CUDC: A Curiosity-Driven Unsupervised Data Collection Method with Adaptive Temporal Distances for Offline Reinforcement Learning

Chenyu Sun[1,2,3], **Hangwei Qian**[4], **Chunyan Miao**[1,2,3]
[1]Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University (NTU), Singapore
[2]School of Computer Science and Engineering, NTU, Singapore
[3]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), NTU, Singapore
[4]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore
chenyu002@e.ntu.edu.sg, qian_hangwei@cfar.a-star.edu.sg, ascymiao@ntu.edu.sg

Abstract

Offline reinforcement learning (RL) aims to learn an effective policy from a pre-collected dataset. Most existing works are to develop sophisticated learning algorithms, with less emphasis on improving the data collection process. Moreover, it is even challenging to extend the single-task setting and collect a task-agnostic dataset that allows an agent to perform multiple downstream tasks. In this paper, we propose a **C**uriosity-driven **U**nsupervised **D**ata **C**ollection (CUDC) method to expand feature space using adaptive temporal distances for task-agnostic data collection in multi-task offline RL. To achieve this, CUDC estimates the probability of the k-step future states being reachable from the current states, and adapts how many steps into the future that the dynamics model should predict. With this adaptive reachability mechanism in place, the feature representation can be diversified, and the agent can navigate itself to collect higher-quality data with curiosity. Empirically, CUDC surpasses existing unsupervised methods in efficiency and learning performance in various downstream offline RL tasks of the DeepMind control suite.

Introduction

Deep reinforcement learning has achieved remarkable breakthroughs in various fields, such as games, robotics, and navigation in virtual environments (Kiran et al. 2021; Singh, Kumar, and Singh 2022; Sun, Qian, and Miao 2022a). However, real-time interaction with the environment under online RL settings may not always be feasible due to cost, safety, or ethical concerns (Kiran et al. 2021; Singh, Kumar, and Singh 2022). As a result, offline RL has gained popularity in recent years to cope with limited interactions, where agents learn a policy exclusively from a previously-collected dataset. The popular offline RL benchmarks such as D4RL (Fu et al. 2020) and RL Unplugged (Gulcehre et al. 2020) combine data from supervised online RL training runs with expert demonstrations, exploratory agents, and hand-coded controllers. However, collecting expert data can be time-consuming and expensive, and it may not always be available. In such cases, unsupervised methods, such as those described by ExORL (Yarats et al. 2022), can be used to collect data as a distinct contribution for offline RL (Prudencio, Maximo, and Colombini 2022). These methods aim to explore the environment and learn from the intrinsic rewards generated by the agent, without the need for supervision, to collect diverse data.

Despite the popularity of offline RL, existing works have mainly focused on model-centric practices, continually developing new algorithms (Kumar et al. 2020, 2022). These algorithms are typically evaluated on the same task for which the dataset was collected, and the learned policy can be pessimistic in out-of-distribution states and actions, leading to poor generalization in unseen downstream tasks. Recently, data-centric approaches have become emerging, emphasizing the importance of training data quality over algorithmic advances (Motamedi, Sakharnykh, and Kaldewey 2021; Patel et al. 2022). To improve training data quality, researchers have explored selecting the most critical samples or re-weighting (Wu et al. 2021) all samples in the offline RL algorithms. However, these methods are restricted to a single training data distribution and cannot be applied to multi-task settings with distribution shifts. To address this challenge, we propose to improve the data collection process directly through feature space expansion, where the distributions naturally span during diverse exploration. This approach is applicable to the multi-task setting, enabling us to obtain more diverse and high-quality data for offline RL.

Upon analyzing the current challenges faced in offline RL, the benchmark ExORL (Yarats et al. 2022) has shown that unsupervised RL methods are more effective than supervised methods in collecting datasets that allow the vanilla off-policy RL algorithm to learn and acquire different skills as an offline RL agent. However, we discovered that all these methods rely on a fixed temporal distance k between current and future states during data collection. This practice is sub-optimal and restricts the diversity of the learned feature representation, as illustrated in Figure 1 (left). To address this limitation, we propose to adapt the temporal distance as a simple yet effective way to enhance the feature representation, as it has a direct connection with the feature space.

To facilitate adaptation, exploiting reachability to more distant future states is desired. Reachability-based methods in RL aim to learn safe and efficient policies by considering reachable states under the current policy or value function (Savinov et al. 2019; Péré et al. 2018; Ivanovic et al. 2019; Yu et al. 2022), but these approaches are not directly applicable. For example, Savinov et al. (2019) only consider binary reachability, and extensively compares to stored embeddings

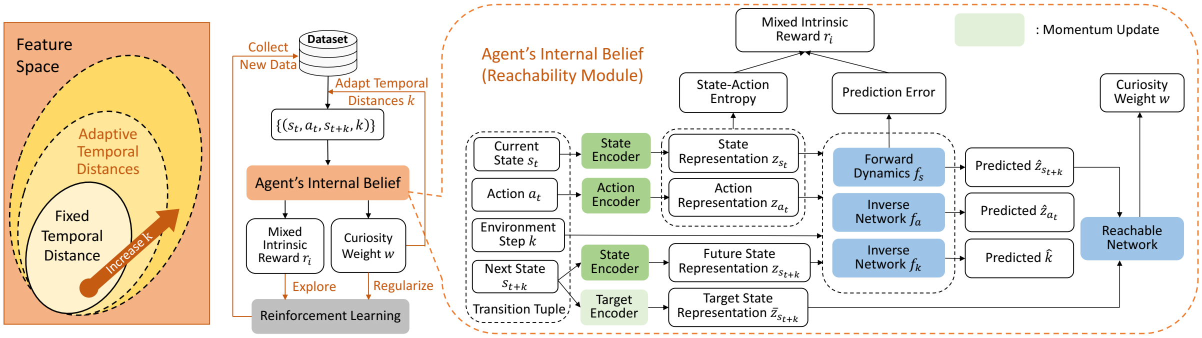

Figure 1: Curiosity-driven Unsupervised Data Collection (CUDC): The left diagram depicts the relationship between fixing (existing works) and adapting (CUDC) temporal distance in feature space. The middle diagram outlines the CUDC framework, measuring reachability between k-step future and current states using the agent's internal belief. It generates mixed intrinsic rewards for diverse exploration and curiosity weight to adapt temporal distance, regulating the RL backbone. This process continues until the capacity is reached. The right diagram illustrates how the agent assesses and updates its internal belief regarding the probability of k-step future states being reachable from current states.

