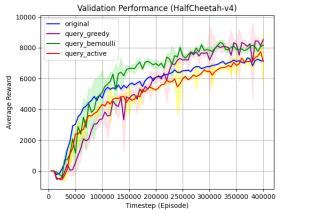
Active Reinforcement Learning from Demonstration in Continuous Action Spaces

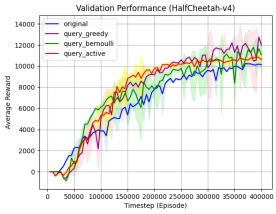
Ming-Hsin Chen [1] **Si-An Chen** [1] **Hsuan-Tien Lin** [1]

Abstract

Learning from Demonstration (LfD) is a human-in-the-loop paradigm that aims to overcome the limitations of safety considerations and weak data efficiency in Reinforcement Learning (RL). Active Reinforcement Learning from Demonstration (ARLD) takes LfD a step further by actively involving the human expert only during critical moments, reducing the costs associated with demonstrations. While successful ARLD strategies have been developed for RL environments with discrete actions, their potential in continuous action environments has not been thoroughly explored. In this work, we propose a novel ARLD strategy specifically designed for continuous environments. Our strategy involves estimating the uncertainty of the current RL agent directly from the variance of the stochastic policy within the state-of-the-art Soft Actor-Critic RL model. We demonstrate that our strategy outperforms both a naive attempt to adapt existing ARLD strategies to continuous environments and the passive LfD strategy. These results validate the potential of ARLD in continuous environments and lay the foundation for future research in this direction.



(a) Noisy TD3 validation performance



(b) Bootstrapped TD3 validation performance

Figure 1. The left figure represents the results obtained using the Noisy TD3 algorithm. The right figure displays the results of Bootstrapped TD3 algorithm.

1. Introduction

Reinforcement Learning (RL) is a powerful approach for tackling sequential decision-making problems by learning from experience through interactions with environments. While RL is shown to be effective in simulated environments, adapting RL to real-world robotics tasks faces challenges related to safety concerns when RL agents explore the environment. Recent studies emphasize the importance of mitigating the risk of causing harm during exploration (Amodei et al., 2016; Choi et al., 2020; Marvi & Kiumarsi, 2021). Combining Learning from Demonstration (LfD) with RL offers a potential solution, allowing computers to learn from human demonstrations, refine decision-making, and ensure safety in dynamic environments with human-computer interaction (Nair et al., 2018; Thananjeyan et al., 2020; Yang et al., 2022). This integration improves performance and promotes safe collaboration between humans and machines, with a cost of acquiring numerous human demonstrations, which can be laborious and expensive (Kang et al., 2018).

Active learning (Settles, 2009) allows labeling fewer data for supervised learning by interactively querying experts for unlabelled data, and has been extended to address the demonsration cost in LfD (Chen et al., 2020) as the Active Reinforcement Learning from Demonstration (ARLD) paradigm. The paradigm enhances demonstration efficacy and achieves competitive performance. The original ARLD paradigm is designed to select the states that the agent is uncertain about to ask for human demonstration. The design is successful in the context of discrete action spaces, where two variants of the Deep Q-learning (DQN, Mnih et al. 2013) agent, namely bootstrapped DQN (Osband et al., 2016) and noisy DQN (Fortunato et al., 2017), are shown to produce uncertainty measures that are sufficiently

[1]Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. Correspondence to: Hsuan-Tien Lin <htlin@csie.ntu.edu.tw>.

AI & HCI Workshop at the 40^{th} International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

meaningful to ask for effective demonstration.

Inspired by this success, we aim to extend ARLD to continuous action spaces. The extension is highly non-trivial. For instance, after extending a representative agent in the continuous action space, Twin-Delayed Deep Deterministic Policy Gradient (TD3, Fujimoto et al. 2018), to its noisy or bootstrapped version, we find that ARLD can be inferior to no demonstration, passive demonstration (Bournoulli, or greedily asking for demonstration, as shown in Figure 1. The results suggest that more research is needed to understand the design of uncertainty measures in continuous action spaces.

