

```

!pip install gymnasium[atari]
!pip install gymnasium[accept-rom-license]
!pip install stable_baselines3

Collecting gymnasium[atari]
  Downloading gymnasium-0.29.1-py3-none-any.whl (953 kB)
  _____ 0.0/953.9 kB ? eta -:-:--
  _____ 225.3/953.9 kB 6.7 MB/s eta
0:00:01 _____ 953.9/953.9 kB 16.9
MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.21.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[atari])
(1.25.2)
Requirement already satisfied: cloudpickle>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[atari])
(2.2.1)
Requirement already satisfied: typing-extensions>=4.3.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[atari])
(4.11.0)
Collecting farama-notifications>=0.0.1 (from gymnasium[atari])
  Downloading Farama_Notifications-0.0.4-py3-none-any.whl (2.5 kB)
Collecting shimmy[atari]<1.0,>=0.1.0 (from gymnasium[atari])
  Downloading Shimmy-0.2.1-py3-none-any.whl (25 kB)
Collecting ale-py~0.8.1 (from shimmy[atari]<1.0,>=0.1.0-
>gymnasium[atari])
  Downloading ale_py-0.8.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
  _____ 1.7/1.7 MB 68.1 MB/s eta
0:00:00
Requirement already satisfied: importlib-resources in
/usr/local/lib/python3.10/dist-packages (from ale-py~0.8.1-
>shimmy[atari]<1.0,>=0.1.0->gymnasium[atari]) (6.4.0)
Installing collected packages: farama-notifications, gymnasium, ale-
py, shimmy
Successfully installed ale-py-0.8.1 farama-notifications-0.0.4
gymnasium-0.29.1 shimmy-0.2.1
Requirement already satisfied: gymnasium[accept-rom-license] in
/usr/local/lib/python3.10/dist-packages (0.29.1)
Requirement already satisfied: numpy>=1.21.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[accept-rom-
license]) (1.25.2)
Requirement already satisfied: cloudpickle>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[accept-rom-
license]) (2.2.1)
Requirement already satisfied: typing-extensions>=4.3.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[accept-rom-
license]) (4.11.0)
Requirement already satisfied: farama-notifications>=0.0.1 in
/usr/local/lib/python3.10/dist-packages (from gymnasium[accept-rom-
license]) (0.0.4)

```

```
Collecting autorom[accept-rom-license]~=0.4.2 (from gymnasium[accept-rom-license])
  Downloading AutoROM-0.4.2-py3-none-any.whl (16 kB)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license]) (8.1.7)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license]) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license]) (4.66.2)
Collecting AutoROM.accept-rom-license (from autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license])
  Downloading AutoROM.accept-rom-license-0.6.1.tar.gz (434 kB)


---


434.7/434.7 kB 9.1 MB/s eta
0:00:00
ents to build wheel ... etadata (pyproject.toml) ... ent already
satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-
packages (from requests->autorom[accept-rom-license]~=0.4.2-
>gymnasium[accept-rom-license]) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license])
(3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license])
(2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>autorom[accept-rom-license]~=0.4.2->gymnasium[accept-rom-license])
(2024.2.2)
Building wheels for collected packages: AutoROM.accept-rom-license
  Building wheel for AutoROM.accept-rom-license (pyproject.toml) ... -
license: filename=AutoROM.accept_rom_license-0.6.1-py3-none-any.whl
size=446659
sha256=cf5f01ac6cb0276ca22ee6b7225e6b86f213ddf30600f7937a67e4abbe8937a
3
  Stored in directory:
/root/.cache/pip/wheels/6b/1b/ef/a43ff1a2f1736d5711faalba4c1f61be1131b
8899e6a057811
Successfully built AutoROM.accept-rom-license
Installing collected packages: AutoROM.accept-rom-license, autorom
Successfully installed AutoROM.accept-rom-license-0.6.1 autorom-0.4.2
Collecting stable_baselines3
  Downloading stable_baselines3-2.3.2-py3-none-any.whl (182 kB)


---


182.3/182.3 kB 4.7 MB/s eta
0:00:00
```

```
ent already satisfied: gymnasium<0.30,>=0.28.1 in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(0.29.1)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(1.25.2)
Requirement already satisfied: torch>=1.13 in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(2.2.1+cu121)
Requirement already satisfied: cloudpickle in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(2.2.1)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(2.0.3)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from stable_baselines3)
(3.7.1)
Requirement already satisfied: typing-extensions>=4.3.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium<0.30,>=0.28.1-
>stable_baselines3) (4.11.0)
Requirement already satisfied: farama-notifications>=0.0.1 in
/usr/local/lib/python3.10/dist-packages (from gymnasium<0.30,>=0.28.1-
>stable_baselines3) (0.0.4)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (3.14.0)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (1.12)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (3.3)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (3.1.3)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (2023.6.0)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.13-
>stable_baselines3)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-
manylinux1_x86_64.whl (23.7 MB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.13-
>stable_baselines3)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-
manylinux1_x86_64.whl (823 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.13-
>stable_baselines3)
```

```
Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-
manylinux1_x86_64.whl (14.1 MB)
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-
manylinux1_x86_64.whl (731.7 MB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_cublas_cu12-12.1.3.1-py3-none-
manylinux1_x86_64.whl (410.6 MB)
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_cufft_cu12-11.0.2.54-py3-none-
manylinux1_x86_64.whl (121.6 MB)
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_curand_cu12-10.3.2.106-py3-none-
manylinux1_x86_64.whl (56.5 MB)
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_cusolver_cu12-11.4.5.107-py3-none-
manylinux1_x86_64.whl (124.2 MB)
Collecting nvidia-cuspars-cu12==12.1.0.106 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_cuspars-cu12-12.1.0.106-py3-none-
manylinux1_x86_64.whl (196.0 MB)
Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl
(166.0 MB)
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.13-
>stable_baselines3)
Using cached nvidia_nvtx_cu12-12.1.105-py3-none-
manylinux1_x86_64.whl (99 kB)
Requirement already satisfied: triton==2.2.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.13-
>stable_baselines3) (2.2.0)
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-
cu12==11.4.5.107->torch>=1.13->stable_baselines3)
Using cached nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (21.1 MB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (1.2.1)
Requirement already satisfied: cycycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
```

```

>stable_baselines3) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (24.0)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>stable_baselines3) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas-
>stable_baselines3) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/dist-packages (from pandas-
>stable_baselines3) (2024.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib->stable_baselines3) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.13-
>stable_baselines3) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.13-
>stable_baselines3) (1.3.0)
Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-
cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-
cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12,
nvidia-cublas-cu12, nvidia-cusparse-cu12, nvidia-cudnn-cu12, nvidia-
cusolver-cu12, stable_baselines3
Successfully installed nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-
cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-
cu12-12.1.105 nvidia-cudnn-cu12-8.9.2.26 nvidia-cufft-cu12-11.0.2.54
nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-11.4.5.107 nvidia-
cusparse-cu12-12.1.0.106 nvidia-nccl-cu12-2.19.3 nvidia-nvjitlink-
cu12-12.4.127 nvidia-nvtx-cu12-12.1.105 stable_baselines3-2.3.2

```

```

import gymnasium as gym
import seaborn as sns
import os
from collections import deque, Counter, namedtuple, defaultdict
import random
from matplotlib import pyplot as plt
import warnings

```

```

warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)
import torch
from torch import nn
from torch.nn import init
import torch.nn.functional as F
from torch.distributions import Categorical
import math
from itertools import count
from tqdm import tqdm
import numpy as np
import time
import uuid

from stable_baselines3.common.atari_wrappers import ClipRewardEnv,
FireResetEnv, MaxAndSkipEnv, NoopResetEnv

import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

ENV_ARGS = {
    'id': "BreakoutNoFrameskip-v4"
}
NUM_ENVS = 3
SEED = 1
LR = 3e-4
NUM_STEPS = 2048
NUM_ITERATIONS = 4000
GAMMA = 0.99
GAE_LAMBDA = 0.95
UPDATE_EPOCHS = 10
CLIP_COEF = 0.2 # the epsilon in KL divergece in PPO paper
ENTROPY_COEF = 0.0
VF_COEF = 0.5
MAX_GRAD_NORM = 0.5
MINI_BATCH_COUNT = 64
UPDATE_PLOTS = 10
DEVICE = 'cuda' if torch.cuda.is_available() else 'cpu'
print('device = ', DEVICE)

#output directory
ROOT = os.getcwd()
OUTPUT = os.path.join(ROOT, 'output')

if os.path.exists(OUTPUT) == False:
    os.makedirs(OUTPUT)

#seeding
random.seed(SEED)

