Deep Deterministic Policy Gradient (DDPG)

```
!apt-get update && apt-get install -y xvfb
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link
```

```
!pip install swig
```

```
Collecting swig
  Downloading swig-4.2.1.post0-py2.py3-none-manylinux_2_5_x86_64.manylinux1_x86_64.wh
Downloading swig-4.2.1.post0-py2.py3-none-manylinux_2_5_x86_64.manylinux1_x86_64.whl
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 22.5 MB/s eta 0:00:00
Installing collected packages: swig
Successfully installed swig-4.2.1.post0
```

```
!pip install gym[box2d]==0.23.1
```

```
Collecting gym==0.23.1 (from gym[box2d]==0.23.1)
  Downloading gym-0.23.1.tar.gz (626 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 626.2/626.2 kB 13.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: numpy>=1.18.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: cloudpickle>=1.2.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: gym-notices>=0.0.4 in /usr/local/lib/python3.10/dist-p
Collecting box2d-py==2.3.5 (from gym[box2d]==0.23.1)
  Downloading box2d-py-2.3.5.tar.gz (374 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 374.4/374.4 kB 28.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting pygame==2.1.0 (from gym[box2d]==0.23.1)
  Downloading pygame-2.1.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Downloading pygame-2.1.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18.3/18.3 MB 59.6 MB/s eta 0:00:00
Building wheels for collected packages: gym, box2d-py
  Building wheel for gym (pyproject.toml) ... done
  Created wheel for gym: filename=gym-0.23.1-py3-none-any.whl size=701343 sha256=45cd
  Stored in directory: /root/.cache/pip/wheels/1a/00/fb/fe5cf2860fb9b7bc860e28f00095a
  Building wheel for box2d-py (setup.py) ... done
  Created wheel for box2d-py: filename=box2d_py-2.3.5-cp310-cp310-linux_x86_64.whl si
  Stored in directory: /root/.cache/pip/wheels/db/8f/6a/eaaadf056fba10a98d986f6dce954
Successfully built gym box2d-py
Installing collected packages: box2d-py, pygame, gym
  Attempting uninstall: pygame
    Found existing installation: pygame 2.6.1
    Uninstalling pygame-2.6.1:
      Successfully uninstalled pygame-2.6.1
  Attempting uninstall: gym
    Found existing installation: gym 0.25.2
    Uninstalling gym-0.25.2:
      Successfully uninstalled gym-0.25.2
Successfully installed box2d-py-2.3.5 gym-0.23.1 pygame-2.1.0
```

```
!pip install pytorch_lightning
```

```
Collecting pytorch_lightning
  Downloading pytorch_lightning-2.4.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: torch>=2.1.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: tqdm>=4.57.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: PyYAML>=5.4 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: fsspec>=2022.5.0 in /usr/local/lib/python3.10/dist-pac
Collecting torchmetrics>=0.7.0 (from pytorch_lightning)
  Downloading torchmetrics-1.5.1-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: typing-extensions>=4.4.0 in /usr/local/lib/python3.10/
Collecting lightning-utilities>=0.10.0 (from pytorch_lightning)
  Downloading lightning_utilities-0.11.8-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: numpy<2.0,>1.20.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: yarl<2.0,>=1.12.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-pac