In this work, we take a state-of-the-art RL agent for continuous action spaces, Soft Actor-Critic (SAC, Haarnoja et al. 2018), to study the design of uncertainty measures for ARLD. SAC employs a stochastic policy parameterized as a distribution over the action space, which can be naturally used to compute an *intrinsic* uncertainty measure from the variance of the distribution. On the other hand, similar to TD3, SAC can be extended to a noisy version to compute an *extrinsic* uncertainty measure like Noisy TD3 or Noisy DQN. Comparing the intrinsic and extrinsic measures reveals that the intrinsic one work better for ARLD as it matches the intent of the RL agent better. ARLD coupled with the intrinsic uncertainty measure of SAC reaches super-expert level performance after strategically asking for the expert's demonstration. The promising results lay the foundation for future research for ARLD in continuous action spaces. Our main contributions are as follows.

- extending ARLD to continuous action spaces successfully, outperforming passive and greedy baselines
- comparing possible uncertainty measures fairly for the SAC agent to justify the design choices
- analyzing the success of the intrinsic uncertainty measure to deepen our understanding of the ARLD task

2. Background

2.1. Reinforcement Learning

In the context of reinforcement learning (RL), the agent selects actions in a sequential manner across multiple timesteps to interact with the environment and obtain rewards. The objective of the agent is to maximize the discounted cumulative reward. We can model the problem as a Markov decision process (MDP), characterized by a tuple $M = (S, A, r, P, \gamma)$. S is the state space; A is the action space; R: $S \times A \to \mathbb{R}$ is the reward function; $\gamma \in [0, 1)$ is the discount factor. The agent starts from an initial state distribution with density $P(s_1)$, and transit to the next state in a stationary manner with a conditional density $P(s_{t+1}|s_t, a_t)$ that satisfies the Markov property $P(s_{t+1}|s_1, a_1, ..., s_t, a_t) = P(s_{t+1}|s_t, a_t)$ for any trajectory $s_1, a_1, s_2, a_2, ..., s_T, a_T$ in state-action space.

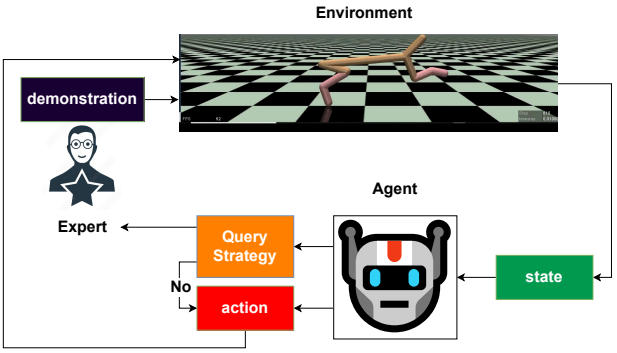


Figure 2. The ARLD process: At each time step (t), the agent observes the current state s_t and calculates an action based on its current policy. The agent then consults the query strategy to determine whether it should seek guidance from the expert. If the query strategy suggests querying the expert, the expert takes control and performs a series of demonstrations, interacting with the environment to provide guidance. On the other hand, the agent proceeds to execute its own action and interacts with the environment accordingly.

2.2. RL in Discrete Action Spaces

In discrete action spaces, one prominent RL agent called Deep Q-Network (DQN, Mnih et al. 2015). DQN approximates the Q-value function, which estimates the cumulative reward for any action, using a deep neural network (DNN). Training the DQN agent commonly involves a replay buffer to learn from past experiences repeatedly and the epsilon-greedy technique to either explore a random action or exploit the action with the largest estimated Q-value. Extensions of DQN involve Double Q-learning (Van Hasselt et al., 2016) that uses another network to stabilize the estimation process, Noisy DQN (Fortunato et al., 2017) that injects noise to the action-decision layer to facilitate exploration, and Bootstrapped DQN (Osband et al., 2016) that uses bootstrapping instead of noise injection.

The combination of Learning from Demonstration (LfD) and Reinforcement Learning (RL) has garnered significant attention recently. Deep Q-learning from Demonstrations (DQfD, Hester et al. 2018) capitalizes on demonstrations to expedite the learning process. By integrating the standard RL loss with a supervised loss based on demonstrations, DQfD simultaneously maximizes the reward and emulates expert behavior.