```

```
np.random.seed(SEED)
torch.manual_seed(SEED)

device = cuda

<torch._C.Generator at 0x7ed5549c5d10>

gym.envs.registration.registry.keys()

dict_keys(['CartPole-v0', 'CartPole-v1', 'MountainCar-v0',
'MountainCarContinuous-v0', 'Pendulum-v1', 'Acrobot-v1',
'phys2d/CartPole-v0', 'phys2d/CartPole-v1', 'phys2d/Pendulum-v0',
'LunarLander-v2', 'LunarLanderContinuous-v2', 'BipedalWalker-v3',
'BipedalWalkerHardcore-v3', 'CarRacing-v2', 'Blackjack-v1',
'FrozenLake-v1', 'FrozenLake8x8-v1', 'CliffWalking-v0', 'Taxi-v3',
'tabular/Blackjack-v0', 'tabular/CliffWalking-v0', 'Reacher-v2',
'Reacher-v4', 'Pusher-v2', 'Pusher-v4', 'InvertedPendulum-v2',
'InvertedPendulum-v4', 'InvertedDoublePendulum-v2',
'InvertedDoublePendulum-v4', 'HalfCheetah-v2', 'HalfCheetah-v3',
'HalfCheetah-v4', 'Hopper-v2', 'Hopper-v3', 'Hopper-v4', 'Swimmer-v2',
'Swimmer-v3', 'Swimmer-v4', 'Walker2d-v2', 'Walker2d-v3', 'Walker2d-
v4', 'Ant-v2', 'Ant-v3', 'Ant-v4', 'Humanoid-v2', 'Humanoid-v3',
'Humanoid-v4', 'HumanoidStandup-v2', 'HumanoidStandup-v4',
'GymV26Environment-v0', 'GymV21Environment-v0', 'Adventure-v0',
'AdventureDeterministic-v0', 'AdventureNoFrameskip-v0', 'Adventure-
v4', 'AdventureDeterministic-v4', 'AdventureNoFrameskip-v4',
'Adventure-ram-v0', 'Adventure-ramDeterministic-v0', 'Adventure-
ramNoFrameskip-v0', 'Adventure-ram-v4', 'Adventure-ramDeterministic-
v4', 'Adventure-ramNoFrameskip-v4', 'AirRaid-v0',
'AirRaidDeterministic-v0', 'AirRaidNoFrameskip-v0', 'AirRaid-v4',
'AirRaidDeterministic-v4', 'AirRaidNoFrameskip-v4', 'AirRaid-ram-v0',
'AirRaid-ramDeterministic-v0', 'AirRaid-ramNoFrameskip-v0', 'AirRaid-
ram-v4', 'AirRaid-ramDeterministic-v4', 'AirRaid-ramNoFrameskip-v4',
'Alien-v0', 'AlienDeterministic-v0', 'AlienNoFrameskip-v0', 'Alien-
v4', 'AlienDeterministic-v4', 'AlienNoFrameskip-v4', 'Alien-ram-v0',
'Alien-ramDeterministic-v0', 'Alien-ramNoFrameskip-v0', 'Alien-ram-
v4', 'Alien-ramDeterministic-v4', 'Alien-ramNoFrameskip-v4', 'Amidar-
v0', 'AmidarDeterministic-v0', 'AmidarNoFrameskip-v0', 'Amidar-v4',
'AmidarDeterministic-v4', 'AmidarNoFrameskip-v4', 'Amidar-ram-v0',
'Amidar-ramDeterministic-v0', 'Amidar-ramNoFrameskip-v0', 'Amidar-ram-
v4', 'Amidar-ramDeterministic-v4', 'Amidar-ramNoFrameskip-v4',
'Assault-v0', 'AssaultDeterministic-v0', 'AssaultNoFrameskip-v0',
'Assault-v4', 'AssaultDeterministic-v4', 'AssaultNoFrameskip-v4',
'Assault-ram-v0', 'Assault-ramDeterministic-v0', 'Assault-
ramNoFrameskip-v0', 'Assault-ram-v4', 'Assault-ramDeterministic-v4',
'Assault-ramNoFrameskip-v4', 'Asterix-v0', 'AsterixDeterministic-v0',
'AsterixNoFrameskip-v0', 'Asterix-v4', 'AsterixDeterministic-v4',
'AsterixNoFrameskip-v4', 'Asterix-ram-v0', 'Asterix-ramDeterministic-
v0', 'Asterix-ramNoFrameskip-v0', 'Asterix-ram-v4', 'Asterix-
ramDeterministic-v4', 'Asterix-ramNoFrameskip-v4', 'Asteroids-v0',
```

'AsteroidsDeterministic-v0', 'AsteroidsNoFrameskip-v0', 'Asteroids-v4', 'AsteroidsDeterministic-v4', 'AsteroidsNoFrameskip-v4', 'Asteroids-ram-v0', 'Asteroids-ramDeterministic-v0', 'Asteroids-ramNoFrameskip-v0', 'Asteroids-ram-v4', 'Asteroids-ramDeterministic-v4', 'Asteroids-ramNoFrameskip-v4', 'Atlantis-v0', 'AtlantisDeterministic-v0', 'AtlantisNoFrameskip-v0', 'Atlantis-v4', 'AtlantisDeterministic-v4', 'AtlantisNoFrameskip-v4', 'Atlantis-ram-v0', 'Atlantis-ramDeterministic-v0', 'Atlantis-ramNoFrameskip-v0', 'Atlantis-ram-v4', 'Atlantis-ramDeterministic-v4', 'Atlantis-ramNoFrameskip-v4', 'BankHeist-v0', 'BankHeistDeterministic-v0', 'BankHeistNoFrameskip-v0', 'BankHeist-v4', 'BankHeistDeterministic-v4', 'BankHeistNoFrameskip-v4', 'BankHeist-ram-v0', 'BankHeist-ramDeterministic-v0', 'BankHeist-ramNoFrameskip-v0', 'BankHeist-ram-v4', 'BankHeist-ramDeterministic-v4', 'BankHeist-ramNoFrameskip-v4', 'BattleZone-v0', 'BattleZoneDeterministic-v0', 'BattleZoneNoFrameskip-v0', 'BattleZone-v4', 'BattleZoneDeterministic-v4', 'BattleZoneNoFrameskip-v4', 'BattleZone-ram-v0', 'BattleZone-ramDeterministic-v0', 'BattleZone-ramNoFrameskip-v0', 'BattleZone-ram-v4', 'BattleZone-ramDeterministic-v4', 'BattleZone-ramNoFrameskip-v4', 'BeamRider-v0', 'BeamRiderDeterministic-v0', 'BeamRiderNoFrameskip-v0', 'BeamRider-v4', 'BeamRiderDeterministic-v4', 'BeamRiderNoFrameskip-v4', 'BeamRider-ram-v0', 'BeamRider-ramDeterministic-v0', 'BeamRider-ramNoFrameskip-v0', 'BeamRider-ram-v4', 'BeamRider-ramDeterministic-v4', 'BeamRider-ramNoFrameskip-v4', 'Berzerk-v0', 'BerzerkDeterministic-v0', 'BerzerkNoFrameskip-v0', 'Berzerk-v4', 'BerzerkDeterministic-v4', 'BerzerkNoFrameskip-v4', 'Berzerk-ram-v0', 'Berzerk-ramDeterministic-v0', 'Berzerk-ramNoFrameskip-v0', 'Berzerk-ram-v4', 'Berzerk-ramDeterministic-v4', 'Berzerk-ramNoFrameskip-v4', 'Bowling-v0', 'BowlingDeterministic-v0', 'BowlingNoFrameskip-v0', 'Bowling-v4', 'BowlingDeterministic-v4', 'BowlingNoFrameskip-v4', 'Bowling-ram-v0', 'Bowling-ramDeterministic-v0', 'Bowling-ramNoFrameskip-v0', 'Bowling-ram-v4', 'Bowling-ramDeterministic-v4', 'Bowling-ramNoFrameskip-v4', 'Boxing-v0', 'BoxingDeterministic-v0', 'BoxingNoFrameskip-v0', 'Boxing-v4', 'BoxingDeterministic-v4', 'BoxingNoFrameskip-v4', 'Boxing-ram-v0', 'Boxing-ramDeterministic-v0', 'Boxing-ramNoFrameskip-v0', 'Boxing-ram-v4', 'Boxing-ramDeterministic-v4', 'Boxing-ramNoFrameskip-v4', 'Breakout-v0', 'BreakoutDeterministic-v0', 'BreakoutNoFrameskip-v0', 'Breakout-v4', 'BreakoutDeterministic-v4', 'BreakoutNoFrameskip-v4', 'Breakout-ram-v0', 'Breakout-ramDeterministic-v0', 'Breakout-ramNoFrameskip-v0', 'Breakout-ram-v4', 'Breakout-ramDeterministic-v4', 'Breakout-ramNoFrameskip-v4', 'Carnival-v0', 'CarnivalDeterministic-v0', 'CarnivalNoFrameskip-v0', 'Carnival-v4', 'CarnivalDeterministic-v4', 'CarnivalNoFrameskip-v4', 'Carnival-ram-v0', 'Carnival-ramDeterministic-v0', 'Carnival-ramNoFrameskip-v0', 'Carnival-ram-v4', 'Carnival-ramDeterministic-v4', 'Carnival-ramNoFrameskip-v4', 'Centipede-v0', 'CentipedeDeterministic-v0', 'CentipedeNoFrameskip-v0', 'Centipede-v4', 'CentipedeDeterministic-v4', 'CentipedeNoFrameskip-v4', 'Centipede-ram-v0', 'Centipede-

