

## ✓ Deep Q Learning with RL-Baselines3-Zoo

- Environments to be used - Space Invaders
  - Link: [https://ale.farama.org/environments/space\\_invaders/](https://ale.farama.org/environments/space_invaders/)
- RL-Library - RL-Baselines3-Zoo
  - Link: <https://github.com/DLR-RM/rl-baselines3-zoo>

## ✓ Install the RL-Baselines3 Zoo and its dependencies

- Use git+ when the package to be installed are not published on PyPI

```
1 !pip install git+https://github.com/DLR-RM/rl-baselines3-zoo
```

 [Show hidden output](#)

## ✓ Install the game environment

```
1 !pip install gymnasium[atari]
2 !pip install gymnasium[accept-rom-license]
```

 [Show hidden output](#)

## Step 1: Create a Hyperparameter Config

- Since the model to be trained is Deep Q Learning model, may refer to dqn documentation to understand the hyperparameter that will be needed.
  - Link: <https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html>
- A template for the config can be found in: <https://github.com/DLR-RM/rl-baselines3-zoo/blob/master/hyperparams/dqn.yml>. Here is a snippet copied from there. Create the config file and save as dqn.yml.

```
SpaceInvadersNoFrameskip-v4:
env_wrapper:
  - stable_baselines3.common.atari_wrappers.AtariWrapper
frame_stack: 4 #Every 4 frame as 1 input to allow the model to learn the trajectories of the object.
policy: 'CnnPolicy'
n_timesteps: !!float 1e2 # 1e6 (Recommended, but shortened in this notebook as its only for demo)
buffer_size: 100000
learning_rate: !!float 1e-4
batch_size: 32
learning_starts: 100000
target_update_interval: 1000
train_freq: 4
gradient_steps: 1
exploration_fraction: 0.1
exploration_final_eps: 0.01
# If True, you need to deactivate handle_timeout_termination
# in the replay_buffer_kwargs
optimize_memory_usage: False
```

- Some side notes on the config syntax
  1. "!!float" - Clearly indicate the data type to be read in
  2. "-" - Create a list item

## ✓ Step 2: Train the agent

- `algo` - The algorithm name, such as `dqn`, `ppo` ... (refer to the stable-baselines3 documentation for these algo, such as <https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html> for `dqn`)
- `env` - The environment id. (refer to the environment documentation, such as [https://ale.farama.org/environments/space\\_invaders/](https://ale.farama.org/environments/space_invaders/) for `SpaceInvadersNoFrameskip-v0`)
- `f` - The save folder.
- `c` - The config file.

```
1 !python -m rl_zoo3.train --algo dqn --env SpaceInvadersNoFrameskip-v4 -f logs/ -c dqn.yml
```

 [Show hidden output](#)

## ✓ Step 3: Evaluate the agent

- `algo` - The algorithm name, such as `dqn`, `ppo` ... (refer to the stable-baselines3 documentation for these algo, such as <https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html> for `dqn`)
- `env` - The environment id. (refer to the environment documentation, such as [https://ale.farama.org/environments/space\\_invaders/](https://ale.farama.org/environments/space_invaders/) for `SpaceInvadersNoFrameskip-v0`)
- `no-render` - Do not visually render the environment during evaluation / playback.
- `n-timesteps` - Number of timesteps to evaluate the agent.
- `f` - The save folder.


```
1 !python -m rl_zoo3.enjoy --algo dqn --env SpaceInvadersNoFrameskip-v4 --no-render --n-timesteps 5000 -f logs
```

 [Show hidden output](#)

## ✓ Step 4: Push the trained agent to Hub

- Create a new token here: <https://huggingface.co/settings/tokens>

```
1 from huggingface_hub import notebook_login
2
3 notebook_login()
4 !git config --global credential.helper store
```

### ✓ Step 4.1 Push to Hub

- `algo` - The algorithm name, such as `dqn`, `ppo` ... (refer to the stable-baselines3 documentation for these algo, such as <https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html> for `dqn`)
- `env` - The environment id. (refer to the environment documentation, such as [https://ale.farama.org/environments/space\\_invaders/](https://ale.farama.org/environments/space_invaders/) for `SpaceInvadersNoFrameskip-v0`)
- `repo-name` - Name for the repository to be uploaded to.
- `orga` - Hugging Face username.
- `f` - The save folder.

```
1 !python -m rl_zoo3.push_to_hub --algo dqn --env SpaceInvadersNoFrameskip-v4 --repo-name dqn-SpaceInvaderNoFrameskipv4-Attempt2 -orga
```