2.3. RL in Continuous Action Spaces

However, Q-learning is not directly applicable to continuous action spaces, as it struggles to generate state-action values in such settings, resulting in significant performance drops. To address this limitation, the actor-critic framework has emerged as a powerful approach. For tackling intricate and high-dimensional continuous real-world tasks, the actor-critic framework manifests as the predominant approach of RL. Deep Deterministic Policy Gradient (DDPG, Lillicrap et al. 2015) combines actor-critic techniques with insights from DQN. It employs an Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930) to generate temporally correlated exploration, enhancing exploration efficiency. Consequently, DDPG demonstrates robust performance across diverse domains with continuous action spaces. In continuous action space, the problem of overestimation also arises. Twin-Delayed DDPG (TD3, Fujimoto et al. 2018) introduces a novel variant of Double Q-learning, featuring delayed policy updates, clipped Double Q-learning, and target policy smoothing, collectively contributing to improved performance. Soft Actor-Critic (SAC, Haarnoja et al. 2018) incorporates an entropy regularization term into the objective function, encouraging the actor to succeed while promoting random behavior.

In real-world scenarios, many tasks lack a simulator, necessitating learning directly in the physical domain with safe and proficient performance early in training. DDPG from Demonstration (DDPGfD, Vecerik et al. 2017) extends the concept of DQfD to continuous action domains. It incorporates demonstrations into a refined prioritized replay buffer with higher priority and combines supervised loss to effectively leverage the demonstrations, resulting in state-of-the-art performance across various tasks. Several studies have successfully expedited the RL process by leveraging task-agnostic experience from extensive datasets (Nair et al., 2020; Singh et al., 2020; Pertsch et al., 2021). Notably, recent research (Liu et al., 2022) introduces novel RL from demonstration techniques to enhance performance and accelerate training in the autonomous car driving using SAC.

2.4. Active Reinforcement Learning from Demonstration

Active Reinforcement Learning from Demonstration (ARLD, Chen et al. 2020) is an RL framework that addresses the challenge of demonstration efficiency by allowing the agent to request demonstrations actively. The overall process of ARLD is depicted in Figure 2. In ARLD, we assume the existence of experts who can generate demonstrations promptly when queried. During each step of the learning process, the agent observes the state obtained from the environment and calculates the uncertainty. This uncertainty is then passed to the query strategy to determine whether to request a demonstration from the expert.

2.4.1. ARLD in Continuous Action Space

Motivated by the significant advancements in the application of ARLD to discrete action space, our objective is to extend the efficiency of demonstrations to continuous environments. These continuous environments present greater complexity and better reflect real-world scenarios, making it crucial to adapt and enhance demonstration efficacy in such settings.

However, directly applying the techniques of ADQN to continuous spaces poses challenges due to the discrepancy in uncertainty measurement, which directly impacts the active learning process for querying. In the transition from discrete to continuous space, it becomes crucial to carefully define the uncertainty measurement, fine-tune the associated parameters, and effectively leverage demonstrations to provide substantial benefits to the agent. To address these differences and capitalize on the contributions of ADQN, we strive to adapt and extend its methodologies to the continuous space, with the ultimate goal of enhancing the effectiveness of ARLD in real-world settings.

2.4.2. Soft Actor-Critic

To solve ARLD in continuous space, we employ soft actor-critic (SAC) as the RL agent. SAC is a widely used and effective algorithm specifically designed for continuous action spaces. SAC is policy maximum entropy actor-critic algorithm which provides for both sample-efficient learning and stability. This algorithm extends readily to very complex, high-dimensional tasks. SAC consistently outperforms state-of-the-art model-free deep reinforcement learning methods, including the off-policy DDPG algorithm and the on-policy PPO algorithm in continuous control tasks (like robotic locomotion and manipulation).

A key aspect of SAC is entropy regularization, which draws inspiration from the maximum entropy framework. This regularization enhances the conventional objective of maximizing cumulative rewards in reinforcement learning by incorporating an entropy maximization term with a temperature parameter α. This objective offers the advantage of motivating the agent to explore diverse possibilities while disregarding gloomy trails. Additionally, the policy is capable of capturing a variety of near-optimal solutions.

SAC is an off-policy algorithm that optimizes a stochastic policy, bridging the gap between stochastic policy optimization and DDPG-style approaches. It incorporates the clipped double-Q trick and benefits from target policy smoothing due to the inherent stochasticity of the policy. This formulation prevents the policy from prematurely converging to suboptimal local optima.

