Incremental Reinforcement Learning with Dual-Adaptive ϵ-Greedy Exploration

Wei Ding[*], **Siyang Jiang**[*], **Hsi-Wen Chen**[*], **Ming-Syan Chen**
Graduate Institute of Electrical Engineering, National Taiwan University, Taiwan
{wding, syjiang, hwchen}@arbor.ee.ntu.edu.tw, mschen@ntu.edu.tw

Abstract

Reinforcement learning (RL) has achieved impressive performance in various domains. However, most RL frameworks oversimplify the problem by assuming a fixed-yet-known environment and often have difficulty being generalized to real-world scenarios. In this paper, we address a new challenge with a more realistic setting, *Incremental Reinforcement Learning*, where the search space of the Markov Decision Process continually expands. While previous methods usually suffer from the lack of efficiency in exploring the unseen transitions, especially with increasing search space, we present a new exploration framework named *Dual-Adaptive ϵ-greedy Exploration (DAE)* to address the challenge of Incremental RL. Specifically, DAE employs a *Meta Policy* and an *Explorer* to avoid redundant computation on those sufficiently learned samples. Furthermore, we release a testbed based on a synthetic environment and the Atari benchmark to validate the effectiveness of any exploration algorithms under Incremental RL. Experimental results demonstrate that the proposed framework can efficiently learn the unseen transitions in new environments, leading to notable performance improvement, i.e., an average of more than 80%, over eight baselines examined.

Introduction

Reinforcement learning (RL) methods mainly aim at training agents to conduct continuous control and decision-making tasks and have demonstrated encouraging performance improvement in various domains, including game playing (Baker et al. 2019; Hu et al. 2021), autonomous driving (Kiran et al. 2021), and robot controlling (Won, Gopinath, and Hodgins 2021). Despite the recent progress in RL, it remains challenging to train the RL agents through extensive interactions with the environments. Thus, several early efforts focus on improving the sample efficiency (Schulman et al. 2017; Van Hasselt, Hessel, and Aslanides 2019; Kaiser et al. 2019), exploration strategies (Burda et al. 2018; Zhang et al. 2021; Ermolov and Sebe 2020) and value estimation (Hessel et al. 2018; Badia et al. 2020) to strengthen the capability of RL agents. However, most prior works assume that the environment remains unchanged, hence having difficulty being generalized to new scenarios. In contrast, the sets of states and actions would usually be enlarged by means that we cannot foresee in the future since real-world applications update from time to time (Wang et al. 2019). One straightforward idea is to retrain the agents as the environments vary, which is undesirable due to the computational overhead of the training process (Kaiser et al. 2019). For example, an autonomous vehicle company (Kiran et al. 2021) may first train an agent to drive the car following the lanes. In the next version, they may want to launch a new desirable feature on this agent to recognize traffic lights and intersections. Under the conventional RL setting, the training process has to start over again when developing new states and actions. Note, however, that humans can effectively learn a task from their previous experience and inference the strategies to unseen situations (Wang et al. 2020).

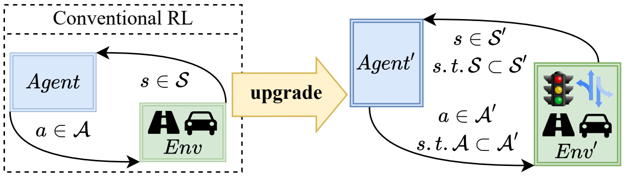


Figure 1: Illustration of Incremental RL. Conventional RL (left) has fixed state and action spaces. Whereas in Incremental RL, state and action spaces can be upgraded into larger ones whenever the task of interest is updated.

Based on these observations, we formulate a new RL problem, named *Incremental Reinforcement Learning (Incremental RL)*, where the agents can adapt their behavior incrementally as the environment changes, utilizing previous knowledge to benefit the future decision-making process. A close topic, i.e., lifelong reinforcement learning (Brunskill and Li 2014; Abel et al. 2018), trains an agent for a sequence of similar tasks and encourages the agent to transfer the experience from previous tasks to the new one faster. However, the works mentioned above learn the policy from the environment with *fixed-yet-known* state and action sets. Thus, it is nontrivial to solve Incremental RL because the search space of the solution would grow exponentially, corresponding to the size of the state and action spaces. While one can initialize the parameters of a function approximation

[*] These authors contributed equally.

from the previous environment, the previous global optimum may be a local one that barely fits the latter environment. In light of this, we investigate how to utilize prior knowledge to quickly adapt the agent to new state and action spaces. As illustrated in Fig. 1, Incremental RL emerges by adjusting the previously learned policy to new environments with the state and action spaces continually enlarged incrementally.

To address the challenges mentioned above, Incremental RL can be regarded as a special exploration problem of RL under an inherent bias from the previous training process. Precisely, after an agent has been trained within a specific environment and the corresponding state, action spaces increase, the goal of that agent would then be to maintain past experiences while exploring new transitions, i.e., state-action pairs. Thus, aiming at this target, we propose a novel exploration algorithm named *Dual-Adaptive ϵ-greedy Exploration (DAE)*. DAE takes advantage of two strategies, the *Meta Policy* and the *Explorer*, where the Meta Policy adaptively sets the value of ϵ by assessing the exploration convergence of the current state; and the Explorer estimates the occurrence of actions, given the current state, and adaptively explore the least-tried actions. Furthermore, we release a new testbed based on an exponential-growing environment and the Atari benchmark (Mnih et al. 2013) to evaluate the efficiency of any algorithms under Incremental RL, including the one we proposed, DAE.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to formally model and formulate the challenge of *Incremental Reinforcement Learning (Incremental RL)*.
- Accordingly, we propose a novel exploration algorithm for Incremental RL called *Dual-Adaptive ϵ-greedy Exploration (DAE)*, which can (1) adaptively make a trade-off between exploitation/exploration, and (2) give adaptive exploration guidance to the agent.
- Two benchmarks of Incremental RL and the corresponding baselines are proposed to accompany the challenges. Experiments show that DAE can efficiently solve Incremental RL compared to the eight baselines examined.