Downloading pytorch_lightning-2.4.0-py3-none-any.whl (815 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 815.2/815.2 kB 21.3 MB/s eta 0:00:00
Downloading lightning_utilities-0.11.8-py3-none-any.whl (26 kB)
Downloading torchmetrics-1.5.1-py3-none-any.whl (890 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 890.6/890.6 kB 54.1 MB/s eta 0:00:00
Installing collected packages: lightning-utilities, torchmetrics, pytorch_lightning
Successfully installed lightning-utilities-0.11.8 pytorch_lightning-2.4.0 torchmetric
```

```
!pip install pyvirtualdisplay
```

```
Collecting pyvirtualdisplay
  Downloading PyVirtualDisplay-3.0-py3-none-any.whl.metadata (943 bytes)
Downloading PyVirtualDisplay-3.0-py3-none-any.whl (15 kB)
Installing collected packages: pyvirtualdisplay
Successfully installed pyvirtualdisplay-3.0
```

```
!pip install brax==0.10.5
```

```
Requirement already satisfied: tensorstore in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: rich>=11.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: PyYAML>=5.4.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: cloudpickle>=1.2.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: gym-notices>=0.0.4 in /usr/local/lib/python3.10/dis
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (fro
Collecting contextlib2 (from ml-collections->brax==0.10.5)
  Downloading contextlib2-21.6.0-py2.py3-none-any.whl.metadata (4.1 kB)
Collecting glfw (from mujoco->brax==0.10.5)
  Downloading glfw-2.7.0-py2.py27.py3.py30.py31.py32.py33.py34.py35.py36.py37.py38
Requirement already satisfied: pyopengl in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: chex>=0.1.86 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: humanize in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: toolz>=0.9.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.1
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: importlib_resources in /usr/local/lib/python3.10/di
Requirement already satisfied: zipp in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packag
Downloading brax-0.10.5-py3-none-any.whl (998 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 998.9/998.9 kB 14.2 MB/s eta 0:00:00
Downloading dm_env-1.6-py3-none-any.whl (26 kB)
Downloading Flask_Cors-5.0.0-py2.py3-none-any.whl (14 kB)
Downloading jaxopt-0.8.3-py3-none-any.whl (172 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 172.3/172.3 kB 12.2 MB/s eta 0:00:00
Downloading mujoco-3.2.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.3/6.3 MB 74.5 MB/s eta 0:00:00
Downloading mujoco_mjx-3.2.5-py3-none-any.whl (6.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.7/6.7 MB 64.2 MB/s eta 0:00:00
Downloading pytinyrenderer-0.0.14-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 54.0 MB/s eta 0:00:00
Downloading tensorboardX-2.6.2.2-py2.py3-none-any.whl (101 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.7/101.7 kB 6.8 MB/s eta 0:00:00
Downloading trimesh-4.5.2-py3-none-any.whl (704 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 704.4/704.4 kB 20.0 MB/s eta 0:00:00