3. Method

In this section, we first describe an attempt to extend ARLD to the continuous domains in Section 3.1. The attempt mimics the change from DQN to Noisy DQN on top of SAC. That is, we propose a Noisy SAC variant. The variant allows us to design an uncertainty measure similar to the one used in ARLD. Nevertheless, we observe that the attempt was unsuccessful as the measurement does not seem strongly related to the epistemic uncertainty of the model and does not gradually decrease during RL training. We then design another uncertainty measure directly from SAC instead in Section 3.2.

3.1. Noisy Soft Actor-Critic for Uncertainty Measurement

As discussed in Section 2.4, ARLD is based on asking for a demonstration when the model uncertainty exceeds a certain level. Model uncertainty, also known as the epistemic uncertainty (Hüllermeier & Waegeman, 2021), arises from a lack of knowledge or understanding of the RL model. The uncertainty should decrease as the model training on more data over additional iterations. While the concept of model uncertainty is intuitive, there is no universal definition of the measure of model uncertainty.

In the SAC algorithm, we utilize a stochastic actor to select actions based on the given state, incorporating an entropy regularization term that encourages a trade-off between exploration and exploitation, enhancing the agent's exploration ability. We typically model the policy π_θ as a Gaussian distribution, with the parameterized mean μ_θ and standard deviation σ_θ.

ARLD in discrete action spaces builds upon Noisy DQN, a noisy variant of DQN that injects parameter noise to alter action decisions. For the TD3 model discussed in Section 1, existing work (Plappert et al., 2017) similarly extend it to Noisy DDPG by injecting the parameter noise on the actor. We follow the same principle to extend SAC to its noisy variant. The perturbed mean ($\tilde{\mu_\theta}$) is formulated as

$$\tilde{\mu_\theta} = (\mu_w + \sigma_w \odot \epsilon_w)\mu_\theta + \mu_b + \sigma_b \odot \epsilon_b$$

where $(\mu_w, \sigma_w, \mu_b, \sigma_b)$ is a set of vectors of learnable parameters, ϵ is a vector of zero-mean noise sampled from the standard normal distribution, and $\odot$ stands for element-wise multiplication.

$$\begin{aligned} \text{Var}[\tilde{\mu_\theta}] &= \text{Var}[w\phi(s) + b] \\ &= \text{Var}[w\phi(s)] + \text{Var}[b] \\ &= \phi(s)^T \Sigma \phi(s) + \sigma_b \end{aligned}$$

where $\phi(s)$ is the input to the $\mu(\cdot)$, $w \sim N(\mu_w, \Sigma), \Sigma = \text{diag}((\sigma_w)^2)$ and $b \sim N(\mu_b, (\sigma_b)^2)$.

With the design of Noisy SAC, we can then take the variance of the perturbed mean of the actor to the state s as an uncertainty measure.

$$u(s) = \text{Var}[\tilde{\mu_\theta}].$$

This uncertainty indicates the variation of the actor output under the noise perturbation.

3.2. Soft Actor-Critic for Uncertainty Measurement

The uncertainty measure defined by the Noisy SAC can be viewed as an *extrinsic* uncertainty measure. Our careful study in Section 4 will reveal that the measure does not match the need of ARLD. We thus seek to devise another uncertainty measure that is ideally more connected with the *intrinsic* uncertainty of the SAC model.

During the training process, the parameterized actor samples an action $\tilde{a}_\theta(s)$ from a Gaussian distribution and subsequently applies an activation function to facilitate its interaction with the environment.

$$\tilde{a}_\theta(s) = \tanh(\mu_\theta(s) + \sigma_\theta(s) \cdot \xi), \xi \sim \mathcal{N}(0, 1)$$

Notice that when validating, we will remove the noise to make a deterministic actor for getting an optimal policy.

In this work, we consider the standard deviation layer output σ_θ of the actor as a measure of uncertainty u, which denotes the state-dependent parameter vectors that served as the noise.

$$u(s) = \sigma_\theta(s)$$

When the σ_θ gets higher, the sampled actions exhibit a higher probability of deviating from the mean, promoting increased exploration. By estimating the uncertainty measured through σ_θ, we can identify and avoid querying unimportant states during the active learning process.

