时间连续的决策量：（最大总体系数，乘以归一化的功率谱密度微元然后求和）；

先不做MARL，依旧基于PPO 训练单个智能体：

假设：

1. 只做抗干扰，只有一个地面用户，不存在示假、链路和同频问题。
2. 航迹固定，用户位置固定。
3. 功率谱密度函数服从高斯分布

框架：

1. 敌方初始：扫频；

Agent躲避；save；

1. 训练敌方，load我方的模型

训练敌方的干扰agent；

敌方优化：窄带；

3、load敌方智能，训练出最终的agent；

**1.1 agent训练**

决策量：，（最大总体系数）；

**State：**当前时刻我方的功率谱密度函数 ，敌方的

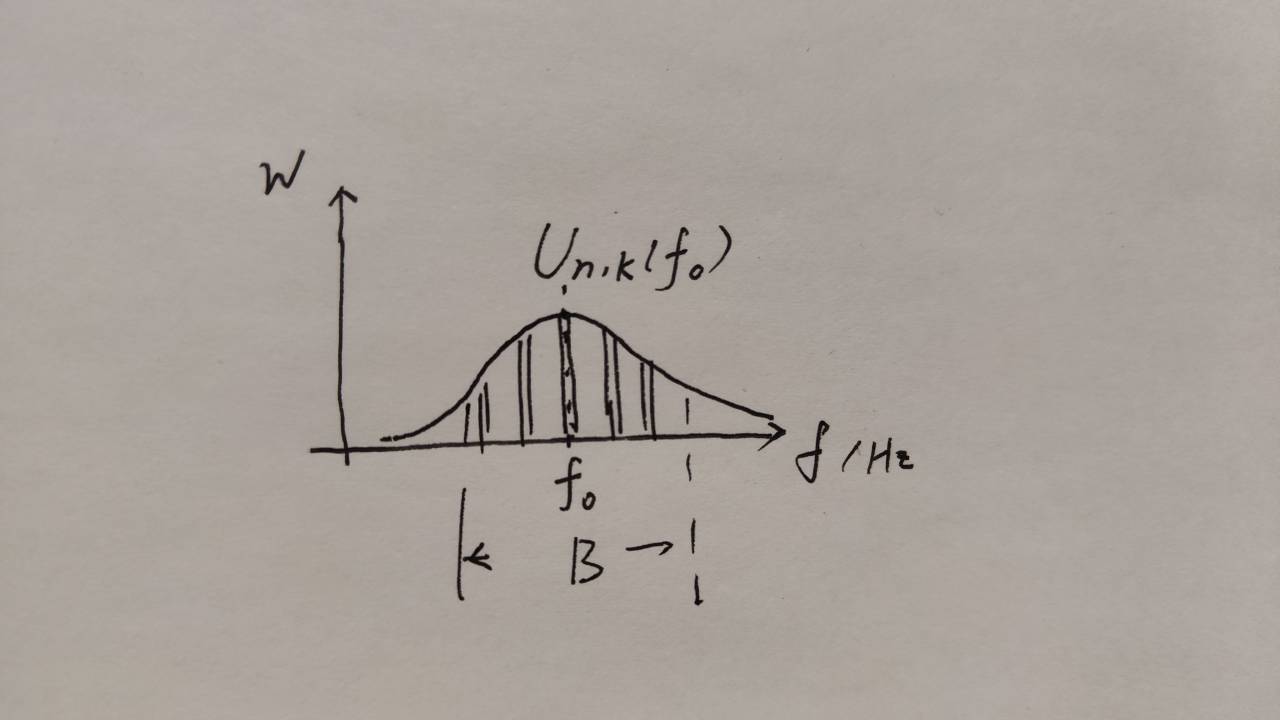
敌方的依旧需要作为输入，以用于下一轮训练敌方的对齐。即便是已知的全局变量。

——部分观测空间？

**Action：**下一时刻我方的功率谱密度函数

Tt时隙；

**Reward**：（最大化）通信链路所有时刻的总吞吐量

~~P：状态转移概率~~

设有第个时隙的功率谱密度分布：

离散功率谱定义（微元面积）：

有效功率计算（有效带宽假设）：

信道系数：无人机在*n*时隙传输数据时的信道系数，可表示为：

其中为当发射功率是1W时，在距离1m处的接收功率；为无人机在*n*时隙的三维位置坐标，为地面用户在*n*时隙的三维位置坐标。

敌方干扰功率J：（信道系数根据干扰源位置和无人机位置同理计算得出）

交叉面积；

（又不理解了。）

白噪声：PSD = 常数1

信号带宽；项目

**1.2 jamming训练**

决策量：，（最大总体系数）；

**State：**当前时刻敌方的功率谱密度函数 ，当前时刻我方的

（训练完全已知，执行的时候加上干扰/赋予观测概率）

**Action：**下一时刻敌方的功率谱密度函数 ；

下一时刻我方的功率谱密度也属于动作空间，用model.predict得到，更新state；

**Reward**：（最小化）通信链路的吞吐量

### 实现：

12.14

信道系数 =1，不考虑任何位置移动；

干扰扫频；移动中心频点，干扰带宽、功率强度不变；

这有一个问题，同时改变mu, sigma，而没有约束，就不需要赋功率强度的值了。自然就改变了强度。但是应该加约束。Random函数的问题？

移动位置，改变形状，改变高度和总面积

时刻要怎么推进。每次训练包括所有时刻吗？

干扰功率计算？

12.15

两轮训练模型观测空间的对齐。部分/完全观测？

A和S的定义，对齐问题

用函数生成代替random高斯。

面积（功率）的计算？代值得到微元高度直接求和？

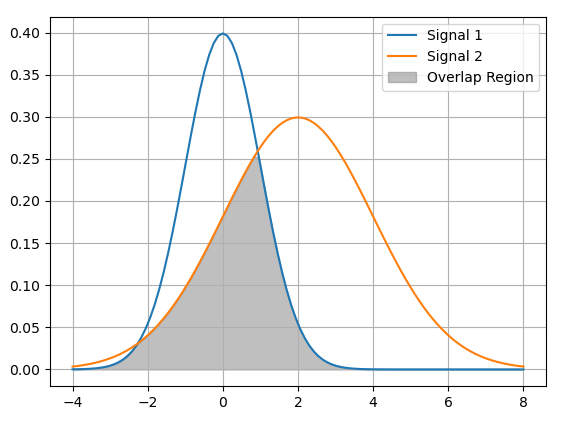
12.18

干扰求和；

时隙计数；

action没有变化；可能是位置原因。

