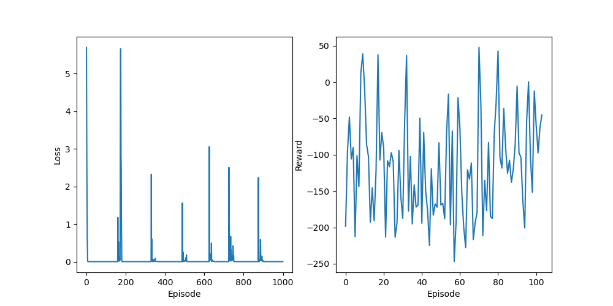
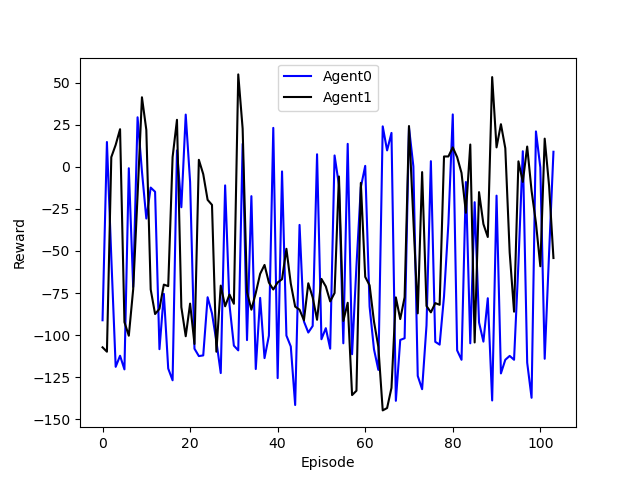
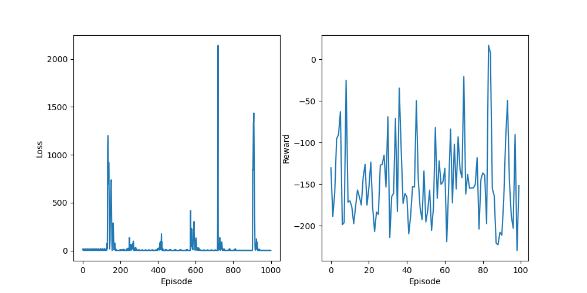
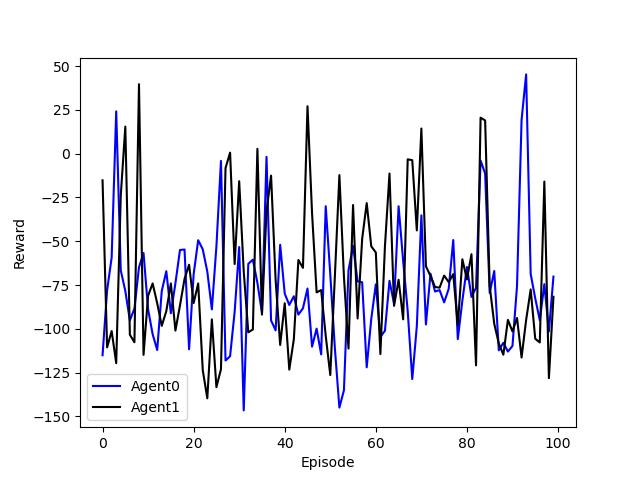
9.14 仿真结果总结

今天早上回收结果，发现两种设置（累计奖励，结尾补0）结果都有问题

结尾补零  


累计奖励



这两组都是加入了通信模块后的结果。

对结果和代码进行了分析  
代码分析：发现了一个bug, 为了节省内存空间，实验中设置仅记录最近N个episode的状态转移经验序列（报告中的所有结果均设定N=12）, 这里犯了个错误，旧经验没有被删除而是直接被加入到了原来的经验末尾，这导致了原先的序列被增长，智能体学习到了错误的被加长了的历史经验。

结果分析：为了解决“懒惰智能体”问题，我对智能体share的joint reward的计算方式进行了修改，在原先取各个智能体的平均值的基础上，加入了标准差作为惩罚项，希望能让他们的表现更加平均，然而从这两组结果上看，各个智能体之间的表现还是差很多，而且因为考虑到标准差，智能体还有向表现更差的一方靠拢的趋势，因此，目前奖励函数的设置也不合理。

9.15修改与目前结果

为了解决上述问题，首先是修改了bug, 然后将loss的展示变成了纵向平均。

**Joint reward**

对于joint reward函数的构造，目前有以下三种方案：

1. 取最大的individual reward作为 team reward， 并用标准差进行修正
2. 取最小的individual reward作为 team reward， 并用标准差进行修正
3. 取individual rewards 的平均值