4. Experiments

In this section, our primary focus is to assess the effectiveness of each query strategy employed. The query strategy plays a crucial role in determining the effectiveness of demonstrations throughout the training process. A core component of the query strategy is budget pacing, which involves allocating the query budgets over a fixed-length training period. This allocation can be uniform, ensuring equal representation of queries before budget depletion. Alternatively, more sophisticated pacing strategies can be employed, focusing on querying critical steps during specific phases of the training process while assigning lower priority to other periods. We employed four methods based on SAC framework:

1. SAC: This method represents the baseline and involves training SAC without demonstrations.

2. GQSAC: The Greedy Query SAC strategy queries all states until the budget is exhausted.
3. BQSAC: The Bernoulli Query SAC strategy randomly selects states based on a fixed probability.
4. AQSAC: The Active SAC strategy utilizes previous proposed active learning query strategy and uncertainty estimation to intelligently select states for querying.

We first demonstrate how the uncertainty measurement affects the active learning strategy. Subsequently, we present the experimental results of the methods obtained. Last, we evaluate the impact of the query proportion threshold of the adaptive active learning strategy, providing insightful findings on its efficacy.

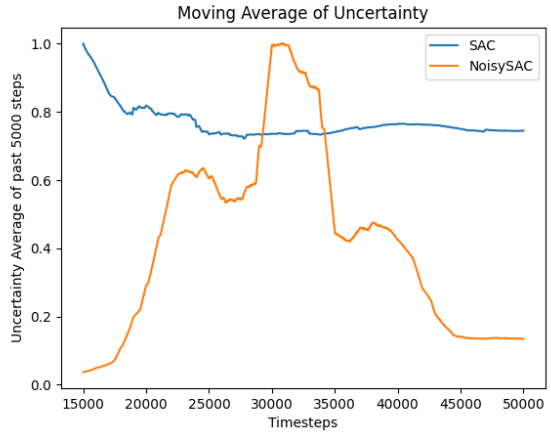


Figure 3. Moving average of the normalized uncertainty of SAC and NoisySAC before budget depletion.

4.1. Uncertainty Comparison

In Noisy DQN, the uncertainty measurement with predictive variance indicates the agent's lack of confidence in the chosen action.

Regarding the uncertainty estimation in SAC and NoisySAC, both measures convey a similar meaning by capturing the sensitivity to state-dependent noise for exploration. The epistemic uncertainty correlates with the level of exploration in the state. Higher epistemic uncertainty implies that the agent is uncertain about the policy or value with this state, indicating an underexplored region.

Figure 3 illustrates the moving average trend of normalized uncertainty before exhausting the query budget. We observe that the uncertainty measurement of SAC demonstrates a decreasing trend throughout the training process before budget depletion, suggesting that querying states with high uncertainty is more critical (see Figure 4). On the other hand, the uncertainty measurement of NoisySAC exhibits several peaks without any discernible correlation with performance or other factors, and the results of active learning align closely with the Bernoulli approach (see Figure 5).

The trend in SAC uncertainty can be attributed to the entropy regularization term H in the objective function of the actor (π), which aims to maximize the expected value $V^\pi(s)$ defined as follows:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{a_t\sim\pi}[Q^\pi(s,a) + \alpha H(\pi(\cdot|s))] \\ &= \mathbb{E}_{a_t\sim\pi}[Q^\pi(s,a) - \alpha \log(\pi(a|s))] \end{aligned}$$

The reduction in variance indicates that the policy becomes more focused and deterministic, leveraging the acquired knowledge to select actions that are more likely to yield higher rewards. The alignment of the presented observations with the description of epistemic uncertainty and their correspondence to expected outcomes provides further evidence of the suitability and accuracy of the characterization.

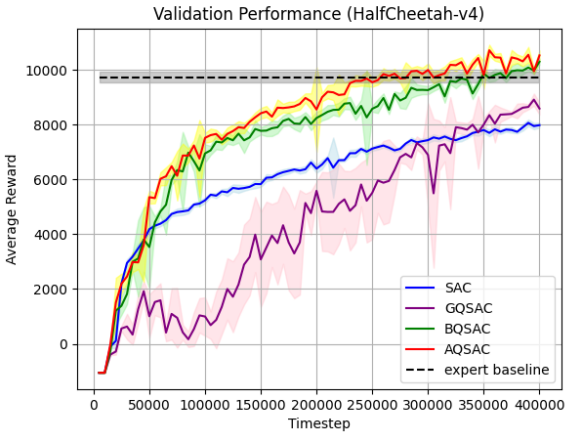


Figure 4. For AQSAC and BQSAC, we select the parameter with best performance for comparison. BQSAC with Bernoulli probability 0.2, and AQSAC with proportion threshold 0.1. The black line demonstrates the expert policy performance.