ramDeterministic-v0', 'Centipede-ramNoFrameskip-v0', 'Centipede-ram-v4', 'Centipede-ramDeterministic-v4', 'Centipede-ramNoFrameskip-v4', 'ChopperCommand-v0', 'ChopperCommandDeterministic-v0', 'ChopperCommandNoFrameskip-v0', 'ChopperCommand-v4', 'ChopperCommandDeterministic-v4', 'ChopperCommandNoFrameskip-v4', 'ChopperCommand-ram-v0', 'ChopperCommand-ramDeterministic-v0', 'ChopperCommand-ramNoFrameskip-v0', 'ChopperCommand-ram-v4', 'ChopperCommand-ramDeterministic-v4', 'ChopperCommand-ramNoFrameskip-v4', 'CrazyClimber-v0', 'CrazyClimberDeterministic-v0', 'CrazyClimberNoFrameskip-v0', 'CrazyClimber-v4', 'CrazyClimberDeterministic-v4', 'CrazyClimberNoFrameskip-v4', 'CrazyClimber-ram-v0', 'CrazyClimber-ramDeterministic-v0', 'CrazyClimber-ramNoFrameskip-v0', 'CrazyClimber-ram-v4', 'CrazyClimber-ramDeterministic-v4', 'CrazyClimber-ramNoFrameskip-v4', 'Defender-v0', 'DefenderDeterministic-v0', 'DefenderNoFrameskip-v0', 'Defender-v4', 'DefenderDeterministic-v4', 'DefenderNoFrameskip-v4', 'Defender-ram-v0', 'Defender-ramDeterministic-v0', 'Defender-ramNoFrameskip-v0', 'Defender-ram-v4', 'Defender-ramDeterministic-v4', 'Defender-ramNoFrameskip-v4', 'DemonAttack-v0', 'DemonAttackDeterministic-v0', 'DemonAttackNoFrameskip-v0', 'DemonAttack-v4', 'DemonAttackDeterministic-v4', 'DemonAttackNoFrameskip-v4', 'DemonAttack-ram-v0', 'DemonAttack-ramDeterministic-v0', 'DemonAttack-ramNoFrameskip-v0', 'DemonAttack-ram-v4', 'DemonAttack-ramDeterministic-v4', 'DemonAttack-ramNoFrameskip-v4', 'DoubleDunk-v0', 'DoubleDunkDeterministic-v0', 'DoubleDunkNoFrameskip-v0', 'DoubleDunk-v4', 'DoubleDunkDeterministic-v4', 'DoubleDunkNoFrameskip-v4', 'DoubleDunk-ram-v0', 'DoubleDunk-ramDeterministic-v0', 'DoubleDunk-ramNoFrameskip-v0', 'DoubleDunk-ram-v4', 'DoubleDunk-ramDeterministic-v4', 'DoubleDunk-ramNoFrameskip-v4', 'ElevatorAction-v0', 'ElevatorActionDeterministic-v0', 'ElevatorActionNoFrameskip-v0', 'ElevatorAction-v4', 'ElevatorActionDeterministic-v4', 'ElevatorActionNoFrameskip-v4', 'ElevatorAction-ram-v0', 'ElevatorAction-ramDeterministic-v0', 'ElevatorAction-ramNoFrameskip-v0', 'ElevatorAction-ram-v4', 'ElevatorAction-ramDeterministic-v4', 'ElevatorAction-ramNoFrameskip-v4', 'Enduro-v0', 'EnduroDeterministic-v0', 'EnduroNoFrameskip-v0', 'Enduro-v4', 'EnduroDeterministic-v4', 'EnduroNoFrameskip-v4', 'Enduro-ram-v0', 'Enduro-ramDeterministic-v0', 'Enduro-ramNoFrameskip-v0', 'Enduro-ram-v4', 'Enduro-ramDeterministic-v4', 'Enduro-ramNoFrameskip-v4', 'FishingDerby-v0', 'FishingDerbyDeterministic-v0', 'FishingDerbyNoFrameskip-v0', 'FishingDerby-v4', 'FishingDerbyDeterministic-v4', 'FishingDerbyNoFrameskip-v4', 'FishingDerby-ram-v0', 'FishingDerby-ramDeterministic-v0', 'FishingDerby-ramNoFrameskip-v0', 'FishingDerby-ram-v4', 'FishingDerby-ramDeterministic-v4', 'FishingDerby-ramNoFrameskip-v4', 'Freeway-v0', 'FreewayDeterministic-v0', 'FreewayNoFrameskip-v0', 'Freeway-v4', 'FreewayDeterministic-v4', 'FreewayNoFrameskip-v4', 'Freeway-ram-v0', 'Freeway-ramDeterministic-v0', 'Freeway-ramNoFrameskip-v0', 'Freeway-ram-v4', 'Freeway-ramDeterministic-v4',

'Freeway-ramNoFrameskip-v4', 'Frostbite-v0', 'FrostbiteDeterministic-v0', 'FrostbiteNoFrameskip-v0', 'Frostbite-v4', 'FrostbiteDeterministic-v4', 'FrostbiteNoFrameskip-v4', 'Frostbite-ram-v0', 'Frostbite-ramDeterministic-v0', 'Frostbite-ramNoFrameskip-v0', 'Frostbite-ram-v4', 'Frostbite-ramDeterministic-v4', 'Frostbite-ramNoFrameskip-v4', 'Gopher-v0', 'GopherDeterministic-v0', 'GopherNoFrameskip-v0', 'Gopher-v4', 'GopherDeterministic-v4', 'GopherNoFrameskip-v4', 'Gopher-ram-v0', 'Gopher-ramDeterministic-v0', 'Gopher-ramNoFrameskip-v0', 'Gopher-ram-v4', 'Gopher-ramDeterministic-v4', 'Gopher-ramNoFrameskip-v4', 'Gravitar-v0', 'GravitarDeterministic-v0', 'GravitarNoFrameskip-v0', 'Gravitar-v4', 'GravitarDeterministic-v4', 'GravitarNoFrameskip-v4', 'Gravitar-ram-v0', 'Gravitar-ramDeterministic-v0', 'Gravitar-ramNoFrameskip-v0', 'Gravitar-ram-v4', 'Gravitar-ramDeterministic-v4', 'Gravitar-ramNoFrameskip-v4', 'Hero-v0', 'HeroDeterministic-v0', 'HeroNoFrameskip-v0', 'Hero-v4', 'HeroDeterministic-v4', 'HeroNoFrameskip-v4', 'Hero-ram-v0', 'Hero-ramDeterministic-v0', 'Hero-ramNoFrameskip-v0', 'Hero-ram-v4', 'Hero-ramDeterministic-v4', 'Hero-ramNoFrameskip-v4', 'IceHockey-v0', 'IceHockeyDeterministic-v0', 'IceHockeyNoFrameskip-v0', 'IceHockey-v4', 'IceHockeyDeterministic-v4', 'IceHockeyNoFrameskip-v4', 'IceHockey-ram-v0', 'IceHockey-ramDeterministic-v0', 'IceHockey-ramNoFrameskip-v0', 'IceHockey-ram-v4', 'IceHockey-ramDeterministic-v4', 'IceHockey-ramNoFrameskip-v4', 'Jamesbond-v0', 'JamesbondDeterministic-v0', 'JamesbondNoFrameskip-v0', 'Jamesbond-v4', 'JamesbondDeterministic-v4', 'JamesbondNoFrameskip-v4', 'Jamesbond-ram-v0', 'Jamesbond-ramDeterministic-v0', 'Jamesbond-ramNoFrameskip-v0', 'Jamesbond-ram-v4', 'Jamesbond-ramDeterministic-v4', 'Jamesbond-ramNoFrameskip-v4', 'JourneyEscape-v0', 'JourneyEscapeDeterministic-v0', 'JourneyEscapeNoFrameskip-v0', 'JourneyEscape-v4', 'JourneyEscapeDeterministic-v4', 'JourneyEscapeNoFrameskip-v4', 'JourneyEscape-ram-v0', 'JourneyEscape-ramDeterministic-v0', 'JourneyEscape-ramNoFrameskip-v0', 'JourneyEscape-ram-v4', 'JourneyEscape-ramDeterministic-v4', 'JourneyEscape-ramNoFrameskip-v4', 'Kangaroo-v0', 'KangarooDeterministic-v0', 'KangarooNoFrameskip-v0', 'Kangaroo-v4', 'KangarooDeterministic-v4', 'KangarooNoFrameskip-v4', 'Kangaroo-ram-v0', 'Kangaroo-ramDeterministic-v0', 'Kangaroo-ramNoFrameskip-v0', 'Kangaroo-ram-v4', 'Kangaroo-ramDeterministic-v4', 'Kangaroo-ramNoFrameskip-v4', 'Krull-v0', 'KrullDeterministic-v0', 'KrullNoFrameskip-v0', 'Krull-v4', 'KrullDeterministic-v4', 'KrullNoFrameskip-v4', 'Krull-ram-v0', 'Krull-ramDeterministic-v0', 'Krull-ramNoFrameskip-v0', 'Krull-ram-v4', 'Krull-ramDeterministic-v4', 'Krull-ramNoFrameskip-v4', 'KungFuMaster-v0', 'KungFuMasterDeterministic-v0', 'KungFuMasterNoFrameskip-v0', 'KungFuMaster-v4', 'KungFuMasterDeterministic-v4', 'KungFuMasterNoFrameskip-v4', 'KungFuMaster-ram-v0', 'KungFuMaster-ramDeterministic-v0', 'KungFuMaster-ramNoFrameskip-v0', 'KungFuMaster-ram-v4', 'KungFuMaster-ramDeterministic-v4', 'KungFuMaster-ramNoFrameskip-v4', 'MontezumaRevenge-v0',

