Actor Free critic Update for Off-policy and Offline learning

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

Introduction

Related Work

Reinforcement learning algorithms for continuous control domains have evolved significantly in recent years. The development of actor-critic architectures has proven particularly effective, with algorithms like DDPG [4] and SAC [5] establishing themselves as standard approaches. These methods combine the advantages of policy gradient methods with value-based learning, enabling sample-efficient learning in continuous action spaces.

DDPG employs a deterministic actor that maximizes a learned Q-function, operating entirely off-policy to improve sample efficiency. However, DDPG presents stability issues, demonstrating sensitivity to hyperparameter choices and exploration strategies. SAC extends this framework by incorporating entropy maximization, encouraging exploration while learning a stochastic policy. The entropy term provides additional stability, allowing SAC to achieve both better sample efficiency and final performance compared to DDPG in many domains.

The challenge of addressing the distribution shift when transitioning from offline to online reinforcement learning has received increasing attention. Offline reinforcement learning methods train agents using previously collected datasets without environment interaction, eliminating exploration costs but introducing optimization difficulties. IQL [1] approaches this problem by completely avoiding direct querying of the learned Q-function with unseen actions during training, using expectile regression to estimate the maximum value of Q-functions. Similarly, CALQL [2] applies a constraint to the conservative Q-learning framework to reduce overestimation bias during offline learning while enabling efficient online fine-tuning. Both methods aim to mitigate performance degradation observed when agents trained offline begin interacting with environments directly.

The ability to learn truly off-policy—from data generated by arbitrary or random policies—represents another research direction. Traditional actor-critic methods are categorized as off-policy but often struggle when presented with data significantly different from their current policy distribution. This limitation arises because their critic updates depend on actions sampled by the actor, creating an implicit coupling that restricts genuine off-policy learning. Some algorithms have attempted to address this issue through several methods but the fundamental actor-critic interdependence remains.

AFU [3] introduces a structural departure from previous approaches by maintaining critic updates that remain entirely independent from the actor. Unlike other algorithms derived from Q-learning for continuous control, AFU aims to solve the maximization problem inherent in Q-learning through a mechanism based on value and advantage decomposition, employing conditional gradient scaling. This approach potentially enables more effective learning from arbitra-

ry data distributions without requiring explicit constraints during the critic learning phase.

Preleminaries

We consider a discounted infinite horizon Markov Decision Problem (MDP) defined as a tuple $< S, A, T, R, \gamma >$, where S represents the state space, A denotes a continuous action space, T is a stochastic transition function, $R : S \times A \to \mathbb{R}$ is a reward function, and $0 \leq \gamma < 1$ is a discount factor. When an agent performs an action $a \in A$ in state $s \in S$, it transitions to a new state s' according to the transition probability $T(s'|s, a)$ and receives a reward $r = R(s, a)$. We denote transitions as tuples (s, a, r, s').

The goal in reinforcement learning is to find a policy $\pi : S \to A$ that maximizes the expected sum of discounted rewards. The optimal Q-function Q^* is defined as:

$$Q^*(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a, \pi^*\right]$$

where the policy used from $t = 1$ onwards is π^*, which selects actions optimally in every state. The optimal value function V^* satisfies $V^*(s) = \max_{\{a \in A\}}(Q^*(s, a))$.

In deep reinforcement learning, we approximate the value functions using neural networks. We denote by V_φ a function approximators for the value function, and by Q_ψ a function approximator for the Q-function (the critic), where φ, and ψ are the parameter vectors of neural networks.

On-policy and off-policy learning represent two distinct approaches in reinforcement learning. In on-policy learning, the agent learns from data collected using its current policy. In contrast, off-policy learning allows the agent to learn from data collected using any policy, including random exploration or previously stored experiences. This distinction is crucial as truly off-policy algorithms can potentially learn more efficiently by reusing diverse experiences.

Offline reinforcement learning takes the off-policy concept further by learning entirely from a fixed dataset of previously collected transitions without any environment interaction during training. When transitioning from offline to online learning (where the agent begins to interact with the environment), algorithms often suffer from distribution shift problems as the learned policy encounters states and actions not represented in the offline dataset.

