CCPPO and QMIX RL algorithms on Google Research Football Environment

Edmund Shieh

***Abstract*—In this paper, we describe an implementation of two RL algorithms Centralized Critic Proximal Policy Optimization (CCPPO) and QMIX, in an attempt to beat three provided baseline agents in the google football 3v3 environment. We also collect 3 metrics to demonstrate coordination in a multi agent setting**

I. 3v3 Google Research Football Environment

In this project, agents will be trained to play a 3v3 version of football, with each team consisting of two field players and a goalkeeper. The aim is to train a team to maximize their chances of scoring and minimize their chances of conceding goals against an opposing team. The game is stochastic and terminates when a goal is scored, the ball goes out of bounds, or the maximum time-step limit of 500 is reached.

The state representation in the 3v3 football environment is a 43-dimensional vector, which includes the coordinates and direction of each team's players, the ball's position and direction, who possesses the ball, the active player, and the game mode. The action space consists of 19 discrete actions related to player movement, ball handling, and strategy.

The reward function primarily centers around scoring goals, with additional reward shaping for player's crossing distance checkpoints towards the opponent's goal. The maximum reward possible for each player is +2, +1 for scoring a goal and +1 for crossing all distance checkpoints.

II. Reward Shaping

As part of the iterative process of improving agent performance, reward shaping was undertaken to encourage certain behaviours. This was implemented A custom reward wrapped was used with the following:

- **Time Penalty:** At each time step, a penalty of 0.01 is subtracted from the reward to encourage faster completion of the game.
- **Ball Possession Reward/Penalty:** If the current ball team is 0 (left team), a reward of 0.01 is added. If the current ball team is 1 (right team), a penalty of 0.01 is subtracted.
- **Change of Possession Reward/Penalty:** If the ball changes possession from one team to another, a reward/penalty of 0.05 is added/subtracted in the event of a change of possession.
- **Successful Pass Reward:** If the ball is passed successfully within the same team (the ball owner changes but the team doesn't), and the ball has advanced more than 10m past the furthest progress point within the same possession then a reward of 0.1 is given. This is to incentivize forward passes that lead to advancement in position.
- **Shot on Goal Reward/Penalty:** If a shot action (action 12) is taken and the ball is near the opponent's goal (x $>$ 0.8 for the left team and x $<$ -0.8 for the right team, with y within 0.1 of the center), a reward of 0.1 is added for the left team and subtracted for the right team.

III. Centralized Training with Decentralized Execution (CDTE)

The provided baseline agents trained in the multi-agent environment described above are trained using the Proximal Policy Optimization (PPO) algorithm on each agent's policies independently. The main limitations to this approach are:

1) **Non-Stationarity:** In multi-agent settings, each agent perceives the environment as non-stationary. This is due to the concurrent learning and policy updates of the other agents, thereby violating the assumption of stationarity, which is integral to most RL algorithms, including PPO.
2) **Credit Assignment Problem:** PPO, like other value-based methods, struggles with the credit assignment problem in multi-agent settings. Determining the contribution of each agent's actions to a given outcome can be particularly challenging, especially in scenarios where rewards are delayed or where agents' contributions are highly intertwined.

To combat these pitfalls, CDTE is proposed in [2]. The main concept proposed is a principle where training is centralized, meaning each agent's critic is augmented with extra information about the policies of other agents. This provides a global view of the environment during the learning process, thereby enabling the agents to make decisions that are aware of the others. However, during execution, the policy used by each agent to select actions is only based on its own local observations, thereby maintaining the decentralization at the execution phase. This scheme is particularly useful when there are constraints on communication or privacy in real-world scenarios.

One of the algorithms used in this project applies the principle of CDTE through a multi-agent variant of the actor-critic method called Centralized Critic PPO. In the actor-critic architecture, the actor is responsible for determining the optimal policy (action), and the critic is responsible for evaluating the action taken by the actor. During training, the centralized critic uses the joint action-state information from all the actors to evaluate the state-action values. This additional information from the global state helps to address some of

the difficulties that traditional methods face when dealing with multi-agent environments, such as non-stationarity and complex environment dynamics.

The other algorithm (QMIX) used in this project also employs the principle of CDTE and is summarized below.

IV. Algorithms

A. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning

QMIX aims to address the challenge of multi-agent environments where each agent has a partial observation of the total environment state. To do this, it employs a centralized training procedure where the total state is accessible during the learning phase, but only allows each agent to use its own action-value function during execution [1].