in memory. Additionally, the reachability in goal space exploration (Péré et al. 2018) often requires kernel density estimation, which can increase computational cost substantially. Different from these, we propose a **C**uriosity-driven **U**nsupervised **D**ata **C**ollection (CUDC) method with a novel reachability module. Inspired by the fact that human curiosity can foster learning and is driven by novel knowledge beyond one's perception (Markey and Loewenstein 2014; Sun, Qian, and Miao 2022b; Sun and Miao 2022), CUDC facilitates data collection curiously without any task-specific reward. In particular, the reachability module estimates the probability of a k-step future state being reachable from the current state, with no episodic memory or feature space density modeling required. This module enables the agent to adaptively determine how many steps into the future that the dynamics model should predict, allowing for an enhanced feature representation to be learned. Compared with the existing unsupervised methods, it refrains from learning a fixed feature space. With this enhanced representation, CUDC utilizes a mixed intrinsic reward that encourages the agent to curiously explore meaningful state-action spaces and under-learned states. As a result, the collected dataset can lead to improved computational efficiency, sample efficiency, and learning performances in various downstream offline RL tasks.

Our contributions can be summarized as follows. 1) We are the first to introduce reachability for improving data collection in offline RL, which is defined in a more efficient way and can enable the agent to navigate curiosity-driven learning coherently. 2) We point out a common drawback of fixing the temporal distance in existing approaches, and empirically show that adapting the temporal distance in the reachability analysis can enhance feature representation by expanding the feature space.3) With the enhanced representations, CUDC additionally incentivizes the agent to explore diverse state-action space as well as the under-learned states with high prediction errors through a mixed intrinsic reward and regularization. 4) Under the ExORL benchmark setting (Yarats et al. 2022), CUDC outperforms other unsupervised methods when collecting the task-agnostic dataset that can be used for offline learning in multiple downstream tasks from the DeepMind control suite (Tassa et al. 2018).

Related Works

Reachability in RL Savinov et al. (2019) devised a reachability network to estimate how many environment steps to take for reaching a particular state. It intrinsically rewards the agent to explore the state that is unreachable from other states in memory. However, this approach only considers the binary case of reachability, potentially being inefficient when comparing with stored states. In goal exploration tasks, Péré et al. (2018) defined the reachability of a goal with an estimated density and proposed to sample increasingly difficult goals to reach during exploration. While this approach can learn the goal space in an unsupervised manner, its sampling process requires a kernel density estimator, which can substantially increase computational cost. Following the similar idea, BARC (Ivanovic et al. 2019) adapts the initial state distribution gradually from easy-to-reach to challenging-to-reach goals with physical priors in hard robotic control tasks. Recently, RCRL (Yu et al. 2022) shows that leveraging reachability analysis can help learn an optimal safe policy by expanding the limited conservative feasible set to the largest feasible set of the state space. Different from these works, CUDC is efficient and easy to implement, as it directly adapts the temporal distance to perform increasingly challenging reachability analysis without extensive comparisons, kernel density estimation or physical priors.

Curiosity-Driven RL Curiosity-driven RL is essential for encouraging agents to explore tasks in a human-like manner, especially when task-specific rewards are sparse or absent (Sun, Qian, and Miao 2022b). The main approach to curiosity-driven RL involves incorporating intrinsic rewards that motivate agents to explore based on different aspects of the state, including novelty, entropy (Seo et al. 2021; Liu and Abbeel 2021b), reachability (Savinov et al. 2019), prediction errors (Pathak et al. 2017), complexity (Campero et al. 2020), and uncertainty (Pathak, Gandhi, and Gupta 2019). Another approach is to prioritize experience replay towards under-explored states (Jiang, Grefenstette, and Rocktäschel 2021). Curiosity can also be used to explore other components of RL, as seen in CCLF (Sun, Qian, and Miao 2022a). CUDC is the first method to curiously adapt the temporal distance to explore more distant future states in offline RL, which enhances the learned representation space with increasingly challenging prediction. In addition, CUDC also regularizes Q-learning with a curiosity weight as the sample importance to focus more on under-learned tuples.

Unsupervised Data Collection The ExORL benchmark (Yarats et al. 2022) evaluates 9 unsupervised data collection algorithms, demonstrating superiority over supervised methods for multi-task offline learning. These methods include knowledge-driven models like ICM (Pathak et al. 2017), Disagreement (Pathak, Gandhi, and Gupta 2019), and RND (Burda et al. 2019), which encourage exploration by maximizing prediction errors. Data-driven models like APT (Liu and Abbeel 2021b) and ProtoRL (Yarats et al. 2021) incentivize agents to uniformly explore the entire state space. Competence-based models like DIAYN (Eysenbach et al. 2019), SMM (Lee et al. 2019), and APS (Liu and Abbeel 2021a) encourage agents to learn diverse skills by leveraging prior information. However, all of these methods were originally designed for online pretraining and fine-tuning (Laskin et al. 2021), not tailored for data collection. In contrast, CUDC is a novel method that gradually expands the feature space by exploiting reachability into more distant future states, rather than a fixed temporal distance. Additionally, CUDC exploits importance weights to focus more on under-learned tuples, which is not considered in Explore2Offline (Lambert et al. 2022), another recent method that leverages intrinsic model predictive control for simulating trajectories.