Downloading contextlib2-21.6.0-py2.py3-none-any.whl (13 kB)
Downloading glfw-2.7.0-py2.py27.py3.py30.py31.py32.py33.py34.py35.py36.py37.py38-n
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.8/211.8 kB 13.7 MB/s eta 0:00:00
Building wheels for collected packages: ml-collections
  Building wheel for ml-collections (setup.py) ... done
  Created wheel for ml-collections: filename=ml_collections-0.1.1-py3-none-any.whl
  Stored in directory: /root/.cache/pip/wheels/7b/89/c9/a9b87790789e94aadcfc393c28
Successfully built ml-collections
Installing collected packages: pytinyrenderer, glfw, trimesh, tensorboardX, dm-env
Successfully installed brax-0.10.5 contextlib2-21.6.0 dm-env-1.6 flask-cors-5.0.0
```

Setup virtual display

```
from pyvirtualdisplay import Display
Display(visible=False, size=(1400, 900)).start()
```

```
<pyvirtualdisplay.display.Display at 0x7c7d486b56c0>
```

Import the necessary code libraries

```
import copy
import gym
import torch
import random
import functools

import numpy as np
import torch.nn.functional as F

from collections import deque, namedtuple
from IPython.display import HTML
from base64 import b64encode

from torch import nn
from torch.utils.data import DataLoader
from torch.utils.data.dataset import IterableDataset
from torch.optim import AdamW

from pytorch_lightning import LightningModule, Trainer

import brax
from brax import envs
from brax.envs.wrappers import gym as gym_wrapper
from brax.envs.wrappers import torch as torch_wrapper
from brax.io import html

device = 'cuda' if torch.cuda.is_available() else 'cpu'
num_gpus = torch.cuda.device_count()
```

```
def display_video(episode=0):
  video_file = open(f'/content/videos/rl-video-episode-{episode}.mp4', "r+b").read()
  video_url = f"data:video/mp4; base64,{b64encode(video_file).decode()}"
  return HTML(f"<video width=600 controls><source src='{video_url}'></video>")
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning
  and should_run_async(code)
```

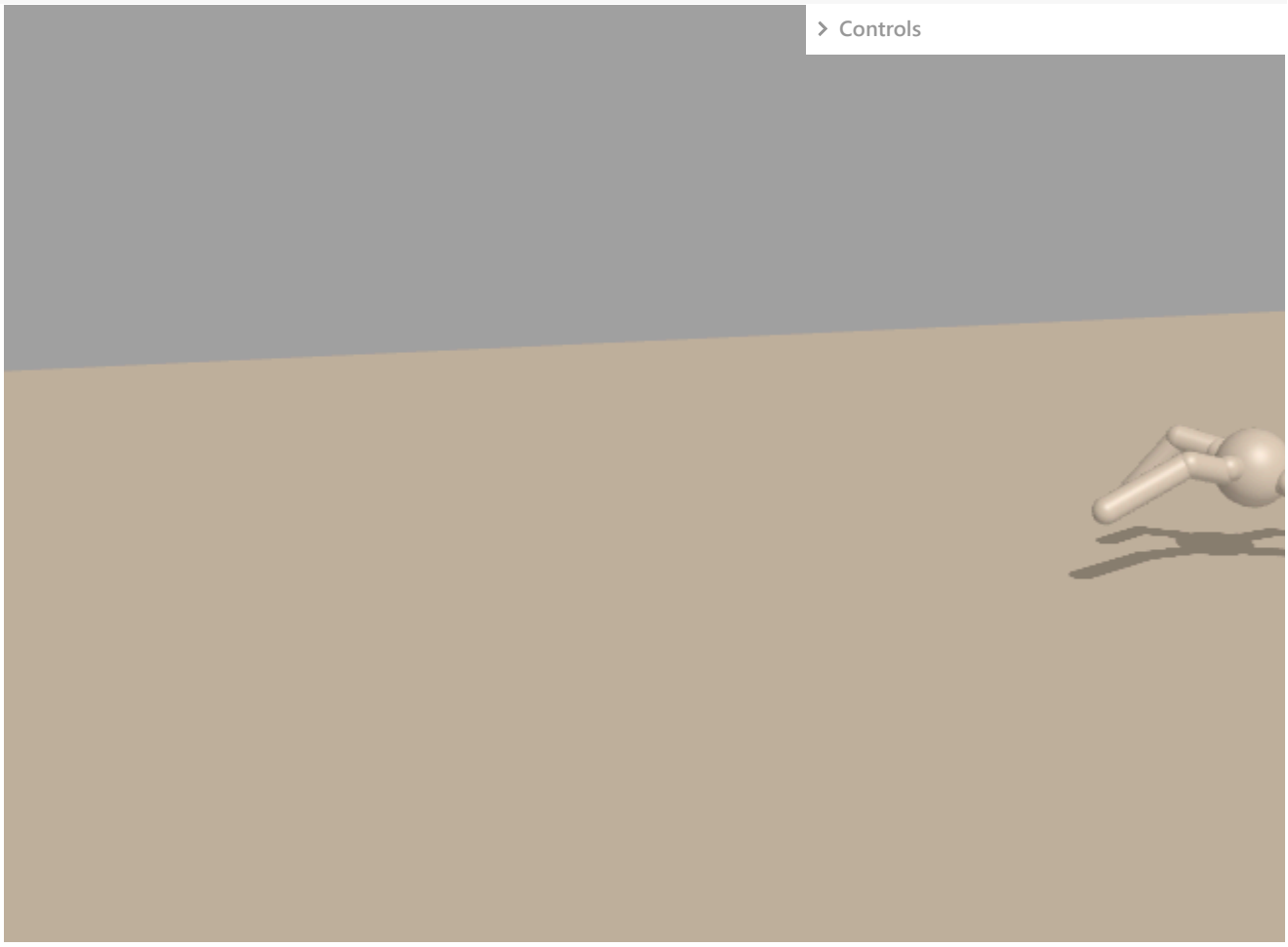
```
def create_environment(env_name, num_envs=256, episode_length=1000):
    env = envs.create(env_name, batch_size=num_envs, episode_length=episode_length, backe
    env = gym_wrapper.VectorGymWrapper(env)
    env = torch_wrapper.TorchWrapper(env, device=device)
    return env
```

```
import pytorch_lightning as pl
import warnings
warnings.filterwarnings('ignore')

@torch.no_grad()
```

```
def test_env(env_name, policy=None):
  env = envs.create(env_name, episode_length=1000, backend='spring')
  env = gym_wrapper.GymWrapper(env)
  env = torch_wrapper.TorchWrapper(env, device=device)
  ps_array = []
  state = env.reset()
  for i in range(1000):
    if policy:
      action = algo.policy.net(state.unsqueeze(0)).squeeze()
    else:
      action = torch.from_numpy(env.action_space.sample()).to(device)
    state, _, _, _ = env.step(action)
    ps_array.extend([env.unwrapped._state.pipeline_state]*5)
  return HTML(html.render(env.unwrapped._env.sys, ps_array))
```

```
test_env('ant')
```



Create the gradient policy

```
class GradientPolicy(nn.Module):

  def __init__(self, hidden_size, obs_size, out_dims, min, max):
    super().__init__()
    self.min = torch.from_numpy(min).to(device)
