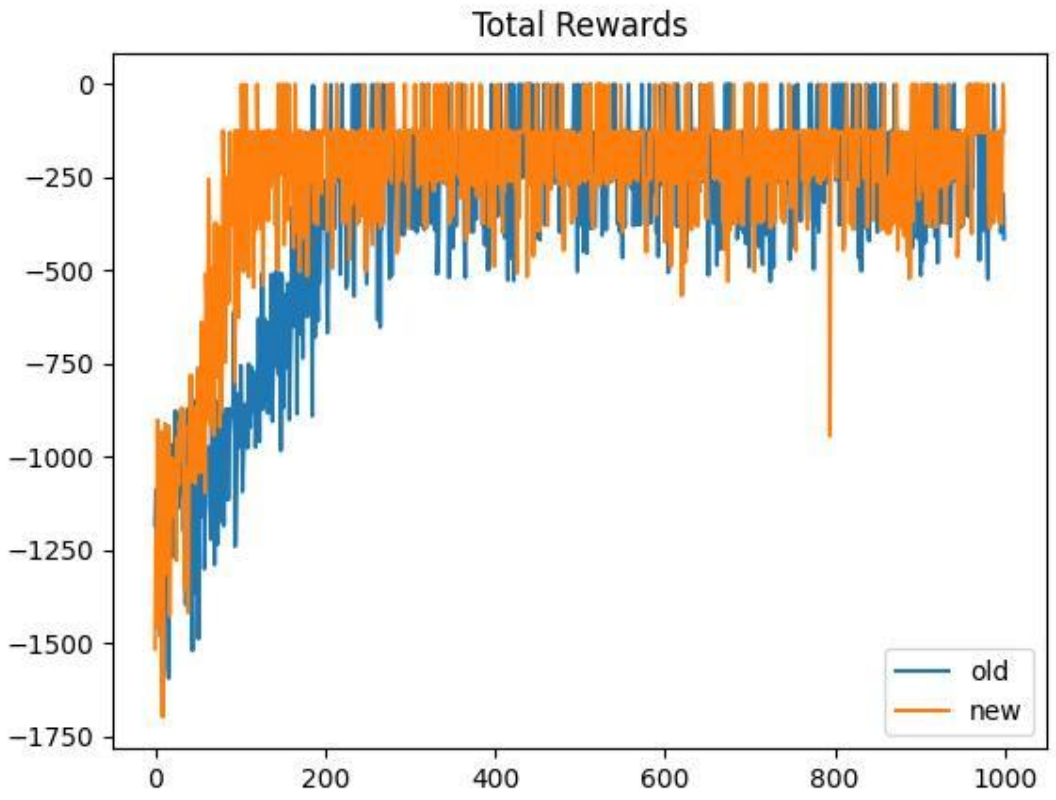
Отчет по домашнему заданию

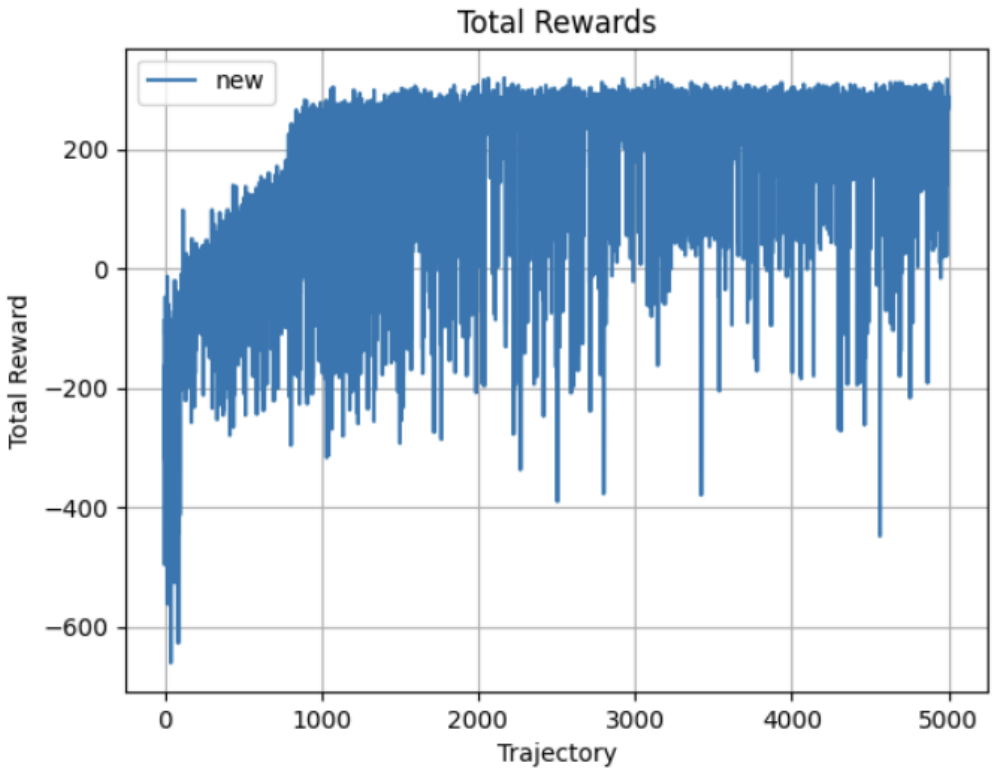
1. Как было сказано на занятиях. Advantage функцию в PPO можно считать и учить по-разному. В задании предлагается написать и исследовать другой способ делать это. А именно использовать представление A(s,a) = r + \gamma V(s') - V(s), где s' - следующее состояние. То есть returns в данном случае использовать не нужно. Необходимо сравнить кривые обучения алгоритма с этим “новым” способом и “старым” способом (из практики) на задаче Pendulum.

   В задании нужно модифицировать метод, который был написан на практике. И сравнить со “старым” методом.



   Вывод: Видно, что модель по новому методу обучилась лучше. Вышла раньше на плато.

2. На практике мы написали PPO для случая одномерного пространства действий. Использование же его для многомерного пространства действий требует небольших технических изменений в коде (при этом содержательно ничего не меняется). Задание заключается в том, чтобы внести эти изменения (т.е. модифицировать PPO для работы в средах с многомерным пространством действий) и решить с его помощью LunarLander (результат должен быть больше 100). Для того, чтобы сделать LunarLander с непрерывным пространством действий нужно положить continuous=True (см. пояснения в Lunar Lander - Gym Documentation (gymlibrary.dev))



Вывод: Модель,примерно после 1000, траектории смогла выйти на плато, но при этом процесс обучения получился достаточно волатильным.

3. Написать PPO для работы в средах с конечным пространствам действий и решить Acrobot. Для решения можно использовать Categorial из torch.distributions (см. pytorch документацию).

   Не успел подобрать параметры для модели, после 20 эпизодов, модель не взлетела.