Curiosity-Driven Unsupervised Data Collection (CUDC)

Problem Setting

We consider the problem of multi-task offline learning, which consists of three main steps: data collection, reward relabeling, and downstream offline learning, as described in both ExORL (Yarats et al. 2022) and Explore2Offline (Lambert et al. 2022). In the data collection phase, the exploratory agent (data collector) has access to a Markov Decision Process (MDP) environment with a state $s \in \mathcal{S}$, an action $a \in \mathcal{A}$ based on a policy $\pi(s)$, a transition probability $p(s'|s, a)$ mapping from the current state s and action a to the next state s', a reward r, and a discount factor $\gamma \in [0, 1)$ weighting future rewards. The exploratory agent collects a dataset $\mathcal{D}$ of unlabeled tuples (s, a, s') by interacting with the environment. The second phase is to relabel the collected dataset $\mathcal{D}$ using the given reward function $r_\tau(s, a)$ about the downstream task τ for each tuple. It transfers information from task-agnostic exploration to downstream tasks. The last step is to perform multiple downstream tasks with an offline RL agent on the labeled dataset, without interacting with the environment to collect additional experiences. In this paper, we focus on the most challenging part of this problem, which is the task-agnostic data collection and we evaluate the quality of the collected dataset $\mathcal{D}$ in multiple downstream tasks.

Framework Overview

In Figure 1 (mid), we present our **C**uriosity-driven **U**nsupervised **D**ata **C**ollection (CUDC) method, employing DDPG (Lillicrap et al. 2015) as the base RL algorithm for the exploratory agent. To foster diverse exploration, our novel reachability module (Figure 1 right) calculates the likelihood of reaching a future state k steps ahead. The exploratory agent is incentivized to curiously explore through a mixed intrinsic reward, simultaneously regularizing the critic-actor update to prioritize under-learned tuples. Most importantly, the temporal distance of k-step between current and future states is adaptively increased to incorporate the dynamics information in the learned feature representation. This adaptation leads to a more diverse exploration and improved data collection quality. Further details are outlined in Algorithm 1.

The Reachability Module

In ExORL (Yarats et al. 2022), existing unsupervised methods are limited by fixing the temporal distance $k = 3$ between current and future states, as illustrated in Figure 1 (left). To overcome this limitation and expand the feature space for improved representation learning, an intuitive approach is to employ reachability analysis for adaptive adjustment of k. However, existing reachability implementations are not desired due to limited binary classification of reachable states (Savinov et al. 2019) or their reliance on costly density estimation of goal space (Péré et al. 2018). To address these issues, we propose a self-supervised reachability estimation method in CUDC, which estimates the probability of a k-step future state s_{t_i+k} being reachable from the current state s_{t_i} without requiring expensive density estimation or manual labeling. Consequently, our method can effectively enhance feature representation by expanding the feature space through an adaptive k-step. This approach has also been demonstrated to be effective in other works on reachability, such as constrained RL (Yu et al. 2022) and robotics (Ivanovic et al. 2019).

Given a batch of unlabeled tuples $(s_{t_i}, a_{t_i}, s_{t_i+k}, k)_{i=1}^n$, existing methods in ExORL benchmark (Yarats et al. 2022) simply fix the temporal distance $k = 3$ throughout the data collection. In contrast, CUDC considers k as a parameter and incorporates it explicitly into the tuples. We start by encoding the state features $z_{s_{t_i}} = \phi_s(s_{t_i})$, $z_{s_{t_i+k}} = \phi_s(s_{t_i+k})$, and the action feature $z_{a_{t_i}} = \phi_a(a_{t_i})$ using a state encoder $\phi_s(\cdot)$ and an action encoder $\phi_a(\cdot)$. We then perform one-hot encoding for the temporal distance k. To enable reachability analysis, we construct a forward dynamic network $\hat{z}_{s_{t_i+k}} = f_s(z_{s_{t_i}}, z_{a_{t_i}}, k; \theta_s)$ that takes as input $z_{s_{t_i}}$, $z_{a_{t_i}}$, and the encoded k to predict the future state feature $\hat{z}_{st_i+k}$, fully utilizing dynamics information. The network can be trained by minimizing the l_2 norm loss $||z_{s_{t_i+k}} - \hat{z}_{s_{t_i+k}}||_2$.

To quantify the reachability, CUDC enforces $\hat{z}_{s_{t_i+k}}$ to match with its own $z_{s_{t_i+k}}$ as much as possible, while keeping apart from the other future states within the same batch. This contrastive intuition is that each future state should be most reachable from its own current state, and it can quantify the reachability in a simple and efficient way. Self-supervised contrastive learning has been shown to be capable of learning rich representations with more semantic latents in RL (Srinivas, Laskin, and Abbeel 2020; Liu and Abbeel 2021b), and CUDC follows this intuition to estimate the probability l_i of s_{t_i+k} being reachable from s_{t_i} by:

$$l_i = \frac{\text{sim}(\hat{z}_{s_{t_i+k}}, m_{s_{t_i+k}})}{\text{sim}(\hat{z}_{s_{t_i+k}}, m_{s_{t_i+k}}) + \sum_{j=1, j\neq i}^{n} \text{sim}(\hat{z}_{s_{t_i+k}}, m_{s_{t_j+k}})}, \tag{1}$$

where $\text{sim}(a, b) = \exp(h(a)^T W \bar{h}(b))$, n is the batch size, $h(\cdot)$ is a deterministic projection function, W is a hidden weight to compute the similarity between the two projections, and $\bar{h}(\cdot)$ as well as $m(\cdot)$ are respectively the momentum-based moving average of the projection and state feature to ensure consistency and stability (He et al. 2020). The reachability network is updated by minimizing the contrastive loss function $\mathcal{L}_{\text{reach}} = -\sum_{i=1}^{n} \log l_i$ in a self-supervised manner, without manual labeling.

