```
!git clone https://github.com/leggedrobotics/rsl_rl.git
```

```
fatal: destination path 'rsl_rl' already exists and is not an empty directory.
```

```
import os

# Change to the rsl_rl directory
os.chdir('/content/rsl_rl')
print(f"Current working directory: {os.getcwd()}")
```

Show hidden output

```
pip_install_command = '!pip install -e .'

# Execute the pip install command
import subprocess
result = subprocess.run(pip_install_command[1:].split(), capture_output=True, text=True, shell=False)

# Print stdout and stderr for visibility
print(result.stdout)
print(result.stderr)

# Verify tensordict installation
try:
    import tensordict
    print("tensordict installed successfully.")
except ImportError:
    print("tensordict not found. Please ensure it's installed.")
```

```
Requirement already satisfied: torchvision>=0.5.0 in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (0.2
Requirement already satisfied: tensordict>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (0.10
Requirement already satisfied: numpy>=1.16.4 in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (2.0.2)
Requirement already satisfied: GitPython in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (3.1.45)
Requirement already satisfied: onnx in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (1.20.0)
Requirement already satisfied: onnxscript>=0.5.4 in /usr/local/lib/python3.12/dist-packages (from rsl-rl-lib==3.2.0) (0.5.
Requirement already satisfied: ml_dtypes in /usr/local/lib/python3.12/dist-packages (from onnxscript>=0.5.4->rsl-rl-lib==3
Requirement already satisfied: onnx_ir<2,>=0.1.12 in /usr/local/lib/python3.12/dist-packages (from onnxscript>=0.5.4->rsl-
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from onnxscript>=0.5.4->rsl-rl-lib==3
Requirement already satisfied: typing_extensions>=4.10 in /usr/local/lib/python3.12/dist-packages (from onnxscript>=0.5.4-
Requirement already satisfied: protobuf>=4.25.1 in /usr/local/lib/python3.12/dist-packages (from onnx->rsl-rl-lib==3.2.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.12/dist-packages (from tensordict>=0.7.0->rsl-rl-lib=
Requirement already satisfied: importlib_metadata in /usr/local/lib/python3.12/dist-packages (from tensordict>=0.7.0->rsl-
Requirement already satisfied: orjson in /usr/local/lib/python3.12/dist-packages (from tensordict>=0.7.0->rsl-rl-lib==3.2.
Requirement already satisfied: pyvers<0.2.0,>=0.1.0 in /usr/local/lib/python3.12/dist-packages (from tensordict>=0.7.0->rs
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.2.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.2.0
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.2.0) (3
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0-
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rs
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0-
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.6.0->rsl-rl-lib==3.
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.12/dist-packages (from torchvision>=0.5.0->
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.12/dist-packages (from GitPython->rsl-rl-lib==3.2
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from gitdb<5,>=4.0.1->GitPython
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=2
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib_metadata->tensordict>
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch>=2.6.0->rsl-
Building wheels for collected packages: rsl-rl-lib
  Building editable for rsl-rl-lib (pyproject.toml): started
```

```
import gymnasium as gym
import torch
from tensordict import TensorDict

# Import RSL-RL components
from rsl_rl.modules import ActorCritic
from rsl_rl.storage import RolloutStorage
from rsl_rl.algorithms import PPO
from rsl_rl.utils import resolve_obs_groups
from rsl_rl.modules import ActorCritic
print("rsl_rl components imported successfully.")
```

```
rsl_rl components imported successfully.
```

