# XuanCe: A Comprehensive and Unified Deep Reinforcement Learning Library

**XuanCe** is an open-source ensemble of Deep Reinforcement Learning (DRL) algorithm implementations.

We call it as **Xuan-Ce (玄策)** in Chinese. "**Xuan (玄)**" means incredible and magic box, "**Ce (策)**" means policy.

DRL algorithms are sensitive to hyper-parameters tuning, varying in performance with different tricks, and suffering from unstable training processes, therefore, sometimes DRL algorithms seems elusive and "Xuan". This project gives a thorough, high-quality and easy-to-understand implementation of DRL algorithms, and hope this implementation can give a hint on the magics of reinforcement learning.

We expect it to be compatible with multiple deep learning toolboxes( **PyTorch**, **TensorFlow**, and **MindSpore**), and hope it can really become a zoo full of DRL algorithms.

Paper link: <https://arxiv.org/pdf/2312.16248.pdf>

:book: **Full Documentation** | **中文文档** :book:

## Why XuanCe?

### Features of XuanCe

* :school\_satchel: Highly modularized.
* :thumbsup: Easy to [learn](https://xuance.readthedocs.io/en/latest/), easy for [installation](https://xuance.readthedocs.io/en/latest/documents/usage/installation.html), and easy for [usage](https://xuance.readthedocs.io/en/latest/documents/usage/basic_usage.html).
* :twisted\_rightwards\_arrows: Flexible for model combination.
* :tada: Abundant [algorithms](https://xuance.readthedocs.io/en/latest/documents/api/agents/agents.html) with various tasks.
* :couple: Supports both DRL and MARL tasks.
* :key: High compatibility for different users. (PyTorch, TensorFlow2, MindSpore, CPU, GPU, Linux, Windows, MacOS, etc.)
* :zap: Fast running speed with parallel environments.
* :chart\_with\_upwards\_trend: Good visualization effect with [tensorboard](https://www.tensorflow.org/tensorboard) or [wandb](https://wandb.ai/site) tool.

## Currently Included Algorithms

### :point\_right: DRL

* Deep Q Network - DQN [[Paper](https://www.nature.com/articles/nature14236)]
* DQN with Double Q-learning - Double DQN [[Paper](https://ojs.aaai.org/index.php/AAAI/article/view/10295)]
* DQN with Dueling network - Dueling DQN [[Paper](http://proceedings.mlr.press/v48/wangf16.pdf)]
* DQN with Prioritized Experience Replay - PER [[Paper](https://arxiv.org/pdf/1511.05952.pdf)]
* DQN with Parameter Space Noise for Exploration - NoisyNet [[Paper](https://arxiv.org/pdf/1706.01905.pdf)]
* Deep Recurrent Q-Netwrk - DRQN [[Paper](https://cdn.aaai.org/ocs/11673/11673-51288-1-PB.pdf)]
* DQN with Quantile Regression - QRDQN [[Paper](https://ojs.aaai.org/index.php/AAAI/article/view/11791)]
* Distributional Reinforcement Learning - C51 [[Paper](http://proceedings.mlr.press/v70/bellemare17a/bellemare17a.pdf)]
* Vanilla Policy Gradient - PG [[Paper](https://proceedings.neurips.cc/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf)]
* Phasic Policy Gradient - PPG [[Paper](http://proceedings.mlr.press/v139/cobbe21a/cobbe21a.pdf)] [[Code](https://github.com/openai/phasic-policy-gradient)]
* Advantage Actor Critic - A2C [[Paper](http://proceedings.mlr.press/v48/mniha16.pdf)] [[Code](https://github.com/openai/baselines/tree/master/baselines/a2c)]
* Soft actor-critic based on maximum entropy - SAC [[Paper](http://proceedings.mlr.press/v80/haarnoja18b/haarnoja18b.pdf)] [[Code](http://github.com/haarnoja/sac)]
* Soft actor-critic for discrete actions - SAC-Discrete [[Paper](https://arxiv.org/pdf/1910.07207.pdf)] [[Code](https://github.com/p-christ/Deep-Reinforcement-Learning-Algorithms-with-PyTorch)]
* Proximal Policy Optimization with clipped objective - PPO-Clip [[Paper](https://arxiv.org/pdf/1707.06347.pdf)] [[Code](https://github.com/berkeleydeeprlcourse/homework/tree/master/hw4)]
* Proximal Policy Optimization with KL divergence - PPO-KL [[Paper](https://arxiv.org/pdf/1707.06347.pdf)] [[Code](https://github.com/berkeleydeeprlcourse/homework/tree/master/hw4)]
* Deep Deterministic Policy Gradient - DDPG [[Paper](https://arxiv.org/pdf/1509.02971.pdf)] [[Code](https://github.com/openai/baselines/tree/master/baselines/ddpg)]
* Twin Delayed Deep Deterministic Policy Gradient - TD3 [[Paper](http://proceedings.mlr.press/v80/fujimoto18a/fujimoto18a.pdf)][[Code](https://github.com/sfujim/TD3)]
* Parameterised deep Q network - P-DQN [[Paper](https://arxiv.org/pdf/1810.06394.pdf)]
* Multi-pass parameterised deep Q network - MP-DQN [[Paper](https://arxiv.org/pdf/1905.04388.pdf)] [[Code](https://github.com/cycraig/MP-DQN)]
* Split parameterised deep Q network - SP-DQN [[Paper](https://arxiv.org/pdf/1810.06394.pdf)]