To further improve representation learning, the reachability module includes two inverse models for predicting action feature $\hat{z}_{a_{t_i}}$ and temporal distance $\hat{k}$. Similar to ICM (Pathak et al. 2017) and Disagreement (Pathak, Gandhi, and Gupta 2019), we define $\hat{z}_{a_{t_i}} = f_a(z_{s_{t_i}}, z_{s_{t_i+k}}, k; \theta_a)$ with a backward loss of $||z_{a_{t_i}} - \hat{z}_{a_{t_i}}||_2$. This loss ensures that the encoded features are robust to environment variations that are uncontrollable by the agent. For the inverse model of the k-step, $\hat{k} = f_k(z_{s_{t_i}}, z_{s_{t_i+k}}; \theta_k)$ characterizes the prediction with a distribution $\mathbb{P}(k)$. The inverse model is updated through a cross-entropy loss, which enables the encoders to capture the dynamics information in the encoded features.

By updating its internal belief in a self-supervised way, the agent can learn without the expensive labeling required in supervised learning. Additionally, the proposed reachability module allows the k-step temporal distance to adapt during learning, rather than relying on a fixed value in many existing unsupervised methods. This adaptability is important, as the feature representations of both states and actions become more informative and robust when adjusting the temporal distance of k-step.

The reachability module also computes a curiosity weight w_i for each tuple i as $w_i = 1 - l_i \in [0, 1]$, where l_i is the contrastive loss defined in Equation 1. Intuitively, a large value of w_i means that the agent does not believe the true future state is reachable from the current state, which induces high curiosity due to the conflict with current internal belief. It further indicates that this under-learned transition tuple contains novel information, and the encoders are not capable of extracting meaningful features yet. With this reachability module in place, we can seamlessly enable the agent to perform the task-agnostic dataset collection in a curious manner, which shall be illustrated in the next subsection.

Curiosity-Driven Learning

To clarify, prior works on reachability such as (Savinov et al. 2019) only incorporate reachability as an intrinsic reward to encourage diverse exploration. In contrast, our proposed CUDC leverages reachability in multiple stages of learning to promote curiosity-driven learning coherently. Firstly, it adapts the temporal distance, i.e. k-step, to expand the feature space and enhance feature representation with the prediction of future states. Secondly, it incorporates a mixed intrinsic reward to encourage effective exploration in under-learned state-action space with the enhanced representation. Lastly, it regularizes the critic-actor update for the backbone DDPG algorithm by utilizing the curiosity weights to focus more on under-learned tuples. Unlike the eight existing methods evaluated in ExORL that only utilize intrinsic rewards as curiosity, our CUDC extends curiosity-driven learning to different RL components, improving task-agnostic data collection coherently.

Enhance Feature Representation with Adaptive Temporal Distances It is worth noting that the eight methods evaluated in ExORL limit the autonomy of the feature space by requiring the agent to reach future states exactly three steps away, i.e., $(s_{t_i}, a_{t_i}, s_{t_i+3})_{i=1}^{n}$. Recent online pre-training RL methods, such as SPR (Schwarzer et al. 2020) and SGI (Schwarzer et al. 2021), predict the agent's own latent state representations multiple steps into the future, improving sample efficiency. However, these methods require iterative predictions by calling the forward dynamic network k times. In contrast, our proposed CUDC enables automatic adjustment of the temporal distance k and performs k-step future state estimation directly, without substantially increasing computational complexity. The key idea is to keep the reachability estimation increasingly challenging with an adaptive k-step, thereby expanding the feature space to learn more meaningful reachability information.

In our approach, we dynamically adjust k to impose more challenging reachability predictions, by leveraging the agent's level of curiosity. Specifically, we increase k by 1 if the agent's curiosity level is low in the current reachability analysis, and we define a threshold C_w for low curiosity and a threshold C_k for the proportion of tuples with low curiosity. Thus, the agent adapts k when the average value of w_i is below C_w for more than C_k of the tuples in the batch, as represented by:

$$\tfrac{1}{n} \textstyle\sum_{i}^{n} \mathbb{1}_{\mathrm{w_i < C_w}} > C_k. \tag{2}$$

The rationale behind this approach is that when the agent can estimate the current k-step reachability well for the majority of tuples in the batch, it should be encouraged to explore further. By expanding the feature space to learn the dynamics of more distant future states, the feature representation can be enhanced, leading to more informative and diverse task-agnostic data collection. It is worth noting that there are other possible ways to vary the k-step, such as by sampling from a probabilistic distribution. To validate the effectiveness of our proposed curiosity-driven method compared to other sampling-based methods, we conduct an ablation study in Figure 4.

Incorporate a Mixed Intrinsic Reward CUDC utilizes a mixed intrinsic reward that combines state-action entropy and prediction error of future states. While previous methods like APT (Liu and Abbeel 2021b) and RE3 (Seo et al. 2021) have demonstrated that particle-based k-nearest neighbors state entropy can encourage agents to explore the state space more uniformly, we believe that exploration should not be limited to the state space alone, but should also extend to the action space. To achieve this, CUDC expands state embedding to state-action embedding and shows that entropy maximization can be applied to the k-nearest neighbor entropy estimation in the state-action representation space in Lemma 0.1. This approach encourages the agent to explore both the state and action spaces more diversely, leading to more effective and informative data collection.

Lemma 0.1. *Let $u = (z_s, z_a)$ represent the state-action representation. The particle-based entropy $\mathcal{H}(u)$ is proportional to a K-nearest neighbor (K-NN) distance,*

$$\mathcal{H}(u) \propto \sum_{i=1}^{n} \log ||u_i - u_i^{\textit{K-NN}}||_2.$$

Proof. A proof is provided in Appendix[1] B. □

We build on the idea of treating each tuple as a particle (Liu and Abbeel 2021b; Seo et al. 2021) and propose an intrinsic reward to estimate particle-based entropy, defined as $r_{\mathcal{H}}(s_{t_i}, a_{t_i}) = \log(\frac{1}{N_K} \sum ||u_i - u_i^{\text{K-NN}}||_2 + 1)$, where $u_i = (\phi_s(s_{t_i}), (\phi_a(a_{t_i}))$, N_K is the number of K-NN, and ϕ_s and ϕ_a are state and action encoders respectively. Since the encoded features are constantly updated to capture the dynamics of more distant future states in the reachability module, the proposed $r_{\mathcal{H}}$ promotes diverse state-action space exploration. This is consistent with the entropy maximization principle (Singh et al. 2003) and has been shown to be effective in the state space using the state-of-the-art off-policy RL algorithm SAC (Haarnoja et al. 2018).

