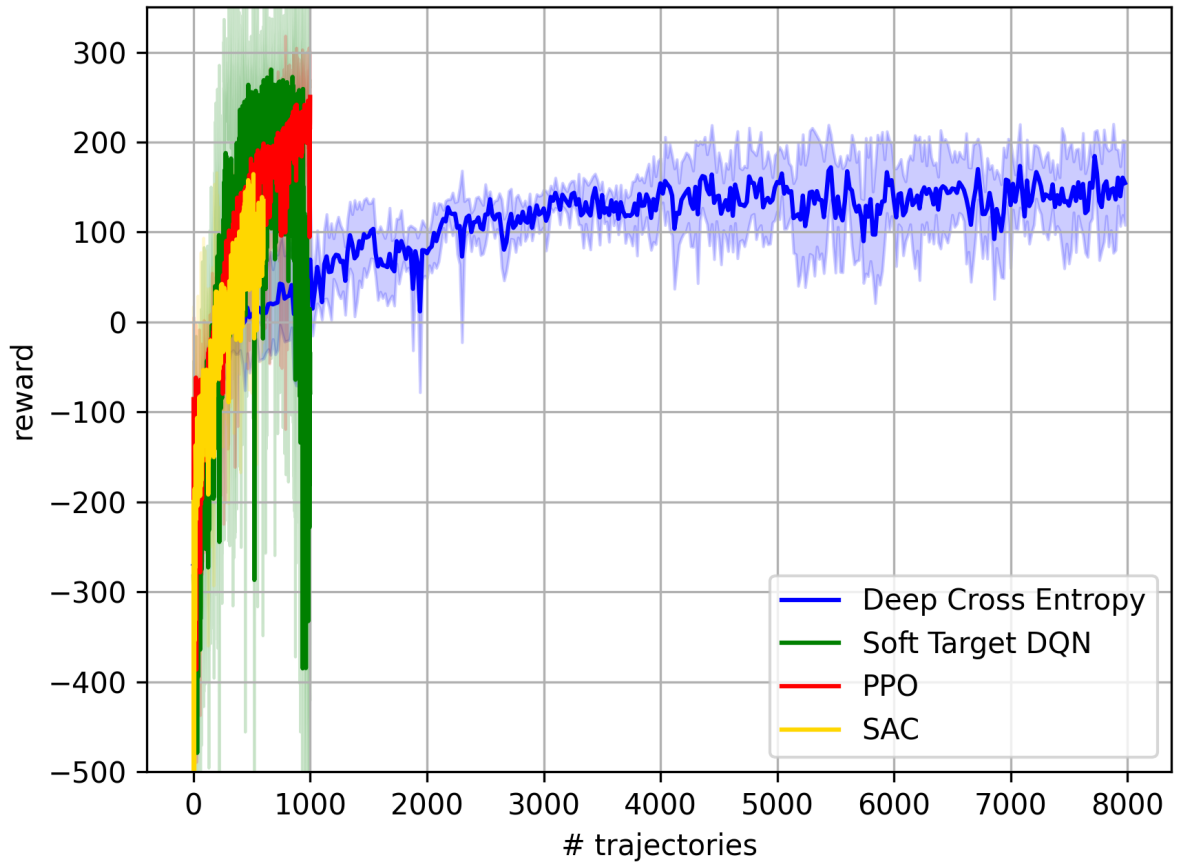
При выполнении домашнего задания 7 было необходимо сравнить 4 алгоритма машинного обучения с подкреплением. Сравнивались Deep Cross Entropy, DQN (мной была выбрана реализация с использованием soft target модели), PPO и SAC.
Для сравнения использовалась среда LunarLander с непрерывным пространством действий.