    self.max = torch.from_numpy(max).to(device)
    self.net = nn.Sequential(
```

```
        nn.Linear(obs_size, hidden_size),
        nn.ReLU(),
        nn.Linear(hidden_size, hidden_size),
        nn.ReLU(),
        nn.Linear(hidden_size, out_dims),
        nn.Tanh()
    )

  def mu(self, x):
    if isinstance(x, np.ndarray):
      x = torch.from_numpy(x).to(device)
    return self.net(x.float()) * self.max

  def forward(self, x, epsilon=0.0):
    mu = self.mu(x)
    mu = mu + torch.normal(0, epsilon, mu.size(), device=mu.device)
    action = torch.max(torch.min(mu, self.max), self.min)
    return action
```

Create the Deep Q-Network

```
class DQN(nn.Module):

  def __init__(self, hidden_size, obs_size, out_dims):
    super().__init__()
    self.net = nn.Sequential(
        nn.Linear(obs_size + out_dims, hidden_size),
        nn.ReLU(),
        nn.Linear(hidden_size, hidden_size),
        nn.ReLU(),
        nn.Linear(hidden_size, 1),
    )

  def forward(self, state, action):
    if isinstance(state, np.ndarray):
      state = torch.from_numpy(state).to(device)
    if isinstance(action, np.ndarray):
      action = torch.from_numpy(action).to(device)
    in_vector = torch.hstack((state, action))
    return self.net(in_vector.float())
```

```
class ReplayBuffer:

  def __init__(self, capacity):
    self.buffer = deque(maxlen=capacity)

  def __len__(self):
    return len(self.buffer)

  def append(self, experience):
```

```
    self.buffer.append(experience)

  def sample(self, batch_size):
    return random.sample(self.buffer, batch_size)
```

```
class RLDataset(IterableDataset):

  def __init__(self, buffer, sample_size=400):
    self.buffer = buffer
    self.sample_size = sample_size

  def __iter__(self):
    for experience in self.buffer.sample(self.sample_size):
      yield experience
```

```
def polyak_average(net, target_net, tau=0.01):
    for qp, tp in zip(net.parameters(), target_net.parameters()):
        tp.data.copy_(tau * qp.data + (1 - tau) * tp.data)
```

```
class DDPG(LightningModule):

  def __init__(self, env_name, capacity=500, batch_size=8192, actor_lr=1e-3,
               critic_lr=1e-3, hidden_size=256, gamma=0.99, loss_fn=F.smooth_l1_loss,
               optim=AdamW, eps_start=1.0, eps_end=0.2, eps_last_episode=500,
               samples_per_epoch=10, tau=0.005):

    super().__init__()

    self.env = create_environment(env_name, num_envs=batch_size)
    self.obs = self.env.reset()
    self.videos = []

    obs_size = self.env.observation_space.shape[1]
    action_dims = self.env.action_space.shape[1]
    max_action = self.env.action_space.high
    min_action = self.env.action_space.low

    self.q_net = DQN(hidden_size, obs_size, action_dims)
    self.policy = GradientPolicy(hidden_size, obs_size, action_dims, min_action, max_acti

    self.target_policy = copy.deepcopy(self.policy)
    self.target_q_net = copy.deepcopy(self.q_net)

    self.buffer = ReplayBuffer(capacity=capacity)

    self.save_hyperparameters()

    self.automatic_optimization = False

    while len(self.buffer) < self.hparams.samples_per_epoch:
      print(f"{len(self.buffer)} samples in experience buffer. Filling...")