'MontezumaRevengeDeterministic-v0', 'MontezumaRevengeNoFrameskip-v0',
'MontezumaRevenge-v4', 'MontezumaRevengeDeterministic-v4',
'MontezumaRevengeNoFrameskip-v4', 'MontezumaRevenge-ram-v0',
'MontezumaRevenge-ramDeterministic-v0', 'MontezumaRevenge-
ramNoFrameskip-v0', 'MontezumaRevenge-ram-v4', 'MontezumaRevenge-
ramDeterministic-v4', 'MontezumaRevenge-ramNoFrameskip-v4', 'MsPacman-
v0', 'MsPacmanDeterministic-v0', 'MsPacmanNoFrameskip-v0', 'MsPacman-
v4', 'MsPacmanDeterministic-v4', 'MsPacmanNoFrameskip-v4', 'MsPacman-
ram-v0', 'MsPacman-ramDeterministic-v0', 'MsPacman-ramNoFrameskip-v0',
'MsPacman-ram-v4', 'MsPacman-ramDeterministic-v4', 'MsPacman-
ramNoFrameskip-v4', 'NameThisGame-v0', 'NameThisGameDeterministic-v0',
'NameThisGameNoFrameskip-v0', 'NameThisGame-v4',
'NameThisGameDeterministic-v4', 'NameThisGameNoFrameskip-v4',
'NameThisGame-ram-v0', 'NameThisGame-ramDeterministic-v0',
'NameThisGame-ramNoFrameskip-v0', 'NameThisGame-ram-v4',
'NameThisGame-ramDeterministic-v4', 'NameThisGame-ramNoFrameskip-v4',
'Phoenix-v0', 'PhoenixDeterministic-v0', 'PhoenixNoFrameskip-v0',
'Phoenix-v4', 'PhoenixDeterministic-v4', 'PhoenixNoFrameskip-v4',
'Phoenix-ram-v0', 'Phoenix-ramDeterministic-v0', 'Phoenix-
ramNoFrameskip-v0', 'Phoenix-ram-v4', 'Phoenix-ramDeterministic-v4',
'Phoenix-ramNoFrameskip-v4', 'Pitfall-v0', 'PitfallDeterministic-v0',
'PitfallNoFrameskip-v0', 'Pitfall-v4', 'PitfallDeterministic-v4',
'PitfallNoFrameskip-v4', 'Pitfall-ram-v0', 'Pitfall-ramDeterministic-
v0', 'Pitfall-ramNoFrameskip-v0', 'Pitfall-ram-v4', 'Pitfall-
ramDeterministic-v4', 'Pitfall-ramNoFrameskip-v4', 'Pong-v0',
'PongDeterministic-v0', 'PongNoFrameskip-v0', 'Pong-v4',
'PongDeterministic-v4', 'PongNoFrameskip-v4', 'Pong-ram-v0', 'Pong-
ramDeterministic-v0', 'Pong-ramNoFrameskip-v0', 'Pong-ram-v4', 'Pong-
ramDeterministic-v4', 'Pong-ramNoFrameskip-v4', 'Pooyan-v0',
'PooyanDeterministic-v0', 'PooyanNoFrameskip-v0', 'Pooyan-v4',
'PooyanDeterministic-v4', 'PooyanNoFrameskip-v4', 'Pooyan-ram-v0',
'Pooyan-ramDeterministic-v0', 'Pooyan-ramNoFrameskip-v0', 'Pooyan-ram-
v4', 'Pooyan-ramDeterministic-v4', 'Pooyan-ramNoFrameskip-v4',
'PrivateEye-v0', 'PrivateEyeDeterministic-v0', 'PrivateEyeNoFrameskip-
v0', 'PrivateEye-v4', 'PrivateEyeDeterministic-v4',
'PrivateEyeNoFrameskip-v4', 'PrivateEye-ram-v0', 'PrivateEye-
ramDeterministic-v0', 'PrivateEye-ramNoFrameskip-v0', 'PrivateEye-ram-
v4', 'PrivateEye-ramDeterministic-v4', 'PrivateEye-ramNoFrameskip-v4',
'Qbert-v0', 'QbertDeterministic-v0', 'QbertNoFrameskip-v0', 'Qbert-
v4', 'QbertDeterministic-v4', 'QbertNoFrameskip-v4', 'Qbert-ram-v0',
'Qbert-ramDeterministic-v0', 'Qbert-ramNoFrameskip-v0', 'Qbert-ram-
v4', 'Qbert-ramDeterministic-v4', 'Qbert-ramNoFrameskip-v4',
'Riverraid-v0', 'RiverraidDeterministic-v0', 'RiverraidNoFrameskip-
v0', 'Riverraid-v4', 'RiverraidDeterministic-v4',
'RiverraidNoFrameskip-v4', 'Riverraid-ram-v0', 'Riverraid-
ramDeterministic-v0', 'Riverraid-ramNoFrameskip-v0', 'Riverraid-ram-
v4', 'Riverraid-ramDeterministic-v4', 'Riverraid-ramNoFrameskip-v4',
'RoadRunner-v0', 'RoadRunnerDeterministic-v0', 'RoadRunnerNoFrameskip-
v0', 'RoadRunner-v4', 'RoadRunnerDeterministic-v4',

'RoadRunnerNoFrameskip-v4', 'RoadRunner-ram-v0', 'RoadRunner-ramDeterministic-v0', 'RoadRunner-ramNoFrameskip-v0', 'RoadRunner-ram-v4', 'RoadRunner-ramDeterministic-v4', 'RoadRunner-ramNoFrameskip-v4', 'Robotank-v0', 'RobotankDeterministic-v0', 'RobotankNoFrameskip-v0', 'Robotank-v4', 'RobotankDeterministic-v4', 'RobotankNoFrameskip-v4', 'Robotank-ram-v0', 'Robotank-ramDeterministic-v0', 'Robotank-ramNoFrameskip-v0', 'Robotank-ram-v4', 'Robotank-ramDeterministic-v4', 'Robotank-ramNoFrameskip-v4', 'Seaquest-v0', 'SeaquestDeterministic-v0', 'SeaquestNoFrameskip-v0', 'Seaquest-v4', 'SeaquestDeterministic-v4', 'SeaquestNoFrameskip-v4', 'Seaquest-ram-v0', 'Seaquest-ramDeterministic-v0', 'Seaquest-ramNoFrameskip-v0', 'Seaquest-ram-v4', 'Seaquest-ramDeterministic-v4', 'Seaquest-ramNoFrameskip-v4', 'Skiing-v0', 'SkiingDeterministic-v0', 'SkiingNoFrameskip-v0', 'Skiing-v4', 'SkiingDeterministic-v4', 'SkiingNoFrameskip-v4', 'Skiing-ram-v0', 'Skiing-ramDeterministic-v0', 'Skiing-ramNoFrameskip-v0', 'Skiing-ram-v4', 'Skiing-ramDeterministic-v4', 'Skiing-ramNoFrameskip-v4', 'Solaris-v0', 'SolarisDeterministic-v0', 'SolarisNoFrameskip-v0', 'Solaris-v4', 'SolarisDeterministic-v4', 'SolarisNoFrameskip-v4', 'Solaris-ram-v0', 'Solaris-ramDeterministic-v0', 'Solaris-ramNoFrameskip-v0', 'Solaris-ram-v4', 'Solaris-ramDeterministic-v4', 'Solaris-ramNoFrameskip-v4', 'SpaceInvaders-v0', 'SpaceInvadersDeterministic-v0', 'SpaceInvadersNoFrameskip-v0', 'SpaceInvaders-v4', 'SpaceInvadersDeterministic-v4', 'SpaceInvadersNoFrameskip-v4', 'SpaceInvaders-ram-v0', 'SpaceInvaders-ramDeterministic-v0', 'SpaceInvaders-ramNoFrameskip-v0', 'SpaceInvaders-ram-v4', 'SpaceInvaders-ramDeterministic-v4', 'SpaceInvaders-ramNoFrameskip-v4', 'StarGunner-v0', 'StarGunnerDeterministic-v0', 'StarGunnerNoFrameskip-v0', 'StarGunner-v4', 'StarGunnerDeterministic-v4', 'StarGunnerNoFrameskip-v4', 'StarGunner-ram-v0', 'StarGunner-ramDeterministic-v0', 'StarGunner-ramNoFrameskip-v0', 'StarGunner-ram-v4', 'StarGunner-ramDeterministic-v4', 'StarGunner-ramNoFrameskip-v4', 'Tennis-v0', 'TennisDeterministic-v0', 'TennisNoFrameskip-v0', 'Tennis-v4', 'TennisDeterministic-v4', 'TennisNoFrameskip-v4', 'Tennis-ram-v0', 'Tennis-ramDeterministic-v0', 'Tennis-ramNoFrameskip-v0', 'Tennis-ram-v4', 'Tennis-ramDeterministic-v4', 'Tennis-ramNoFrameskip-v4', 'TimePilot-v0', 'TimePilotDeterministic-v0', 'TimePilotNoFrameskip-v0', 'TimePilot-v4', 'TimePilotDeterministic-v4', 'TimePilotNoFrameskip-v4', 'TimePilot-ram-v0', 'TimePilot-ramDeterministic-v0', 'TimePilot-ramNoFrameskip-v0', 'TimePilot-ram-v4', 'TimePilot-ramDeterministic-v4', 'TimePilot-ramNoFrameskip-v4', 'Tutankham-v0', 'TutankhamDeterministic-v0', 'TutankhamNoFrameskip-v0', 'Tutankham-v4', 'TutankhamDeterministic-v4', 'TutankhamNoFrameskip-v4', 'Tutankham-ram-v0', 'Tutankham-ramDeterministic-v0', 'Tutankham-ramNoFrameskip-v0', 'Tutankham-ram-v4', 'Tutankham-ramDeterministic-v4', 'Tutankham-ramNoFrameskip-v4', 'UpNDown-v0', 'UpNDownDeterministic-v0', 'UpNDownNoFrameskip-v0', 'UpNDown-v4', 'UpNDownDeterministic-v4', 'UpNDownNoFrameskip-v4', 'UpNDown-ram-v0', 'UpNDown-ramDeterministic-v0', 'UpNDown-