#### 干扰功率：



import numpy as np

import matplotlib.pyplot as plt

def calGauss(mu, sigma, para, cur\_x):

    cur\_y = para \* np.exp(-(cur\_x - mu) \*\* 2 / (2 \* sigma \*\* 2)) / (np.sqrt(2 \* np.pi) \* sigma)

    return cur\_y

# 定义两个高斯信号

u1, sig1, p1 = 0, 1, 1

u2, sig2, p2 = 2, 2, 1.5

x = np.linspace(min(u1 - 3 \* sig1, u2 - 3 \* sig2), max(u1 + 3 \* sig1, u2 + 3 \* sig2), 100)  # 有效域

y1 = calGauss(u1, sig1, p1, x)

y2 = calGauss(u2, sig2, p2, x)

# 叠加两个高斯信号

y\_overlap = y1 + y2

# 找到相互重叠的区域

overlap\_region = np.minimum(y1, y2)

# 可视化两个高斯信号及其相互重叠的区域

plt.plot(x, y1, label='Signal 1')

plt.plot(x, y2, label='Signal 2')

plt.fill\_between(x, overlap\_region, color='gray', alpha=0.5, label='Overlap Region')

plt.legend()

plt.grid(True)

plt.show()

# 计算相互重叠区域的面积

overlap\_area = np.sum(overlap\_region) \* (x[1] - x[0])

print(f"相互重叠的面积：{overlap\_area}")

#### 时隙循环 ：

内循环处理六十个时隙

不写循环；记录时隙计数器tt即可，到达59就重置为0；

12.25：

训练中不处理时隙；

Jamming中不需要tt计数；

动作空间是我方的动作，state的对方信息需要在step里面合并？

——learn的时候是step，predict的时候写在外面就行。

12.31

敌方的目标函数：最小化信干比，还是最大化敌方的信干比？

计算总功率的B：总带宽还是当前带宽？

Reward与时隙的关系：step中不处理？

训练输出正常，model.load()报错。

尝试修改，或者直接换tensorboard可视化。不行，必须改好，敌方的训练需要。

ABCMeta object got multiple values for keyword argument 'use\_sde'

