## Первые шаги

Заполнил все пропуски “TODO” и преступил к экспериментам.

В качестве исследуемых сред были выбраны Ant и Humanoid.

## Конфиги и Аугментации

Ведем конфиги запусков:

Для Behavior Cloning:

class Args:

  def \_\_getitem\_\_(self, key):

    return getattr(self, key)

  def \_\_setitem\_\_(self, key, val):

    setattr(self, key, val)

  #@markdown expert data

  expert\_policy\_file = 'cds\_rl/policies/experts/Ant.pkl' #@param

  expert\_data = 'cds\_rl/expert\_data/expert\_data\_Ant-v2.pkl' #@param

  env\_name = 'Ant-v2' #@param ['Ant-v2', 'Humanoid-v2', 'Walker2d-v2', 'HalfCheetah-v2', 'Hopper-v2']

  exp\_name = 'test\_bc\_ant' #@param

  do\_dagger = False #@param {type: "boolean"}

  ep\_len = 1000 #@param {type: "integer"}

  save\_params = True #@param {type: "boolean"}

  num\_agent\_train\_steps\_per\_iter = 1000 #@param {type: "integer"})

  n\_iter = 1 #@param {type: "integer"})

  #@markdown batches & buffers

  batch\_size = 1000 #@param {type: "integer"})

  eval\_batch\_size = 10000 #@param {type: "integer"}

  train\_batch\_size = 100 #@param {type: "integer"}

  max\_replay\_buffer\_size = 1000000 #@param {type: "integer"}

  #@markdown network

  n\_layers = 2 #@param {type: "integer"}

  size =  64#@param {type: "integer"}

  learning\_rate = 5e-3 #@param {type: "number"}

  #@markdown logging

  video\_log\_freq =  -1#@param {type: "integer"}

  scalar\_log\_freq = 1 #@param {type: "integer"}

  #@markdown gpu & run-time settings

  no\_gpu = False #@param {type: "boolean"}

  which\_gpu = 0 #@param {type: "integer"}

  seed = 1 #@param {type: "integer"}

Для Dagger:

class Args:

  def \_\_getitem\_\_(self, key):

    return getattr(self, key)

  def \_\_setitem\_\_(self, key, val):

    setattr(self, key, val)

  #@markdown expert data

  expert\_policy\_file = 'cds\_rl/policies/experts/Ant.pkl' #@param

  expert\_data = 'cds\_rl/expert\_data/expert\_data\_Ant-v2.pkl' #@param

  env\_name = 'Ant-v2' #@param ['Ant-v2', 'Humanoid-v2', 'Walker2d-v2', 'HalfCheetah-v2', 'Hopper-v2']

  exp\_name = 'test\_da\_ant' #@param

  do\_dagger = True #@param {type: "boolean"}

  ep\_len = 1000 #@param {type: "integer"}

  save\_params = True #@param {type: "boolean"}

  num\_agent\_train\_steps\_per\_iter = 1000 #@param {type: "integer"})

  n\_iter = 10 #@param {type: "integer"})

  #@markdown batches & buffers

  batch\_size = 1000 #@param {type: "integer"})

  eval\_batch\_size = 10000 #@param {type: "integer"}

  train\_batch\_size = 100 #@param {type: "integer"}

  max\_replay\_buffer\_size = 1000000 #@param {type: "integer"}

  #@markdown network

  n\_layers = 2 #@param {type: "integer"}

  size =  64#@param {type: "integer"}

  learning\_rate = 5e-3 #@param {type: "number"}

  #@markdown logging

  video\_log\_freq =  -1#@param {type: "integer"}

  scalar\_log\_freq = 1 #@param {type: "integer"}

  #@markdown gpu & run-time settings

  no\_gpu = False #@param {type: "boolean"}

  which\_gpu = 0 #@param {type: "integer"}

  seed = 1 #@param {type: "integer"}

А также укажем набор аугментаций конфига:

Orig:

- no\_changes

Mod\_1:

- size = 128

Mod\_2:

- size = 256

Mod\_3:

- size = 32

Mod\_4:

- num\_agent\_train\_steps\_per\_iter = 10000

## Важные замечания

Замечания по графикам:

* Для Ant Average и Std строятся на отдельных графиках, т.к. Std на порядок меньше и делает совместный график нечитаемым.
* Для Humanoid не строится , т.к. он на порядок больше и делает совместный график нечитаемым. Он будет указываться отдельным числом.