      self.play(epsilon=self.hparams.eps_start)
```

```
  @torch.no_grad()
  def play(self, policy=None, epsilon=0.):
    if policy:
      action = policy(self.obs, epsilon=epsilon)
    else:
        action = torch.from_numpy(self.env.action_space.sample()).to(device)
    next_obs, reward, done, info = self.env.step(action)
    exp = (self.obs, action, reward, done, next_obs)
    self.buffer.append(exp)
    self.obs = next_obs
    return reward.mean()

  def forward(self, x):
    output = self.policy(x)
    return output

  def configure_optimizers(self):
    q_net_optimizer = self.hparams.optim(self.q_net.parameters(), lr=self.hparams.critic_
    policy_optimizer = self.hparams.optim(self.policy.parameters(), lr=self.hparams.actor
    return [q_net_optimizer, policy_optimizer]

  def train_dataloader(self):
    dataset = RLDataset(self.buffer, self.hparams.samples_per_epoch)
    dataloader = DataLoader(
        dataset=dataset,
        batch_size=1,
    )
    return dataloader

  def training_step(self, batch, batch_idx):

    epsilon = max(
        self.hparams.eps_end,
        self.hparams.eps_start - self.current_epoch / self.hparams.eps_last_episode
    )

    mean_reward = self.play(policy=self.policy, epsilon=epsilon)
    self.log("episode/mean_reward", mean_reward)

    states, actions, rewards, dones, next_states = map(torch.squeeze, batch)
    rewards = rewards.unsqueeze(1)
    dones = dones.unsqueeze(1).bool()

    polyak_average(self.q_net, self.target_q_net, tau=self.hparams.tau)
    polyak_average(self.policy, self.target_policy, tau=self.hparams.tau)

    # Manual Optimization:
    q_net_optimizer, policy_optimizer = self.optimizers() # Access optimizers

    # Update Q-Network
    q_net_optimizer.zero_grad() # Clear gradients for Q-Network
    state_action_values = self.q_net(states, actions)
    next_state_values = self.target_q_net(next_states, self.target_policy.mu(next_states)
    next_state_values[dones] = 0.0
    expected_state_action_values = rewards + self.hparams.gamma * next_state_values
```

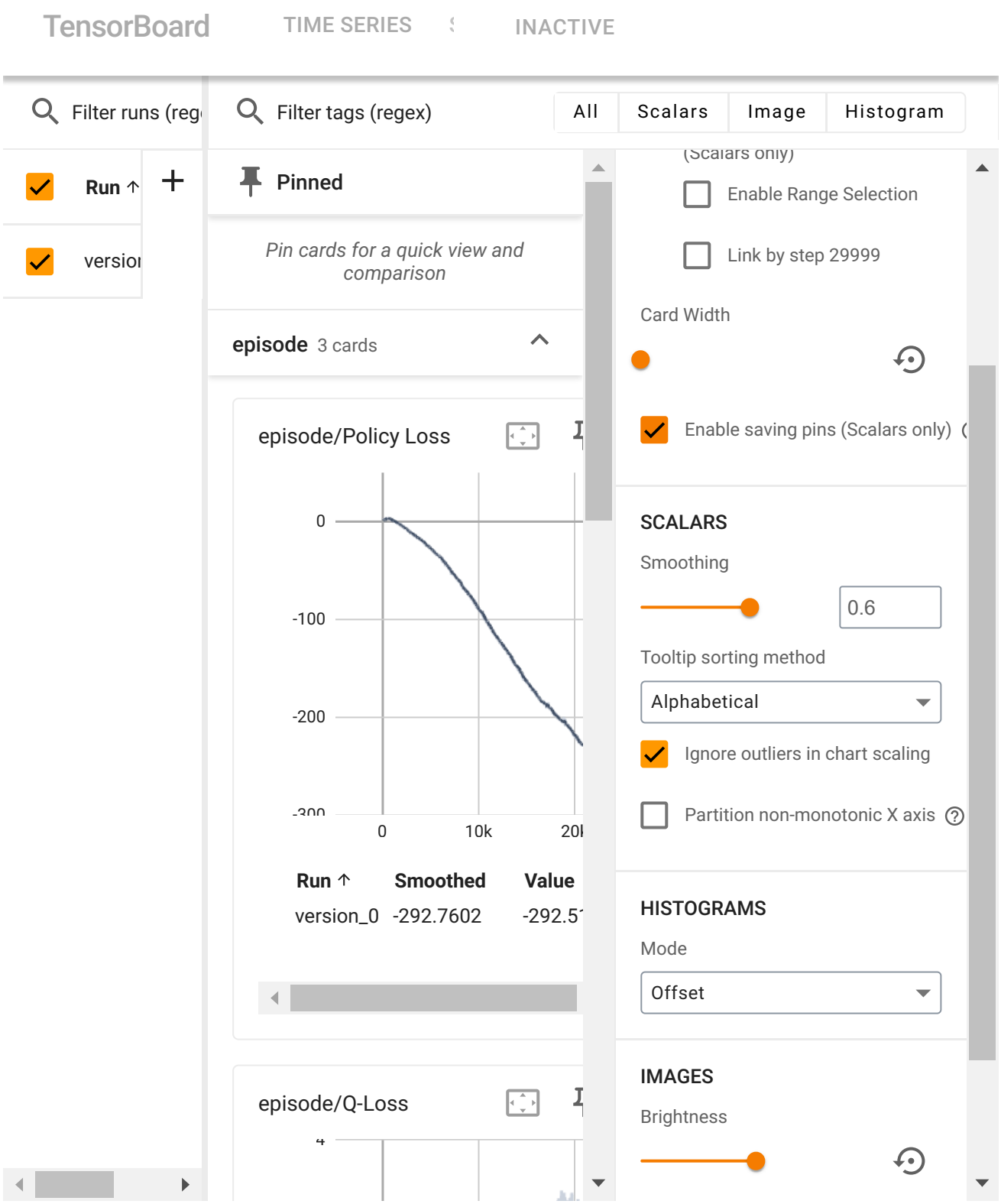
```
    q_loss = self.hparams.loss_fn(state_action_values, expected_state_action_values)
    self.manual_backward(q_loss)  # Manually compute gradients
    q_net_optimizer.step() # Update Q-Network parameters
