

## A Dataset Statements

- Dataset documentation and intended uses.

### Environments.

- **Gym-MuJoCo** We set `EXCLUDE_CURRENT_POSITIONS_FROM_OBSERVATION` to false to include the first dimension of the position in `HalfCheetah-v3`, `Walker2d-v3`, and `Hopper-v3`. We use Gym-MuJoCo: <https://gym.openai.com/envs/#mujoco>.
- **IB** IB simulates the characteristics presented in various industrial control tasks, such as wind or gas turbines, chemical reactors, etc. The raw system output for each time step is a 6-dimensional vector including velocity, gain, shift, setpoint, consumption, and fatigue. To enhance the Markov property, the authors stitch the system outputs of the last  $K$  timesteps as observations ( $K = 30$  by default). The action space is three-dimensional. Each action can be interpreted as three proposed changes to the three observable state variables called current steerings. Original codes can be found at <https://github.com/siemens/industrialbenchmark>.
- **FinRL** FinRL contains 30 stocks in the pool and the trading histories over the past 10 years. Each stock is represented as a 6-dimensional feature vector, where one dimension is the number of stocks currently owned, another five dimensions are the factor information of that stock. The observation has one dimension of information representing the current account cash balance. The dimension of the action space is 30, corresponding to the transactions of each of the thirty stocks. Original codes can be found at <https://github.com/AI4Finance-LLC/FinRL-Library>.
- **CityLearn** The CityLearn (CL) environment reshapes the aggregation curve of electricity demand by controlling energy storage in different types of buildings. Domestic hot water (DHW) and solar power demands are modeled in the CL environment. High electricity demand raises the price of electricity and the overall cost of the distribution network. Flattening, smoothing, and narrowing the electricity demand curve help to reduce the operating and capital costs of generation, transmission, and distribution. The observation encodes the states of buildings, including time, outdoor temperature, indoor temperature, humidity, solar radiation, power consumption, charging status of the cooling and heating storage units, etc. The action is to control each building to increase or decrease the amount of energy stored in its own heat storage and cooling equipment. Original codes can be found at <https://github.com/intelligent-environments-lab/CityLearn>.

**Dataset description.** The datasets are firstly generated from three locomotion tasks from Gym-MuJoCo domains, i.e., `Hopper-v3`, `HalfCheetah-v3`, `Walker2d-v3`. Then, the datasets consist of trajectories from the industrial benchmark (IB), financial trading (FinRL), and city management (CityLearn) domains. Each item of a dataset consists of  $(s_t, a_t, r_t, s_{t+1})$  tuples. The datasets of each task have been split into training and test dataset.

The name (or identifier) of dataset consists of the domain name (i.e., `Hopper-v3`), the data quality (L for low, M for medium and H for high), and the trajectories of the dataset (100, 1, 000, 10, 000). Domain features can be found in Table 1. The trajectory length in domains with *Done* may be less than the maximum timesteps.

Table 1: Configuration of environments.

Environment	Observation Shape	Action Shape	Have Done	Max Timesteps
HalfCheetah-v3	18	6	False	1000
Hopper-v3	12	3	True	1000
Walker2d-v3	18	6	True	1000
IB	180	3	False	1000
FinRL	181	30	False	2516
CL	74	14	False	1000

**Data generation** The datasets are collected by Polixir Technologies from Gym-MuJoCo, IB, FinRL and CL environments. For each domain, we use an RL algorithm to train

the policies (modeled by a Gaussian distribution) on the environment until convergence and record a policy at every epoch. The policy with the highest episode return during the whole training are denoted as the expert policy. Another three levels of policies with around 25%, 50%, 75% expert returns are stored to simulate multi-level sub-optimal policies, denoted by low, medium, and high respectively. For each level, 4 policies with similar returns are selected, among which three policies are randomly selected to collect the training data used for offline RL policy training, and the left one produces the test data. The size of the test data is 1/10 of the training data for each task. The extra test dataset can be used to design the offline evaluation method for the model selection during training and hyper-parameter selection. Because of human manipulation or sensory errors, demonstrations are noisy in general, to reproduce this phenomenon, with probability 20%, we sample from the trained Gaussian policies to execute, otherwise, use the mean of Gaussian.

We provide training data with a maximum of  $10^4$  trajectories and three-level sizes of  $10^2$ ,  $10^3$ , and  $10^4$  trajectories by default, in order to cover both the limited data setting and when data are enough.

**Intended uses.** Our datasets are provided for offline reinforcement learning with a special focus on real-world applications. The design follows real-world properties like the conservative of behavior policies, limited amounts of data, high-dimensional state and action spaces, and the highly stochastic nature of the environments. The datasets include robotics, industrial control, finance trading and city management tasks with real-world properties, containing three-level sizes of dataset, three-level quality of data to mimic the dataset we will meet in offline RL scenarios. Users can use the dataset to evaluate offline RL algorithms with near real-world application nature.

- URL to website/platform where the dataset/benchmark can be viewed and downloaded by the reviewers.

The datasets is now accessible at <http://polixir.ai/research/neorl>.

Besides, repositories describing the dataset and reference library is available at <https://github.com/polixir/NeoRL/tree/master> or <https://agit.ai/Polixir/NeoRL>.

You can download the dataset from the web page or follow the instructions of the repositories to access the data.

- Author statement that they bear all responsibility in case of violation of rights, etc., and confirmation of the data license.

We bear all responsibility in case of violation of rights, etc. The datasets are under the Creative Commons Attribution 4.0 License (CC BY). The re-implemented codes of offline RL algorithms and OPE methods are under the Apache 2.0 License.

- Hosting, licensing, and maintenance plan.

The datasets are hosted by Polixir Technologies and are publicly available at <http://polixir.ai/research/neorl>. Polixir Technologies will provide long-term maintenance. The current license for the datasets is CC BY.

- Links to access the dataset and its metadata.

The datasets is now accessible at <http://polixir.ai/research/neorl>. The following link is for the metadata on the paperswithcode. <https://paperswithcode.com/dataset/neorl>

- Long-term preservation: It must be clear that the dataset will be available for a long time, either by uploading to a data repository or by explaining how the authors themselves will ensure this.

Since the volume of current datasets are too large, e.g., the file size of each of the FinRL 1000 tasks can be more than 3GB, we store them on our company data center and provide a persistent public link to access it. We are considering moving it to a data repository.

- Explicit license: Authors must choose a license, ideally a CC license for datasets, or an open source license for code (e.g. RL environments).

The datasets are under the Creative Commons Attribution 4.0 License (CC BY). The re-implemented codes of offline RL algorithms and OPE methods are under the Apache 2.0 License.

- Highly recommended: a persistent dereferenceable identifier (e.g. a DOI minted by a data repository or a prefix on identifiers.org) for datasets, or a code repository (e.g. GitHub, GitLab,...) for code. If this is not possible or useful, please explain why.

The codes for NeoRL are available at <https://github.com/polixir/NeoRL/tree/master> or <https://agit.ai/Polixir/NeoRL>, including the loading of the data, our re-implemented offline RL algorithms and OPE methods. The datasets are hosted on our company's server, so they do not contain an identifier currently.

- For benchmarks, the supplementary materials must ensure that all results are easily reproducible.

We list the implementation details in the Appendix D, E.