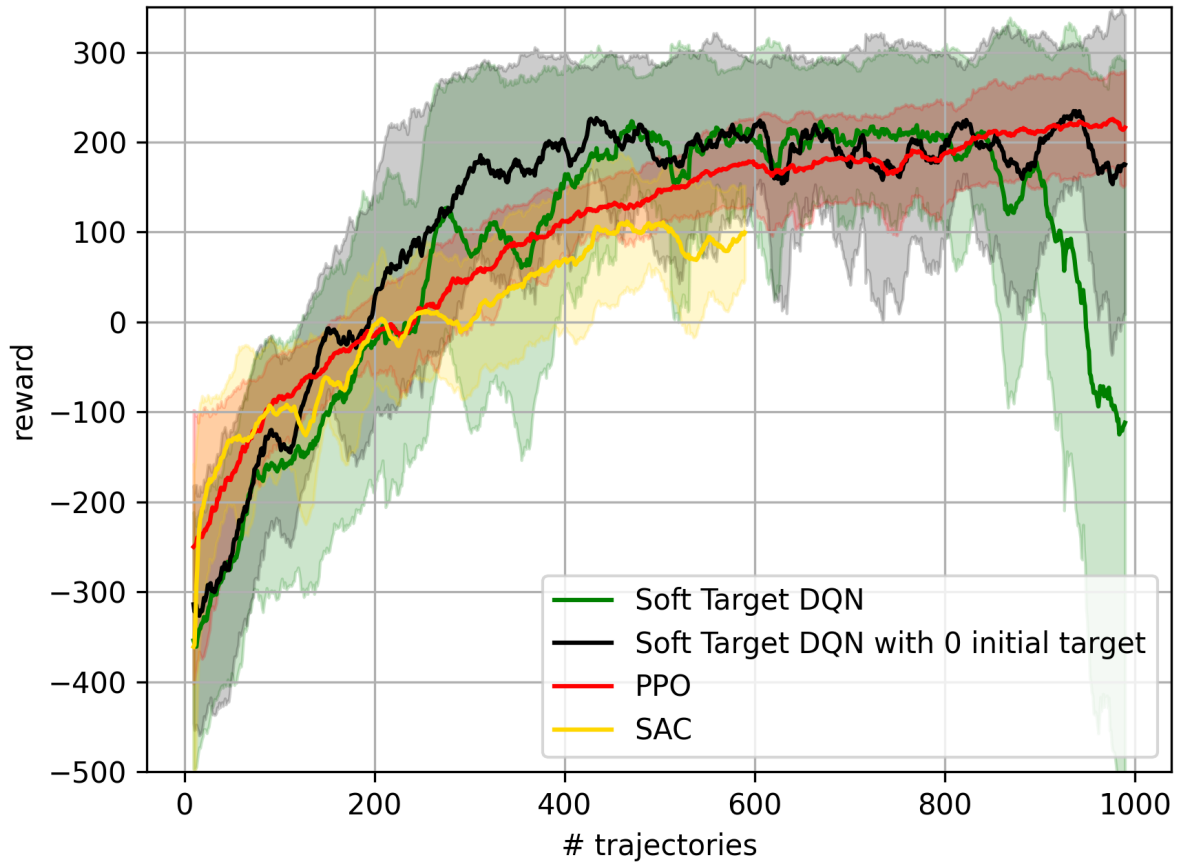
Для сравнения каждый алгоритм был запущен 6 раз, чтобы уменьшить влияние случайной инициализации весов сетей. На графике ниже будут представлены средние кривые награды в зависимости от числа траекторий.

Награды считались в момент обучения без валидации, тк все алгоритмы кроме SAC либо напрямую позволяли менять степень исследования, либо стремились энтропию минимизировать (PPO), что в конечном итоге на последних эпизодах позволяло видеть практически точную производительность агентов (во всех алгоритмах исследование к середине обучения было близко к нулю). Однако как я упомянул ранее SAC такими свойствами не обладает (я считал среднюю предсказанную дисперсию за траекторию и она не стремилась к нулю со временем). Но ввиду ограничения по времени (дедлайн сдачи дз), вычислительно было слишком накладно полноценно валидировать SAC (алгоритм оказался самым долгим по времени вычисления) и тоже не совсем правильно было бы этот график сопоставлять остальным, тк в том числе мы сравниваем не только конечную производительность, но и скорость сходимости в зависимости от числа траекторий. Поэтому на графике будет представлена награда вычисленная по обучающей выборке для всех алгоритмов.



Из графика выше очевидно, что DCE показывает куда более слабую производительность по отношению к другим алгоритмам, что ожидаемо поскольку он обычно использует лишь малую часть сгенерированных траекторий для обучение сети (остальные траектории тоже используются, но в качестве критерия отбора).

Для наглядности ниже изображен график без DCE, но с добавлением модификации к Soft Target DQN, которая заключается в изначальном обнулении всех весов таргетной сети.