Problem Formulation

Markov Decision Process

Markov Decision Process (MDP) (Puterman 1990) can be defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the set of all possible states, $\mathcal{A}$ is the action space of all available actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition function that maps states and actions to the probability distribution of state transitions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to r$ is the predefined reward function, and lastly, $\gamma \in [0, 1]$ is the discount factor. For each time step t, an agent with policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is to interact with the environment as follows: the current state $s_t \in \mathcal{S}$ is returned by the environment, and an action sampled from the policy $a_t \sim \pi(s_t)$ is conducted. Following, the reward r_t and next state $s_{t+1} \sim \mathcal{T}(s_t, a_t)$ are yielded back from the environment. Overall, the goal of the agent is to maximize the discounted accumulated reward, the estimation function of which is called the value function $\mathcal{V}$ or the action-value function $\mathcal{Q}$ in value-based reinforcement learning:

$$\mathcal{V}_\pi(s) = \max_{a \sim \mathcal{A}} \mathcal{Q}_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s \right] \quad (1)$$

Deep Q-Learning

In order to maximize the discounted accumulated reward, we first obtain a function of the $\mathcal{Q}$-value. With sufficient state-action pairs, we can learn the $\mathcal{Q}$-value of the policy π by satisfying the *Bellman optimality equation*:

$$\mathcal{Q}_\pi(s_t, a_t) = \mathcal{R}(s_t, a_t) + \gamma \max_{a_{t+1} \sim \mathcal{A}} \mathcal{Q}_\pi(s_{t+1}, a_{t+1}) \quad (2)$$

In fact, the combinations of $\mathcal{S}$ and $\mathcal{A}$ are usually countless to be tabularly memorized by a machine. Thence, agents of deep architectures dominate the domain of reinforcement learning by the general $\mathcal{Q}$-value of all state-action pairs. Under the deep Q-learning setting, one can update $\mathcal{Q}_\pi$ of the policy π_θ by performing gradient descent with MSE of Equation (2) and with respect to the parameters θ of the policy. After building a model to estimate the $\mathcal{Q}$-value, we can optimize the policy with experiences collected by interacting with the environment and acting greedily.

While greedily exploiting highest-rewarded actions may lead to sub-optimal behavior, the exploration allows an agent to improve its current knowledge about each state-action pair, leading to long-term benefits. ϵ-greedy (Sutton 1995) is a simple but widely-adopted exploration method in RL, which searches the unseen transitions in the environment by choosing random actions under a fixed and small probability ϵ. Nonetheless, it suffers from the lack of efficiency (Dabney, Ostrovski, and Barreto 2021), which could bring the failure of Incremental RL with increasing search space.

Incremental Reinforcement Learning

As the environment continually grows in real-world applications, we introduce a new challenge named *Incremental Reinforcement Learning (Incremental RL)*, which can be formally defined as an MDP. A near-optimal policy π^* is firstly trained on a MDP $\mathcal{M}$ of tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ with $|\mathcal{S}| = n$ and $|\mathcal{A}| = m$. Based on Q-learning, the search space of $\mathcal{M}$ is the number of transitions (state-action pairs): $|\mathcal{S} \times \mathcal{A}| = n*m$. Yet, the MDP $\mathcal{M}$ is extended to $\mathcal{M}'$ afterward, where $\mathcal{M}'$ is a tuple $(\mathcal{S}', \mathcal{A}', \mathcal{R}', \mathcal{T}', \gamma')$ and $\mathcal{S} \subset \mathcal{S}'$, $\mathcal{A} \subset \mathcal{A}'$ and $|\mathcal{S}'| = q > n, |\mathcal{A}'| = k > m$. It is not ambiguous that the search space of the new MDP $\mathcal{M}'$ is strictly greater than the original one, $|\mathcal{S}' \times \mathcal{A}'| = q*k > n*m$. Last but not least, existing transitions are assumed unchanged to focus this study on learning new transitions while preserving learned behavior. Thereby, the reward function $\mathcal{R}'$ and transition function $\mathcal{T}'$ are augmented from the original ones such that the corresponding outputs of new transitions are defined, i.e., $\mathcal{R} \subset \mathcal{R}'$ and $\mathcal{T} \subset \mathcal{T}'$.

While prior RL algorithms assume a fixed-yet-known set of actions to explore the changeless environment (Ostrovski et al. 2017; Burda et al. 2018), the state and action spaces of the agent usually continually increase in practice. When state and action spaces are enlarged, it is inefficient to re-train the policy by re-collecting and re-learning for billions of interactions between the agent and the environment (Abel et al. 2018). To avoid power and time-consuming training, we should better utilize the trained policy and directly finetune the model. In our implementation, to reuse the value estimation of the trained policy, if the input dimension expands, new input neurons are appended into the deep Q-network and initialized with (He et al. 2015) and old states can be regarded as zero-padded (features missing), maintaining their $\mathcal{Q}$-values. This strategy is also applied to other components of the agent. Especially, the $\mathcal{Q}$-values of newly-added actions shall be no larger than existing $\mathcal{Q}$-values to avoid the influence on prior estimations due to the Bellman optimality equation mentioned in Section . Accordingly, new output neurons are inserted into the policy network with their weights and biases initialized by a zero-mean and small-variance normal distribution.