Additionally, we integrate prediction error of future states as another component of the intrinsic reward to incentivize the agent to explore surprising states beyond its expectations (Pathak et al. 2017; Burda et al. 2018). Specifically, we use $r_{\mathcal{E}}(s_{t_i}, a_{t_i}) = ||z_{s_{t_i+k}} - \hat{z}_{s_{t_i+k}}||_2$, where the reachability module is conveniently re-used without additional networks. Finally, the mixed intrinsic reward in CUDC is given by

$$r_i(s_{t_i}, a_{t_i}) = r_{\mathcal{H}}(s_{t_i}, a_{t_i}) + \alpha r_{\mathcal{E}}(s_{t_i}, a_{t_i}) + \beta, \quad (3)$$

where α prioritizes under-learned state exploration and β is a constant for numerical stability.

Regularize the critic-actor update Furthermore, CUDC utilizes the curiosity weight w_i to adaptively regularize the backbone DDPG algorithm, allowing it to focus more on under-learned tuples. The weight $w = (w_1, w_2, \cdots, w_n)$ quantitatively characterizes the curiosity weight of each transition tuple, which can be used to determine sample importance and regularize both critic and actor updates. Therefore, the Q-learning in DDPG can be performed by minimizing the following objective,

$$\mathbb{E}_{\cdot\sim\mathcal{D}}\left[w\left(Q(s_t, a_t) - (r_i(s_t, a_t) + \gamma Q_{\text{target}}(s_{t+k}, \pi(s_{t+k})))\right)^2\right]. \quad (4)$$

Meanwhile, the policy can be updated by maximizing $\mathbb{E}_{\cdot\sim\mathcal{D}}\left[wQ(s_t, \pi(s_t))\right]$. In this way, CUDC enables the agent to adapt its learning process in a self-supervised manner by using the conceptualized curiosity to exploit sample importance.

```
Algorithm 1: Implementation of the proposed CUDC
Initialize parameters of encoders $\phi_s$ and $\phi_a$, forward dynamic $f_s$, inverse models $f_a$ and $f_k$, projection $h$, critic $Q$, policy $\pi$, hidden weight $W$, temporal distance $k$, batch size $n$, and an empty dataset $\mathcal{D} = \emptyset$
  for each time step $t$ do
    // COLLECT TRANSITIONS
    Interact with the environment using the policy $a_t \sim \pi(s_t)$ and observe $s_{t+1}$
    $\mathcal{D} \cup (s_t, a_t, s_{t+1}) \rightarrow \mathcal{D}$
    // UPDATE INTERNAL BELIEF
    Sample a minibatch $\{(s_{t_i}, a_{t_i}, s_{t_i+k}, k)\}_{i=1}^{n} \sim \mathcal{D}$
    for each tuple $i$ in the minibatch do
      Encode the state and action, and predict the $t_i + k$'s future state feature $\hat{z}_{s_{t_i+k}}$
      Evaluate the curiosity weight $w_i = 1 - l_i$ by Eq. (1)
      Compute the intrinsic reward $r_i$ using Eq. (3)
    end for
    Update the internal belief of the reachability module
    //ADAPT THE K-STEP TO PREDICT
    if $\frac{1}{n}\sum_i^n \mathbb{1}_{\mathrm{w_i < C_w}} > C_k$ then
      Increase the temporal distance by $k = k + 1$
    end if
    //REGULARIZE CRITIC-ACTOR UPDATE
    Update the critic $Q$ with regularization by Eq. (4)
    Update the actor $\pi$ with regularization
    Perform the momentum update for $\bar{h}$ and $m$
  end for
```

Experiments

Environments We evaluated on a set of challenging continuous control tasks with state observations, drawn from the DeepMind control suite (Tassa et al. 2018). The suite contains 12 downstream tasks, organized into three main domains: Walker, Quadruped, and Jaco Arm. Walker is a controllable entity with locomotion-related balancing controls, where it can learn to walk, run, flip, and stand. Quadruped is a passively stable body in a more challenging 3D environment, which requires learning various locomotion skills such as walking, running, standing, and jumping. Jaco Arm is a six-degree-of-freedom robotic arm with a three-finger gripper for object manipulation, where the downstream tasks require it to reach different positions. Note that the PointMass Maze task is not included, as most baseline methods in ExORL have already demonstrated excellent performances on it.

Baseline Models We compare CUDC against state-of-the-art unsupervised methods across all three categories as benchmarked in ExORL, i.e., a knowledge-driven baseline of ICM (Pathak et al. 2017), data-driven baselines of APT (Liu and Abbeel 2021b) and ProtoRL (Yarats et al. 2021), and a competence-driven baseline of APS (Liu and Abbeel 2021a). Meanwhile, a random data collector is also included, which collects the data by performing randomly sampled actions. The other four methods discussed in ExORL are excluded since their performance are less competitive. We use the same hyperparameters and model architecture as reported in ExORL to ensure a fair comparison. To demonstrate that all proposed components play important roles in the performance, we also compare four versions of CUDC. CUDC$^{\text{ICM}}_{\text{vary}}$

[1] Full appendix is available at https://arxiv.org/abs/2312.12191.