4.2. Comparison between Query Strategy

We compared the results of four methods: SAC, GQSAC, BQSAC, and AQSAC. Figure 4 depicts the validation performance of each method in solving the task HalfCheetah. AQSAC exhibits a more effective utilization of demonstrations, shortening the exploration phase in the early stage and surpassing the expert performance by the end, outperforming other heuristic approaches. This highlights the superiority of the proposed active learning strategy in leveraging demonstrations to enhance the agent's performance.

The greedy query strategy continually queries until the budget is depleted, initially filling the replay buffer with demon-

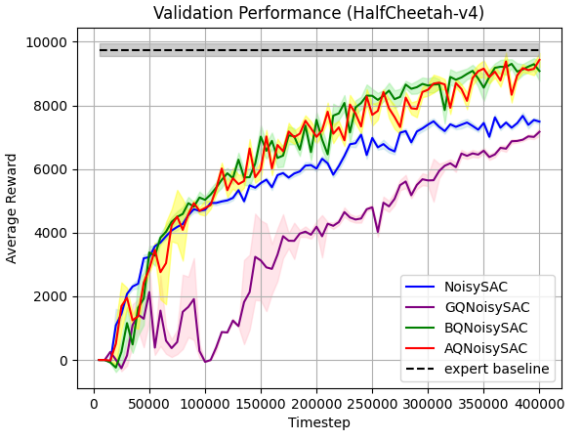


Figure 5. For AQNoisySAC and BQNoisySAC, we select the parameter with the best performance for comparison. BQNoisySAC with Bernoulli probability 0.4, and AQNoisySAC with proportion threshold 0.1. The black line demonstrates the expert policy performance.

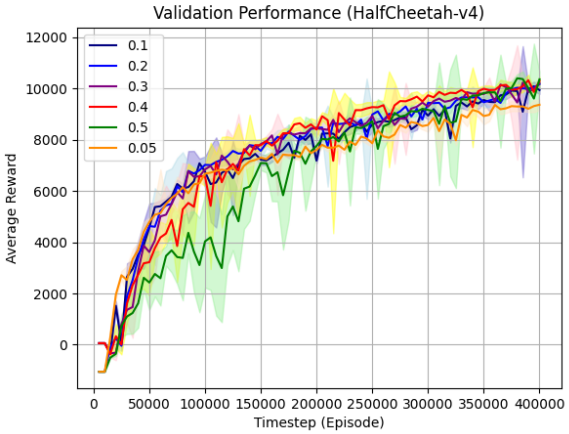


Figure 6. Episode rewards and number of timesteps for each Bernoulli probability.

strations. Therefore, in the subsequent steps, the transition distribution of the replay buffer deviates from the distribution of states that the current policy would encounter. This deviation can result in the inaccurate estimation of Q-values for underexplored states, potentially leading to corruption of both the policy and critic.

Although we have observed promising outcomes, we still lack a comprehensive understanding of why the selected demonstrations lead to improved agent performance over others.

4.3. Effect of Query Proportion Threshold and Bernoulli Probability

In Figure 4 shows that Bernoulli query strategy already stands for a strong baseline compare to the original model without any demonstrations. Previous study (Tifrea et al., 2022) shows that active learning with uncertainty sometimes leads to worse performance compare to the passive learning. We take BQSAC and AQSAC to discuss the parameter that effects the performance. The Bernoulli probability b and Active learning proportion threshold t_{query} are two main parameters that have impact on the performance. Figure 6 shows that different choices of probability b perform similarly, besides $b = 0.5$. Figure 7 shows that applying lower t_{query} will obtain better performance. With larger value of b and t_{query} implies spending budget in the early stage, two figures show the same trend of applying lower probability to sample obtain better performance in the early stage. The peak performance is near $b = 0.1$ for BQSAC and $t_{query} = 0.1$ for AQSAC.