Dual-Adaptive ϵ-greedy Exploration

As we defined in Section , the new MDP $\mathcal{M}'$ is the old MDP $\mathcal{M}$ with an incremental expansion. Intuitively, the dynamics of the state-action pairs observed in the early MDP remain unchanged in the latter MDP, and only novel transitions are worth exploring. A naive but effective way to solve Incremental RL may be to maintain the behavior of the agent in transitions $\mathcal{S} \times \mathcal{A}$ and to explore those unseen transitions $(\mathcal{S}' \times \mathcal{A}') - (\mathcal{S} \times \mathcal{A})$. Nevertheless, the previously trained agent has its default trajectory in $\mathcal{M}$, which can be seen as an *initialization bias* of $\mathcal{M}'$ (Dabney, Ostrovski, and Barreto 2021). It could result in the difficulty of finding a near-optimal policy of $\mathcal{M}'$ based on and against the greedy policy of $\mathcal{M}$. Thence, it is one straightforward perspective to deem Incremental RL as an *exploration problem under the strong inductive bias of prior experience.*

Compared to the conventional RL problems, several challenges arise in Incremental RL. First, the algorithm has to automatically determine when to conduct exploitation and exploration as the environment changes. Second, the exploration scheme has to correctly estimate the least but worth-tried actions to reduce the sampling overhead from repeated trying. To efficiently solve Incremental RL, we transform it into a special exploration problem and propose a new exploration framework, *Dual-Adaptive ϵ-greedy Exploration (DAE)*, to address the above challenges.

Adaptive Exploitation-Exploration Trade-off

In vanilla ϵ-greedy Exploration, ϵ is often a scalar that is fixed throughout the training. Though some prior works use decaying ϵ to gradually reduce the global exploration (Dabney, Ostrovski, and Barreto 2021), these methods neglect the differences between states and could cause over-exploration to those well-explored states and under-exploration to the others. While Value-Difference Based Exploration (Tokic 2010) considers state-dependent exploration rate, the static ϵ-TD-Error mapping could be inapplicable to the expanding state space and reward function in Incremental RL.

To adaptively make a trade-off between exploitation and exploration, we propose the **Meta Policy** Ψ, which is a heuristic method that determines the continuous variable ϵ_t, taking states as inputs: the Meta Policy would yield a smaller exploration probability ϵ_t if a state is well-explored, otherwise, a larger ϵ_t would be returned,

$$\epsilon_t = \Psi(s_t),\ s.t.\ 0 \leq \Psi(s_t) \leq 1, \forall s_t \in \mathcal{S} \tag{3}$$

, which is fashioned into a binary classification task (labels of 0 for well-explored states and 1 for under-explored states) and learned by gradient descent with Binary Cross-Entropy Loss. Since there is a lack of ground truth to determine whether the model should prefer exploitation or exploration, our Ψ is a trainable model with a pseudo ground truth y for the given state defined as follows:

$$y = \begin{cases} 1, & \text{if TD-Error rate } > \tau \\ 0, & \text{otherwise} \end{cases},$$

Especially, *TD-Error rate* refers to the absolute error rate of the LHS and the RHS of Eq. 2, indicating the value-estimation convergence (Tokic 2010), i.e., a high *TD-Error rate* implies that the value estimation may not have converged, and more exploration might be required to fulfill the agent's knowledge in the current state. (Note that we cannot directly utilize TD-Error rates since they are available only after action selection.) Then, y is set to 1 to allow more exploration if the *TD-Error rate* is higher than a hyper-parameter τ, serving as a threshold of uncertainty, or 0, otherwise. In particular, we only update Ψ when exploration is conducted because the Meta Policy is to evaluate whether a state is well-explored or not and encourage the agent to be more explorative in those states with more uncertainty.

Adaptive Action Exploration

ϵ-greedy uniformly selects random actions when exploring, which causes inefficient sampling. To avoid the aforementioned issue and to only explore the rarely-taken actions, we propose an **Explorer** Φ to estimate the occurrence number of all available actions given a state. Unfortunately, the combinations of the state and action spaces are often innumerable and even increasing in Incremental RL thus hard to be analyzed in practice. So instead, given the current state, we estimate the underlying counting of available actions in a normalized form, referred to as the *Relative Frequency (RF).*

$$\begin{aligned} &\Phi(a|s_t) \sim RF(a|s_t), \\ &s.t. \sum \Phi(a|s_t) = 1, \Phi(a|s_t) >= 0, \forall a \in \mathcal{A}. \end{aligned} \tag{4}$$

We adopt a deep neural network followed by a Softmax function as the Explorer Φ, which encodes the states to predict the relative frequency of each action. Note that Φ can be end-to-end trained by gradient ascent with the objective function set to the logarithmic probability of taken actions. An intuition to this update is to increase the relative frequency of any taken action a_t, given the state s_t.