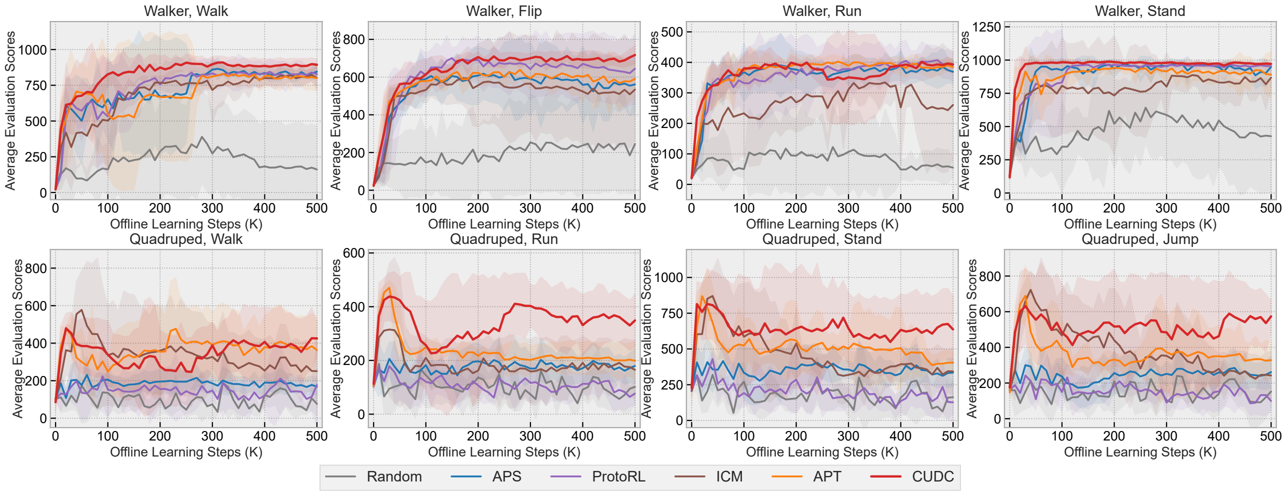

Figure 2: Learning curves of the offline RL agent on the task-agnostic dataset collected by different methods. CUDC demonstrates the superior capability of improving the computational efficiency and learning performances of the offline RL agent.

and $\text{CUDC}^{\text{APT}}_{\text{vary}}$: adapting the temporal distance of k-step by the intrinsic rewards based on the original ICM and APT methods. $\text{CUDC}_{\text{reward}}$: extending to state-action entropy with a mixed intrinsic reward based on $\text{CUDC}^{\text{APT}}_{\text{vary}}$. $\text{CUDC}_{\text{reach}}$: adding the full reachability module without regularization based on $\text{CUDC}_{\text{reward}}$. The detailed implementation and differences from the full model are summarized in Appendix A.

Model Training and Evaluation To ensure model stability during learning, we have restricted the temporal distance k to be increased from 3 to 6 and have set upper and lower bounds for the regularization weights to guarantee stability. For further details regarding the network implementation and hyperparameter setting of the proposed CUDC, readers can refer to Appendix A. During data collection, all methods have been trained using a DDPG (Lillicrap et al. 2015) agent as the backbone to ensure fairness. They have interacted with 3 domain environments in the absence of extrinsic rewards for 1M steps. For the main results, a total of 90 datasets (6 algorithms $\times$ 3 main tasks $\times$ 5 seeds) have been collected. Afterwards, relabeling has been performed for each downstream task. During the evaluation, a TD3 (Fujimoto, Hoof, and Meger 2018) agent learns offline from each relabeled dataset for 500K steps. We report the performance score at 100K steps for computational efficiency and at 500K steps for learning performance.

Main Results on 12 Downstream Tasks Figure 2 indicates that ProtoRL performs well in the Walker domain but fails in the Quadruped domain. Similarly, all the other baseline methods cannot collect consistent high-quality datasets for all domains. In contrast, the dataset collected by CUDC demonstrates a higher quality with an expanded feature space, as the offline agent's learning performances at 500K steps are enhanced in all 12 downstream tasks across the 3 challenging domains, as highlighted in Table 3 of Appendix C. Specifically, CUDC outperforms the competence-based method (APS) in the Walker domain by 6%, outperforms the data-based method (APT) in the Quadruped domain by 51%, and outperforms the knowledge-based method (ICM) in the Jaco Arm domain by 10%. In terms of efficiency, Figure 2 shows significant improvements of CUDC on 3 downstream tasks of the Quadruped domain, indicating improved computational efficiency. In the easiest domain of Walker, CUDC helps the offline agent to converge faster in 3 downstream tasks. However, the computational efficiency in the Jaco Arm domain is unsatisfactory. This could be due to too much complexity in this most challenging environment, increasing the difficulty of reachability analysis. A visualization of the quality for the collected datasets is provided in Appendix C.1, where our proposed method has collected higher-quality dataset with increasingly more rewarding states being visited. For the sample efficiency, the offline RL agent can perform well with significantly less data collected by the proposed CUDC as discussed in Appendix C.2. Additional results are presented in Appendix C.1 and consistent results are obtained by evaluating with another offline RL algorithm of CQL (Kumar et al. 2020) in Appendix C.3.

Effects of Adapting the k-Step We empirically show that adapting the temporal distance to explore more distant future states can enhance the feature representation, and thereby improve the data collection process. By comparing the results in Figure 3, $\text{CUDC}^{\text{ICM}}_{\text{vary}}$ has outperformed ICM significantly, with on average 1.25 $\times$ computational efficiency at 100K step and 1.16 $\times$ offline learning performance at 500K step. Similarly, $\text{CUDC}^{\text{APT}}_{\text{vary}}$ obtains respectively 1.12 $\times$ and 1.04 $\times$ scores at 100K and 500K steps across 4 downstream tasks, compared with APT. Note that the standard deviation increases slightly, which may be due to the introduced complexity of considering more distant future states in improving the learned representation. Thus, it is important to find an adaptive way to smooth this process, such as by incorporating the other proposed components coherently.