Actor-critic methods combine policy-based and value-based approaches. The actor (policy network) determines which actions to take, while the critic (value network) evaluates these actions. In traditional actor-critic architectures like SAC and DDPG, the critic's updates depend on actions sampled by the actor, creating an interdependence between the two components. The AFU algorithm represents a departure from this approach by updating the critic independently of the actor.

The temporal difference (TD) learning used in many algorithms aims to minimize the following loss function for the critic:

$$L_Q(\psi) = \mathbb{E}_{(s,a,r,s') \sim B}\left[\left(Q_\psi(s, a) - r - \gamma V_\varphi(s')\right)^2\right]$$

where B represents a mini-batch of transitions sampled from an experience replay buffer.

Methods

Environment. For our experiments, we utilized the *Gymnasium* framework, a widely adopted benchmark for reinforcement learning research. Gymnasium provides standardized environments, allowing for reproducible evaluation of algorithmic performance. We primarily focused on continuous control tasks including Pendulum, CartPole (continuous version), LunarLander (continuous version) and MountainCar (continuous version).

To facilitate our research on off-policy learning properties, we extended these environments with additional functionality. Each envi-

ronment was modified to support direct state manipulation through the implementation of a `_set_state()` method. This modification enables precise control over the system state, allowing us to sample uniformly from the state-action space during our off-policy learning experiments. It should be noted that such direct state manipulation represents a research tool rather than a practical capability in real-world scenarios, where complete state control is rarely possible.

Algorithms. We implemented several state-of-the-art deep reinforcement learning algorithms using PyTorch and the BBRL (BlackBoard Reinforcement Learning) framework. BBRL provides a modular architecture where agents interact through a shared workspace, facilitating the implementation of complex algorithms with clear separation of components. Each algorithm implementation follows the same structure, comprising neural network architectures, training procedures, and evaluation methods.

(DDPG) Deep Deterministic Policy Gradient combines the deterministic policy gradient algorithm with deep neural networks. It consists of two primary networks. The actor network implements a deterministic policy $\pi(s)$ that maps states to specific actions. The critic network evaluates the quality of state-action pairs by approximating the Q-function $Q(s, a)$. For stable learning, DDPG employs target networks for both actor and critic, which are updated using soft updates: $\theta^{\text{target}} \leftarrow \tau\theta + (1-\tau)\theta^{\text{target}}$, where τ is a small value controlling the update rate. DDPG also uses a replay buffer to store and randomly sample transitions, breaking the correlation between consecutive samples and stabilizing learning.

(SAC) Soft Actor-Critic extends the actor-critic architecture by incorporating entropy maximization, encouraging exploration through stochastic policies. Our SAC implementation includes a policy network, a Q-network and a V-network. Unlike DDPG's deterministic policy, SAC's policy is stochastic, outputting a Gaussian distribution over actions. The network produces both the mean $\mu(s)$ and log standard deviation $\log(\sigma(s))$ of this distribution. Actions are sampled using the reparameterization trick and squashed through a tanh function to bound them. SAC employs two Q-networks to mitigate overestimation bias, a common issue in Q-learning. Both networks approximate $Q(s, a)$ and the minimum of their predictions is used for updates. The value network estimates the state value $V(s)$, which represents the expected future return starting from state s when following the policy. SAC optimizes the expected return plus the entropy of the policy using a temperature parameter that determines the relative importance of entropy versus reward. This parameter is automatically adjusted during training to achieve a target entropy.

(AFU) The Actor-Free Updates algorithm represents a significant departure from traditional actor-critic methods. While algorithms like DDPG and SAC update their critic using actions generated by the actor, AFU decouples these components, enabling critic updates that are truly independent of the actor. AFU's architecture consists of two value networks, two advantage networks, a Q-network and a policy network. The advantage networks estimates the the advantage function $A(s, a)$, which represents how much better taking action a in state s is compared to the average action.

The key innovation in AFU lies in how it computes the value and advantage functions. Instead of relying on the actor to estimate the maximum value of $Q(s, a)$ over actions, AFU directly decomposes $Q(s, a)$ into $V(s) + A(s, a)$. This decomposition, combined with a conditional gradient scaling mechanism, allows AFU to solve the maximization problem in Q-learning without depending on the actor. This mechanism allows AFU to maintain stable learning by adaptively adjusting the gradient flow depending on whether the current estimate falls short of the target.

