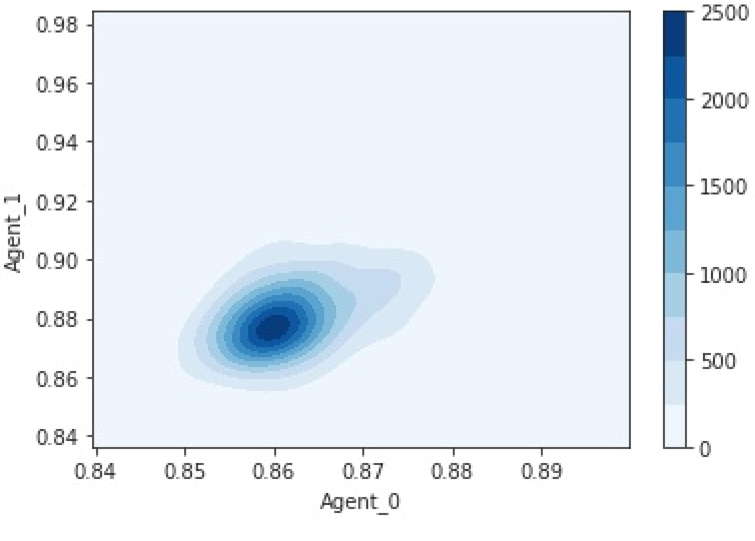
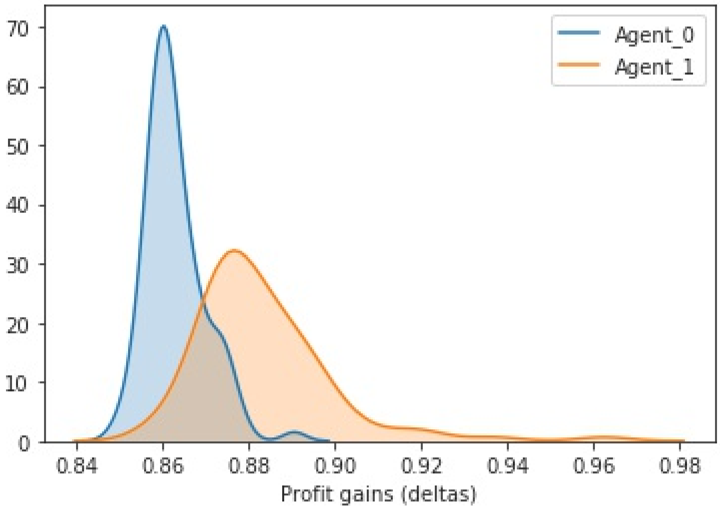
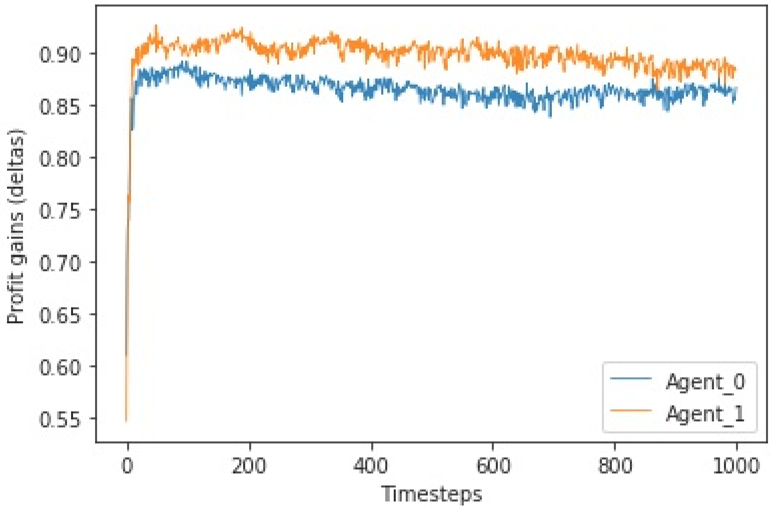
```
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.403,
max=1.403, mean=1.403)},
(pid=24815)   2: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.444,
max=1.444, mean=1.444),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.444,
max=1.444, mean=1.444)},
(pid=24815)   3: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.236,
max=1.236, mean=1.236),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.236,
max=1.236, mean=1.236)},
(pid=24815)   4: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.395,
max=1.395, mean=1.395),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.395,
max=1.395, mean=1.395)},
(pid=24815)   5: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.663,
max=1.663, mean=1.663),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.663,
max=1.663, mean=1.663)},
(pid=24815)   6: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.671,
max=1.671, mean=1.671),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.671,
max=1.671, mean=1.671)},
(pid=24815)   7: { 'agent_0': np.ndarray((2,), dtype=float16, min=1.717,
max=1.717, mean=1.717),
(pid=24815)        'agent_1': np.ndarray((2,), dtype=float16, min=1.717,
max=1.717, mean=1.717)}}
(pid=24815) 2019-09-21 12:41:58,392        INFO sampler.py:305 -- Info return from
env: { 0: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   1: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   2: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   3: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   4: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   5: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   6: {'agent_0': {}, 'agent_1': {}},
(pid=24815)   7: {'agent_0': {}, 'agent_1': {}}}
(pid=24815) 2019-09-21 12:41:58,392        INFO sampler.py:403 -- Preprocessed
obs: np.ndarray((2,), dtype=float16, min=1.95, max=1.95, mean=1.95)
(pid=24815) 2019-09-21 12:41:58,392        INFO sampler.py:407 -- Filtered obs:
np.ndarray((2,), dtype=float64, min=0.0, max=0.0, mean=0.0)
(pid=24815) 2019-09-21 12:41:58,401        INFO sampler.py:521 -- Inputs to
compute_actions():
(pid=24815)
(pid=24815) { 'agent_0': [ { 'data': { 'agent_id': 'agent_0',
(pid=24815)                            'env_id': 0,
(pid=24815)                            'info': {},
(pid=24815)                            'obs': np.ndarray((2,), dtype=float64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                            'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                            'prev_reward': 0.0,
(pid=24815)                            'rnn_state': []},
(pid=24815)                  'type': 'PolicyEvalData'},
(pid=24815)                { 'data': { 'agent_id': 'agent_0',
(pid=24815)                            'env_id': 1,
(pid=24815)                            'info': {},
(pid=24815)                            'obs': np.ndarray((2,), dtype=float64,
min=-0.707, max=-0.707, mean=-0.707),