В папке run\_logs лежит 20 логов:

* 5 – Ant Behavioral Cloning
* 5 – Ant Dagger
* 5 – Humanoid Behavioral Cloning
* 5 – Humanoid Dagger

## Задание BC.1

Метрики Ant в зависимости от аугментации:

|  |  |  |  |
| --- | --- | --- | --- |
| Augmentations | Eval\_AverageReturn | Eval\_StdReturn | Expert\_AverageReturn |
| Orig | 4722.0615 | 99.0428 | 4713.6533 |
| Mod\_1 | 4299.5518 | 1167.1285 |
| Mod\_2 | 4093.5215 | 1169.6318 |
| Mod\_3 | 4623.3975 | 82.7196 |
| Mod\_4 | 4790.5366 | 64.0650 |

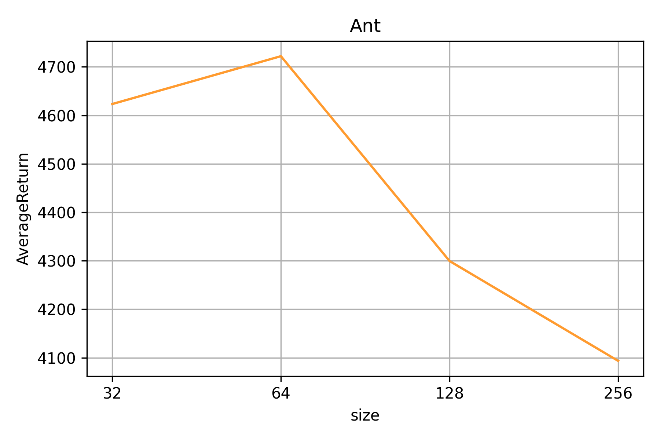
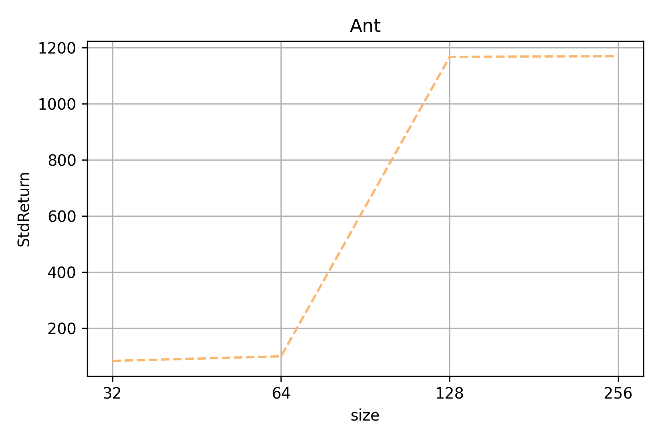
Метрики Humanoid в зависимости от аугментации:

|  |  |  |  |
| --- | --- | --- | --- |
| Augmentations | Eval\_AverageReturn | Eval\_StdReturn | Expert\_AverageReturn |
| Orig | 272.9457 | 15.1833 | 10344.5176 |
| Mod\_1 | 268.3560 | 12.5534 |
| Mod\_2 | 273.2079 | 19.5259 |
| Mod\_3 | 281.2399 | 16.2930 |
| Mod\_4 | 280.8163 | 56.3695 |