### :point\_right: Multi-Agent Reinforcement Learning (MARL)

* Independent Q-learning - IQL [[Paper](https://hal.science/file/index/docid/720669/filename/Matignon2012independent.pdf)] [[Code](https://github.com/oxwhirl/pymarl)]
* Value Decomposition Networks - VDN [[Paper](https://arxiv.org/pdf/1706.05296.pdf)] [[Code](https://github.com/oxwhirl/pymarl)]
* Q-mixing networks - QMIX [[Paper](http://proceedings.mlr.press/v80/rashid18a/rashid18a.pdf)] [[Code](https://github.com/oxwhirl/pymarl)]
* Weighted Q-mixing networks - WQMIX [[Paper](https://proceedings.neurips.cc/paper/2020/file/73a427badebe0e32caa2e1fc7530b7f3-Paper.pdf)] [[Code](https://github.com/oxwhirl/wqmix)]
* Q-transformation - QTRAN [[Paper](http://proceedings.mlr.press/v97/son19a/son19a.pdf)] [[Code](https://github.com/Sonkyunghwan/QTRAN)]
* Deep Coordination Graphs - DCG [[Paper](http://proceedings.mlr.press/v119/boehmer20a/boehmer20a.pdf)] [[Code](https://github.com/wendelinboehmer/dcg)]
* Independent Deep Deterministic Policy Gradient - IDDPG [[Paper](https://proceedings.neurips.cc/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf)]
* Multi-agent Deep Deterministic Policy Gradient - MADDPG [[Paper](https://proceedings.neurips.cc/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf)] [[Code](https://github.com/openai/maddpg)]
* Counterfactual Multi-agent Policy Gradient - COMA [[Paper](https://ojs.aaai.org/index.php/AAAI/article/view/11794)] [[Code](https://github.com/oxwhirl/pymarl)]
* Multi-agent Proximal Policy Optimization - MAPPO [[Paper](https://proceedings.neurips.cc/paper_files/paper/2022/file/9c1535a02f0ce079433344e14d910597-Paper-Datasets_and_Benchmarks.pdf)] [[Code](https://github.com/marlbenchmark/on-policy)]
* Mean-Field Q-learning - MFQ [[Paper](http://proceedings.mlr.press/v80/yang18d/yang18d.pdf)] [[Code](https://github.com/mlii/mfrl)]
* Mean-Field Actor-Critic - MFAC [[Paper](http://proceedings.mlr.press/v80/yang18d/yang18d.pdf)] [[Code](https://github.com/mlii/mfrl)]
* Independent Soft Actor-Critic - ISAC
* Multi-agent Soft Actor-Critic - MASAC [[Paper](https://arxiv.org/pdf/2104.06655.pdf)]
* Multi-agent Twin Delayed Deep Deterministic Policy Gradient - MATD3 [[Paper](https://arxiv.org/pdf/1910.01465.pdf)]