(pid=24815)                            'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 2,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.509, max=-0.509, mean=-0.509),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 3,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.884, max=-0.884, mean=-0.884),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 4,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.336, max=-0.336, mean=-0.336),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 5,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=0.583, max=0.583, mean=0.583),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 6,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=0.559, max=0.559, mean=0.559),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_0',
(pid=24815)                             'env_id': 7,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=0.682, max=0.682, mean=0.682),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'}],
(pid=24815)   'agent_1': [ { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 0,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 1,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.707, max=-0.707, mean=-0.707),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 2,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.509, max=-0.509, mean=-0.509),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 3,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.884, max=-0.884, mean=-0.884),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 4,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=-0.336, max=-0.336, mean=-0.336),
(pid=24815)                             'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                             'prev_reward': 0.0,
(pid=24815)                             'rnn_state': []},
(pid=24815)                   'type': 'PolicyEvalData'},
(pid=24815)                 { 'data': { 'agent_id': 'agent_1',
(pid=24815)                             'env_id': 5,
(pid=24815)                             'info': {},
(pid=24815)                             'obs': np.ndarray((2,), dtype=float64,
min=0.583, max=0.583, mean=0.583),
(pid=24815)                            'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                            'prev_reward': 0.0,
(pid=24815)                            'rnn_state': []},
(pid=24815)                  'type': 'PolicyEvalData'},
(pid=24815)                { 'data': { 'agent_id': 'agent_1',
(pid=24815)                            'env_id': 6,
(pid=24815)                            'info': {},
(pid=24815)                            'obs': np.ndarray((2,), dtype=float64,
min=0.559, max=0.559, mean=0.559),
(pid=24815)                            'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                            'prev_reward': 0.0,
(pid=24815)                            'rnn_state': []},
(pid=24815)                  'type': 'PolicyEvalData'},
(pid=24815)                { 'data': { 'agent_id': 'agent_1',
(pid=24815)                            'env_id': 7,
(pid=24815)                            'info': {},
(pid=24815)                            'obs': np.ndarray((2,), dtype=float64,
min=0.682, max=0.682, mean=0.682),
(pid=24815)                            'prev_action': np.ndarray((), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                            'prev_reward': 0.0,
(pid=24815)                            'rnn_state': []},
(pid=24815)                  'type': 'PolicyEvalData'}]}
(pid=24815)
(pid=24815) 2019-09-21 12:41:58,402        INFO tf_run_builder.py:92 -- Executing
TF run without tracing. To dump TF timeline traces to disk, set the
TF_TIMELINE_DIR environment variable.