ramNoFrameskip-v0', 'UpNDown-ram-v4', 'UpNDown-ramDeterministic-v4',
'UpNDown-ramNoFrameskip-v4', 'Venture-v0', 'VentureDeterministic-v0',
'VentureNoFrameskip-v0', 'Venture-v4', 'VentureDeterministic-v4',
'VentureNoFrameskip-v4', 'Venture-ram-v0', 'Venture-ramDeterministic-
v0', 'Venture-ramNoFrameskip-v0', 'Venture-ram-v4', 'Venture-
ramDeterministic-v4', 'Venture-ramNoFrameskip-v4', 'VideoPinball-v0',
'VideoPinballDeterministic-v0', 'VideoPinballNoFrameskip-v0',
'VideoPinball-v4', 'VideoPinballDeterministic-v4',
'VideoPinballNoFrameskip-v4', 'VideoPinball-ram-v0', 'VideoPinball-
ramDeterministic-v0', 'VideoPinball-ramNoFrameskip-v0', 'VideoPinball-
ram-v4', 'VideoPinball-ramDeterministic-v4', 'VideoPinball-
ramNoFrameskip-v4', 'WizardOfWor-v0', 'WizardOfWorDeterministic-v0',
'WizardOfWorNoFrameskip-v0', 'WizardOfWor-v4',
'WizardOfWorDeterministic-v4', 'WizardOfWorNoFrameskip-v4',
'WizardOfWor-ram-v0', 'WizardOfWor-ramDeterministic-v0', 'WizardOfWor-
ramNoFrameskip-v0', 'WizardOfWor-ram-v4', 'WizardOfWor-
ramDeterministic-v4', 'WizardOfWor-ramNoFrameskip-v4', 'YarsRevenge-
v0', 'YarsRevengeDeterministic-v0', 'YarsRevengeNoFrameskip-v0',
'YarsRevenge-v4', 'YarsRevengeDeterministic-v4',
'YarsRevengeNoFrameskip-v4', 'YarsRevenge-ram-v0', 'YarsRevenge-
ramDeterministic-v0', 'YarsRevenge-ramNoFrameskip-v0', 'YarsRevenge-
ram-v4', 'YarsRevenge-ramDeterministic-v4', 'YarsRevenge-
ramNoFrameskip-v4', 'Zaxxon-v0', 'ZaxxonDeterministic-v0',
'ZaxxonNoFrameskip-v0', 'Zaxxon-v4', 'ZaxxonDeterministic-v4',
'ZaxxonNoFrameskip-v4', 'Zaxxon-ram-v0', 'Zaxxon-ramDeterministic-v0',
'Zaxxon-ramNoFrameskip-v0', 'Zaxxon-ram-v4', 'Zaxxon-ramDeterministic-
v4', 'Zaxxon-ramNoFrameskip-v4', 'ALE/Adventure-v5', 'ALE/Adventure-
ram-v5', 'ALE/AirRaid-v5', 'ALE/AirRaid-ram-v5', 'ALE/Alien-v5',
'ALE/Alien-ram-v5', 'ALE/Amidar-v5', 'ALE/Amidar-ram-v5',
'ALE/Assault-v5', 'ALE/Assault-ram-v5', 'ALE/Asterix-v5',
'ALE/Asterix-ram-v5', 'ALE/Asteroids-v5', 'ALE/Asteroids-ram-v5',
'ALE/Atlantis-v5', 'ALE/Atlantis-ram-v5', 'ALE/Atlantis2-v5',
'ALE/Atlantis2-ram-v5', 'ALE/Backgammon-v5', 'ALE/Backgammon-ram-v5',
'ALE/BankHeist-v5', 'ALE/BankHeist-ram-v5', 'ALE/BasicMath-v5',
'ALE/BasicMath-ram-v5', 'ALE/BattleZone-v5', 'ALE/BattleZone-ram-v5',
'ALE/BeamRider-v5', 'ALE/BeamRider-ram-v5', 'ALE/Berzerk-v5',
'ALE/Berzerk-ram-v5', 'ALE/Blackjack-v5', 'ALE/Blackjack-ram-v5',
'ALE/Bowling-v5', 'ALE/Bowling-ram-v5', 'ALE/Boxing-v5', 'ALE/Boxing-
ram-v5', 'ALE/Breakout-v5', 'ALE/Breakout-ram-v5', 'ALE/Carnival-v5',
'ALE/Carnival-ram-v5', 'ALE/Casino-v5', 'ALE/Casino-ram-v5',
'ALE/Centipede-v5', 'ALE/Centipede-ram-v5', 'ALE/ChopperCommand-v5',
'ALE/ChopperCommand-ram-v5', 'ALE/CrazyClimber-v5', 'ALE/CrazyClimber-
ram-v5', 'ALE/Crossbow-v5', 'ALE/Crossbow-ram-v5', 'ALE/Darkchambers-
v5', 'ALE/Darkchambers-ram-v5', 'ALE/Defender-v5', 'ALE/Defender-ram-
v5', 'ALE/DemonAttack-v5', 'ALE/DemonAttack-ram-v5', 'ALE/DonkeyKong-
v5', 'ALE/DonkeyKong-ram-v5', 'ALE/DoubleDunk-v5', 'ALE/DoubleDunk-
ram-v5', 'ALE/Earthworld-v5', 'ALE/Earthworld-ram-v5',
'ALE/ElevatorAction-v5', 'ALE/ElevatorAction-ram-v5', 'ALE/Enduro-v5',
'ALE/Enduro-ram-v5', 'ALE/Entombed-v5', 'ALE/Entombed-ram-v5',

```
'ALE/Et-v5', 'ALE/Et-ram-v5', 'ALE/FishingDerby-v5',  
'ALE/FishingDerby-ram-v5', 'ALE/FlagCapture-v5', 'ALE/FlagCapture-ram-  
v5', 'ALE/Freeway-v5', 'ALE/Freeway-ram-v5', 'ALE/Frogger-v5',  
'ALE/Frogger-ram-v5', 'ALE/Frostbite-v5', 'ALE/Frostbite-ram-v5',  
'ALE/Galaxian-v5', 'ALE/Galaxian-ram-v5', 'ALE/Gopher-v5',  
'ALE/Gopher-ram-v5', 'ALE/Gravitar-v5', 'ALE/Gravitar-ram-v5',  
'ALE/Hangman-v5', 'ALE/Hangman-ram-v5', 'ALE/HauntedHouse-v5',  
'ALE/HauntedHouse-ram-v5', 'ALE/Hero-v5', 'ALE/Hero-ram-v5',  
'ALE/HumanCannonball-v5', 'ALE/HumanCannonball-ram-v5',  
'ALE/IceHockey-v5', 'ALE/IceHockey-ram-v5', 'ALE/Jamesbond-v5',  
'ALE/Jamesbond-ram-v5', 'ALE/JourneyEscape-v5', 'ALE/JourneyEscape-  
ram-v5', 'ALE/Kaboom-v5', 'ALE/Kaboom-ram-v5', 'ALE/Kangaroo-v5',  
'ALE/Kangaroo-ram-v5', 'ALE/KeystoneKapers-v5', 'ALE/KeystoneKapers-  
ram-v5', 'ALE/KingKong-v5', 'ALE/KingKong-ram-v5', 'ALE/Klax-v5',  
'ALE/Klax-ram-v5', 'ALE/Koolaid-v5', 'ALE/Koolaid-ram-v5', 'ALE/Krull-  
v5', 'ALE/Krull-ram-v5', 'ALE/KungFuMaster-v5', 'ALE/KungFuMaster-ram-  
v5', 'ALE/LaserGates-v5', 'ALE/LaserGates-ram-v5', 'ALE/LostLuggage-  
v5', 'ALE/LostLuggage-ram-v5', 'ALE/MarioBros-v5', 'ALE/MarioBros-ram-  
v5', 'ALE/MiniatureGolf-v5', 'ALE/MiniatureGolf-ram-v5',  
'ALE/MontezumaRevenge-v5', 'ALE/MontezumaRevenge-ram-v5', 'ALE/MrDo-  
v5', 'ALE/MrDo-ram-v5', 'ALE/MsPacman-v5', 'ALE/MsPacman-ram-v5',  
'ALE/NameThisGame-v5', 'ALE/NameThisGame-ram-v5', 'ALE/Othello-v5',  
'ALE/Othello-ram-v5', 'ALE/Pacman-v5', 'ALE/Pacman-ram-v5',  
'ALE/Phoenix-v5', 'ALE/Phoenix-ram-v5', 'ALE/Pitfall-v5',  
'ALE/Pitfall-ram-v5', 'ALE/Pitfall2-v5', 'ALE/Pitfall2-ram-v5',  
'ALE/Pong-v5', 'ALE/Pong-ram-v5', 'ALE/Pooyan-v5', 'ALE/Pooyan-ram-  
v5', 'ALE/PrivateEye-v5', 'ALE/PrivateEye-ram-v5', 'ALE/Qbert-v5',  
'ALE/Qbert-ram-v5', 'ALE/Riverraid-v5', 'ALE/Riverraid-ram-v5',  
'ALE/RoadRunner-v5', 'ALE/RoadRunner-ram-v5', 'ALE/Robotank-v5',  
'ALE/Robotank-ram-v5', 'ALE/Seaquest-v5', 'ALE/Seaquest-ram-v5',  
'ALE/SirLancelot-v5', 'ALE/SirLancelot-ram-v5', 'ALE/Skiing-v5',  
'ALE/Skiing-ram-v5', 'ALE/Solaris-v5', 'ALE/Solaris-ram-v5',  
'ALE/SpaceInvaders-v5', 'ALE/SpaceInvaders-ram-v5', 'ALE/SpaceWar-v5',  
'ALE/SpaceWar-ram-v5', 'ALE/StarGunner-v5', 'ALE/StarGunner-ram-v5',  
'ALE/Superman-v5', 'ALE/Superman-ram-v5', 'ALE/Surround-v5',  
'ALE/Surround-ram-v5', 'ALE/Tennis-v5', 'ALE/Tennis-ram-v5',  
'ALE/Tetris-v5', 'ALE/Tetris-ram-v5', 'ALE/TicTacToe3D-v5',  
'ALE/TicTacToe3D-ram-v5', 'ALE/TimePilot-v5', 'ALE/TimePilot-ram-v5',  
'ALE/Trondead-v5', 'ALE/Trondead-ram-v5', 'ALE/Turmoil-v5',  
'ALE/Turmoil-ram-v5', 'ALE/Tutankham-v5', 'ALE/Tutankham-ram-v5',  
'ALE/UpNDown-v5', 'ALE/UpNDown-ram-v5', 'ALE/Venture-v5',  
'ALE/Venture-ram-v5', 'ALE/VideoCheckers-v5', 'ALE/VideoCheckers-ram-  
v5', 'ALE/VideoChess-v5', 'ALE/VideoChess-ram-v5', 'ALE/VideoCube-v5',  
'ALE/VideoCube-ram-v5', 'ALE/VideoPinball-v5', 'ALE/VideoPinball-ram-  
v5', 'ALE/WizardOfWor-v5', 'ALE/WizardOfWor-ram-v5', 'ALE/WordZapper-  
v5', 'ALE/WordZapper-ram-v5', 'ALE/YarsRevenge-v5', 'ALE/YarsRevenge-  
ram-v5', 'ALE/Zaxxon-v5', 'ALE/Zaxxon-ram-v5']])
```

```
env = gym.make(**ENV_ARGS)
```

```

def make_env(**env_args):
    env = gym.make(**env_args)
    # env = gym.wrappers.FlattenObservation(env)
    env = gym.wrappers.RecordEpisodeStatistics(env)
    env = NoopResetEnv(env, noop_max=30)
    env = MaxAndSkipEnv(env, skip = 4)
    env = ClipRewardEnv(env)
    env = gym.wrappers.ResizeObservation(env, (84,84))
    env = gym.wrappers.GrayScaleObservation(env)
    env = gym.wrappers.FrameStack(env, 4)
    return env

# Test env
envs = gym.vector.SyncVectorEnv(
    [lambda : make_env(**ENV_ARGS) for _ in range(NUM_ENVS)]
)

assert isinstance(envs.single_action_space, gym.spaces.Discrete),
'Only discrete action is supported'

def layer_init(layer: nn.Linear, std = np.sqrt(2), bias_const = 0.0):
    torch.nn.init.orthogonal_(layer.weight, std)
    torch.nn.init.constant_(layer.bias, bias_const)
    return layer

class Agent(nn.Module):

    def __init__(self, envs: gym.Env, hidden_size: int = 512):
        super().__init__()

        self.network = nn.Sequential(
            layer_init(nn.Conv2d(4, 32, 8, stride = 4)),
            nn.ReLU(),
            layer_init(nn.Conv2d(32, 64, 4, stride = 2)),
            nn.ReLU(),
            layer_init(nn.Conv2d(64, 64, 3, stride = 1)),
            nn.ReLU(),
            nn.Flatten(),
            layer_init(nn.Linear(64 * 7 * 7, hidden_size)),
            nn.ReLU(),
        )

        self.actor = layer_init(nn.Linear(hidden_size,
envs.single_action_space.n), std = 0.01)
        self.critic = layer_init(nn.Linear(hidden_size,1 ), std = 1.0)

    def get_value(self, x):
        return self.critic(self.network(x/255.0))