Overall Framework

After introducing the two strategies proposed above, we hereby illustrate the overall framework of *Dual-Adaptive ϵ-greedy Exploration (DAE).*[1] During the training process, the

[1] https://github.com/weiding98/DAE

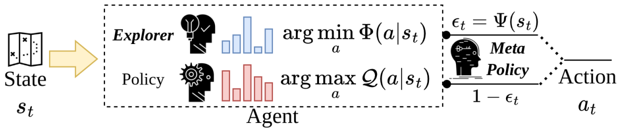


Figure 2: Dual-Adaptive ϵ-greedy Exploration.

exploitation-exploration trade-off is controlled by the value of ϵ as we often see in the regular ϵ-greedy Exploration. Nonetheless, in DAE, ϵ is adaptively inferenced by the Meta Policy Ψ, instead of a global and static value. Then, with the probability of $\epsilon_t = \Psi(s_t)$, the agent will explore by taking the least-tried actions estimated by the Explorer Φ, rather than picking random actions; and with the odd of $1 - \epsilon_t$, greedy actions that exploit current policy will be conducted. It is worth noting that both the above methods are state-conditional, ensuring adaptive exploration for distinct states. Overall, DAE explores more in states with uncertain $\mathcal{Q}$-values by taking rare moves, thus reducing redundant samples of vanilla ϵ-greedy.

To address Incremental RL, the new states in the new environments would be granted higher exploration probability by the Meta Policy because these unseen states would not have been learned and would only be noise to the Meta Policy. The old states would be explored under relatively low ϵ, assuming the previous environment is well-learned and the value estimation has converged. The emergence of new actions may cause the $\mathcal{Q}$-value of old states to be inaccurate and trigger the Meta Policy to explore more in these states again. As a result, the Meta Policy adapts the agent in Incremental RL by exploring more in the new states while exploiting more in the old ones. Besides, the $\mathcal{Q}$-values of those newly-added actions are usually initialized with small values, thus the agents would not likely try those newly-added actions and suffer from the strong inductive bias from prior experience. To overcome such bias, the Explorer would encourage the agent to sample those new actions more frequently by initializing new output neurons with small values, i.e., $\Phi(a|s) \approx 0, \forall a \in \mathcal{A}' - \mathcal{A}$. While conventional exploration methods cannot adapt to the changes of state-action space and reward functions through time (Badia et al. 2020; Tang et al. 2017; Burda et al. 2018), the Explorer is state-sensitive and value-independent, ensuring least-tried actions are sampled and benefiting long-term planning in the expanding environment of Incremental RL. Even though the Explorer favors least-tried actions, these actions will no longer be the least-tried after the relative frequencies of which are increased, preventing another sampling bias. The detailed pseudo codes are presented in Algorithm 1.

Expanding World

We aim to answer the following questions in our experiments (Section , ,). **(1)** As the environment changes, does our method incrementally adjust the previously learned policy to fit the new environment? **(2)** How efficient and performant is our method compared to prior work?

Algorithm 1: Dual-Adaptive ϵ-greedy Exploration

```
Function DAE($\mathcal{Q}, \mathcal{M}$)
    Initialize the Explorer $\Phi$
    Initialize the Meta Policy $\Psi$
    E ← number of episodes
    T ← max steps of an episode
    for episode ← 1 to E do
        Observe $s_0$ from $\mathcal{M}$
        for time step t ← 1 to T do
            if $random(0,1) < \epsilon_t = \Psi(s_t)$ then
                Select novel action
                  $a_t \leftarrow \arg\min_a \Phi(a|s_t)$
                Update $\Psi = \Psi + \alpha\nabla(y \log \Psi(s_t) + (1 - y)\log(1 - \Psi(s_t)))$
            else
                Select greedy action
                  $a_t \leftarrow \arg\max_a Q(a|s_t)$
            Update $\Phi = \Phi + \alpha\nabla \log \Phi(a_t|s_t)$
            Take action $a_t$
            Observe $s_{t+1}$ from $\mathcal{M}$
```

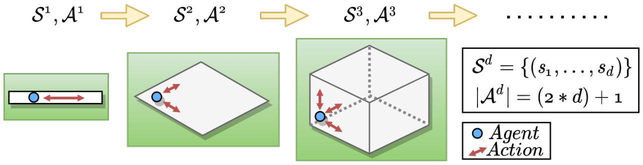

$\mathcal{S}^1, \mathcal{A}^1$
$\mathcal{S}^2, \mathcal{A}^2$
$\mathcal{S}^3, \mathcal{A}^3$
....
$\mathcal{S}^d = \{(s_1, \dots, s_d)\}$
$|\mathcal{A}^d| = (2*d)+1$
Agent
Action

Figure 3: An illustration of Expanding World.

Therefore, we build our first benchmark, Expanding World, to validate whether DAE can efficiently solve Incremental RL.