(pid=24815) 2019-09-21 12:41:58,535        INFO sampler.py:548 -- Outputs of
compute_actions():
(pid=24815)
(pid=24815) { 'agent_0': ( np.ndarray((8,), dtype=int64, min=0.0, max=4.0,
mean=2.75),
(pid=24815)                [],
(pid=24815)                { 'q_values': np.ndarray((8, 5), dtype=float32,
min=-1.808, max=1.219, mean=-0.229)}),
(pid=24815)   'agent_1': ( np.ndarray((8,), dtype=int64, min=0.0, max=4.0,
mean=2.625),
(pid=24815)                [],
(pid=24815)                { 'q_values': np.ndarray((8, 5), dtype=float32,
min=-2.423, max=1.053, mean=-0.47)})}
(pid=24815)
(pid=24815) 2019-09-21 12:41:58,942        INFO sample_batch_builder.py:161 --
Trajectory fragment after postprocess_trajectory():
(pid=24815)
(pid=24815) { 'agent_0': { 'data': { 'actions': np.ndarray((32,), dtype=int64,
min=0.0, max=4.0, mean=2.094),
(pid=24815)                          'agent_index': np.ndarray((32,), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                          'dones': np.ndarray((32,), dtype=bool,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                          'eps_id': np.ndarray((32,), dtype=int64,
min=1802379756.0, max=1802379756.0, mean=1802379756.0),
(pid=24815)                          'infos': np.ndarray((32,), dtype=object,
head={'delta': -0.042803041650380264}),
(pid=24815)                          'new_obs': np.ndarray((32, 2), dtype=float32,
min=-1.379, max=1.976, mean=0.384),
(pid=24815)                          'obs': np.ndarray((32, 2), dtype=float32,
min=-1.379, max=1.976, mean=0.414),
(pid=24815)                          'prev_actions': np.ndarray((32,),
dtype=int64, min=0.0, max=4.0, mean=2.062),
(pid=24815)                          'prev_rewards': np.ndarray((32,),
dtype=float32, min=0.0, max=0.361, mean=0.213),
(pid=24815)                          'q_values': np.ndarray((32, 5),
dtype=float32, min=-1.49, max=0.771, mean=-0.12),
(pid=24815)                          'rewards': np.ndarray((32,), dtype=float32,
min=0.286, max=0.977, mean=0.63),
(pid=24815)                          't': np.ndarray((32,), dtype=int64, min=0.0,
max=31.0, mean=15.5),
(pid=24815)                          'unroll_id': np.ndarray((32,), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                          'weights': np.ndarray((32,), dtype=float32,
min=2.244, max=2.465, mean=2.347)},
(pid=24815)                'type': 'SampleBatch'},
(pid=24815)   'agent_1': { 'data': { 'actions': np.ndarray((32,), dtype=int64,
min=0.0, max=4.0, mean=2.0),
(pid=24815)                          'agent_index': np.ndarray((32,), dtype=int64,
min=1.0, max=1.0, mean=1.0),
(pid=24815)                          'dones': np.ndarray((32,), dtype=bool,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                          'eps_id': np.ndarray((32,), dtype=int64,
min=1802379756.0, max=1802379756.0, mean=1802379756.0),
(pid=24815)                          'infos': np.ndarray((32,), dtype=object,
head={'delta': 1.76945978085831}),
(pid=24815)                          'new_obs': np.ndarray((32, 2), dtype=float32,
min=-1.379, max=1.976, mean=0.384),
(pid=24815)                          'obs': np.ndarray((32, 2), dtype=float32,
min=-1.379, max=1.976, mean=0.414),
(pid=24815)                          'prev_actions': np.ndarray((32,),
dtype=int64, min=0.0, max=4.0, mean=1.938),
(pid=24815)                          'prev_rewards': np.ndarray((32,),