## Currently Supported Environments

### [Classic Control](https://www.gymlibrary.dev/environments/classic_control/)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

### [Box2D](https://www.gymlibrary.dev/environments/box2d/)

|  |  |  |
| --- | --- | --- |
|  |  |  |

### [MuJoCo Environments](https://www.gymlibrary.dev/environments/mujoco/)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

### [Atari Environments](https://www.gymlibrary.dev/environments/atari/)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

### [Minigrid Environments](https://minigrid.farama.org/)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

### [Drones Environments](https://github.com/utiasDSL/gym-pybullet-drones)

[XuanCe's documentation for the installation and usage of gym-pybullet-drones](https://xuance.readthedocs.io/en/latest/documents/api/environments/drones.html).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

### [MPE Environments](https://pettingzoo.farama.org/environments/mpe/)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

### [SMAC](https://github.com/oxwhirl/smac)

### [Google Research Football](https://github.com/google-research/football)

## :point\_right: Installation

:computer: The library can be run at Linux, Windows, MacOS, and EulerOS, etc.

Before installing **XuanCe**, you should install [Anaconda](https://www.anaconda.com/download) to prepare a python environment. (Note: select a proper version of Anaconda from [**here**](https://repo.anaconda.com/archive/).)

After that, open a terminal and install **XuanCe** by the following steps.

**Step 1**: Create a new conda environment (python>=3.7 is suggested):

conda create -n xuance\_env python=3.7

**Step 2**: Activate conda environment:

conda activate xuance\_env

**Step 3**: Install the library:

pip install xuance

This command does not include the dependencies of deep learning toolboxes. To install the **XuanCe** with deep learning tools, you can type pip install xuance[torch] for [PyTorch](https://pytorch.org/get-started/locally/), pip install xuance[tensorflow] for [TensorFlow2](https://www.tensorflow.org/install), pip install xuance[mindspore] for [MindSpore](https://www.mindspore.cn/install/en), and pip install xuance[all] for all dependencies.

Note: Some extra packages should be installed manually for further usage.

## :point\_right: Quickly Start

### Train a Model

import xuance runner = xuance.get\_runner(method='dqn', env='classic\_control', env\_id='CartPole-v1', is\_test=False) runner.run()

### Test the Model

import xuance runner\_test = xuance.get\_runner(method='dqn', env='classic\_control', env\_id='CartPole-v1', is\_test=True) runner\_test.run()

### Visualize the results

#### Tensorboard

You can use tensorboard to visualize what happened in the training process. After training, the log file will be automatically generated in the directory ".results/" and you should be able to see some training data after running the command.

$ tensorboard --logdir ./logs/dqn/torch/CartPole-v0

#### Weights & Biases (wandb)

XuanCe also supports Weights & Biases (wandb) tools for users to visualize the results of the running implementation.

How to use wandb online? :arrow\_right: <https://github.com/wandb/wandb.git/>

How to use wandb offline? :arrow\_right: <https://github.com/wandb/server.git/>

## Community

### [Github Issue](https://github.com/agi-brain/xuance/issues)

You can put your questions, advices, or the bugs you have found in the [Issues](https://github.com/agi-brain/xuance/issues).

### Social Accounts.

Welcome to join the official communication group with QQ app (Group number: 552432695), and the official account ("玄策 RLlib") on WeChat.

|  |  |
| --- | --- |
|  |  |

[@TFBestPractices](https://twitter.com/TFBestPractices/status/1665770204398223361)

### Citations

@article{liu2023xuance, title={XuanCe: A Comprehensive and Unified Deep Reinforcement Learning Library}, author={Liu, Wenzhang and Cai, Wenzhe and Jiang, Kun and Cheng, Guangran and Wang, Yuanda and Wang, Jiawei and Cao, Jingyu and Xu, Lele and Mu, Chaoxu and Sun, Changyin}, journal={arXiv preprint arXiv:2312.16248}, year={2023} }