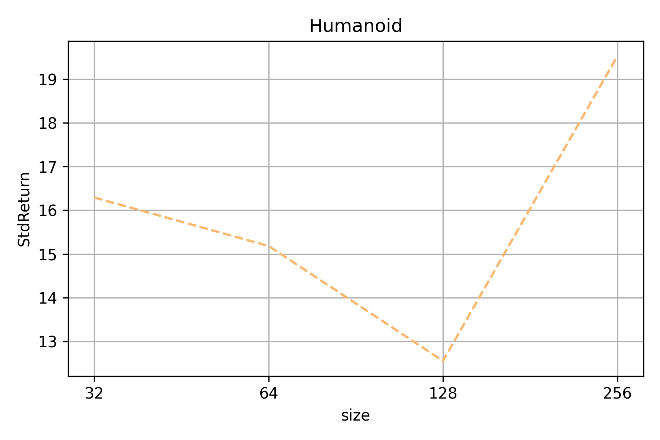
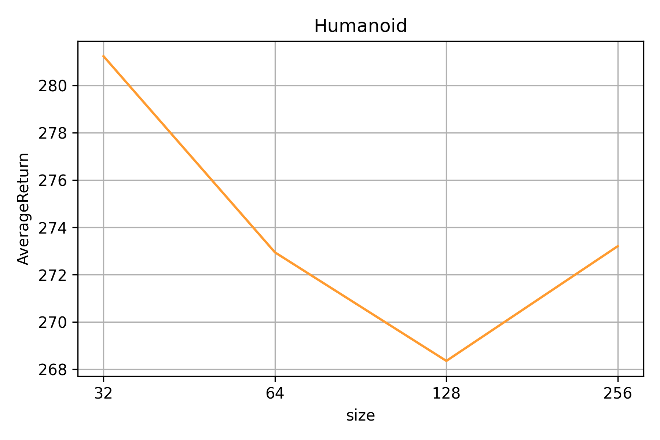
## Задание BC.2

Часть с различными конфигурациями уже была приведена в [Задании BC.1](#_Задание_BC.1).

Была рассмотрена зависимость качества работы от параметра size (Mod\_3, Orig, Mod\_1, Mod\_2).

Можно видеть, что лучшее качество модели наблюдается при .



Для Humanoid лучшим является .

## Задание DA.1

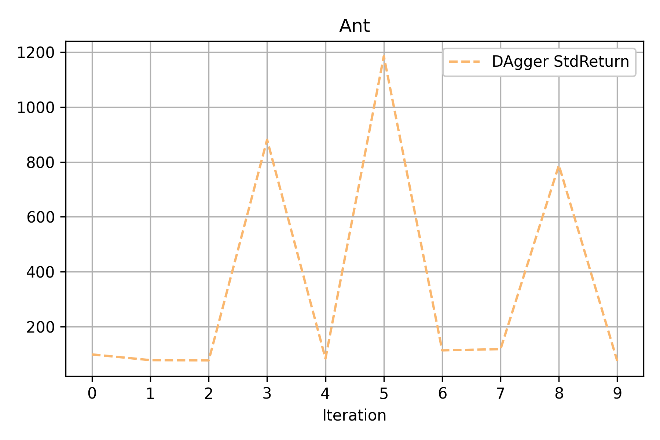
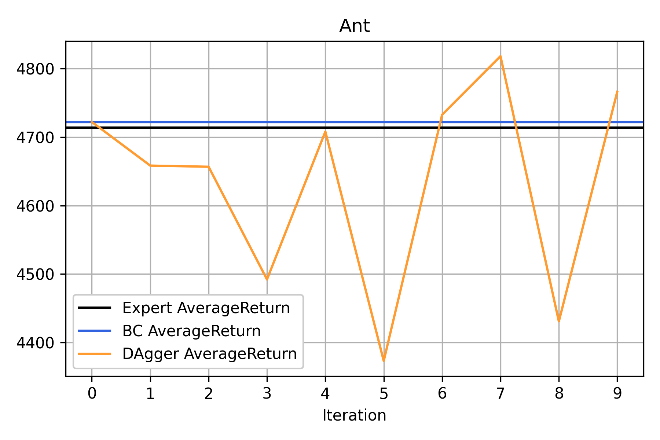
### Ant

Для Ant были произведены эксперименты с каждым набором аугментаций. Для каждого эксперимента будут отдельно указаны значения и и .