dtype=float32, min=0.0, max=0.428, mean=0.32),
(pid=24815)                          'q_values': np.ndarray((32, 5),
dtype=float32, min=-1.668, max=1.335, mean=0.093),
(pid=24815)                          'rewards': np.ndarray((32,), dtype=float32,
min=0.281, max=1.238, mean=0.926),
(pid=24815)                          't': np.ndarray((32,), dtype=int64, min=0.0,
max=31.0, mean=15.5),
(pid=24815)                          'unroll_id': np.ndarray((32,), dtype=int64,
min=0.0, max=0.0, mean=0.0),
(pid=24815)                          'weights': np.ndarray((32,), dtype=float32,
min=2.232, max=2.574, mean=2.333)},
(pid=24815)                'type': 'SampleBatch'}}
(pid=24815)
(pid=24815) 2019-09-21 12:41:58,980        INFO rollout_worker.py:485 -- Completed
sample batch:
(pid=24815)
(pid=24815) { 'count': 256,
(pid=24815)   'policy_batches': { 'agent_0': { 'data': { 'actions':
np.ndarray((256,), dtype=int64, min=0.0, max=4.0, mean=1.965),
(pid=24815)                                              'agent_index':
np.ndarray((256,), dtype=int64, min=0.0, max=0.0, mean=0.0),
(pid=24815)                                              'dones':
np.ndarray((256,), dtype=bool, min=0.0, max=0.0, mean=0.0),
(pid=24815)                                              'eps_id':
np.ndarray((256,), dtype=int64, min=103782451.0, max=1847514065.0,
mean=1427203924.75),
(pid=24815)                                                   'infos':
np.ndarray((256,), dtype=object, head={'delta': -0.042803041650380264}),
(pid=24815)                                                   'new_obs':
np.ndarray((256, 2), dtype=float32, min=-1.853, max=2.08, mean=0.009),
(pid=24815)                                                   'obs': np.ndarray((256,
2), dtype=float32, min=-1.853, max=2.08, mean=0.044),
(pid=24815)                                                   'prev_actions':
np.ndarray((256,), dtype=int64, min=0.0, max=4.0, mean=1.91),
(pid=24815)                                                   'prev_rewards':
np.ndarray((256,), dtype=float32, min=0.0, max=0.428, mean=0.243),
(pid=24815)                                                   'q_values':
np.ndarray((256, 5), dtype=float32, min=-2.461, max=2.082, mean=0.117),
(pid=24815)                                                   'rewards':
np.ndarray((256,), dtype=float32, min=0.129, max=1.245, mean=0.71),
(pid=24815)                                                   't': np.ndarray((256,),
dtype=int64, min=0.0, max=31.0, mean=15.5),
(pid=24815)                                                   'unroll_id':
np.ndarray((256,), dtype=int64, min=0.0, max=0.0, mean=0.0),
(pid=24815)                                                   'weights':
np.ndarray((256,), dtype=float32, min=2.213, max=2.774, mean=2.411)},
(pid=24815)                                         'type': 'SampleBatch'},
(pid=24815)                            'agent_1': { 'data': { 'actions':
np.ndarray((256,), dtype=int64, min=0.0, max=4.0, mean=2.066),
(pid=24815)                                                   'agent_index':
np.ndarray((256,), dtype=int64, min=1.0, max=1.0, mean=1.0),
(pid=24815)                                                   'dones':
np.ndarray((256,), dtype=bool, min=0.0, max=0.0, mean=0.0),
(pid=24815)                                                   'eps_id':
np.ndarray((256,), dtype=int64, min=103782451.0, max=1847514065.0,