Setup As illustrated in Fig. 3, we propose a new benchmark of Incremental RL, *Expanding World*, that the state and action spaces continuously expand. This environment consists of a course of length N where the dimension of the state space $d \in \{1, ..., N\}$ increases. The value of each dimension is restricted in $\{0, 1, 2\}$, resulting in the state space.

$$\mathcal{S}^d = \{\mathbf{s}^d\}, \forall\, \mathbf{s}^d = (s_1, ..., s_d) \\ s.t.\ s_i \in \{0,1,2\}, \forall\, i \in \{1, ..., d\} \tag{5}$$

When the environment expands, the corresponding action space becomes

$$\mathcal{A}^d = \{a_1^+, a_1^-, ..., a_d^+, a_d^-, \text{NOOP}\}, \tag{6}$$

including the increment a_*^+ and decrement a_*^- on each dimension, and a no-operation action NOOP. Thus, the number of combinations of state and action spaces exponentially grows up with respect to the number of dimensions, i.e., $|\mathcal{S}^d \times \mathcal{A}^d| = [3^d] * [(2*d)+1]$. Besides, the reward function is defined as follows.

$$\mathcal{R}^d(\mathbf{s}^d) = \begin{cases} 1, & \text{if } \sum_{i=1}^{d} s_i = 2, s.t.\ \forall s_i \neq 1 \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

which indicates a reward of value 1 will be given if and only if an arbitrary dimension is 2 and the others are 0. Each reward will only be given once before the environment resets,

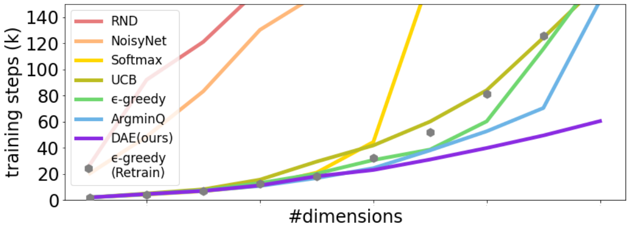


Figure 4: Training overhead on Expanding World: number of training steps taken to attain near-optimal policies while the state and action spaces increase. All agents are NOT initialized when the dimension increases, except for the ϵ-greedy (Retrain) method. The results are averaged over 20 random seeds and clipped at $150K$ training steps.

hence the maximal total reward is equal to the current dimension d. The dimension increases if and only if the maximum reward is achieved in testing. The length of the course N is set to 10 and the environment resets every 500 steps or if the max total reward is reached. The agent will be trained for $200K$ steps in total and be tested every $2K$ steps. The number of training steps taken to reach and conquer each stage of the course will be the assessment of an algorithm's exploration efficiency.

Baselines We employ *DQN* (Mnih et al. 2013) as the framework of the policy and compare DAE with the other six widely-used, state-of-the-art exploration methods.

(1) ϵ-greedy exploration (Sutton 1995) randomly samples actions from an action set with a small probability ϵ, as we discussed in Section . **(2) Upper Confidence Bound (UCB)** (Strehl and Littman 2008) is a count-based exploration method that considers the occurrence of each action and forces the agent to attempt actions that are seldom chosen. **(3) Softmax** (Bridle 1989) samples actions from a probability distribution based on the Q-value. **(4) Noisy Nets** (Fortunato et al. 2018) leverages a noisy stream on top of the conventional linear function. Every time the environment resets, we reset the noisy stream to start the exploring process again. **(5) Random Network Distillation (RND)** (Zhang et al. 2021) is an intrinsically-motivated exploration as it encourages the agent to explore new situations that have rarely been encountered. Every time the environment resets, the target and predictor networks are enlarged. **(6) ArgminQ** is also a variant of ϵ-greedy, which explores by sampling the action with the least Q-value estimated.

Quantitative Results In Fig. 4, the training time of retraining an agent from scratch grows exponentially along with the dimension of the environment. Naive fine-tuning from a previously trained policy with UCB or ϵ-greedy fails to improve the efficiency of finding optimal solutions either. Although UCB considers the frequency of each action and forces the model to select statistically rare actions, the exploration process is not state-sensitive, i.e., globally counts the occurrence of each action. Most interestingly, ArgminQ, sharing a similar idea with DAE, could not elegantly solve Incremental RL either, showing that actions with less Q-value do not necessarily deserve explored. Softmax seriously suffers from the strong learning bias introduced by Incremental RL because those newly added actions would have a relatively low Q-value compared to those well-learned actions, introducing exploration bias to the agent. Despite the self-annealing, state-conditional exploration it maintains, the noisy stream in Noisy Nets could early converge especially when the search space grows, which happens in Incremental RL. In the meanwhile, RND is a curiosity-based exploration strategy, it explores passively by giving bonus rewards only when novel states are found, rather than directly taking novel actions. Thus, the achievement of RND is still based on primitive ones like ϵ-greedy or Softmax and could not be simply transferred to Incremental RL. These three aforementioned methods can barely pass six stages in Expanding World. On the contrary, despite the exponential-growing environment, DAE can solve this nontrivial challenge in near-linear time, i.e., 76.37% and 73.58% less than retraining and ϵ-greedy (best baseline), by adaptive exploitation-exploration trade-off and adaptive action selection of exploration.

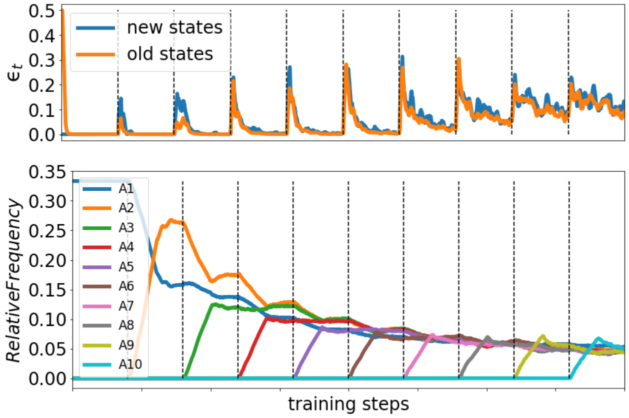


Figure 5: The change of ϵ_t and the *relative frequency* averaged over 20 random seeds. Notably, we consider those newly-available states as the *new states* and others as the *old states* for each stage. Also, newly-added actions are categorized into groups *A1*~*A10*, depending on which stage they were added into the training. The number of training steps for each stage is set to $2 * 10^4$ (grey-dashed lines), for better value convergence. The curves are moving-averaged (Hessel et al. 2018) for more readability.