```

```

def get_action_and_value(self, x, action = None):
    """
    @params:
        x: torch.tensor observation, shape = (N, observation size)
        action: torch.tensor action
    @returns:
        action: torch.tensor, shape = (N, action size)
        log_prob: torch.tensor, shape = (N,)
        entropy: torch.tensor, shape = (N,)
        value: torch.tensor, shape = (N,)
    """

    hidden = self.network(x/255.0)
    logits = self.actor(hidden)
    probs = Categorical(logits=logits)
    if action == None:
        action = probs.sample()

    log_prob = probs.log_prob(action)
    entropy = probs.entropy()
    value = self.critic(hidden)
    return action, log_prob, entropy, value

#Test agent
# Test env
envs = gym.vector.SyncVectorEnv(
    [lambda : make_env(**ENV_ARGS) for _ in range(NUM_ENVS)]
)

assert isinstance(envs.single_action_space, gym.spaces.Discrete),
'Only discrete action is supported'

obs, info = envs.reset()
obs = torch.tensor(obs).float()
print('obs shape = ', obs.shape)

test_agent = Agent(envs)

action, log_prob, entropy, value =
test_agent.get_action_and_value(obs)

print('action shape = ', action.shape)
print('log prob shape = ', log_prob.shape)
print('entropy shape = ', entropy.shape)
print('value shape = ', value.shape)

envs.close()
del test_agent

```



```

obs shape = torch.Size([3, 4, 84, 84])
action shape = torch.Size([3])
log prob shape = torch.Size([3])
entropy shape = torch.Size([3])
value shape = torch.Size([3, 1])

def plot(history, show = False, save_path = None):
    sns.lineplot(y = history['reward'], x =
list(range(len(history['reward']))))

    if save_path != None:
        plt.savefig(save_path)
    if show:
        plt.show()

    plt.clf()
    plt.close()

def evaluate(agent, episodes = 10):
    envs = gym.vector.SyncVectorEnv([lambda: make_env(**ENV_ARGS)])
    agent.eval()
    total_rewards = []
    next_obs, _ = envs.reset()

    while len(total_rewards) < episodes:
        next_obs = torch.Tensor(next_obs)
        with torch.no_grad():
            action, log_prob, _, value =
agent.get_action_and_value(next_obs)

            next_obs, reward, terminated, truncated, info =
envs.step(action.numpy())

            if 'final_info' in info:
                for data in info['final_info']:
                    if data:
                        reward = data['episode']['r'][0]
                        total_rewards.append(reward)

    return total_rewards

# Create env
envs = gym.vector.AsyncVectorEnv(
    [lambda: make_env(**ENV_ARGS) for _ in range(NUM_ENVS)]
)

agent = Agent(envs).to(DEVICE)
optimizer = torch.optim.AdamW(agent.parameters(), lr=LR, eps=1e-5,
amsgrad=True)

M = NUM_STEPS