 [Show hidden output](#)

### ✓ Step 4.2: Create the video manually

- As there was an error with the built-in `push_to_hub` where videos are not generated, the following lines are written to manually generate the video.
- After videos are generated, rename it to `replay.mp4` and upload it manually to the repository on Hub.

```
1 !python -m rl_zoo3.record_video --env SpaceInvadersNoFrameskip-v4 --algo dqn --folder logs --n-eps 1
```

 [Show hidden output](#)

## ✓ Extra challenge - Beam Rider

- Documentation for the environment - [https://gymnasium.farama.org/v0.28.1/environments/atari/beam\\_rider/](https://gymnasium.farama.org/v0.28.1/environments/atari/beam_rider/)

```

1 # Step 1: Create a config
2
3 """
4 # Save as dqn_beam_rider.yaml
5
6 BeamRiderNoFrameskip-v4:
7   env_wrapper:
8     - stable_baselines3.common.atari_wrappers.AtariWrapper
9   frame_stack: 4 # Every 4 frame as 1 input to allow the model to learn the trajectories of the object.
10  policy: 'CnnPolicy'
11  n_timesteps: !!float 1e6
12  buffer_size: 100000
13  learning_rate: !!float 1e-4
14  batch_size: 32
15  learning_starts: 100000
16  target_update_interval: 1000
17  train_freq: 4
18  gradient_steps: 1
19  exploration_fraction: 0.1
20  exploration_final_eps: 0.01
21  # If True, you need to deactivate handle_timeout_termination
22  # in the replay_buffer_kwargs
23  optimize_memory_usage: False
24
25 """
26

```

 Show hidden output

```

1 # Step 2: Train the agent
2
3 !python -m rl_zoo3.train --algo dqn --env BeamRiderNoFrameskip-v4 -f logs/ -c dqn_beam_rider.yaml

```

 Show hidden output

```

1 # Step 3: Evaluate the agent
2
3 !python -m rl_zoo3.enjoy --algo dqn --env BeamRiderNoFrameskip-v4 --no-render --n-timesteps 5000 -f logs

```

 Show hidden output

```

1 # Step 4: Push to Hub (without video)
2
3 from huggingface_hub import notebook_login
4
5 notebook_login()
6 !git config --global credential.helper store
7
8

```

 Show hidden output

```

1 !python -m rl_zoo3.push_to_hub --algo dqn --env BeamRiderNoFrameskip-v4 --repo-name dqn-BeamRiderNoFrameskip-v4 -orga wengti0608 -f

```

 Show hidden output

```

1 # Step 5: Generate video
2
3 !python -m rl_zoo3.record_video --env BeamRiderNoFrameskip-v4 --algo dqn --folder logs --n-eps 1

```

 Show hidden output

## ✓ Load a model from Hub

- `algo` - The algorithm name, such as dqn, ppo ... (refer to the stable-baselines3 documentation for these algo, such as <https://stable-baselines3.readthedocs.io/en/master/modules/dqn.html> for dqn)
- `env` - The environment id. (refer to the environment documentation, such as [https://ale.farama.org/environments/space\\_invaders/](https://ale.farama.org/environments/space_invaders/) for SpaceInvadersNoFrameskip-v0)

- `repo-name` - Name for the repository to be uploaded to.
- `orga` - Hugging Face username.
- `f` - The save folder.

```
1 # Downloading an agent from this repo: https://huggingface.co/markeidsaune/dqn-BeamRiderNoFrameskip-v4
2 # markeidsaune/dqn-BeamRiderNoFrameskip-v4
3
4 !python -m rl_zoo3.load_from_hub --algo dqn --env BeamRiderNoFrameskip-v4 --repo-name dqn-BeamRiderNoFrameskip-v4 -orga markeidsaune
```

[Show hidden output](#)

```
1 # Evaluate the downloaded agent
2
3 !python -m rl_zoo3.enjoy --algo dqn --env BeamRiderNoFrameskip-v4 --no-render --n-timesteps 5000 -f rl_trained/
```

[Show hidden output](#)