Further Analysis In Fig. 5, the desired property of DAE is also showcased: the adaptive ϵ_t of the Meta Policy and the adaptive action selection of the Explorer. We can see that every time the environment expands, the Meta Policy Ψ would adaptively assign a higher exploration probability for the agent to adapt itself to the new environment (not only the new states, but the agent would also be more explorative when in the old states since the value estimation would have become more uncertain). Moreover, this behavior would converge by yielding lower ϵ_t when the value function is not varying anymore. The relative frequencies of newly-added actions, estimated by the Explorer, are ex-

	Method	Mean		Median	
		best	final	best	final
RL	Rainbow	5.57	5.02	3.42	2.46
Incremental RL	Rainbow	3.23	3.23	2.11	2.11
	DAE	**6.11**	**6.11**	**3.97**	**3.97**

Table 1: The normalized scores of Incremental Atari.

tremely low, encouraging the agent to sample more for these novel actions. Yet, the distribution of the relative frequency would be near-uniform after the learning of the Explorer i.e., greedy actions would still have higher relative frequency, preventing exploration of these well-learned actions.

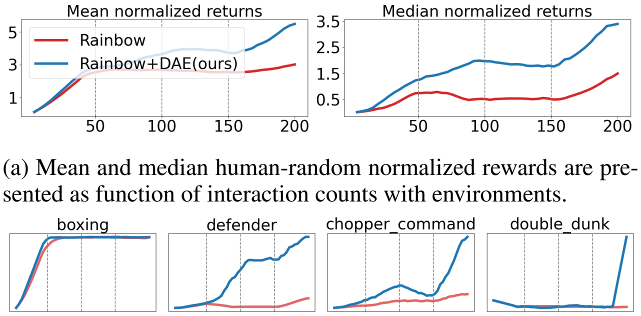
Incremental Atari

Here, we provide another benchmark for Incremental RL based on Atari games (Bellemare et al. 2013) with much more complex dynamics, leading to more training overhead.

Setup Arcade Learning Environment (ALE) is a platform that evaluates RL agents with 57 challenging *Atari2600* games. In Incremental Atari, we carefully select 14 games with different levels of difficulty, each of which has 18 meaningful actions, i.e., all actions in ALE. To simulate the scenario of Incremental RL, only six primitive actions (NOOP, FIRE, UP, RIGHT, LEFT, DOWN) are initially available to enable the agent to play the games. The rest 12 advanced actions are randomly divided into three groups and added into the environment sequentially (any prior knowledge of masked actions is forbidden to meet the setting of Incremental RL). Note that besides the action set, the state set also continually expands as many states need specific actions to be approached. The number of training frames/steps is limited to $200M/50M$, and each group of random extra actions will be included after every $50M$ environment frames (Hessel et al. 2018). Evaluation will be conducted after every $1M$ training steps, where episodic reward averaged over ten individual games will be recorded. For generality, NOOP starts regime (Hessel et al. 2018), which inserts 30 NOOP actions before the agent begins acting, is adopted in testing. We report the mean and median episodic reward of the best and the last agent by human-random normalization since different games achieve different levels of rewards.

Baselines With the complicated dynamics of games, we adopt the state-of-the-art method Rainbow as our baselines in Incremental Atari. Rainbow is widely-adopted for applied-data research (Grigorescu et al. 2020; Zhang, Patras, and Haddadi 2019; Luong et al. 2019) due to its practical hardware requirement, which meets the scenario of Incremental RL in the real world. *Rainbow* (Hessel et al. 2018) consists of 6 extensions of DQN, combined to jointly achieve higher performance in RL, one of them is the Noisy Nets which fails to solve Expanding World in Section . Rainbow and Rainbow with the help of DAE are compared in this benchmark to demonstrate the necessity of DAE.

Quantitative Results Besides Incremental RL, we also include the performance of the baseline method on conventional RL where all actions are available throughout the training, i.e., Rainbow (RL), to demonstrate the exploration overhead Incremental RL may bring. According to Table 1, the performance of Rainbow drops by 35.15% under the setting of Incremental RL, compared to regular RL, showing that conventional exploration strategies cannot properly adjust the policy effectively as the environment changes. In the meanwhile, without being downgraded, our method even outperforms Rainbow (RL) by 22.40%, leading to a 88.76% performance increase upon Rainbow on Incremental Atari averagely. Based on the aforementioned results, our method (DAE) presents the capability of adapting previously learned agents to new environments and the potential of being an exploration-efficient algorithm for conventional RL.



(a) Mean and median human-random normalized rewards are presented as function of interaction counts with environments.

(b) Four representative games in Incremental Atari, i.e., *boxing, defender, chopper command, and double dunk*.

Figure 6: The normalized scores of Incremental Atari. Grey dashed lines indicate increments on the environments. For more readability (Hessel et al. 2018), every curve is smoothed with a moving average of 10.