mean=1427203924.75),
(pid=24815)                                                   'infos':
np.ndarray((256,), dtype=object, head={'delta': 1.76945978085831}),
(pid=24815)                                                   'new_obs':
np.ndarray((256, 2), dtype=float32, min=-1.853, max=2.08, mean=0.009),
(pid=24815)                                                   'obs': np.ndarray((256,
2), dtype=float32, min=-1.853, max=2.08, mean=0.044),
(pid=24815)                                                   'prev_actions':
np.ndarray((256,), dtype=int64, min=0.0, max=4.0, mean=1.988),
(pid=24815)                                                   'prev_rewards':
np.ndarray((256,), dtype=float32, min=0.0, max=0.428, mean=0.227),
(pid=24815)                                                   'q_values':
np.ndarray((256, 5), dtype=float32, min=-2.754, max=1.587, mean=-0.336),
(pid=24815)                                                   'rewards':
np.ndarray((256,), dtype=float32, min=0.143, max=1.238, mean=0.662),
(pid=24815)                                                   't': np.ndarray((256,),
dtype=int64, min=0.0, max=31.0, mean=15.5),
(pid=24815)                                                   'unroll_id':
np.ndarray((256,), dtype=int64, min=0.0, max=0.0, mean=0.0),
(pid=24815)                                                   'weights':
np.ndarray((256,), dtype=float32, min=2.232, max=2.914, mean=2.422)},
(pid=24815)                                         'type': 'SampleBatch'}},
(pid=24815)   'type': 'MultiAgentBatch'}
(pid=24815)
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint8 = np.dtype([("qint8", np.int8, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
```

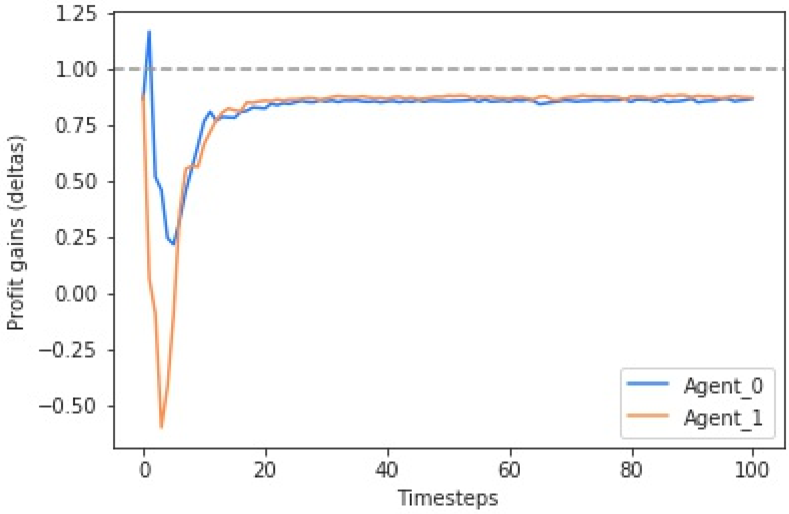
```
tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint16 = np.dtype([("qint16", np.int16, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorflow/python/framework/dtypes.py:519: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorflow/python/framework/dtypes.py:520: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint32 = np.dtype([("qint32", np.int32, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorflow/python/framework/dtypes.py:525: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   np_resource = np.dtype([("resource", np.ubyte, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:541: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint8 = np.dtype([("qint8", np.int8, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:542: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:543: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint16 = np.dtype([("qint16", np.int16, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:544: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:545: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   _np_qint32 = np.dtype([("qint32", np.int32, 1)])
(pid=24826) /home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
tensorboard/compat/tensorflow_stub/dtypes.py:550: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
(pid=24826)   np_resource = np.dtype([("resource", np.ubyte, 1)])
(pid=24826) WARNING:tensorflow:From /home/lorenzo/anaconda3/envs/py36/lib/
python3.6/site-packages/tensorflow/python/compat/v2_compat.py:61:
disable_resource_variables (from tensorflow.python.ops.variable_scope) is
deprecated and will be removed in a future version.