1.2

Load修正。Reward有变化，可视化的问题吧。明天换成tensorboard训练加量级试试。

输出的reward依旧是一个固定值。。大部分时间。。环境不够复杂，调通时隙bug了添加路径试试。

时隙参数的错误影响。

List：

调通bug；

加callback，可视化等；

添加路径，增加问题复杂度；

增加timestep完成训练；

#### 1.2 agent\_luo ：

1.2 师兄改的：

1 增加了训练可视化

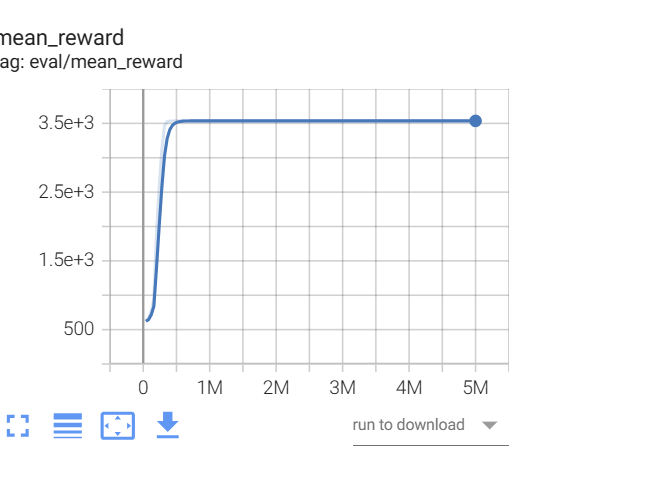
2 改成了并行

3 reset改了一下

1. reward改了一下

——感觉我是废物。我来干嘛的。喵

为什么reward现在不用取负了，神奇。我是废物喵。



收敛很快。1M就够了。毕竟是固定的敌人。明天就改jam。

为什么reward的数值能这么高。。还得单步看看。

List:

配好外星人RL环境用以训练；

单步检验；

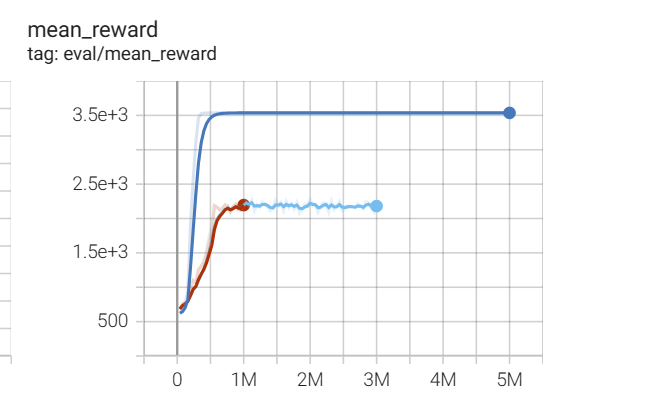
仿照师兄的代码改好jamming代码；

1.3：

改用随机敌方信息训练agent；1e6就有收敛趋势了；

15.15训练到15.35就好了。。。二十分钟

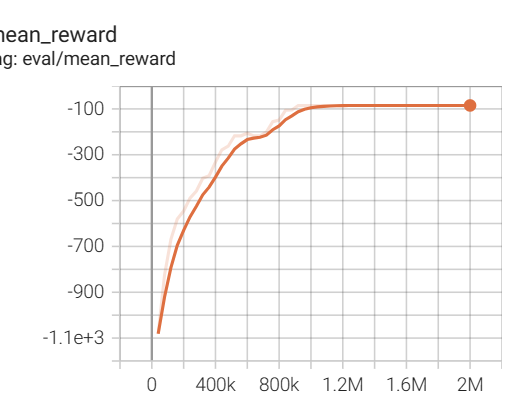
随机敌方训练的reward降低了是合理的，因为敌方强度平均值增大了，干扰强度增加，信干比降低，reward降低。