The QMIX algorithm starts with each agent i having its own local observation o_i and chosen action a_i. It uses these to compute its action-value function $Q_i(o_i, a_i; \theta_i)$. The function reflects the expected discounted rewards of agent i given its observation and action. In QMIX, this function is typically modeled using neural networks with parameters θ_i. This network is trained to minimize the temporal difference (TD) error, which measures the difference between the predicted and actual value.

The goal of QMIX is to approximate the joint action-value function $Q_{\text{tot}}(s, \mathbf{a})$, which represents the expected return given the global state s and the actions of all agents $\mathbf{a} = [a_1, ..., a_n]$. QMIX uses a mixing network to represent this function. The mixing network takes the values of each agent and combines them into a single scalar value that estimates the total value of all agents as shown in 1.

$$Q_{\text{tot}}(s, \mathbf{a}) = f(\mathbf{w}(s) \odot \mathbf{Q}(s, \mathbf{a}), s) \tag{1}$$

where s is the global state, $\mathbf{a}$ is the action profile, $\odot$ is the element-wise product, and $\mathbf{w}(s)$ is a state-dependent weight vector output by a network called the mixing network [1].

To ensure monotonicity, the weights in the mixing network are non-negative and the function f is non-decreasing. This allows the maximum of the joint action-value function to be found by independently maximizing the marginal action-value functions of each agent.

During training, QMIX has access to the actions and observations of all agents. It uses a centralized Q-learning procedure to update the joint action-value function. This involves computing a TD error that measures the difference between the estimated and actual return given the global state and the actions of all agents. The parameters of the agent networks and the mixing network are updated to minimize this TD error. The use of a centralized training procedure allows QMIX to handle scenarios where the reward depends on the actions of all agents.

Even though QMIX uses a centralized training procedure, it can still be executed in a decentralized manner. Each agent chooses its action based only on its local observation and its own Q-function. This means that the actions of the other agents do not need to be known during execution, which makes QMIX applicable to a wide range of multi-agent scenarios.

B. PPO with Centralized Critic (CCPPO)

Proximal Policy Optimization (PPO) is a type of Reinforcement Learning algorithm developed by OpenAI. It is an on-policy method that optimizes a surrogate objective function to ensure the new policy does not deviate significantly from the old policy.

The PPO algorithm alternates between sampling data through interaction with the environment and optimizing the surrogate objective using this sampled data. The objective function for PPO is given by:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{2}$$

where $r_t(\theta)$ is the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, $\hat{A}_t$ is the advantage function estimate at time t, and ϵ is a hyperparameter that controls the degree to which the policy can change in a single update.

In a multi-agent setup, a centralized critic can be used to estimate the value function for all agents. Each agent has its own policy, but the value function used to update these policies is estimated by the centralized critic. The value function $V^\pi(s)$ is updated using the following equation:

$$V^\pi(s) = \mathbb{E}_{a \sim \pi, s' \sim P} \left[r(s, a) + \gamma V^\pi(s') \right] \tag{3}$$

where s' is the next state, a is the action taken under policy π, $r(s, a)$ is the reward, and γ is the discount factor.

This approach combines the benefits of PPO with the benefits of a centralized critic, making it a powerful method for training multi-agent systems.

V. Cooperation metrics

As a result of using these algorithms that incorporate CDTE, the hypothesis is that it allows agents to learn to coordinate their behaviors during training, while still being able to act independently during execution. To measure this hypothesis several metrics have been created to track the level of cooperation in the game.

1) **Spread:** teammate-spread-out which tracks the percentage of timesteps where the teammates are spread at least 5m apart described in [3] is used to measure coordination. Due to the fact that this is a 3v3 game rather than 11v11 the 5m threshold is no longer appropriate as it is much easier for the agents in the 3v3 game to achieve this given there are only 2 players outside of the goalie on the entire field. Indeed, when the original 5m threshold was used, almost always the 2 players on the field (excluding goalie) surpassed this distance. After some testing, 20m was an appropriate threshold to use for this metric.
2) **Passing proportions:** Passing is a key cooperative action in football necessary for team success. The hypothesis is that agents will learn how to use passing

advantageously to demonstrate coordination which will be observed in the proportion of time passing is chosen as an action. Short, long and high pass are all observed as part of this metric.

3) **Same sticky actions:** The final metric tracks the proportion of time agents are performing the same sticky action. An increase or decrease in this could indicate coordination as a centralized critic is used to evaluate the joint action-value function, which takes into account the actions of all agents.