```

```

N = NUM_ENVS

label = str(uuid.uuid4()).split('-')[0]
print('run id = ', label)

SAVE_PATH = os.path.join(OUTPUT, label)
FIG_SAVE_PATH = os.path.join(SAVE_PATH, 'plot.png')
if not os.path.exists(SAVE_PATH):
    os.makedirs(SAVE_PATH)

obs = torch.zeros((M, N) + envs.single_observation_space.shape,
device=DEVICE)
actions = torch.zeros((M, N) + envs.single_action_space.shape,
device=DEVICE)
log_probs = torch.zeros((M, N), device=DEVICE)
rewards = torch.zeros((M, N), device=DEVICE)
dones = torch.zeros((M, N), device=DEVICE) # for masking
values = torch.zeros((M, N), device=DEVICE)

global_step = 0

next_obs, _ = envs.reset()
next_obs = torch.tensor(next_obs, device=DEVICE)
next_done = torch.zeros(N, device=DEVICE) # N is num envs

print('next obs = ', next_obs.shape)
print('next done = ', next_done.shape)

reward_window = deque(maxlen=100)
history = defaultdict(list)

loop = tqdm(range(NUM_ITERATIONS))
agent.train()

best_score = float('-inf')
evaluation = 0
loss = float('inf')

for iter in loop:
    if iter % UPDATE_PLOTS == 0:
        plot(history, save_path=FIG_SAVE_PATH)

    for step in range(M):
        global_step += N

        obs[step] = next_obs
        dones[step] = next_done

        with torch.no_grad():
            action, log_prob, _, value =

```

```

agent.get_action_and_value(next_obs)
    values[step] = value.flatten()

    actions[step] = action
    log_probs[step] = log_prob

    next_obs, reward, terminated, truncated, info =
envs.step(action.cpu().numpy())
    next_done = torch.logical_or(torch.tensor(terminated),
torch.tensor(truncated)).to(DEVICE)

    rewards[step] = torch.tensor(reward, device=DEVICE).view(-1)
    next_obs = torch.tensor(next_obs, device=DEVICE)

    if 'final_info' in info:
        for data in info['final_info']:
            if data:
                reward = data['episode']['r']
                reward_window.append(reward)
                avg_reward =
torch.tensor(list(reward_window)).mean().item()
                history['reward'].append(avg_reward)
                loop.set_description(f"Reward = {avg_reward:.2f},
Global Step = {global_step}, Best Score = {best_score:.2f}, Loss =
{loss:.2f}, Steps = {step}")

                if best_score < avg_reward:
                    best_score = avg_reward
                    torch.save(agent.state_dict(),
os.path.join(SAVE_PATH, 'ppo.checkpoint.torch'))

# Continue with optimization phase
# OPTIMIZE phase:
with torch.no_grad():
    # Bootstrap values, compute returns
    next_value = agent.get_value(next_obs).reshape(1, -1)
    advantages = torch.zeros_like(rewards, device=DEVICE)
    last_gae_lam = 0

    for t in reversed(range(M)):
        if t == M - 1:
            next_non_terminal = 1.0 - next_done.float()
            next_values = next_value
        else:
            next_non_terminal = 1.0 - dones[t + 1].float()
            next_values = values[t + 1]

        # GAE-Lambda advantage calculation
        delta = rewards[t] + GAMMA * next_values *
next_non_terminal - values[t]

```

```

        advantages[t] = last_gae_lam = delta + GAMMA * GAE_LAMBDA
* next_non_terminal * last_gae_lam

        # Compute returns by adding values to advantages
        returns = advantages + values

        # Flatten the tensors to prepare for mini-batch gradient descent
        b_obs = obs.view((-1,)) + envs.single_observation_space.shape
        b_actions = actions.view((-1,)) + envs.single_action_space.shape
        b_log_probs = log_probs.view(-1)
        b_advantages = advantages.view(-1)
        b_returns = returns.view(-1)
        b_values = values.view(-1)

        # Batch indices preparation for mini-batch updates
        batch_size = M * N
        mini_batch_size = batch_size // MINI_BATCH_COUNT
        b_indices = torch.arange(batch_size, device=DEVICE)
        clip_fracs = []

        for epoch in range(UPDATE_EPOCHS):
            # Shuffle batch indices to decorrelate the batches
            b_indices = b_indices[torch.randperm(batch_size)]

            for start in range(0, batch_size, mini_batch_size):
                end = start + mini_batch_size
                mini_indices = b_indices[start:end]

                _, new_log_prob, entropy, new_value =
agent.get_action_and_value(b_obs[mini_indices],
b_actions[mini_indices])

                # Policy gradient loss calculation
                log_ratio = new_log_prob - b_log_probs[mini_indices]
                ratio = torch.exp(log_ratio)

                # Calculate surrogate losses - there is with
torch.no_grad() missing here to approximate KL
                surr1 = ratio * b_advantages[mini_indices]
                surr2 = torch.clamp(ratio, 1.0 - CLIP_COEF, 1.0 +
CLIP_COEF) * b_advantages[mini_indices]
                policy_loss = -torch.min(surr1, surr2).mean()

                # Value loss using mean squared error
                value_loss = 0.5 * (new_value.view(-1) -
b_returns[mini_indices]).pow(2).mean()

                # Total loss
                loss = policy_loss + VF_COEF * value_loss - ENTROPY_COEF *
entropy.mean()

```

```

        # Perform gradient descent step
        optimizer.zero_grad()
        loss.backward()
        nn.utils.clip_grad_norm_(agent.parameters(),
MAX_GRAD_NORM)
        optimizer.step()

        # Optional: collect information about clipping
        clip_frac = ((ratio - 1.0).abs() >
CLIP_COEF).float().mean().item()
        clip_fracs.append(clip_frac)

# Final evaluation and model saving after training
#evaluation = evaluate(agent) # Assuming evaluate function returns a
#scalar or a tensor
#print('Final evaluation score:', evaluation)
torch.save(agent.state_dict(), os.path.join(SAVE_PATH,
'ppo.checkpoint.torch'))

/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning:
os.fork() was called. os.fork() is incompatible with multithreaded
code, and JAX is multithreaded, so this will likely lead to a
deadlock.
    self.pid = os.fork()

run id = c1839c50
next obs = torch.Size([3, 4, 84, 84])
next done = torch.Size([3])

Reward = 46.08, Global Step = 714510, Best Score = 90.75, Loss = -
0.46, Steps = 601: 3%|| | 116/4000 [22:06<12:12:12,
11.31s/it]

import torch

def evaluate(agent, episodes=10):
    # Create a synchronous vector environment
    envs = gym.vector.SyncVectorEnv([lambda: make_env(**ENV_ARGS) for
_ in range(NUM_ENVS)])

    # Put the agent into evaluation mode
    agent.eval()

    total_rewards = []
    episode_rewards = [0.0] * NUM_ENVS # Initialize rewards for each
environment
    episode_counts = [0] * NUM_ENVS # Track the number of episodes
completed per environment

    # Reset environments

```

```

    obs, _ = envs.reset()
    obs = torch.tensor(obs, dtype=torch.float32).to(DEVICE) # Convert
observations to tensors

    while min(episode_counts) < episodes:
        with torch.no_grad():
            action, _, _, _ = agent.get_action_and_value(obs)
            action = action.cpu().numpy() # Convert actions to numpy
array for the environment

            next_obs, rewards, terminated, truncated, infos =
envs.step(action)

            # Update episode rewards and counts
            for i in range(NUM_ENVS):
                episode_rewards[i] += rewards[i]
                if terminated[i] or truncated[i]:
                    total_rewards.append(episode_rewards[i])
                    print(f"Environment {i+1}, Episode {episode_counts[i]
+1}/{episodes}: Reward = {episode_rewards[i]:.2f}")
                    episode_rewards[i] = 0 # Reset the reward counter for
the next episode
                    episode_counts[i] += 1 # Increment the episode count
for this environment

            # Prepare next observations
            obs = torch.tensor(next_obs, dtype=torch.float32).to(DEVICE)

            # If enough episodes have been completed, break early
            if min(episode_counts) >= episodes:
                break

    envs.close() # Always make sure to close environments
    return total_rewards

# Example usage:
test_agent = Agent(NUM_ENVS, envs.single_action_space.n).to(DEVICE)
# Make sure the agent is properly initialized
test_agent = agent
average_reward = np.mean(evaluate(test_agent, episodes=10))
print("Evaluation Average Reward:", average_reward)

Environment 2, Episode 1/10: Reward = 10.00
Environment 3, Episode 1/10: Reward = 50.00
Environment 2, Episode 2/10: Reward = 31.00
Environment 2, Episode 3/10: Reward = 50.00
Environment 3, Episode 2/10: Reward = 81.00
Environment 3, Episode 3/10: Reward = 16.00
Environment 2, Episode 4/10: Reward = 88.00
Environment 3, Episode 4/10: Reward = 39.00