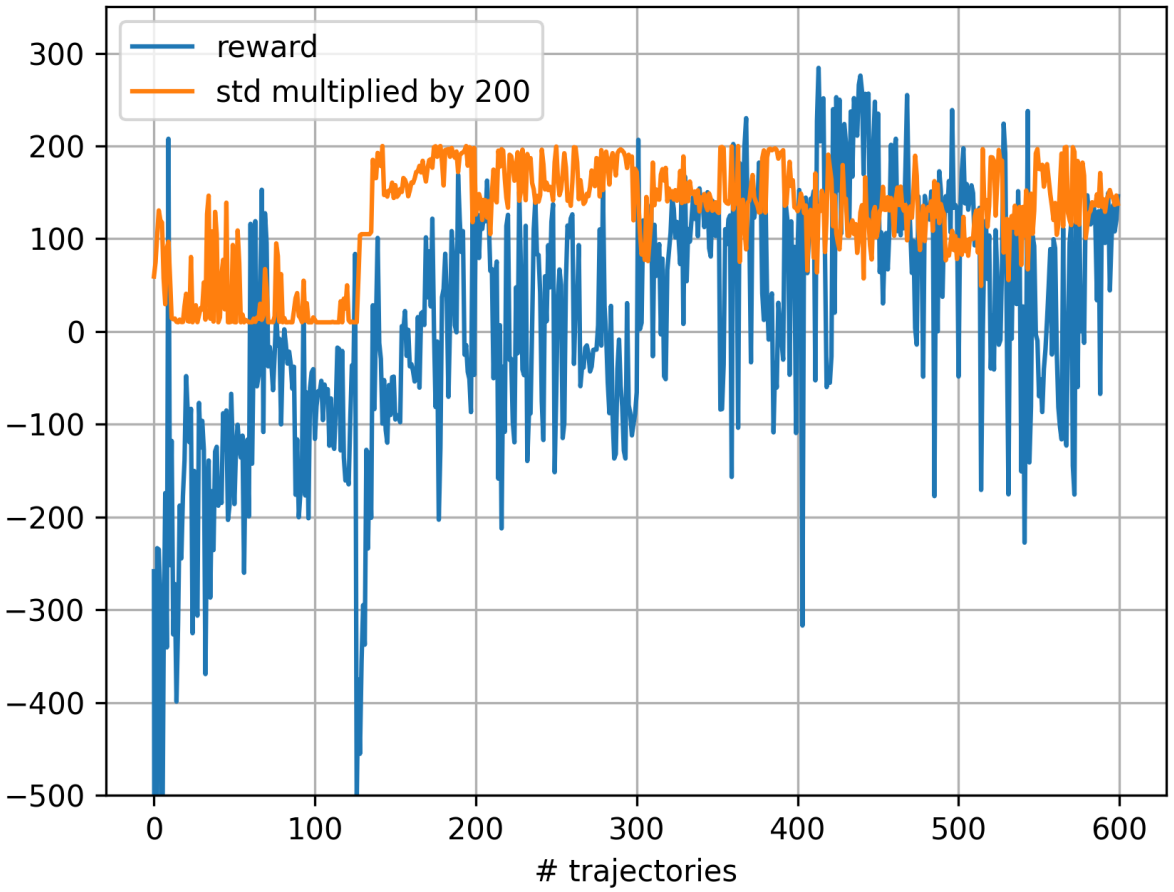
На данном графике мы можем видеть, что для данной среды алгоритмы в целом показали схожую производительность, но очевидно лидирует метод PPO по стабильности и характеру роста награды со временем. Дальше по стабильности располагается SAC и за ними Soft Target и его модификация. При этом по скорости обучения именно Soft Target и модификация немного лидирует, но на уровне погрешности (особенно учитывая не лучшую стабильность этих алгоритмов).

Также относительно производительности SAC (скорости роста награды от траектории) SAC нельзя полностью судить из данного графика, поскольку в этих данных высокий уровень исследования. И валидация конечно выглядит лучше, но не было замечено стабильного достижения награды 200+.

Исходя из графика хочется отметить, что модификация DQN показала себя лучше чем оригинальный подход (отсутствует падение награды в конце и по остальным показателям точно не хуже).