VI. Implementation

The experiment commences by training a reinforcement learning (RL) agent to play a simplified 3v3 version of the Google Research Football game using Ray's RLLib library and QMIX and CCPPO as the algorithms.

Population Based Training (PBT) is employed to find optimal hyperparameters and train the model. PBT is a method introduced by DeepMind [4], and it's somewhat like a combination of genetic algorithms and hyperparameter search. It keeps a population of models with different hyperparameters, and periodically, poorly performing models are replaced with "offspring" of better performing models, which have slightly mutated hyperparameters. The main advantages of PBT are that it is computationally efficient which was necessary for this project given the limited time and budget, as it does not require a separate validation phase unlike most hyperparameter tuning methods, and it can adapt the hyperparameters throughout the training process. In particular PBT was used to optimize the learning rate (lr) and discount factor (γ) of QMIX and various hyperparameters of the PPO algorithm [5] such as the learning rate (lr), discount factor (γ), λ, kl_coeff etc. The 'perturbation_interval' was set at $50,000$ which is the number of training steps before the population is evaluated and the parent/child swapping and perturbation is performed.

Using the above configurations, training occurred for $50m$ timesteps aligning with the timesteps used to train baseline models to allow for a fair comparison. Checkpointing was also conducted every 100 iterations and at the end of the training. The training trial with the highest mean reward across the episodes is considered the best, and its checkpoint is saved and used for evaluation.

Finally, the agents trained using the two algorithms described above (CCPPO and QMIX) were put up against the 3 baseline agents provided and evaluated on 100 episodes of the 3v3 football environment using the policies learned during training with results shown below in figure 2. In this evaluation, the learned policies are evaluated and the action leading to highest value is executed at each timestep as it is in exploitation mode.

VII. Results

A. Win rate

The CCPPO agent was able to outperform the top baseline whereas the QMIX agent was only able to beat 1 baseline agent as shown in figure 2. The success of the CCPPO agent was due to implementing a centralized critic allowing for increased coordination amongst the players. This can be demonstrated by the accelerated learning of CCPPO in scoring goals as shown in figure 1 and coordination metrics in figures 3, 4, 5 and 6. The results of QMIX could be in part be because of a lack of hyperparameter tuning given the limited compute available. It seems that this led to not enough exploration for the QMIX agent as the players were unwilling to take shots over time.

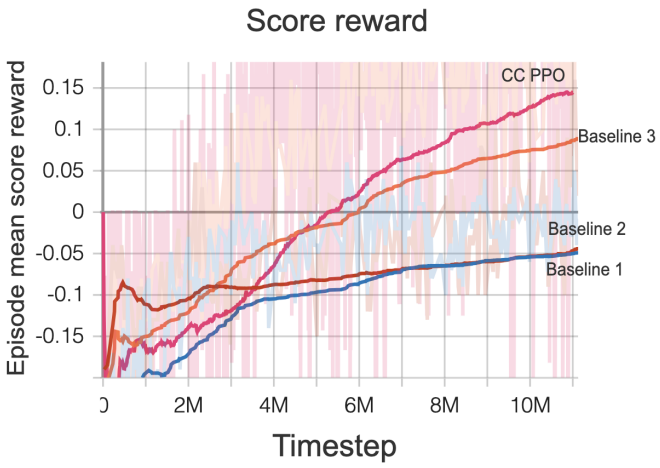

Fig. 1. Episode mean of rewards from scores (+/-1) comparison between Centralized Critic PPO algorithm and 3 baselines over first 10m timesteps

Furthermore it was also observed when visualizing the winning episodes for even the strongest baseline that "hogging the ball" proved to be an effective tactic against the opponent where scores often resulted from a single player taking the ball the entire way and shooting. Given the lack of coordination involved in this successful strategy and in the absence of rewards given for successful forward passes, players are incentivized to take the ball themselves across opponent lines.

The network architecture to model the agent network in QMIX was also different to the one used by the value function in CCPPO as it used a single LSTM layer passed through a dense layer as opposed to a deeper neural network with two fully hidden layers with a tanh activation function followed by the same LSTM layer. The deeper architecture may also have helped the CCPPO in learning more complex relationships from the observations.

B. Coordination metrics

The cooperation metrics were taken from the training episodes of the CCPPO agent.