```

Environment 2, Episode 5/10: Reward = 24.00
Environment 3, Episode 5/10: Reward = 40.00
Environment 2, Episode 6/10: Reward = 45.00
Environment 2, Episode 7/10: Reward = 10.00
Environment 3, Episode 6/10: Reward = 45.00
Environment 2, Episode 8/10: Reward = 19.00
Environment 2, Episode 9/10: Reward = 19.00
Environment 3, Episode 7/10: Reward = 64.00
Environment 2, Episode 10/10: Reward = 41.00
Environment 3, Episode 8/10: Reward = 35.00
Environment 2, Episode 11/10: Reward = 97.00
Environment 3, Episode 9/10: Reward = 78.00
Environment 3, Episode 10/10: Reward = 15.00
Environment 3, Episode 11/10: Reward = 19.00
Environment 2, Episode 12/10: Reward = 47.00
Environment 2, Episode 13/10: Reward = 1.00
Environment 2, Episode 14/10: Reward = 16.00
Environment 3, Episode 12/10: Reward = 39.00
Environment 2, Episode 15/10: Reward = 73.00
Environment 3, Episode 13/10: Reward = 38.00
Environment 2, Episode 16/10: Reward = 27.00
Environment 3, Episode 14/10: Reward = 23.00
Environment 3, Episode 15/10: Reward = 26.00
Environment 2, Episode 17/10: Reward = 29.00
Environment 3, Episode 16/10: Reward = 21.00
Environment 2, Episode 18/10: Reward = 15.00
Environment 2, Episode 19/10: Reward = 20.00
Environment 3, Episode 17/10: Reward = 42.00
Environment 3, Episode 18/10: Reward = 15.00
Environment 2, Episode 20/10: Reward = 90.00
Environment 3, Episode 19/10: Reward = 43.00
Environment 2, Episode 21/10: Reward = 29.00
Environment 2, Episode 22/10: Reward = 28.00
Environment 3, Episode 20/10: Reward = 40.00
Environment 2, Episode 23/10: Reward = 30.00
Environment 3, Episode 21/10: Reward = 66.00
Environment 3, Episode 22/10: Reward = 9.00
Environment 2, Episode 24/10: Reward = 37.00
Environment 3, Episode 23/10: Reward = 47.00
Environment 2, Episode 25/10: Reward = 24.00
Environment 1, Episode 1/10: Reward = 94.00
Environment 1, Episode 2/10: Reward = 8.00
Environment 2, Episode 26/10: Reward = 81.00
Environment 3, Episode 24/10: Reward = 89.00
Environment 1, Episode 3/10: Reward = 45.00
Environment 2, Episode 27/10: Reward = 78.00
Environment 3, Episode 25/10: Reward = 70.00
Environment 1, Episode 4/10: Reward = 38.00
Environment 2, Episode 28/10: Reward = 22.00

```
Environment 3, Episode 26/10: Reward = 19.00
Environment 1, Episode 5/10: Reward = 28.00
Environment 3, Episode 27/10: Reward = 18.00
Environment 2, Episode 29/10: Reward = 23.00
Environment 1, Episode 6/10: Reward = 19.00
Environment 3, Episode 28/10: Reward = 16.00
Environment 2, Episode 30/10: Reward = 18.00
Environment 3, Episode 29/10: Reward = 19.00
Environment 2, Episode 31/10: Reward = 15.00
Environment 1, Episode 7/10: Reward = 32.00
Environment 2, Episode 32/10: Reward = 14.00
Environment 1, Episode 8/10: Reward = 29.00
Environment 3, Episode 30/10: Reward = 43.00
Environment 2, Episode 33/10: Reward = 39.00
```

```
-----
-----
KeyboardInterrupt                                Traceback (most recent call
last)
<ipython-input-339-b2bcf2eb8017> in <cell line: 47>()
      45 #test_agent = Agent(NUM_ENVS,
envs.single_action_space.n).to(DEVICE) # Make sure the agent is
properly initialized
      46 test_agent = agent
--> 47 average_reward = np.mean(evaluate(test_agent, episodes=10))
      48 print("Evaluation Average Reward:", average_reward)

<ipython-input-339-b2bcf2eb8017> in evaluate(agent, episodes)
      21         action = action.cpu().numpy() # Convert actions
to numpy array for the environment
      22
--> 23         next_obs, rewards, terminated, truncated, infos =
envs.step(action)
      24
      25         # Update episode rewards and counts

/usr/local/lib/python3.10/dist-packages/gymnasium/vector/vector_env.py
in step(self, actions)
      202         """
      203         self.step_async(actions)
--> 204         return self.step_wait()
      205
      206         def call_async(self, name, *args, **kwargs):

/usr/local/lib/python3.10/dist-packages/gymnasium/vector/sync_vector_e
nv.py in step_wait(self)
      147         self._truncateds[i],
      148         info,
--> 149         ) = env.step(action)
      150
```



```

151         if self._terminateds[i] or self._truncateds[i]:
/usr/local/lib/python3.10/dist-packages/gymnasium/wrappers/frame_stack
.py in step(self, action)
177         Stacked observations, reward, terminated,
truncated, and information from the environment
178         """
--> 179         observation, reward, terminated, truncated, info =
self.env.step(action)
180         self.frames.append(observation)
181         return self.observation(None), reward, terminated,
truncated, info

/usr/local/lib/python3.10/dist-packages/gymnasium/core.py in
step(self, action)
520     ) -> tuple[WrapperObsType, SupportsFloat, bool, bool,
dict[str, Any]]:
521         """Modifies the :attr:`env` after calling :meth:`step`
using :meth:`self.observation` on the returned observations."""
--> 522         observation, reward, terminated, truncated, info =
self.env.step(action)
523         return self.observation(observation), reward,
terminated, truncated, info
524

/usr/local/lib/python3.10/dist-packages/gymnasium/core.py in
step(self, action)
521         """Modifies the :attr:`env` after calling :meth:`step`
using :meth:`self.observation` on the returned observations."""
522         observation, reward, terminated, truncated, info =
self.env.step(action)
--> 523         return self.observation(observation), reward,
terminated, truncated, info
524
525     def observation(self, observation: ObsType) ->
WrapperObsType:

/usr/local/lib/python3.10/dist-packages/gymnasium/wrappers/resize_obse
rvation.py in observation(self, observation)
78         ) from e
79
--> 80         observation = cv2.resize(
81             observation, self.shape[:-1],
interpolation=cv2.INTER_AREA
82         )

KeyboardInterrupt:

import matplotlib.pyplot as plt

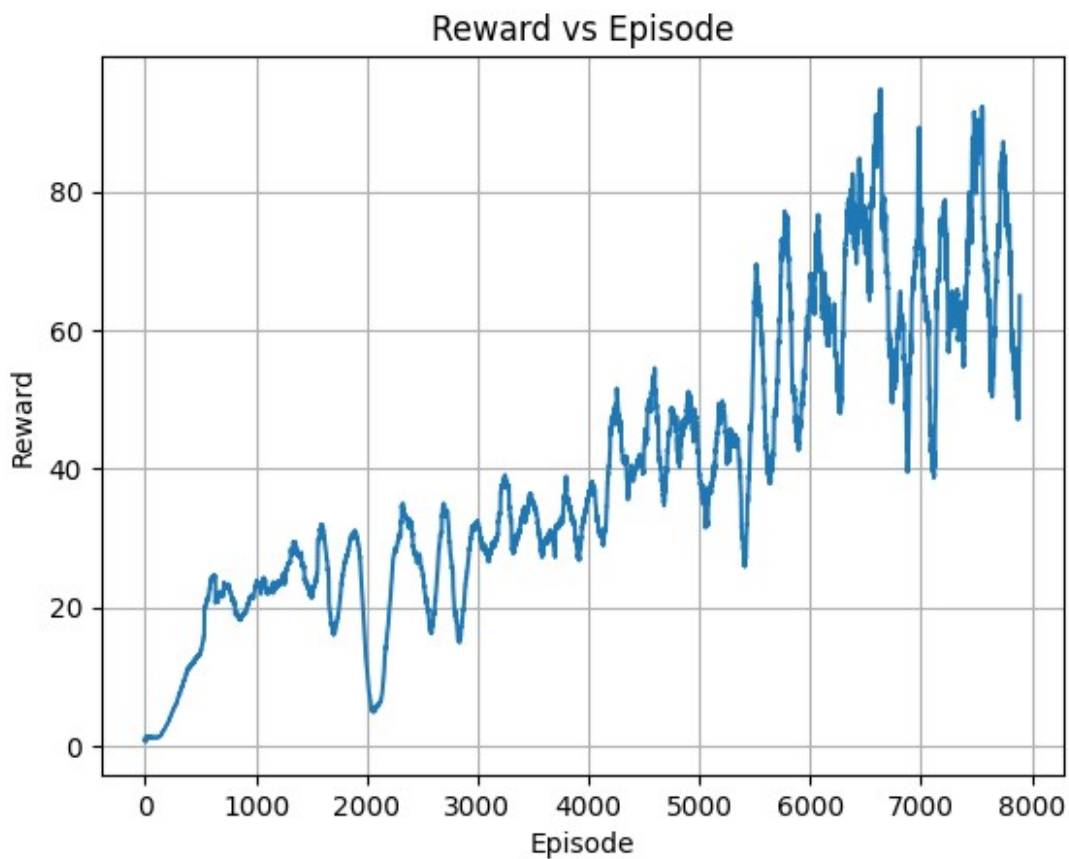
```

```

# Load the training history from the saved file
# Assuming 'history' is a dictionary containing the reward history
# history = ...

# Plot reward versus episode
plt.plot(history['reward'])
plt.xlabel('Episode')
plt.ylabel('Reward')
plt.title('Reward vs Episode')
plt.grid(True)
plt.show()

```



```

plt.figure(figsize=(10, 5))
plt.plot(history['reward'], marker='o', linestyle='--')
plt.xlabel("Episode")
plt.ylabel("Reward")
plt.title("Reward per Episode")
plt.grid(True)

# Show the plot
plt.show()

```