(pid=24826) Instructions for updating:
```

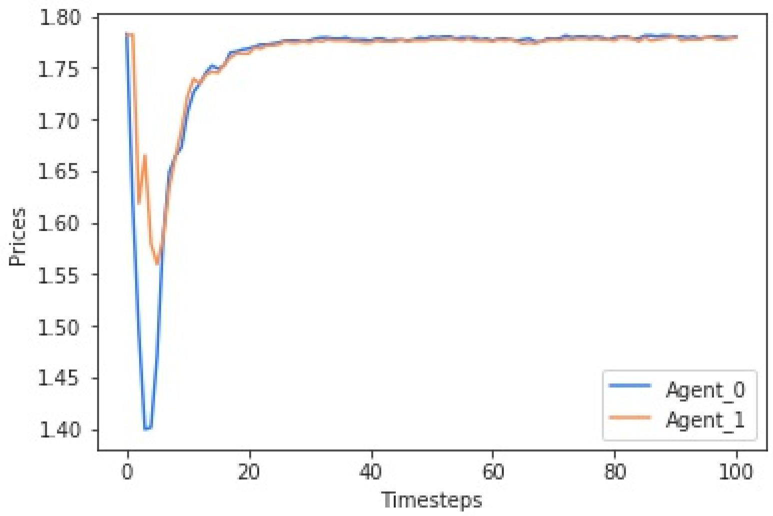
```
(pid=24826) non-resource variables are not supported in the long term
Episode reward 6492.261776493763
Episode 1 of 100
Episode reward 6454.628309068322
Episode 2 of 100
Episode reward 6453.023336687925
Episode 3 of 100
Episode reward 6460.624256761028
Episode 4 of 100
Episode reward 6446.319898075541
Episode 5 of 100
Episode reward 6462.016407910562
Episode 6 of 100
Episode reward 6469.004566093881
Episode 7 of 100
Episode reward 6463.572205063813
Episode 8 of 100
Episode reward 6487.142259706329
Episode 9 of 100
Episode reward 6466.5410214858375
Episode 10 of 100
Episode reward 6445.491307568221
Episode 11 of 100
Episode reward 6504.407948889994
Episode 12 of 100
Episode reward 6464.597901312598
Episode 13 of 100
Episode reward 6464.323295704947
Episode 14 of 100
Episode reward 6462.52863155995
Episode 15 of 100
Episode reward 6454.527204281379
Episode 16 of 100
Episode reward 6447.765205410164
Episode 17 of 100
Episode reward 6486.6555478539185
Episode 18 of 100
Episode reward 6451.116870024328
Episode 19 of 100
Episode reward 6457.841898651727
Episode 20 of 100
Episode reward 6454.746113788852
Episode 21 of 100
Episode reward 6459.430391735519
Episode 22 of 100
Episode reward 6461.13612037922
Episode 23 of 100
Episode reward 6457.99835796449
Episode 24 of 100
Episode reward 6488.059696887117
Episode 25 of 100
Episode reward 6478.891811867995
Episode 26 of 100
Episode reward 6473.943294765498
Episode 27 of 100
Episode reward 6520.523878806123
Episode 28 of 100
Episode reward 6477.974216692803
Episode 29 of 100
Episode reward 6491.039136767895
Episode 30 of 100
Episode reward 6454.400126715989
Episode 31 of 100
Episode reward 6483.2790515274055
Episode 32 of 100
Episode reward 6453.5264525481125
Episode 33 of 100
Episode reward 6468.25063769242
Episode 34 of 100
Episode reward 6487.030036192155
Episode 35 of 100
Episode reward 6470.722729809205
Episode 36 of 100
Episode reward 6434.073917280487
Episode 37 of 100
Episode reward 6452.46685065292
Episode 38 of 100
Episode reward 6491.13563203927
Episode 39 of 100
Episode reward 6471.310040523801
Episode 40 of 100
Episode reward 6449.679338721247
Episode 41 of 100
Episode reward 6438.832444901052
Episode 42 of 100
Episode reward 6455.997262008091
Episode 43 of 100
Episode reward 6478.144030511461
Episode 44 of 100
Episode reward 6458.471984953947
Episode 45 of 100
Episode reward 6463.2735553180155
Episode 46 of 100
Episode reward 6450.808032263873
Episode 47 of 100
Episode reward 6531.109843761016
Episode 48 of 100
Episode reward 6443.49724472755
Episode 49 of 100
Episode reward 6449.385437166058
Episode 50 of 100
Episode reward 6446.1552562184
Episode 51 of 100
Episode reward 6456.932919592272
Episode 52 of 100
Episode reward 6451.506429523891
Episode 53 of 100
Episode reward 6451.171175863464
Episode 54 of 100
Episode reward 6464.2547846375965
Episode 55 of 100
Episode reward 6444.918191703987
Episode 56 of 100
Episode reward 6456.130676717681
Episode 57 of 100
Episode reward 6450.913094650607
Episode 58 of 100
Episode reward 6456.999448606076
Episode 59 of 100
Episode reward 6449.540902303894
Episode 60 of 100
Episode reward 6460.865613318589
Episode 61 of 100
Episode reward 6570.364993876596
Episode 62 of 100
Episode reward 6471.8265365245215
Episode 63 of 100
Episode reward 6450.421085271664
Episode 64 of 100
Episode reward 6485.152550832912
Episode 65 of 100
Episode reward 6472.643675132201
Episode 66 of 100
Episode reward 6489.7362295462535
Episode 67 of 100
Episode reward 6423.4174493851
Episode 68 of 100
Episode reward 6514.635659072953
Episode 69 of 100
Episode reward 6477.469932325338
Episode 70 of 100