Orig:

BC\_AverageReturn = 4722.0615

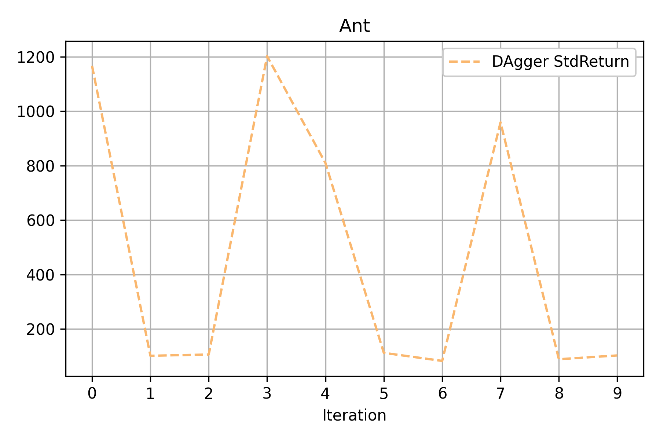
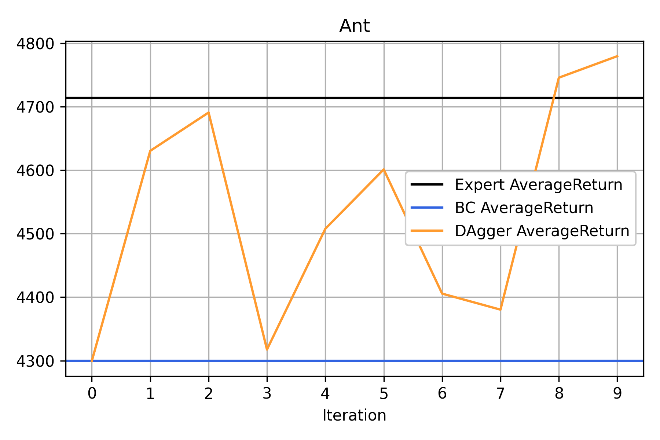
Expert\_AverageReturn = 4713.6533



Mod\_1:

BC\_AverageReturn = 4299.5518

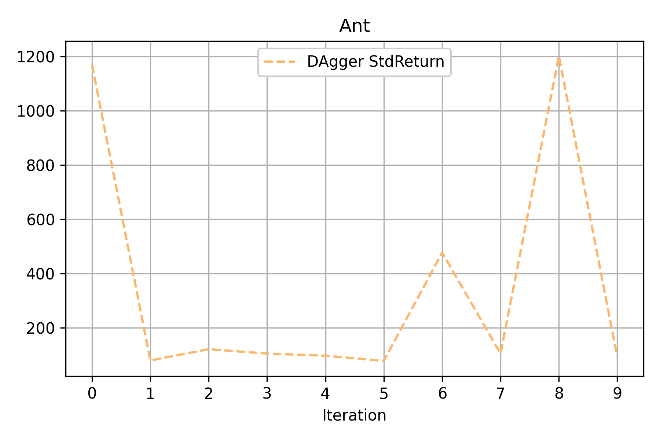
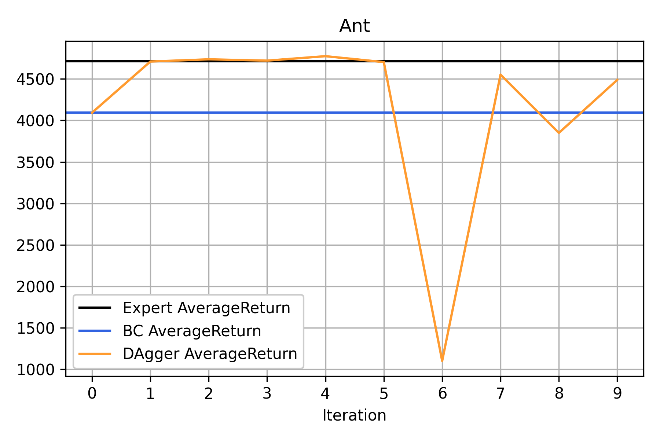
Expert\_AverageReturn = 4713.6533



Mod\_2:

BC\_AverageReturn = 4093.5215

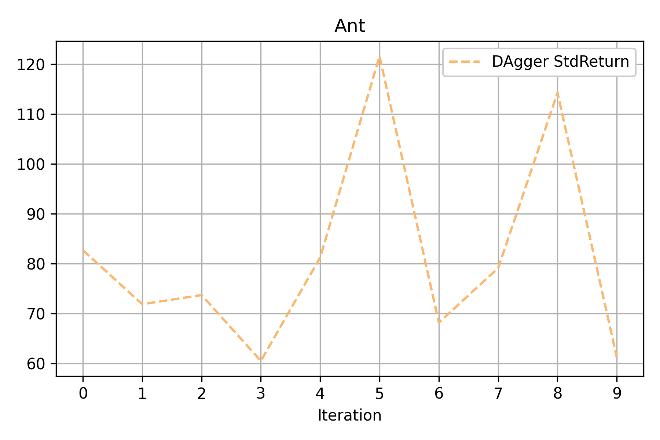
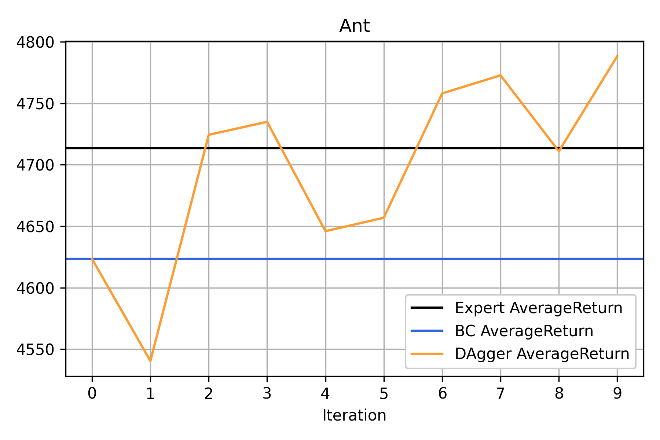
Expert\_AverageReturn = 4713.6533



Mod\_3:

BC\_AverageReturn = 4722.0615

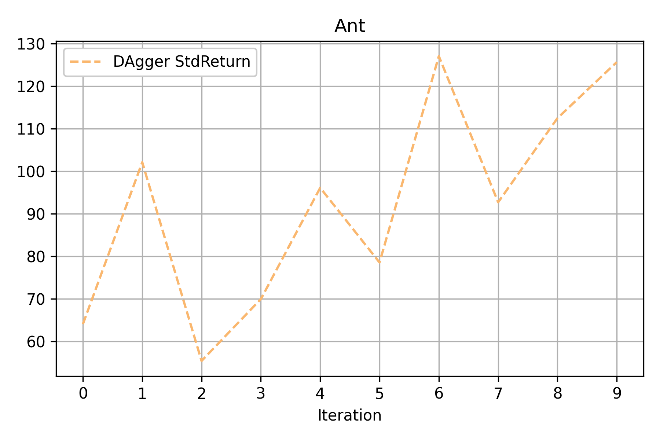
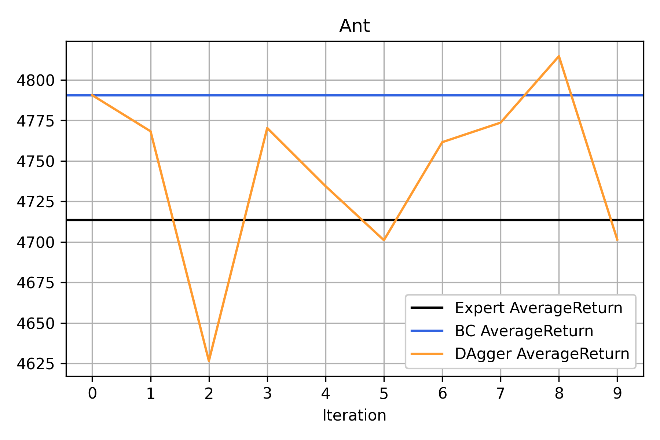
Expert\_AverageReturn = 4713.6533



Mod\_4:

BC\_AverageReturn = 4790.5366

Expert\_AverageReturn = 4713.6533



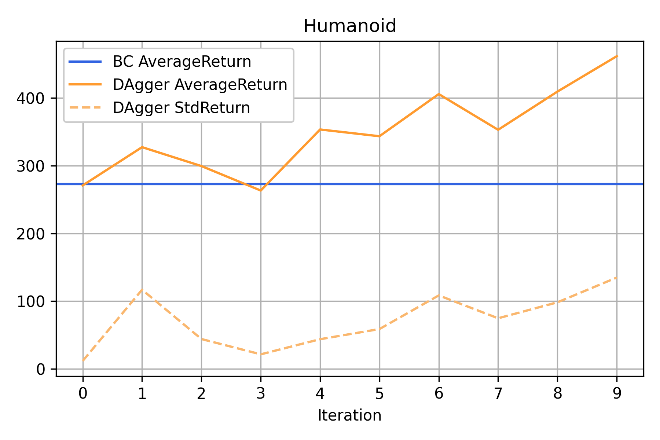
### Humanoid

Результаты для Humanoid так себе при таком наборе аугментаций, но что есть – то есть. Для каждого эксперимента будут отдельно указаны значения и и . не отображен на графиках, т.к. значительно превосходит все другие показатели.

Orig:

BC\_AverageReturn = 272.9457

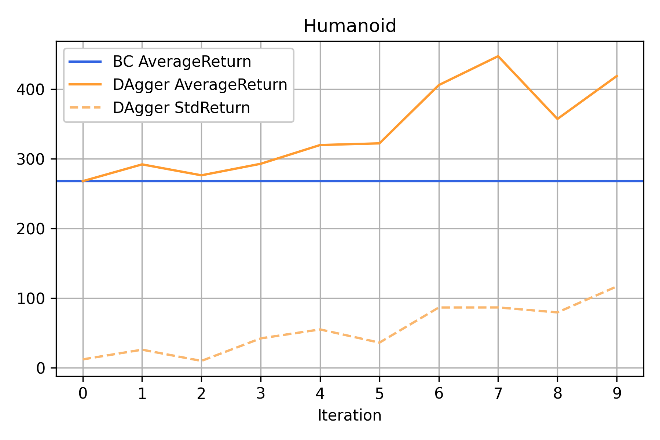
Expert\_AverageReturn = 10344.5176



Mod\_1:

BC\_AverageReturn = 268.3560

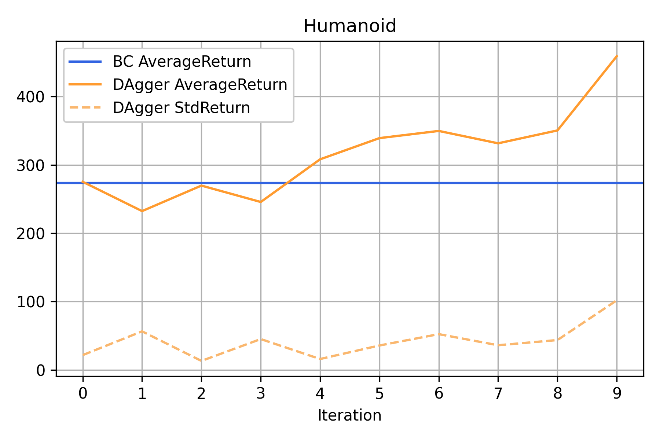
Expert\_AverageReturn = 10344.5176



Mod\_2:

BC\_AverageReturn = 273.2079

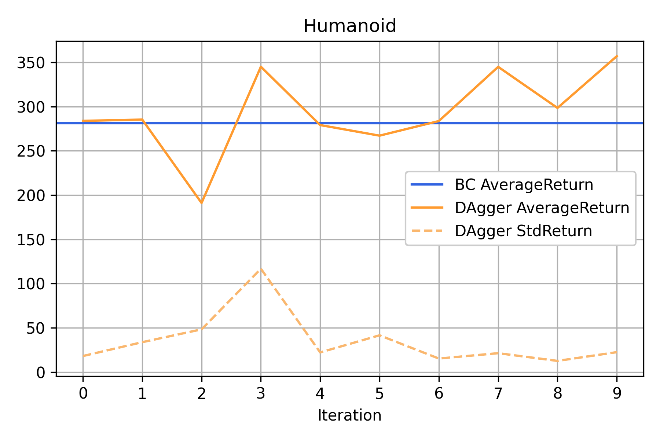
Expert\_AverageReturn = 10344.5176



Mod\_3:

BC\_AverageReturn = 281.2399

Expert\_AverageReturn = 10344.5176



Mod\_4:

BC\_AverageReturn = 280.8163

Expert\_AverageReturn = 10344.5176