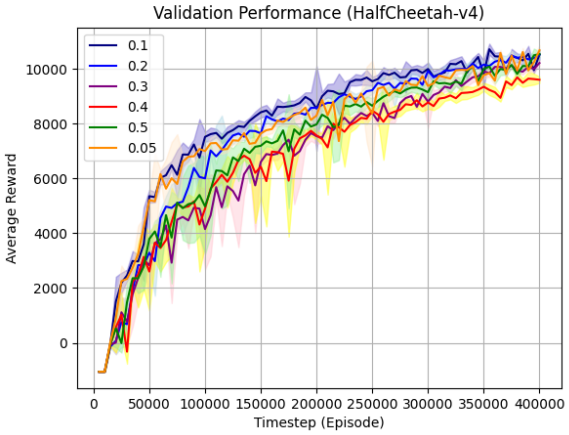


Figure 7. Episode rewards and number of timesteps for each proportion threshold used in active learning.

5. Conclusion and Discussion

In this study, we have made non-trivial progress in extending ARLD to continuous action spaces, resulting in improved performance of the SAC agent and validating demonstration efficacy. We devise an intrinsic uncertainty measure based on SAC and observe its decreasing trend during RL training. Through fair empirical evaluations, we justify the potential of ARLD to outperform the original agent, the greedy strategy, and the passive strategy of asking for demonstrations in continuous action spaces for the first time, to the best of our knowledge.

Our promising results are achieved for only one environment with continuous action space, and more research is needed to confirm the results for more environments. We are in the

process of such a research direction but have not finished all experiments because of computational resource constraints. We hope that our results can inspire deeper studies on other ARLD strategies, other uncertainty measures, and the pacing of the demonstration budget to unlock the full potential of LfD with ARLD in real-world RL applications for human-computer interactions.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work is supported by National Taiwan University Center for Data Intelligence via NTU-112L900901 as well as the Ministry of Science and Technology in Taiwan via MOST 111-2628-E-002-018 and 112-2628-E-002-030. We thank National Center for High-performance Computing (NCHC) in Taiwan for providing computational and storage resources.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Chen, S.-A., Tangkaratt, V., Lin, H.-T., and Sugiyama, M. Active deep q-learning with demonstration. *Machine Learning*, 109:1699–1725, 2020.

Choi, J., Castaneda, F., Tomlin, C. J., and Sreenath, K. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*, 2017.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

Kang, B., Jie, Z., and Feng, J. Policy optimization with demonstrations. In *International conference on machine learning*, pp. 2469–2478. PMLR, 2018.

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

Liu, H., Huang, Z., Wu, J., and Lv, C. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 921–928. IEEE, 2022.

Marvi, Z. and Kiumarsi, B. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control*, 31(6): 1923–1940, 2021.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Pertsch, K., Lee, Y., and Lim, J. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Settles, B. Active learning literature survey. 2009.

Singh, A., Liu, H., Zhou, G., Yu, A., Rhinehart, N., and Levine, S. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

Thananjeyan, B., Balakrishna, A., Rosolia, U., Li, F., McAllister, R., Gonzalez, J. E., Levine, S., Borrelli, F., and Goldberg, K. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.

Tifrea, A., Clarysse, J., and Yang, F. Uniform versus uncertainty sampling: When being active is less efficient than staying passive. *arXiv preprint arXiv:2212.00772*, 2022.

Uhlenbeck, G. E. and Ornstein, L. S. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2016.

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Yang, Y., Chen, L., and Gombolay, M. Safe inverse reinforcement learning via control barrier function. *arXiv preprint arXiv:2212.02753*, 2022.

A. Appendix

A.1. Experiment setup

We conducted our experiments using the Half-Cheetah environment from OpenAI Gym. This environment is a continuous control task object to maximize the accumulated reward within a fixed number of steps.

To ensure reasonable learning progress, we fine-tuned the basic parameters like the reward scale and the weight for soft updates with target networks for SAC in each environment. Subsequently, we fixed these parameters for all four methods. The network architecture employed in all simulated environments remained consistent. It consisted of a policy network with two fully connected hidden layers comprising 256 neurons each, followed by an additional two fully connected layers that outputted the mean and covariance of the Gaussian for each action dimension. The critic network included two fully connected hidden layers with 256 neurons each, followed by a single fully connected layer outputting a one-dimensional Q-value. Rectified linear units (ReLU) were used as the activation function for the hidden layers, while hyperbolic tangent (tanh) activation was applied to the output layers to constrain the values within the range [-1, 1]. We trained the networks using the Adam optimizer, and the temperature parameter α was set to 0.2. The parameters for the prioritized replay buffer were set according to the approach (Schaul et al., 2015). For the SAC networks, we follow the initialization and hyperparameter values from (Haarnoja et al., 2018).