Further Analysis In Fig. 6b, we also visualize the learning process of four representative cases for detailed analysis. In *boxing*, actions available in the first stage are sufficient for the agent to learn to play the game, while DAE still achieves the optimal solution faster than the baseline. In *defender*, the baseline model barely improves after being granted more choices of actions because the greedy policy burdens the exploration process. Contrarily, with the guidance of DAE, the agent explores the novel transitions which are not possible in previous stages, thus deviating from the sub-optimal trajectory and re-searching for superior policy. In *chopper command*, DAE is able to reach an acceptable result, while the baseline model fails as all actions are given. Last, in *double dunk*, the baseline is stuck in local-optima, whereas DAE finds a much better solution within limited training time in stage four where the agent attains the full action set to play well in this environment. Overall, DAE efficiently explores the rare transitions, as the environment changes even in games of various difficulties, demonstrating the generalizability of our method.

First-Visit Visualization

Besides the two aforementioned Incremental RL benchmarks, we further evaluate the exploration efficiency of DAE for *general RL* via conducting the First-Visit Visualization (Dabney, Ostrovski, and Barreto 2021). These tasks

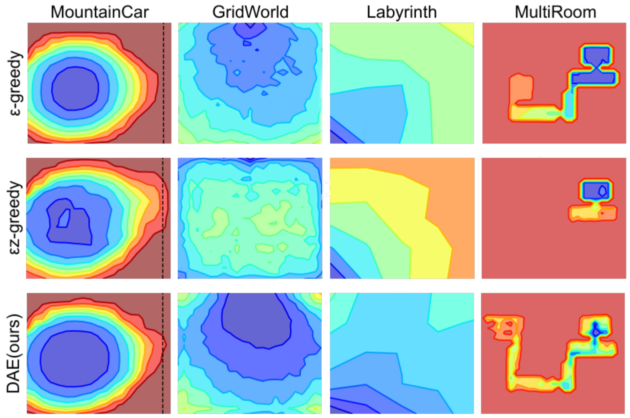


Figure 7: First-Visit Visualization. Blue and green areas take fewer steps to be reached, whereas yellow and red areas take more times.

show the state coverage of an exploration algorithm and how quickly it can discover all of the states. Specifically, the number of steps the agent takes to discover, i.e., first visit, each state are recorded and visualized into heat maps.

Setup Similar to (Dabney, Ostrovski, and Barreto 2021), four small-scale hard-exploration environments are employed, including *MountainCar* (Sutton and Barto 2018), *GridWrold* (Dabney, Ostrovski, and Barreto 2021), *Labyrinth* (Ermolov and Sebe 2020) and *MultiRoom* (Chevalier-Boisvert, Willems, and Pal 2018), to evaluate the state coverage and efficiency.

Baseline The experiments of this section are fully explorative, i.e., $\epsilon = 1$, thus we compare DAE (only the Explorer is active) with only two value-free baselines, *ϵ-greedy* and *ϵz-greedy* (Dabney, Ostrovski, and Barreto 2021). *ϵz-greedy* is another extension of ϵ-greedy, which conducts random actions under probability ϵ with random duration (rather than 1 step in ϵ-greedy) sampled from a heavy-tailed distribution.

Quantitative Results First, *MountainCar* is a power-lack car stuck in a valley, which is a $2D$ environment with the position on the x-axis and the car velocity. Available actions are driving LEFT, RIGHT and NOOP. The goal is to control the car, build momentum, and climb the hill on the right. In Fig. 7, vanilla ϵ-greedy fails to achieve the goal of *MountainCar* (dashed line on the right) within limited steps, while ϵz-greedy and DAE succeed. Though the advantages of action-repeating exploration methods in *MountainCar*, ϵz-greedy still shows some blind spots that cannot be quickly explored. Following, *GridWorld* is a 23×23 grid environment with four actions: move UP, DOWN, LEFT, and RIGHT. The agent is initially placed on the top-middle of the square and aims to explore all the states (grids). DAE shows outstanding exploration coverage compared to the other two methods. Similar to *GridWorld*, *Labyrinth* is a 5×5 maze. Still, walls block between grid and grid, and only a few openings exist for the agent to travel through, making it a trickier environment than *GridWorld* to explore. ϵz-greedy shows its drawback of action repeating, which causes numerous meaningless steps of hitting walls. Whereas DAE still outperforms by higher exploration efficiency. Finally, *MultiRoom*, the most challenging task, consists of six rooms, and closed doors are to be opened before the agent can pass through and get to another room. DAE still demonstrates its exploration efficacy by reaching all six rooms; yet, ϵ-greedy finds only five of them, and ϵz-greedy barely sees two.

Related Works

Lifelong RL In lifelong RL (Wu, Gupta, and Kochenderfer 2020; Tanaka and Yamamura 1997), the agent must solve a sequence of *dynamically changing tasks*. The insight of this challenge is how the agent can improve the sample efficiency by utilizing common knowledge from previous tasks and quickly adapt to the upcoming tasks (some earlier efforts (Wang et al. 2019; Wang, Li, and Chen 2019; Wang, Chen, and Dong 2021) also refer to this line of study as Incremental RL). However, in prior works of lifelong RL, the state and action spaces are fixed throughout the learning process. In contrast, Incremental RL discussed in the paper is answering the question ”how an agent could efficiently learn new transitions while not forgetting what it has learned?”.

