**Status Update – Week 4**

**Clarification of the Research Question:**

Background – Probably the most common training setup for MARL is the CTDE (Centralized Training Decentralized Execution) paradigm, where the agents are trained in a centralized manner, but executed independently. The main method for Centralized Training is deriving the individual utility functions from the **global reward** (which for now I assume is a sum of local rewards) in order to **enforce cooperation between the agents, and create a common goal for them.** This poses a problem in training speed and a seriuous limitation on scalability regarding #agents – since credit assignement from the global reward is often unclear.

My Research Question is as follows: Under what assumptions can we use a combination of global and local rewards in order to speed up training whilst ensuring cooperation? Specifically, can this be done under locality assumptions?

For example, in the multi cart pole scenario, it is obvious that the n’th cartpole has no effect on the first cartpole, and therefore it should not learn from it. This is clear from observation of the local rewards.

**Some Mathematics:**

Let’s assume that we are dealing with an MARL with local rewards per agent that sum up to a global reward.

In this case, the Q function can be split into n partial Q functions that represent the Q-value assuming that the only reward in the environment is

Note that both the partial Q functions and the global are theoretically dependent on the global state and global action. Under sufficient locality assumptions, it has been shown that the partial Q functions depend only on a “small” neighborhood surrounding agent i. This is important because it makes each fairly easy to compute – we just need to consider the actions and state of a small neighborhood. In the multi cartpole scenario, this basically means learning how to balance just the i’th cartpole (fairly easy), using a few cartpoles on either side.

What is the loss for regular centralized learning?

This is fundamentally different from n different Qi function with individual losses, but similar:

Even if we add all of them together and even if we ignore the squaring, they are still fundamentally different:

Shows the difference between the 2 losses – the difference in order of summation and maximization is exactly the difference between individual gain and collective good. So is clearly the correct loss, but hard to compute, and misses the cooperation aspect but is much more scalable and easier to compute since it uses the n local rewards. My research question is is there a middle ground here?

My first thought was using as a warm start, and then fine tuning using the . In setting where cooperation is needed but not too strong this might work. Meaning we define a loss:

Where over the training period we slowly decrease gamma to 0.

I am currently looking for more theoretical and correct ways to incorporate the local rewards as part of the training process.