(IQL)

(Cal-QL)

(Off-policy learning experimental setup) Our first experiment aims to evaluate whether AFU can truly learn from randomly generated data, a capability that would distinguish it from traditional off-policy algorithms like DDPG and SAC. The hypothesis stems from the theoretical properties of AFU's critic update mechanism, which, unlike SARSA-based algorithms, does not rely on the actor's improvement to approximate the max-Q operation.

Traditional actor-critic algorithms like SAC and DDPG are structurally closer to SARSA than to Q-learning. Despite being categorized as off-policy, they depend on the actor to generate actions for the critic's updates, creating an implicit coupling. In contrast, AFU's value and advantage decomposition, combined with conditional gradient scaling, allows it to compute critic updates independently of the actor, potentially enabling true off-policy learning.

To test this hypothesis, we designed a protocol where instead of collecting experience through environment interaction with the current policy, we randomly sample states from the observation space and actions from the action space. This sampling is uniform, representing the extreme case of off-policy data with no correlation to any learning policy. If AFU can converge under these conditions while traditional algorithms struggle, it would validate its theoretical advantage in truly off-policy learning.

(Offline-to-online transition experimental setup) Our second experiment investigates the stability of algorithms during the transition from offline to online learning. This transition presents a significant challenge due to the distribution shift between the fixed offline dataset and the data generated by the current policy during online learning.

Offline reinforcement learning algorithms often employ conservatism to prevent overestimation of out-of-distribution actions. While this conservatism is beneficial during offline learning, it can hinder exploration during subsequent online learning. Algorithms like IQL and CALQL address this through different mechanisms: IQL uses expectile regression to estimate maximum values without querying unseen actions, while CALQL explicitly constrains Q-values for out-of-distribution actions.

We hypothesize that AFU may exhibit superior stability during this transition due to its ability to estimate maximum Q-values without relying on the actor. This capability potentially reduces the need for explicit conservatism, allowing AFU to adapt more smoothly when transitioning to online learning.

To test this hypothesis, we generated datasets from policies at different stages of training, capturing trajectories at fixed intervals during policy learning. We then trained algorithms offline on these datasets before transitioning to online learning. The primary evaluation metric is the area between the learning curve and the maximum stable performance level following the transition point, which quantifies how quickly and stably an algorithm recovers during the online phase.

Offline to Online Training.

(First Results)

We also evaluated AFU in an Offline-to-Online setting, where the agent first learns from a fixed dataset and then transitions to online learning. We used the same datasets as in the previous experiments, but this time we trained AFU for 1 million steps in an offline setting before switching to online training.

Conclusion

Appendix A

Experimental Details.

(Environment Wrappers)

The custom environment wrappers used in this study include the following methods:

– `_set_state`: Allows resetting the environment to a specific state.

– `get_obs`: Retrieves the current observation.
– `get_observation_space`: Returns the observation space of the environment.
– `get_action_space`: Returns the action space of the environment.

These wrappers were implemented to ensure compatibility with the algorithms and to facilitate reproducibility of the experiments.

(Hyperparameters)

The following hyperparameters were used for the experiments:
– Learning rate:
– Discount factor (γ):
– Batch size:
– Replay buffer size:
– Target network update rate (τ):
– Temperature parameter (α):

Reproducibility.

The code for all algorithms and experiments is available at [GitHub Repository]. The repository includes detailed instructions for setting up the environment and running the experiments.

Additional Figures.

Figures referenced in the report, including learning curves and performance histograms, are provided in the supplementary materials. These figures illustrate the comparative performance of the algorithms across different environments and settings.

Appendix B

Cal-QL errors.

Bibliographie

[1] Ilya Kostrikov, Ashvin Nair, et Sergey Levine. 2021. Offline Reinforcement Learning with Implicit Q-Learning. Consulté à l'adresse https://arxiv.org/abs/2110.06169

[2] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, et Sergey Levine. 2024. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. Consulté à l'adresse https://arxiv.org/abs/2303.05479

[3] Nicolas Perrin-Gilbert. 2024. AFU: Actor-Free critic Updates in off-policy RL for continuous control. Consulté à l'adresse https://arxiv.org/abs/2404.16159

[4] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, et Daan Wierstra. 2019. Continuous control with deep reinforcement learning. Consulté à l'adresse https://arxiv.org/abs/1509.02971

[5] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, et Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. Consulté à l'adresse https://arxiv.org/abs/1801.01290