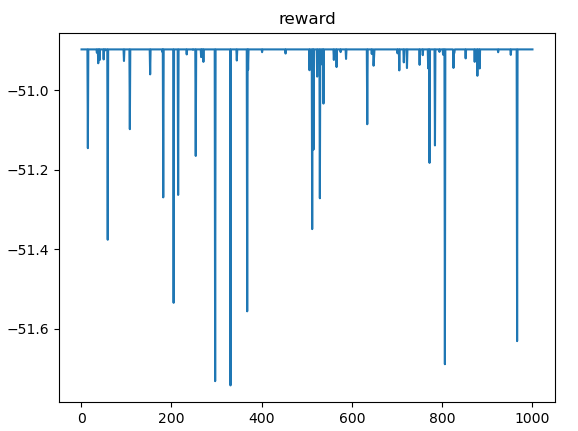
用固定敌方策略的step测试的时候结果一样。

1.4：

敌方收敛也很快：应该是问题写得太简单了



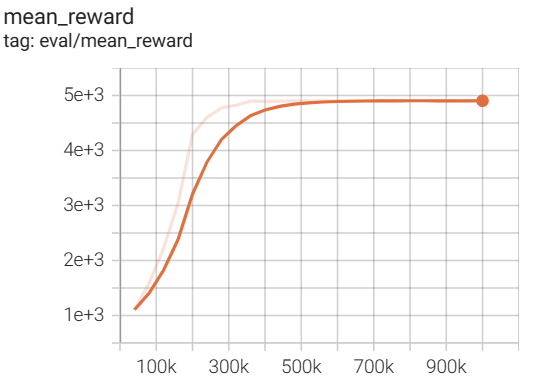
敌方测试：



改成return -reward：

第三次训练：

把敌方的模型写进我方agent的训练，1.3\_final：



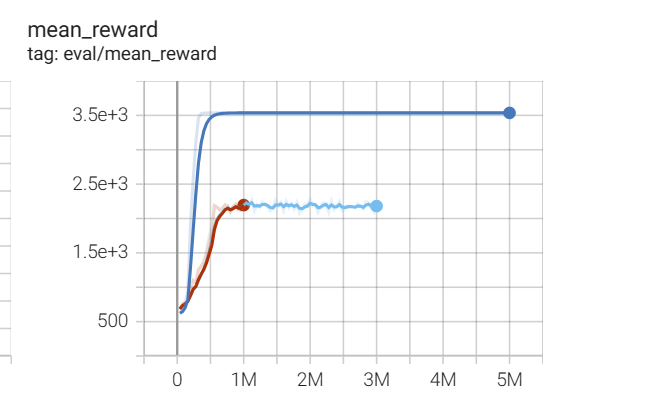
估计是问题写得太简单了。不对，敌方怎么感觉写反了，数值更高了。。

1.5 数值奇奇怪怪的。再逐步看看。

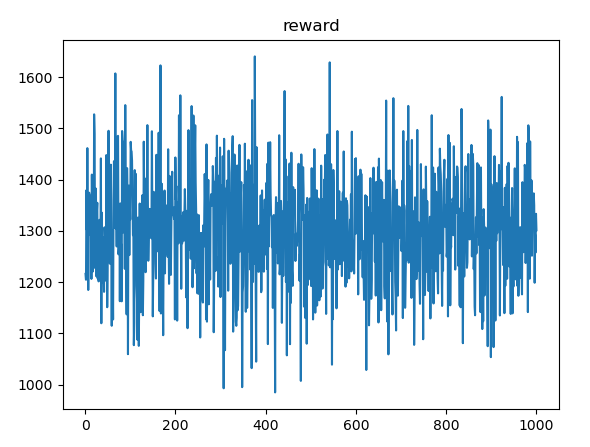
#### 结果：

1.5

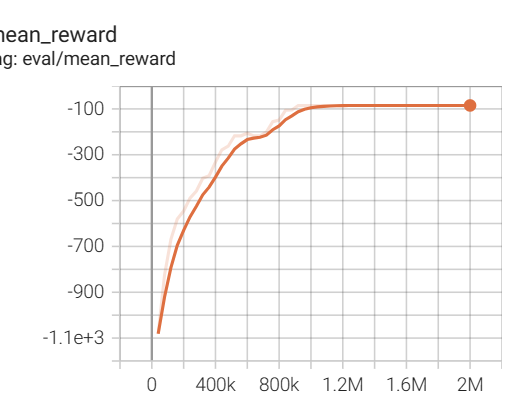