    self.log_dict({"episode/Q-Loss": q_loss})

    # Update Policy
    policy_optimizer.zero_grad()  # Clear gradients for Policy
    mu = self.policy.mu(states)
    policy_loss = - self.q_net(states, mu).mean()
    self.manual_backward(policy_loss) # Manually compute gradients
    policy_optimizer.step() # Update policy parameters
    self.log_dict({"episode/Policy Loss": policy_loss})

  def on_train_epoch_end(self):
    if self.current_epoch % 1000 == 0:
      video = test_env('ant', policy=algo.policy)
      self.videos.append(video)
```

```
# Start tensorboard.
!rm -r /content/lightning_logs/
!rm -r /content/videos/
%load_ext tensorboard
%tensorboard --logdir /content/lightning_logs/
```

```
rm: cannot remove '/content/lightning_logs/': No such file or directory
rm: cannot remove '/content/videos/': No such file or directory
```




```
algo = DDPG('ant')

trainer = pl.Trainer(
    accelerator="gpu" if num_gpus else "cpu", # Use 'gpu' if num_gpus is greater than 0,
    devices=1, # Specify the number of GPUs or 'auto' for automatic detection
    max_epochs=3000,
    log_every_n_steps=10
)
```

```
0 samples in experience buffer. Filling...
1 samples in experience buffer. Filling...
2 samples in experience buffer. Filling...
3 samples in experience buffer. Filling...
4 samples in experience buffer. Filling...
5 samples in experience buffer. Filling...
6 samples in experience buffer. Filling...
7 samples in experience buffer. Filling...
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
8 samples in experience buffer. Filling...
9 samples in experience buffer. Filling...
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name          | Type           | Params | Mode
-----------------------------------------------------------
0 | q_net         | DQN            | 75.3 K | train
1 | policy        | GradientPolicy | 75.0 K | train
2 | target_policy | GradientPolicy | 75.0 K | train
3 | target_q_net  | DQN            | 75.3 K | train
-----------------------------------------------------------
300 K     Trainable params
0         Non-trainable params
300 K     Total params
1.202     Total estimated model params size (MB)
30        Modules in train mode
0         Modules in eval mode
```

Epoch 2999: 0/? [00:00<?, ?it/s, v_num=0]

```
algo.videos[1]
```