1) Spread: The proportion of timesteps where the distance between the two controlled players were more than 20m continually increased over the training indicating that the players strategically decided to be more spread out over the field. After inspecting the evolution of the episodes over time, the increased distance demonstrates a level of cooperation between the two players more than if they acted independently. By spreading out, players can cover more ground and control

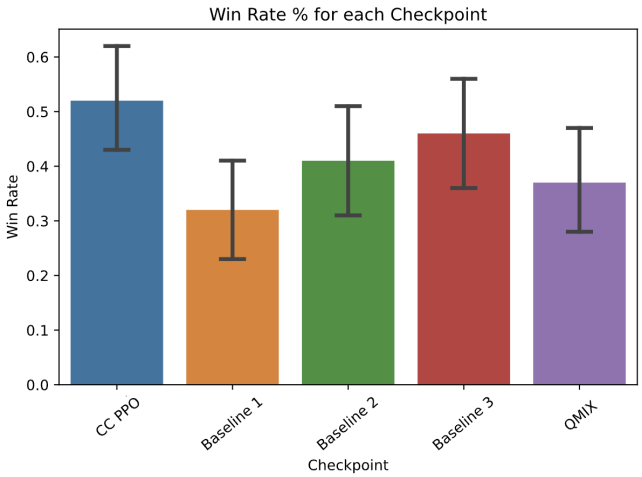

Fig. 2. Win rate comparison between trained Centralized Critic PPO, QMIX and 3 baseline agents evaluated on 100 episodes using learned policies

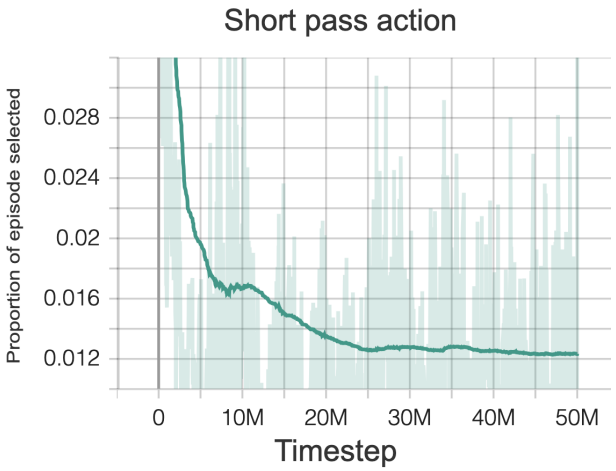

Fig. 4. Episode mean of proportion of timesteps for short pass action selected Centralized Critic PPO over 50m timesteps

more of the field. This can help in both offensive and defensive situations. On offense, spreading out can force the opposing team to also spread their defense, potentially creating gaps that can be exploited. On defense, spreading out can help cover more potential threats. If players are spread out, they can potentially create more passing options. A player with the ball can pass to several different teammates in different parts of the field, making it harder for the opposing team to predict and intercept passes.

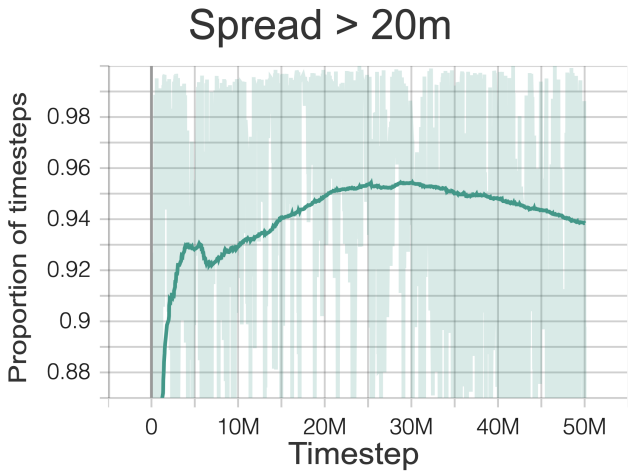

Fig. 3. Episode mean of proportion of timesteps where spread between two players were ¿ 20m using Centralized Critic PPO over 50m timesteps

2) Passing Proportions: The proportion of timesteps spent long passing increased between the two players while short passes decreased shown in figure 4 and figure 5. The centralized critic has access to the full state information and the actions of all agents, which allows it to learn a more global strategy. This could potentially lead to more coordinated play and an increase in long passes/decrease in short passes as the critic learns that this is more effective to maximise rewards.