Exploration in RL The exploration methods can be categorized into reward-free and reward-based methods. In the line of reward-free, the exploration is reward-independent, and their drawback of inefficiency could be magnified under Incremental RL. For instance, blind exploration samples random actions, which cannot avoid redundant sampling. Intrinsically-motivated explorations passively grant extra rewards to transitions that find rare states and cannot be solely adopted without other exploration methods. In another line of reward-based methods, neither randomized action-selection methods nor optimism-based methods can resist the learning bias introduced by Incremental RL since these methods explore the environment based on the value estimation. Consequently, newly involved actions would be greatly ignored by the methods above and could lead to the failure to adapt to new environments. Furthermore, the works mentioned above assume learning the policy from the environment with *fixed-yet-known* action and state sets. In contrast, without the dependency on Q-value, DAE can efficiently explore more in states with uncertain value estimation and discover novel transitions by taking the least-tried actions, which are usually worth trying under Q-learning, for the changes in the new environments.

Conclusions

This paper introduces a new challenge of RL, i.e., *Incremental RL*, with continually expanding state and action spaces. When the search space increases, retraining is computationally infeasible. Thus, we further formulate Incremental RL as a hard exploration problem under strong learning bias and propose a novel algorithm named *Dual-Adaptive ϵ-greedy Exploration (DAE)* to reduce the exploration overhead. In addition, a new testbed with two benchmarks is provided to evaluate potential works for Incremental RL fairly. Extensive experiments demonstrate the stated advantage of DAE compared to the eight baselines examined.

Acknowledgments

This work is in part supported by MOST Project No. 111-2221-E-002-135-MY3 and No. 111-2223-E-002-006, Taiwan.

References

Abel, D.; Jinnai, Y.; Guo, S. Y.; Konidaris, G.; and Littman, M. 2018. Policy and value transfer in lifelong reinforcement learning. In *ICML*.

Badia, A. P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z. D.; and Blundell, C. 2020. Agent57: Outperforming the atari human benchmark. In *ICML*.

Baker, B.; Kanitscheider, I.; Markov, T.; Wu, Y.; Powell, G.; McGrew, B.; and Mordatch, I. 2019. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Bridle, J. 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *NeurIPS*, 2.

Brunskill, E.; and Li, L. 2014. Pac-inspired option discovery in lifelong reinforcement learning. In *ICML*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Chevalier-Boisvert, M.; Willems, L.; and Pal, S. 2018. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid. Accessed: 2023-3-17.

Dabney, W.; Ostrovski, G.; and Barreto, A. 2021. Temporally-Extended ε-Greedy Exploration. In *ICLR*.

Ermolov, A.; and Sebe, N. 2020. Latent World Models For Intrinsically Motivated Exploration. *arXiv preprint arXiv:2010.02302*.

Fortunato, M.; Azar, M. G.; Piot, B.; Menick, J.; Hessel, M.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; et al. 2018. Noisy Networks For Exploration. In *ICLR*.

Grigorescu, S.; Trasnea, B.; Cocias, T.; and Macesanu, G. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3): 362–386.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*.

Hu, J.; Jiang, S.; Harding, S. A.; Wu, H.; and Liao, S.-w. 2021. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*.

Kaiser, L.; Babaeizadeh, M.; Milos, P.; Osinski, B.; Campbell, R. H.; Czechowski, K.; Erhan, D.; Finn, C.; Kozakowski, P.; Levine, S.; et al. 2019. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.

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*.

Luong, N. C.; Hoang, D. T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.-C.; and Kim, D. I. 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4): 3133–3174.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Ostrovski, G.; Bellemare, M. G.; Oord, A.; and Munos, R. 2017. Count-based exploration with neural density models. In *ICML*.

Puterman, M. L. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Strehl, A. L.; and Littman, M. L. 2008. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8): 1309–1331.

Sutton, R. S. 1995. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *NeurIPS*, 8.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Tanaka, F.; and Yamamura, M. 1997. An approach to lifelong reinforcement learning through multiple environments. In *6th European Workshop on Learning Robots*, 93–99.

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2017. # exploration: A study of count-based exploration for deep reinforcement learning. In *31st Conference on Neural Information Processing Systems (NIPS)*, volume 30, 1–18.

Tokic, M. 2010. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, 203–210. Springer.

Van Hasselt, H. P.; Hessel, M.; and Aslanides, J. 2019. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32.

Wang, J.; Cao, J.; Wang, S.; Yao, Z.; and Li, W. 2020. IRDA: incremental reinforcement learning for dynamic resource allocation. *IEEE Transactions on Big Data*.

Wang, Z.; Chen, C.; and Dong, D. 2021. Lifelong incremental reinforcement learning with online Bayesian inference. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, Z.; Chen, C.; Li, H.-X.; Dong, D.; and Tarn, T.-J. 2019. Incremental reinforcement learning with prioritized sweeping for dynamic environments. *IEEE/ASME Transactions on Mechatronics*, 24(2): 621–632.

Wang, Z.; Li, H.-X.; and Chen, C. 2019. Incremental reinforcement learning in continuous spaces via policy relaxation and importance weighting. *IEEE Transactions on Neural Networks and Learning Systems*, 31(6): 1870–1883.

Won, J.; Gopinath, D.; and Hodgins, J. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4): 1–11.

Wu, B.; Gupta, J. K.; and Kochenderfer, M. 2020. Model primitives for hierarchical lifelong reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 34(1): 1–38.

Zhang, C.; Patras, P.; and Haddadi, H. 2019. Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials*, 21(3): 2224–2287.

Zhang, T.; Xu, H.; Wang, X.; Wu, Y.; Keutzer, K.; Gonzalez, J. E.; and Tian, Y. 2021. NovelD: A Simple yet Effective Exploration Criterion. *NeurIPS*, 34.