Episode reward 6459.1645193833665
Episode 71 of 100
Episode reward 6461.112353039997
Episode 72 of 100
Episode reward 6450.227876437484
Episode 73 of 100
Episode reward 6475.84093163456
Episode 74 of 100
Episode reward 6464.295979997814
Episode 75 of 100
Episode reward 6447.89552138415
Episode 76 of 100
Episode reward 6459.233582103466
Episode 77 of 100
Episode reward 6439.306088950851
Episode 78 of 100
Episode reward 6443.94758880319
Episode 79 of 100
Episode reward 6468.677047395509
Episode 80 of 100
Episode reward 6446.667972520414
Episode 81 of 100
Episode reward 6499.193679993485
Episode 82 of 100
Episode reward 6513.127586404025
Episode 83 of 100
Episode reward 6471.42395590374
Episode 84 of 100
Episode reward 6489.912888696567
Episode 85 of 100
Episode reward 6470.579770507194
Episode 86 of 100
Episode reward 6451.429812544289
Episode 87 of 100
Episode reward 6475.26079849445
Episode 88 of 100
Episode reward 6460.49196020606
Episode 89 of 100
Episode reward 6449.540208581222
Episode 90 of 100
Episode reward 6439.982160802234
Episode 91 of 100
Episode reward 6470.1878046639495
Episode 92 of 100
Episode reward 6474.634232724421
Episode 93 of 100
Episode reward 6488.928790540828
Episode 94 of 100
Episode reward 6452.801603541099
Episode 95 of 100
Episode reward 6454.474391006134
Episode 96 of 100
Episode reward 6452.947219543771
Episode 97 of 100
Episode reward 6457.363385006223
Episode 98 of 100
Episode reward 6439.255768331935
Episode 99 of 100
Episode reward 6463.7049193817575
Overall deltas mean: 0.8729 and std: 0.0156
Agent0 deltas mean: 0.8628 and std: 0.0067
Agent1 deltas mean: 0.8830 and std: 0.0155
/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/seaborn/
distributions.py:679: UserWarning: Passing a 2D dataset for a bivariate plot is
deprecated in favor of kdeplot(x, y), and it will cause an error in future
versions. Please update your code.
  warnings.warn(warn_msg, UserWarning)
```




Agent_0
Agent_1
70
60
50
40
30
20
10
0
0.84
0.86
0.88
0.90
0.92
0.94
0.96
0.98
Profit gains (deltas)


Profit gains (deltas)
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
Agent_0
Agent_1
0
200
400
600
800
1000
Timesteps


Profit gains (deltas)
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
Agent_0
Agent_1
0
20
40
60
80
100
Timesteps



```
Traceback (most recent call last):

  File "<ipython-input-1-cfd728b1e1ae>", line 1, in <module>
    runfile('/home/lorenzo/algorithmic-pricing/rollout/rollout.py', args='/home/
lorenzo/algorithmic-pricing/train_results/Azure_ApexDQN_Cont/
azure06_cont_DQN_res2/
APEX_MultiAgentFirmsPricingContinuous_0_2019-09-06_10-17-13df1x7oyx/
checkpoint_740/checkpoint-740 --run APEX --env env_cont', wdir='/home/lorenzo/
algorithmic-pricing/rollout')

  File "/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
spyder_kernels/customize/spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  File "/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/
spyder_kernels/customize/spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "/home/lorenzo/algorithmic-pricing/rollout/rollout.py", line 404, in
<module>
    Deltas_df = pd.DataFrame(d_array)

  File "/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/pandas/core/
frame.py", line 440, in __init__
    mgr = init_ndarray(data, index, columns, dtype=dtype, copy=copy)

  File "/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/pandas/core/
internals/construction.py", line 171, in init_ndarray
    values = prep_ndarray(values, copy=copy)

  File "/home/lorenzo/anaconda3/envs/py36/lib/python3.6/site-packages/pandas/core/
internals/construction.py", line 295, in prep_ndarray
    raise ValueError("Must pass 2-d input")

ValueError: Must pass 2-d input


In [2]:

In [2]:
```