第一轮agent训练，敌方为固定扫频策略或者随机干扰策略：最大化；



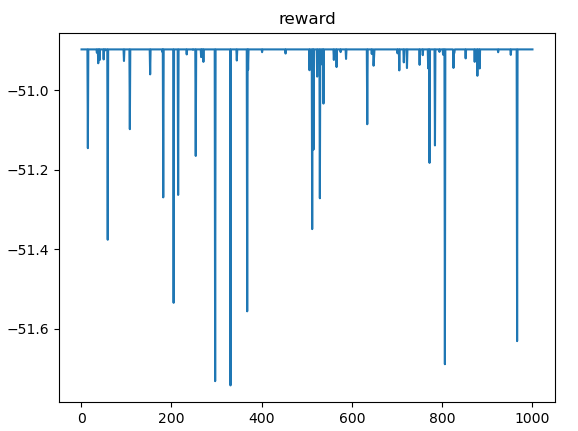
测试：敌方是随机干扰



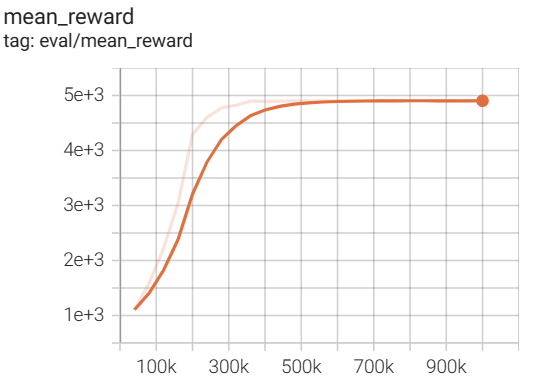
第二轮jamming训练，我方智能体是敌方随机干扰策略训练出的；最小化；



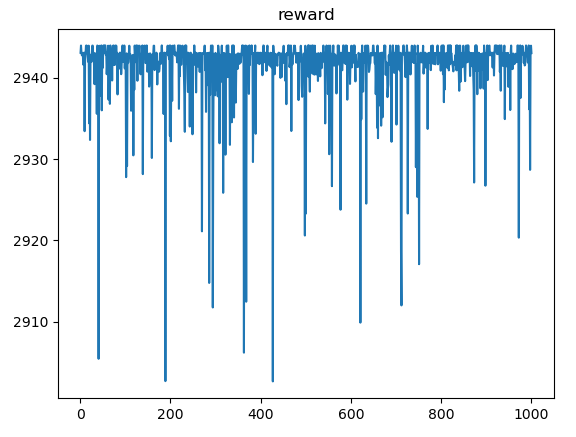
测试：数值更小，两方都是智能体



第三轮我方agent训练，敌方是智能干扰源预测的行动；（数值反而更高了）



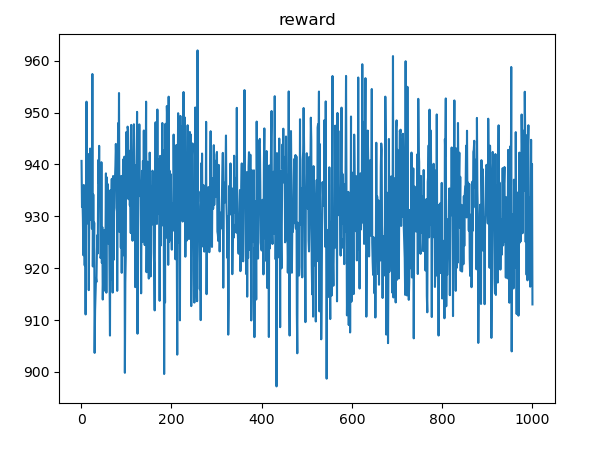
测试：两方都是智能体



每一轮的obs都是完全一样的。。应该是因为敌方信息被完全观测到了。