Effectiveness of the Other Proposed Components We additionally integrated mixed intrinsic reward into $\text{CUDC}^{\text{APT}}_{\text{vary}}$ as $\text{CUDC}_{\text{reward}}$, resulting in further improvements in learn-

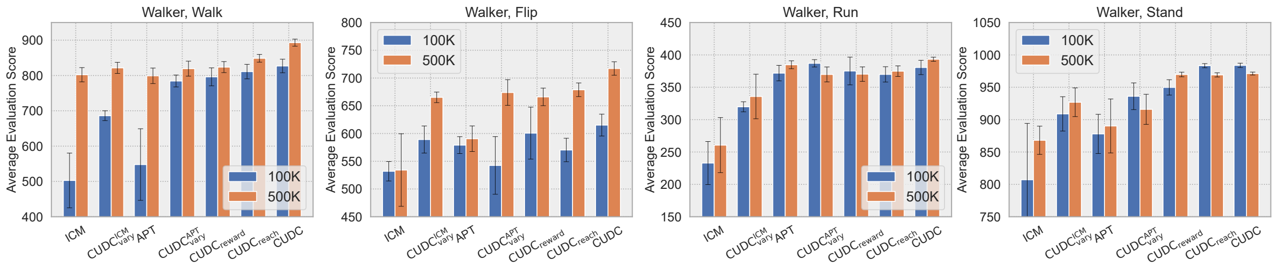

Figure 3: The performance score evaluated at 100K and 500K steps in 4 downstream tasks of Walker. All four versions of CUDC perform better than ICM and APT.

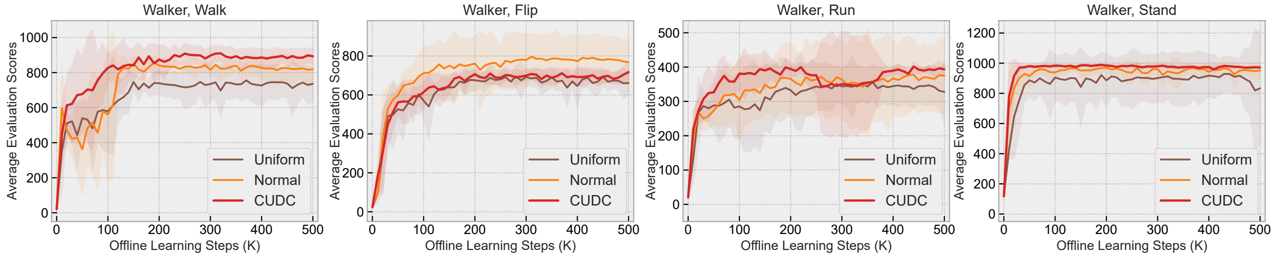

Figure 4: Learning curves of the offline RL agent on 4 downstream tasks of Walker. The k-step adaptation proposed in CUDC outperforms the other two sampling methods in 3 out of 4 downstream tasks.

ing efficiency at 100K steps by 3.3% and capability at 500K steps by 3.0% for the offline RL agent, as presented in Figure 3. However, due to the mixed intrinsic reward's nature of promoting uniform exploration in the state-action space and focusing on under-learned states, the performance at 100K steps became unstable with a 67% increase in standard deviation. Thus, we leveraged the proposed reachability module to function as the agent's internal belief and facilitate the data collection process. Comparing CUDC$_{reach}$ and CUDC$_{reward}$, the dataset collected by CUDC$_{reach}$ reduced standard deviation by 48% and 25% at 100K and 500K steps, respectively, stabilizing offline learning. However, its performance scores decreased slightly in two tasks. The full model, compared to CUDC$_{reach}$, further regularizes the critic-actor update with the curiosity weight to focus more on under-learned tuples, resulting in a 3.2% and 4.0% improvement in learning efficiency and capability, respectively, with the minimum standard deviation at 500K steps. To further investigate the effectiveness, we carry out more experiments by respectively removing each proposed component from the full model in Appendix C.5. It can be concluded that varying the temporal distance is the most crucial factor in collecting a useful dataset with an expanded feature space, while the other components work coherently to yield further improvement.

Adjusting the k-Step in Different Ways One may be curious about how adjusting the temporal distance k in the reachability module affects the feature representation. To investigate this, we conducted an ablation study in the Walker domain by sampling k uniformly (Uniform) from 3 to 6 and normally (Normal) with an increasing mean. The results in Figure 4 show that Uniform performs the worst in all 4 tasks as it cannot adapt the temporal distance in a way that enhances representation learning. At 500K steps, it only achieves 85% overall learning capability with a 300% increase in standard deviation, compared to CUDC. Normal to some extent adapts k through an increasing mean, and it even outperforms CUDC in the Flip task. However, its overall performance is still 4.5% weaker than CUDC, and its standard deviation is 128% higher than CUDC, indicating an instability issue. Overall, the curious adaptation method proposed in CUDC is the best, and there is potential to investigate more adaptive ways in the future.

Limitations and Broader Impacts Despite demonstrating strong empirical performance, CUDC is not without its limitations. Like other unsupervised methods, its scalability to complex environments may be limited, and in safety-critical applications where expert data is crucial, relying solely on unsupervised approaches can pose risks. From an ethical standpoint, the application of CUDC in real-world scenarios, such as robotics, AI video games, or social media platforms, raises concerns. The process of diverse data collection without proper supervision or restrictions can give rise to potential safety and privacy issues. It is important to address these ethical considerations to ensure the responsible and safe implementation of the CUDC method in practical applications.

Conclusion

We propose CUDC, a curiosity-driven unsupervised data collection method for multi-task offline RL. It enhances dataset quality by dynamically expanding the feature space. CUDC's reachability module estimates the probability of reaching a k-step future state from the current state, allowing adaptive exploration of distant future states. This improves feature representation, outperforming existing benchmarks in computational and sample efficiency. Our work offers valuable insights for future research in effective data collection.

Acknowledgments

This research is supported by Alibaba Group and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. H.Qian thanks the support from CFAR.