For each query strategy, after each query, the agents receive C consecutive demonstrations from the experts until the end of the episode, where $C \in \{1, 2, 3, 5, 10\}$.

To integrate demonstrations into our methodology, we adopt the implementation strategies of DDPGfD, as cited in (Vecerik et al., 2017), which introduces several key techniques:

1. We utilize a prioritized replay buffer to sample transitions with higher importance. The probability of sampling a specific transitions is calculated using the priority p_i with the transition i. The priority $p_i = \delta_i^2 + \epsilon + \epsilon_d$, where δ_i represents the last Q-value prediction error for that transition. The term ϵ is a small positive constant ensuring all transitions can be sampled, and ϵ_d is a positive constant for demonstration transitions to increase their sampling probability.

2. Since our simulated environment does not have sparse reward, we do not use n-step returns to update the critic function. Also, we did not do multiple learning updates per environment step for fair comparison with the RL without demonstrations.

The two parameters of AQSAC, query threshold t_{query} is tuned in $\{0.05, 0.1, 0.2, ..., 0.7\}$, the reference size N_r is set to 5000. The query budget is set to 10000.

We employ a pre-trained prioritized SAC model as our expert policy. The expert policy performs interactive demonstrations using various query strategies. The total timesteps are set to 400000. Throughout the evaluation process, we assessed the performance of the models at regular intervals of 5000 timesteps. To ensure the robustness and reliability of the empirical results, we repeated the validation process 10 times using different random seeds.

A.2. Ablation Study

We further examined the impact of the query strategies on the performance by analyzing the budget remaining and the

states queried. Figure 8 illustrates the projection of states onto a 2-dimensional plane, where yellow points represent the queried states. It is evident that the Bernoulli query strategy explores a more diverse range of states compared to the active learning method. Figure 9 reveals that despite following a similar budget spending pattern, the resulting performance still varies among the strategies. These observations highlight the influence of the query strategies on the exploration process and demonstrate the distinct characteristics of each approach.

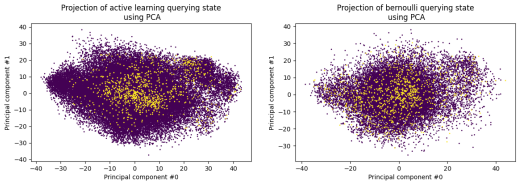
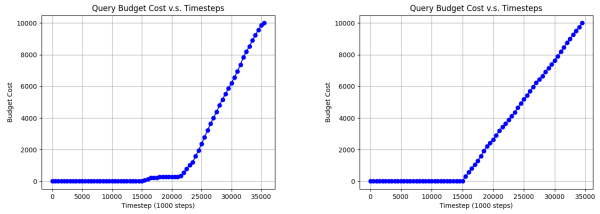


Figure 8. Projection of state being queried by AQSAC and BQSAC. Left: AQSAC. Right: BQSAC.



(a) Active learning with proportion threshold 0.1

(b) Bernoulli with probability 0.1

Figure 9. Trend of budget cost regards to timesteps.

A.3. User Interface

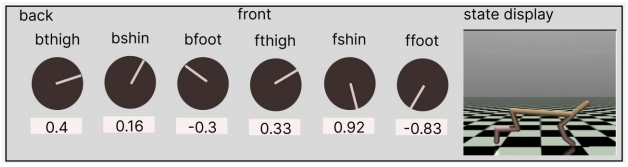


Figure 10. Designed user interface for collecting demonstrations

To make the proposed method more realistic, we designed a user interface for experts to demonstrate and collect the demonstration for further use in the ARLD framework.

- Pros
  1. The screen shows the last recent states, which helps the user demonstrate more appropriately according to the context.
  2. Using knobs for better adapting to the continuous action space.
- Cons
  1. Since all the actions are continuous and be bounded by a little range, slightly different values in each dimension of actions may cause quite different actions, which need accurate manipulation.
  2. When encountering higher dimensional tasks, the interface might become more complicated, and the time of giving out demonstrations becomes the bottleneck.