得改成随机的敌方，或者部分观测。

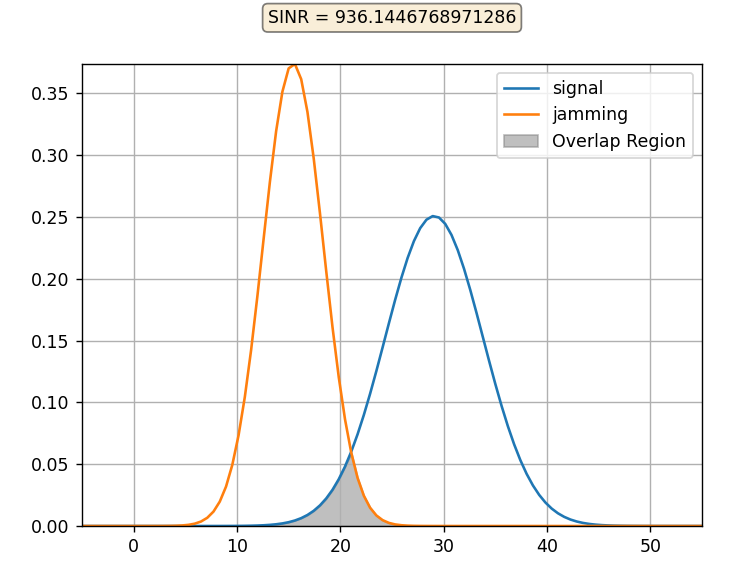
测试：敌方是随机干扰源

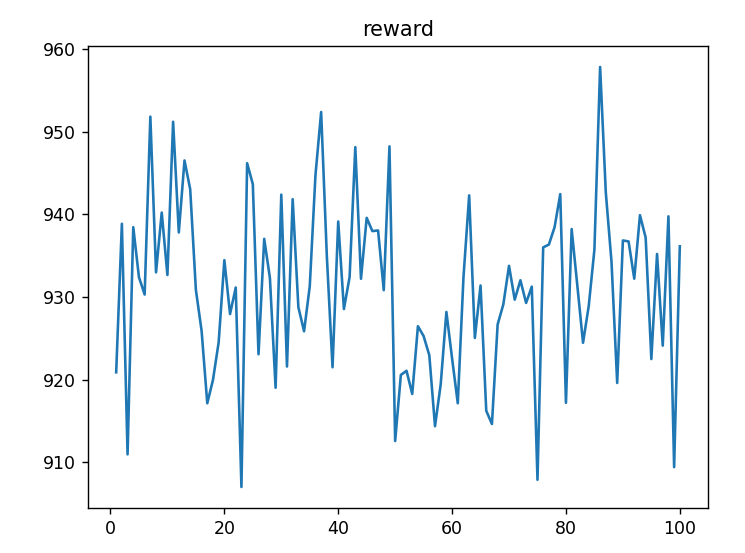


比第一轮效果还差。。不过也提升了一倍。

1.10

可视化：频点移动过于微弱





修改干扰初始频点与信号相同。图像依然如此，干扰和信号都不会移动。只会放缩。

输入敌方智能体，更是都不动了。算法还是有问题。

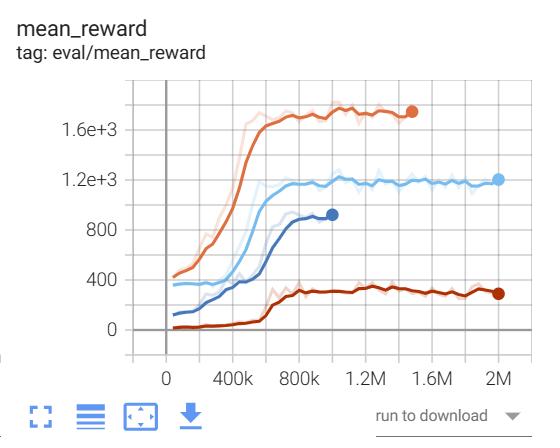
收敛快的原因就是agent直接躲在最右边了。。。。。摆了。

加强惩罚。

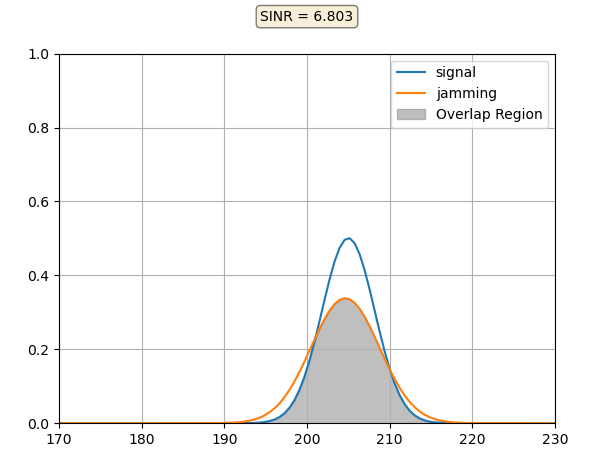
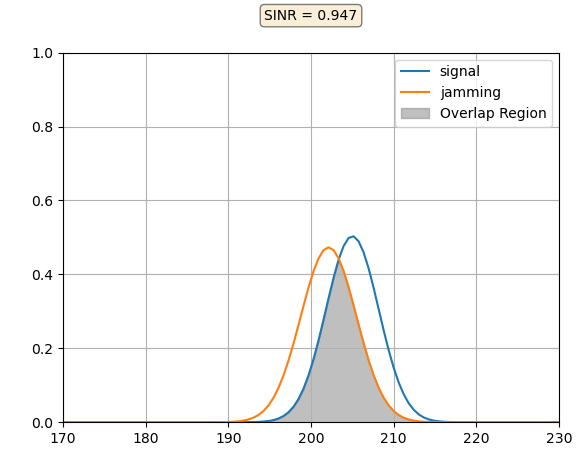
1.15：