General Steps followed on repository

↳ 5 cells hidden

Flipping code

↳ 2 cells hidden

manual flip output

↳ 1 cell hidden

Symmetry Generators

↳ 13 cells hidden

NOrmalised

↳ 5 cells hidden

normalised env

↳ 2 cells hidden

Final output skew generator used

```
import yaml
from types import SimpleNamespace
from rsl_rl.runners import OnPolicyRunner
#import cartpole_symmetry

yaml_config_string = """
runner:
  class_name: OnPolicyRunner
  # General
  num_steps_per_env: 24  # Number of steps per environment per iteration
  max_iterations: 100    # Number of policy updates
  seed: 1
  # Observations
  obs_groups: {"policy": ["policy"], "critic": ["policy", "privileged"]}  # Maps observation groups to sets. See `vec_env.py
  # Logging parameters
  save_interval: 50  # Check for potential saves every `save_interval` iterations
  experiment_name: walking_experiment
  run_name: ""
  # Logging writer
  logger: tensorboard  # tensorboard, neptune, wandb
  neptune_project: legged_gym
  wandb_project: legged_gym
```

```
  # Policy
  policy:
    class_name: ActorCritic
    activation: elu
    actor_obs_normalization: false
    critic_obs_normalization: false
    actor_hidden_dims: [256, 256, 256]
    critic_hidden_dims: [256, 256, 256]
    init_noise_std: 1.0
    noise_std_type: "scalar"  # 'scalar' or 'log'
    state_dependent_std: false

  # Algorithm
  algorithm:
    class_name: PPO
    # Training
    learning_rate: 0.001
    num_learning_epochs: 5
    num_mini_batches: 4  # mini batch size = num_envs * num_steps / num_mini_batches
    schedule: adaptive  # adaptive, fixed
    # Value function
    value_loss_coef: 1.0
    clip_param: 0.2
    use_clipped_value_loss: true
    # Surrogate loss
    desired_kl: 0.01
    entropy_coef: 0.01
    gamma: 0.99
    lam: 0.95
    max_grad_norm: 1.0
    # Miscellaneous
    normalize_advantage_per_mini_batch: false

    # Random network distillation
    rnd_cfg:
      weight: 0.0  # Initial weight of the RND reward
      weight_schedule: null  # This is a dictionary with a required key called "mode". Please check the RND module for more
      reward_normalization: false  # Whether to normalize RND reward
      # Learning parameters
      learning_rate: 0.001  # Learning rate for RND
      # Network parameters
      num_outputs: 1  # Number of outputs of RND network. Note: if -1, then the network will use dimensions of the observati
      predictor_hidden_dims: [-1]  # Hidden dimensions of predictor network
      target_hidden_dims: [-1]  # Hidden dimensions of target network

    # Symmetry augmentation
    symmetry_cfg:
      use_data_augmentation: true  # This adds symmetric trajectories to the batch
      use_mirror_loss: false  # This adds symmetry loss term to the loss function
      data_augmentation_func: "rsl_rl.modules.cartpole_symmetry_utils:cartpole_symmetry"  # String containing the module and
      # Example: "legged_gym.envs.locomotion.anymal_c.symmetry:get_symmetric_states"
      mirror_loss_coeff: 0.0  # Coefficient for symmetry loss term. If 0, no symmetry loss is used
"""


def get_train_cfg():
    raw = yaml.safe_load(yaml_config_string)
    # Optional: turn top-level dict into an object-like cfg
    return SimpleNamespace(**raw)
import os

log_dir = "./logs_cartpole"
os.makedirs(log_dir, exist_ok=True)



# Usage:
train_cfg = get_train_cfg()
print(train_cfg.runner["class_name"])   # 'OnPolicyRunner'
# later:
env = CartPoleVecEnv(num_envs=8, device="cuda:0")
model_symmetry = GroupLatent(num_features=4, num_generators=6).to(device)
model_symmetry.load_state_dict(torch.load("model_symmetry_cartpole.pt"))
env.model_symmetry = model_symmetry
runner = OnPolicyRunner(env, train_cfg.runner, log_dir=log_dir)
runner.learn(num_learning_iterations=train_cfg.runner["max_iterations"])
```

```
                          Mean reward: 159.97
                  Mean episode length: 159.97
                Mean action noise std: 1.01
--------------------------------------------------------------------------------
                       Iteration time: 0.27s
                         Time elapsed: 00:00:29
                                  ETA: 00:00:00

################################################################################
                      Learning iteration 98/100

                          Total steps: 19008
                 Steps per second: 703
                  Collection time: 0.076s
                    Learning time: 0.197s
                   Mean value loss: 2.0668
                Mean surrogate loss: 0.0073
                  Mean entropy loss: 2.8546
                      Mean rnd loss: 0.0003
                 Mean symmetry loss: 0.0255
               Mean extrinsic reward: 159.97
               Mean intrinsic reward: 0.00
                          Mean reward: 159.97
                  Mean episode length: 159.97
               Mean action noise std: 1.01
--------------------------------------------------------------------------------
                       Iteration time: 0.27s
                         Time elapsed: 00:00:30
                                  ETA: 00:00:00

################################################################################
                      Learning iteration 99/100

                          Total steps: 19200
                 Steps per second: 695
                  Collection time: 0.080s
                    Learning time: 0.196s
                   Mean value loss: 1.1878
                Mean surrogate loss: 0.0083
                  Mean entropy loss: 2.8545
                      Mean rnd loss: 0.0003
                 Mean symmetry loss: 0.0086
               Mean extrinsic reward: 167.77
               Mean intrinsic reward: 0.00
                          Mean reward: 167.77
                  Mean episode length: 167.77
               Mean action noise std: 1.01
--------------------------------------------------------------------------------
                       Iteration time: 0.28s
                         Time elapsed: 00:00:30
                                  ETA: 00:00:00
```