References

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2018. Large-Scale Study of Curiosity-Driven Learning. In *ICLR*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by random network distillation. In *ICLR*, 1–17.

Campero, A.; Raileanu, R.; Kuttler, H.; Tenenbaum, J. B.; Rocktäschel, T.; and Grefenstette, E. 2020. Learning with AMIGo: Adversarially Motivated Intrinsic Goals. In *ICLR*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need. In *ICLR*.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *ICML*, 1587–1596. PMLR.

Gulcehre, C.; Wang, Z.; Novikov, A.; Paine, T.; Gómez, S.; Zolna, K.; Agarwal, R.; Merel, J. S.; Mankowitz, D. J.; Paduraru, C.; et al. 2020. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *NeurIPS*, 33: 7248–7259.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Ivanovic, B.; Harrison, J.; Sharma, A.; Chen, M.; and Pavone, M. 2019. Barc: Backward reachability curriculum for robotic reinforcement learning. In *ICRA*, 15–21. IEEE.

Jiang, M.; Grefenstette, E.; and Rocktäschel, T. 2021. Prioritized level replay. In *ICML*, 4940–4950. PMLR.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Kumar, A.; Hong, J.; Singh, A.; and Levine, S. 2022. Should I Run Offline Reinforcement Learning or Behavioral Cloning? In *ICLR*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *NeurIPS*, 33.

Lambert, N.; Wulfmeier, M.; Whitney, W.; Byravan, A.; Bloesch, M.; Dasagi, V.; Hertweck, T.; and Riedmiller, M. 2022. The Challenges of Exploration for Offline Reinforcement Learning. *arXiv preprint arXiv:2201.11861*.

Laskin, M.; Yarats, D.; Liu, H.; Lee, K.; Zhan, A.; Lu, K.; Cang, C.; Pinto, L.; and Abbeel, P. 2021. URLB: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*.

Lee, L.; Eysenbach, B.; Parisotto, E.; Xing, E.; Levine, S.; and Salakhutdinov, R. 2019. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, H.; and Abbeel, P. 2021a. Aps: Active pretraining with successor features. In *ICML*, 6736–6747. PMLR.

Liu, H.; and Abbeel, P. 2021b. Behavior from the void: Unsupervised active pre-training. *NeurIPS*, 34: 18459–18473.

Markey, A.; and Loewenstein, G. 2014. Curiosity. *International handbook of emotions in education*, 238–255.

Motamedi, M.; Sakharnykh, N.; and Kaldewey, T. 2021. A data-centric approach for training deep neural networks with less data. *arXiv preprint arXiv:2110.03613*.

Patel, H.; Guttula, S.; Mittal, R. S.; Manwani, N.; Berti-Equille, L.; and Manatkar, A. 2022. Advances in Exploratory Data Analysis, Visualisation and Quality for Data Centric AI Systems. In *ACM SIGKDD*, 4814–4815.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2778–2787. PMLR.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-supervised exploration via disagreement. In *ICML*, 5062–5071. PMLR.

Péré, A.; Forestier, S.; Sigaud, O.; and Oudeyer, P.-Y. 2018. Unsupervised Learning of Goal Spaces for Intrinsically Motivated Goal Exploration. In *ICLR*.

Prudencio, R. F.; Maximo, M. R.; and Colombini, E. L. 2022. A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. *arXiv preprint arXiv:2203.01387*.

Savinov, N.; Raichuk, A.; Vincent, D.; Marinier, R.; Pollefeys, M.; Lillicrap, T.; and Gelly, S. 2019. Episodic Curiosity through Reachability. In *ICLR*.

Schwarzer, M.; Anand, A.; Goel, R.; Hjelm, R. D.; Courville, A.; and Bachman, P. 2020. Data-Efficient Reinforcement Learning with Self-Predictive Representations. In *ICLR*.

Schwarzer, M.; Rajkumar, N.; Noukhovitch, M.; Anand, A.; Charlin, L.; Hjelm, R. D.; Bachman, P.; and Courville, A. C. 2021. Pretraining representations for data-efficient reinforcement learning. *NeurIPS*, 34: 12686–12699.

Seo, Y.; Chen, L.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2021. State entropy maximization with random encoders for efficient exploration. In *ICML*, 9443–9454. PMLR.

Singh, B.; Kumar, R.; and Singh, V. P. 2022. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, 55(2): 945–990.

Singh, H.; Misra, N.; Hnizdo, V.; Fedorowicz, A.; and Demchuk, E. 2003. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4): 301–321.

Srinivas, A.; Laskin, M.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*.

Sun, C.; and Miao, C. 2022. CD-SLFN: A Curiosity-Driven Online Sequential Learning Framework with Self-Adaptive Hot Cognition. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Sun, C.; Qian, H.; and Miao, C. 2022a. CCLF: A Contrastive-Curiosity-Driven Learning Framework for Sample-Efficient Reinforcement Learning. In *IJCAI-22*, 3444–3450. Main Track.

Sun, C.; Qian, H.; and Miao, C. 2022b. From Psychological Curiosity to Artificial Curiosity: Curiosity-Driven Learning in Artificial Intelligence Tasks. *arXiv preprint arXiv:2201.08300*.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Wu, Y.; Zhai, S.; Srivastava, N.; Susskind, J. M.; Zhang, J.; Salakhutdinov, R.; and Goh, H. 2021. Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning. In *ICML*, 11319–11328. PMLR.

Yarats, D.; Brandfonbrener, D.; Liu, H.; Laskin, M.; Abbeel, P.; Lazaric, A.; and Pinto, L. 2022. Don't Change the Algorithm, Change the Data: Exploratory Data for Offline Reinforcement Learning. *arXiv preprint arXiv:2201.13425*.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021. Reinforcement learning with prototypical representations. In *ICML*, 11920–11931. PMLR.

Yu, D.; Ma, H.; Li, S.; and Chen, J. 2022. Reachability Constrained Reinforcement Learning. In *ICML*, 25636–25655. PMLR.