尝试惩罚：

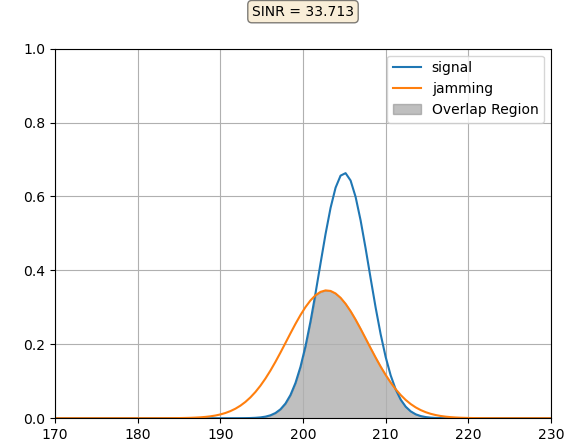
加强干扰强度是无效的。不论是加强干扰功率谱密度还是加强交叉面积代表的干扰强度。如下是PPO\_u的图片，覆写的PPO\_u\_0112.model



结果依旧是躲在右边，我还是觉得是可行的范围太少了。改改看。

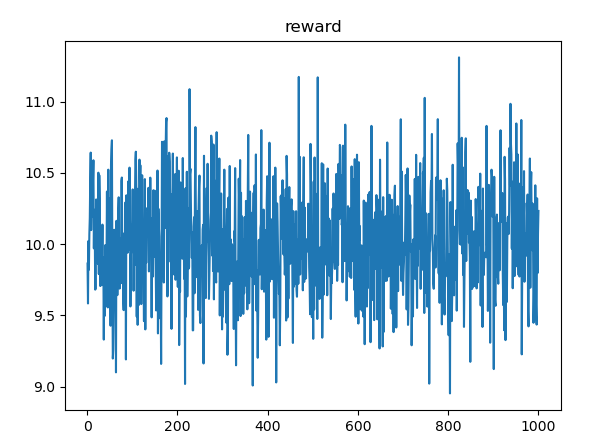


SINR看起来异常。。哦 跳变成几十的应该是积累量，以下是时隙外的绘图

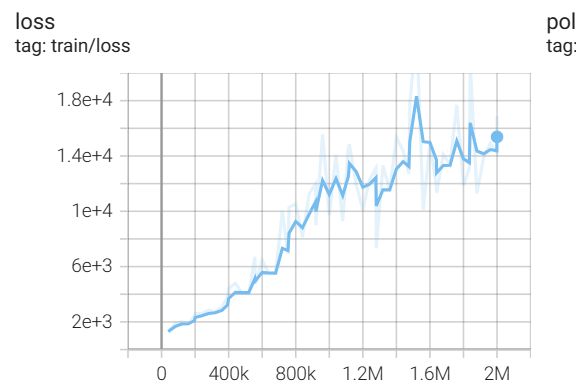


马博说他那边也不动，他打算换个算法。DDPG和SAC。。他那边reward是SINR>10作为1，也可以参考，过低的就不计数。训练的时候ep\_mean\_reward那么高，测试的时候就1.几，就离谱。。马博那边几百几千就收敛，离奇。1e6我都嫌快。。