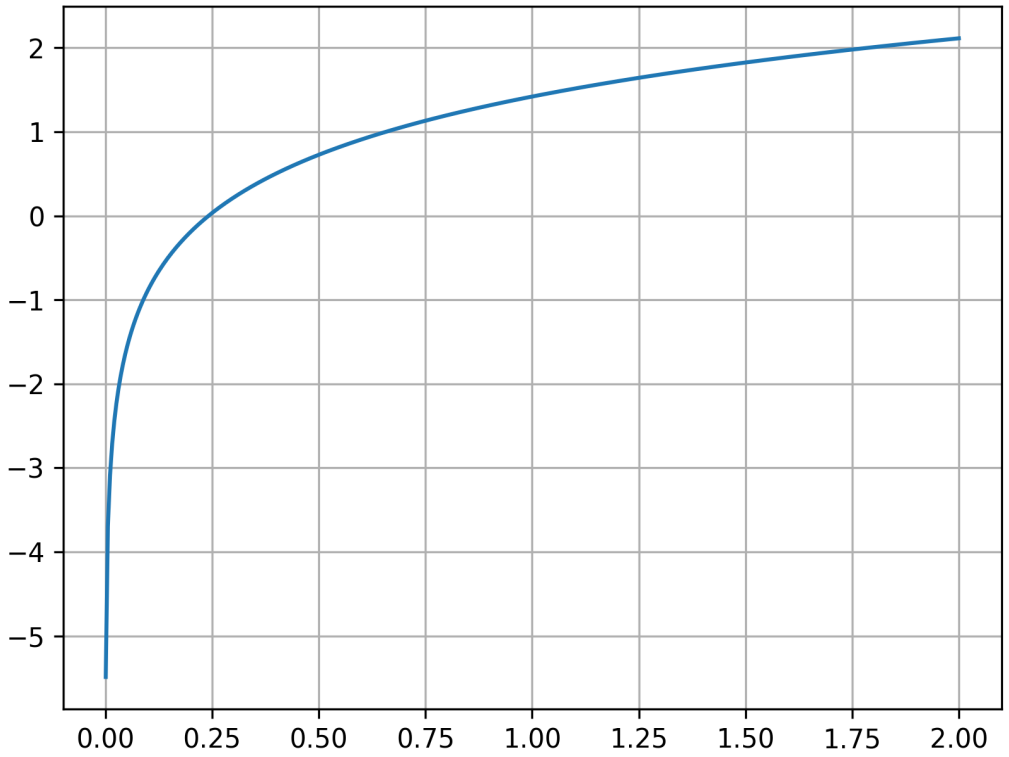
При данных тестах не было в PPO не было добавлено параметра отвечающего за исследование, что мне кажется может улучшить его производительность ещё. Но так же поведение предсказанной дисперсии SAC оставляет впечатление, что этот алгоритм можем увеличивать степень исследования в момент застревания в

локальных минимумах, что возможно делает его более удачным для более сложных сред (данная задача решается линейной моделью с наградой 210+ точно).



Выше на графике показана предсказанная алгоритмом SAC средняя за траекторию дисперсия (помноженная на 200 для наглядности) и награда. Я бы сказал, что тут присутствуют моменты увеличения дисперсии при падениях награды (попадания в локальный минимум). Но на графике это не очень хорошо отражено, потому что судить о попадании в локальный минимум можно только по нулевому градиенту для весов политики, который считается по батчу из всей выборки (значит не всегда текущее состояние политики определяет намерение метода увеличить энтропию или нет).

Так же хочу в защиту SAC сказать, что возможно метод оптимизировал поставленный функционал хорошо, и этот функционал просто отличается от того, что мы отображаем на графике. А именно на графике отсутствует учет дисконта и значение энтропии умноженной на температуру. Ниже приведен график энтропии в зависимости от дисперсии.



Я выставлял температуру для SAC равную 0.1 (при меньших значениях даже на начальных этапах алгоритм минимизировал энтропию и попадал в локальный минимум - зависая в воздухе, чтобы избежать падения). При температуре 0.1 наш штраф колеблется от -0.5 до 0.2. А награда алгоритма от -100 до 250+, что натолкнуло меня на мысль о допустимости такого значения температуры. Однако с учетом дисконта максимальная награда падает до ~50, что увеличивает привлекательность для повышения энтропии. И возможно стоит поработать еще с гипер параметрами (общие параметры с другими алгоритмами были выбраны одинаковые). Так же получение награды 200+ агенту необходимо какое-то время оставаться неподвижным, что не входит в рамки марковской постановки задачи, поэтому и требовать этого от алгоритмов не стоит, но конечно было бы славно:)