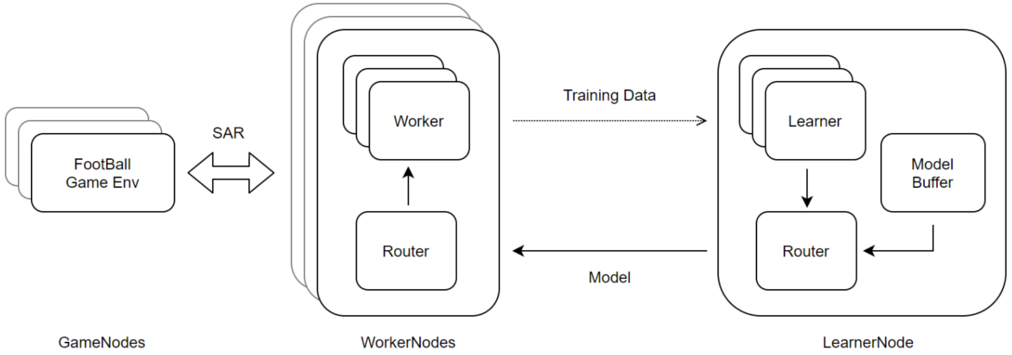


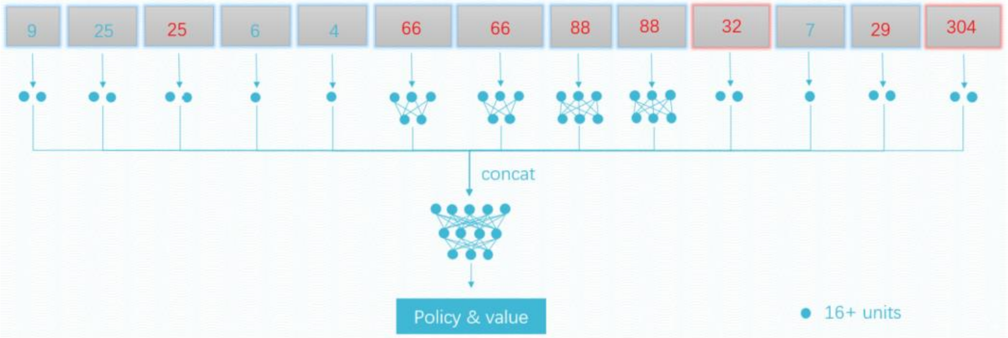
Solutions for the Competition: 2nd SaltyFish

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

General reward shaping because of too sparse score

- Scoring reward is used in self-play training for learning more sophisticated behaviors
- ball possession/ball passing brought nonpromising results





Network Architecture

- A few dense layers for the dimension transformation

Other methods they tried

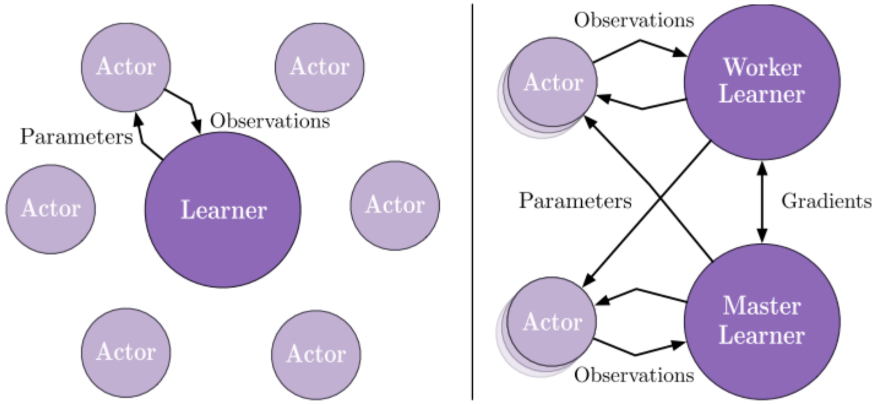
- ELO-based Opponent Selection: similar results with selecting the newest opponent with a certain ratio (70%)
- Curriculum learning and Imitation Learning from WeKick
- GNN-based network for extracting features

Solutions for the Competition: 3th Raw Beast

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

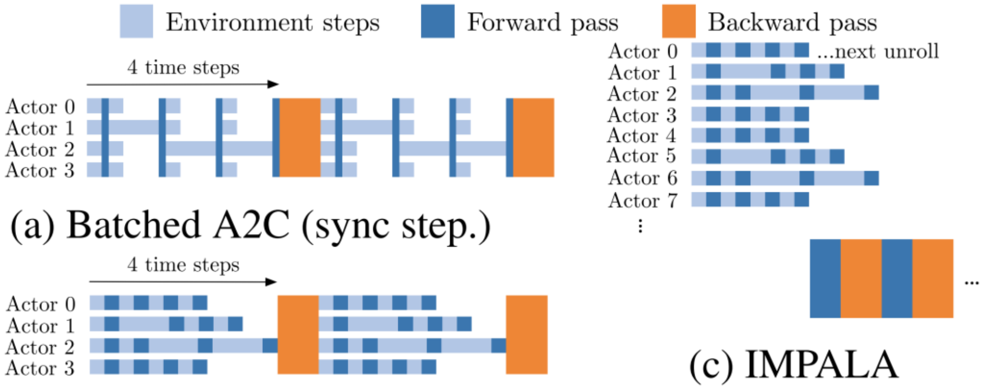
The key techniques in WeKick (based on distributed PPO training)

- Self-play: Maintain an opponent pool based on the LB and add diversity
- Behavior Cloning
- KL term for stabilizing learning and help to add diversity to agents pool



Feature Engineering (simple115 based)

- An encoding of the time since the last pass, shot and sliding.
- Features related to the opponent controlled player – only provided to the value network
- Offside flag to mark potential offside teammates



(a) Batched A2C (sync step.)

(b) Batched A2C (sync traj.)

(c) IMPALA

Solutions for the Competition: 3th Raw Beast

Introduction of the rule of the football competition: Each team only control a single player rather the whole team with 11 players. Other 10 player is the build-in players

General reward shaping because of too sparse score

- Handcrafted rewards for 4 game modes: distant from goal/close to goal attack/defense (unsuitable)
- A single mode reward for scoring a direct goal mapping the real objective

Network Architecture

- A convolutions-based architecture as a balance between speed and size of the model
- Dense layers, 1-D and 2-D convolutions across players
- A deeper network works more efficiently than wider
- TD-lambda value loss, IMPALA policy loss, UPGO policy loss, Entropy bonus, KL loss like in AlphaStar

- "Active player block" with the features from the active player's point of view: location, direction of movement and sticky actions.
- "Opponents block" with the 1-D convolution over 11 opponents using their positions and velocities relative to the active player as features.
- "Teammates block" with the same structure as for the opponents.
- "Pairs block" with a 10×11 tensor containing location and velocity of each opponent relative to each teammate and a 2-D convolution over it with average pooling per teammate.
- "Main opponent" with the features of the opponent closest to the ball using the same features.
- "Game data" with the indicator of the team in possession, ball location and speed, game mode.
- "Historical features" like time since last passing or shooting action
- "Active opponent data" for value function only

Thanks

中国科学院
自动化研究所
INSTITUTE OF AUTOMATION
CHINESE ACADEMY OF SCIENCES