修改了sinr的计算中带宽。用3\*sig\_u代替B。



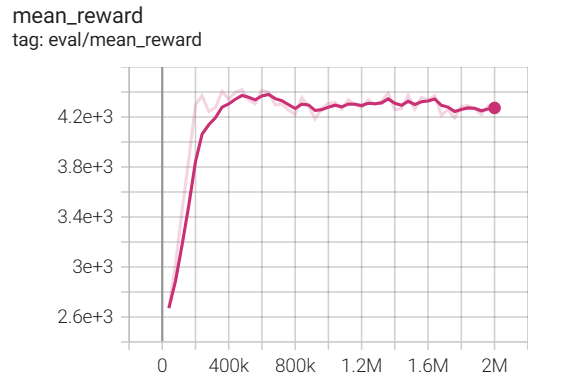
我超，这loss在涨

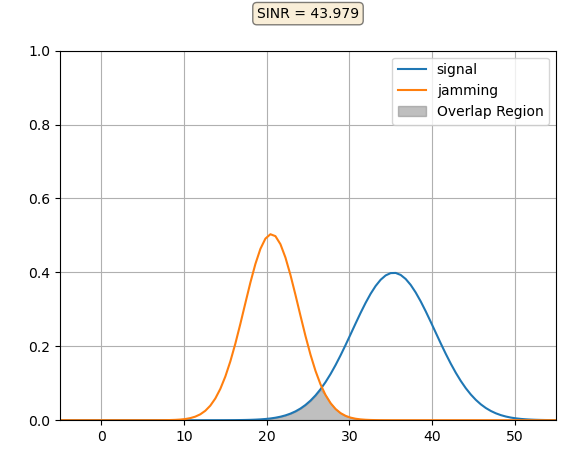


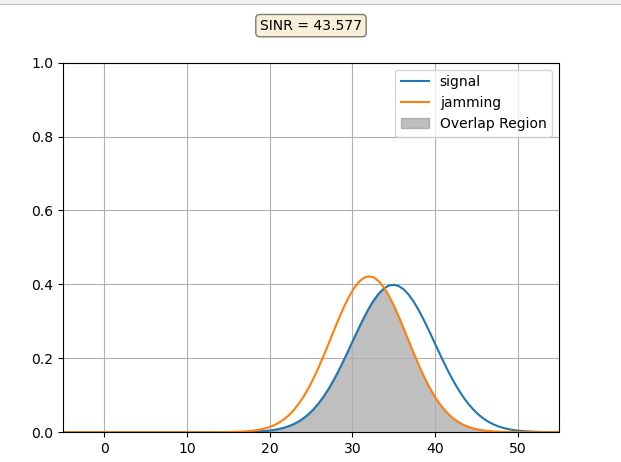
1.22

动了动了！就是之前限制了移动范围的bug！

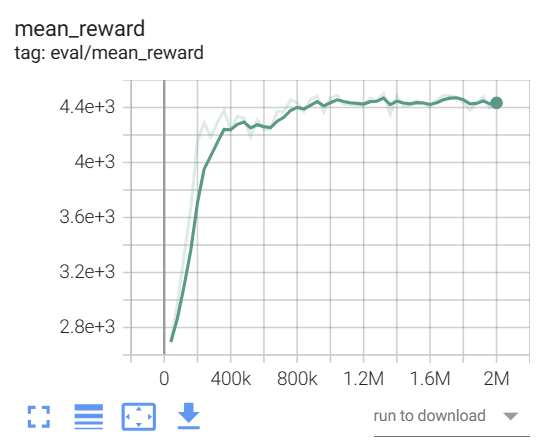
2.1现在是对的。随机干扰，频域对抗移动。







修正后，拓宽reset和归一化，2.1-agent，0112模型，PPO\_6：



波形反而不对。只能拓宽step前面的。。

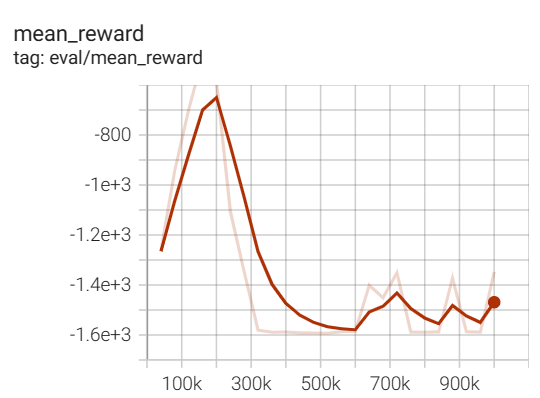
0115模型是对的。0122不对。

2.1jamming

参考agent的经验，只拓宽step开始的频域。载入0115agent。

20min.

依旧不动，收敛反了吧。PPO-3



顺着继续训2.1final试试。18.30

0123：

1、扫频测试 不正确；干扰从右往左，波形依旧停在右端。

SINR的数据显示也有问题，没有重叠的数值不如一开始重叠的高。

1. jamming做差

直接nan爆炸…

因为负数作为reward会出错。需要映射处理。

指数映射。对应模型0124，差分，指数。

还是nan。。继续调试。

是指数映射的小数据被处理成了1 。负数估计也是。梯度就是0了。

用PPO算法的**clip\_range**

**从0.4改成0.1**

### List：

扫频测试

Jamming调试；逐步调试；