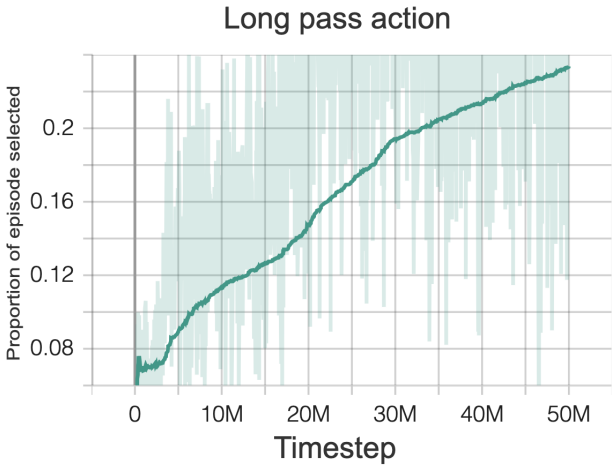

Fig. 5. Episode mean of proportion of timesteps for long pass action selected Centralized Critic PPO over 50m timesteps

3) Same Sticky Actions: Figure 6 shows the decrease in the same sticky action between players over the training. This can indicate a developing of coordination during training. This was validated by looking visualizing the episodes at different times during training. Early on in training, there were more cases where two players performed the same action at the same time e.g. they might be trying to achieve the same goal (like going for the ball or moving to the same spot on the field). This could be a waste of resources and energy, as one player might be able to achieve the goal while the other player could be doing something else useful elsewhere. If the two players' actions conflict with each other (like both trying to take possession of the ball), it could disrupt the flow of the game and potentially lead to negative outcomes (like losing possession of the ball).

VIII. Challenges and Pitfalls

The main challenge was the lack of compute and time available to test various algorithms, hyperparameters, network

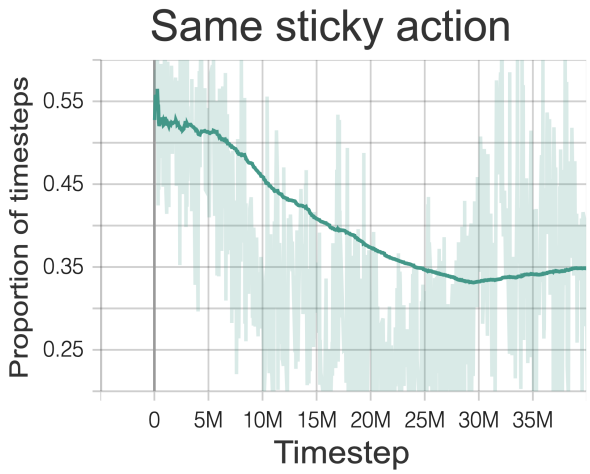

Fig. 6. Episode mean of proportion of same sticky action selected by both players for Centralized Critic PPO over 50m timesteps

architectures and reward shapings.

In particular, PPO, like other policy gradient methods, can be sample-inefficient. This means it may require a large number of episodes to learn an effective policy, which can be computationally expensive and time-consuming in complex environments.

For example, the 50m timesteps for conducting a comparison with baselines would take multiple days to complete with the 16 core CPU google cloud compute option. Realistically without blowing out a reasonable budget, only several iterations and combinations of the above could be attempted.

Furthermore, PPO, like other gradient-based methods, can get stuck in local optima. This means that it might find a policy that is good, but not the best possible policy. This was observed in the training of the baselines as well as the experimented algorithms where the win rate plateaued at around 50%.

For QMIX, one of the pitfalls was that the algorithm imposes a monotonicity constraint on the mixing network to ensure that the joint action-value function is representable as a non-negative linear combination of the individual agents' action-value functions. This constraint can limit the expressiveness of the QMIX architecture and may not be suitable for all environments, particularly those with complex inter-agent interactions.

Another challenge was thinking of rewards and coming up with constraints for those rewards such that agents did exploit the short term rewards in replacement for winning the game.

REFERENCES

[1] Rashid, T., Samvelyan, M., Christianos, F., et al. (2018). QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *CoRR, abs/1803.11485*. Retrieved from http://arxiv.org/abs/1803.11485.

[2] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*, arXiv preprint arXiv:1706.02275, 2017.

[3] Liu, Siqi and Pal, Aditya and Jiang, Yingfeng and Littman, Michael and Isola, Phillip and Roy, Brandon, *Emergent coordination through competition*, arXiv preprint arXiv:1902.07151, 2019.

[4] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K. (2017). Population Based Training of Neural Networks. *arXiv preprint arXiv:1711.09846*. DOI: 10.48550/arXiv.1711.09846 URL: https://arxiv.org/abs/1711.09846

[5] Ray Project, *RLlib Algorithms - PPO*, Ray Project Documentation, https://docs.ray.io/en/latest/rllib/rllib-algorithms.html#ppo