**Padding**

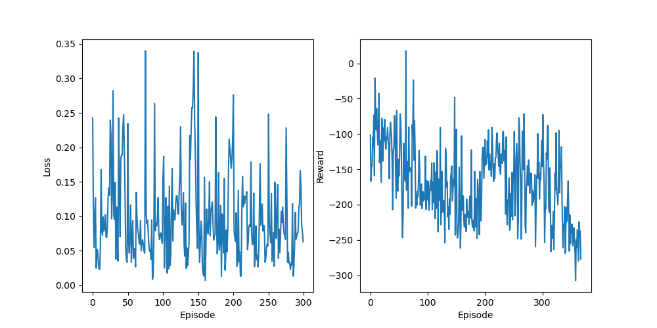
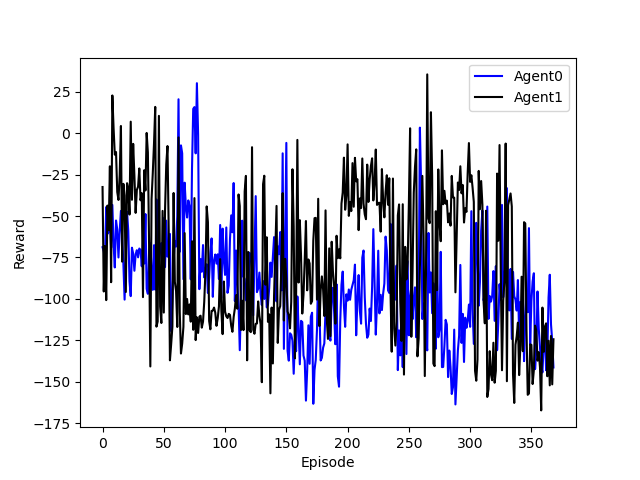
而对于QMIX网络，存在的问题是不同长度的episode在结尾进行填充的方式：

zero\_padding: 填充0到最大长度

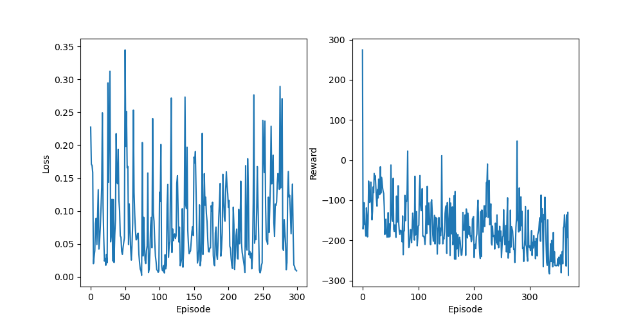
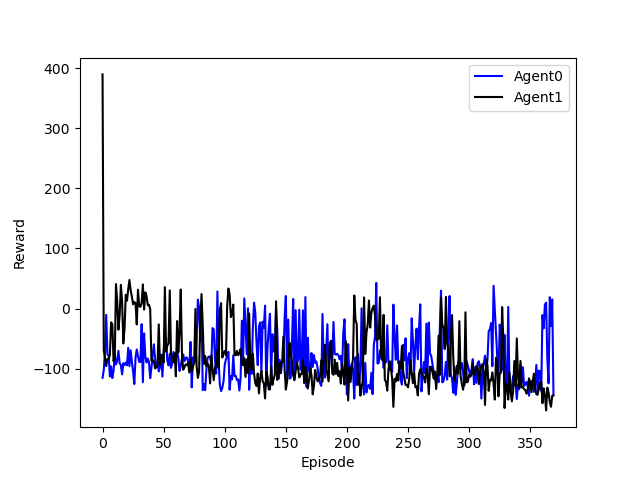
last\_padding: 填充最后一个时刻的数据到最大长度

这两种方法，之前的文章中都有人采用，zero padding 用的更多一点，但是我跑到现在的结果, last\_padding表现更好一点

Bug修改后的last padding



zero-padding



分析reward下降的原因可能是由于一直到不了任务终点，它无法去提高平均值，而转而去使两个智能体的性能相近，且向表现差的一方靠拢了。

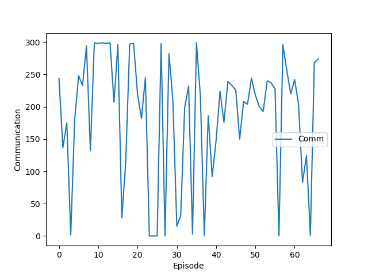
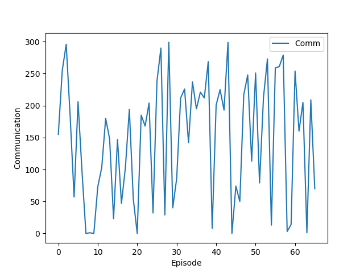
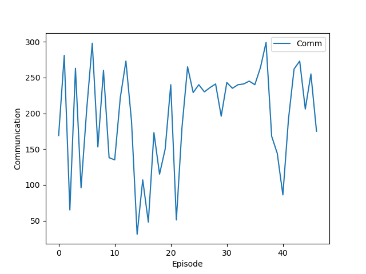
**instant or cumulative reward**

在训练过程中，还存在一个问题，对于每一个timestep，joint reward是这一时刻instant reward还是截止到现在的cumulative reward,？（之前做过的对比中instant reward会好一点，但是看到有文献中认为在轨迹控制的环境下，累计奖励可能会更好）

**episode长度**

最后是环境上的问题，现在每个episode 的 timestep=300，在目前的地图中（10x10x3）的大小下, 应该是可以完成任务的，然而在实验中，智能体几乎不能都完成任务，仅有其中一个完成任务的情况也很少达到，因此timestep可能还需要放长一点。

接下来的实验方向：昨天加入了通信模块之后，虽然任务的表现性能没有增长，但能明显地看到智能体倾向于去通信。



因此，分析通信模块加入后对于智能体的学习还是有不小的影响。

目前， **padding方式**，**joint reward function**, **episode长度**，**instant or cumulative reward**是还需要确定的问题，我想先在不考虑通信模块的情况下，先将这些方式确定下，找到最好的组合后，再加入通信模块。