Dual Critic Reinforcement Learning under Partial Observability

Jinqiu Li[1,2], **Enmin Zhao**[1,2], **Tong Wei**[3], **Junliang Xing**[3*], **Shiming Xiang**[1,2*]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Department of Computer Science and Technology, Tsinghua University
{lijinqiu2021, zhaoenmin2018}@ia.ac.cn, wt22@mails.tsinghua.edu.cn
jlxing@tsinghua.edu.cn, smxiang@nlpr.ia.ac.cn

Abstract

Partial observability in environments poses significant challenges that impede the formation of effective policies in reinforcement learning. Prior research has shown that borrowing the complete state information can enhance sample efficiency. This strategy, however, frequently encounters unstable learning with high variance in practical applications due to the over-reliance on complete information. This paper introduces DCRL, a **D**ual **C**ritic **R**einforcement **L**earning framework designed to adaptively harness full-state information during training to reduce variance for optimized online performance. In particular, DCRL incorporates two distinct critics: an *oracle* critic with access to complete state information and a *standard* critic functioning within the partially observable context. It innovates a synergistic strategy to meld the strengths of the oracle critic for efficiency improvement and the standard critic for variance reduction, featuring a novel mechanism for seamless transition and weighting between them. We theoretically prove that DCRL mitigates the learning variance while maintaining unbiasedness. Extensive experimental analyses across the Box2D and Box3D environments have verified DCRL's superior performance.

1 Introduction

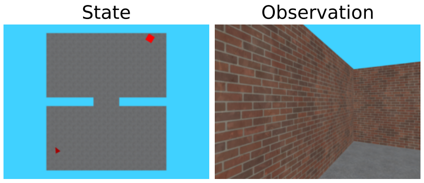

Figure 1: The WallGap environment, a procedurally generated navigation task in MiniWorld. The agent's goal (red triangle) is to reach the target (red box) within as few steps as possible under partial observability.

Real-world issues commonly involve a degree of partial observability, where agents lack access to the full system state and must base decisions on partial observations. These problems are typically defined as partially observable Markov decision processes (POMDPs) [1]. Despite the progress in reinforcement learning (RL) for addressing complex tasks [2–6], learning effective strategies in scenarios with partial observations remains challenging. Traditional RL methods in POMDP contexts often encounter impractical computations in processing the complete history, leading to suboptimal performance. Luckily, full states are accessible during training in many training scenarios. For example, in applications like autonomous driving[7] or robot control tasks [8], where computational and time limitations exist, only low-cost RGB images are available for deployment. At the same time, high-precision sensors can record more accurate and

[*]Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

detailed data during training. Similarly, in card games [9], the private hands of other players are typically hidden during deployment but revealed during training. This distinction of separating available information during training and deployment stages has driven the adoption of an offline learning and online execution paradigm in POMDP settings [10–13].

This study focuses on the asymmetric actor-critic framework [8, 14, 15], where the actor and critic rely on different inputs. The critic, serving as an auxiliary tool for training, can be transformed into an oracle critic to align with the offline learning and online execution paradigm. Specifically, the oracle critic is empowered to access the complete states while the actor remains constrained to partial observations. Using an oracle critic conditioned on the state and observation history promises to enhance performance and theoretically achieve unbiasedness [10]. However, the practical performance of such approaches raises doubts, particularly in domains characterized by high uncertainty and extensive state spaces. For instance, in the illustrated first-person 3D navigation task depicted in Figure 1, the agent is tasked with reaching a dynamic target that varies across episodes within as few steps as possible. This task involves a first-person view observation and an aerial view state. Introducing an oracle critic incorporating state information in this complex environment may increase variance, potentially undermining learning efficiency and overall performance.

To tackle the challenges mentioned above, we argue that the indiscriminate utilization of state information is inadvisable. We introduce a Dual Critic Reinforcement Learning framework (DCRL). DCRL consists of two critics: the *oracle* critic conditioned on the <history, state> tuple, and the *standard* critic conditioned solely on history. It introduces a synergistic strategy to meld the strengths of the oracle critic for efficiency improvement and the standard critic for variance reduction. In summary, the main contributions of this work are as follows:

- We propose a DCRL framework to address partially observable tasks. This framework features a novel mechanism for seamless transition and weighting between the dual critics.
- We theoretically prove that DCRL mitigates learning variance while maintaining unbiasedness.
- The DCRL framework is simple to implement. While our emphasis lies on the Advantage Actor-Critic (A2C) [16], DCRL can also be integrated into other actor-critic architectures, such as Proximal Policy Optimization (PPO) [17].

We extensively validate the DCRL framework based on A2C and PPO. Experimental results conducted on both Box2D and Box3D environments confirm its effectiveness. The source code is available in the supplementary material.

2 Related Work

Most methods can be adapted for partial observability by training memory-based policies to condition past observations. Recurrent Neural Networks (RNNs) [18] are commonly used for their simplicity and strong performance. Deep Recurrent Q-Networks (DRQN) [19] integrates Long Short-Term Memory (LSTM) [20] cells into the Deep Q-Network (DQN) [2] architecture, which demonstrated promising results. In addition to LSTM, other memory mechanisms such as Gated Recurrent Units (GRUs) [21, 22], Transformers [23, 24], and Language Models [25, 26] have been adopted in RL algorithms for memory learning. However, POMDPs are hard to solve when relying solely on memory-based policies because the size of the history grows linearly with the horizon length.

It is crucial that in many environments, complete states are observable during the training phase. Utilizing full observable states during training has emerged as a popular paradigm in RL. We discuss prior works on the utilization of full observability in RL algorithms. Additionally, we provide further discussion about training the world model, applications in multi-agent reinforcement learning, and other double structures in Appendix B.

Using complete states to train the critic. This topic is closely related to our paper and has been applied in many large-scale scenarios, such as DouDizhu [9] and Mahjong [27]. The critic in the actor-critic framework is a training tool not required during deployment, making its structure naturally suitable to exploit the state. This is achieved by critic asymmetry, where the actor and critic receive history and state, respectively. Asymmetric Actor-Critic [8] uses the critic conditioned on ground-truth states to address image-based robot control tasks. AACC [15] employs the critic conditioned on both states and observations to solve contextual Markov decision processes. Theoretically, using the state as the critic input is biased, whereas using the <history, state> tuple is unbiased in expectation [10]. However, it introduces high variance in practice.

Using complete states to train the policy. These methods typically involve training a teacher policy using complete state information and then utilizing this policy to guide the learning of a student policy under partial observability. The forms of guidance vary, such as maintaining consistency of the Q-function [28], the encoder [11], action-distributions [29], and others. In addition to introducing extra loss terms to enforce consistency between the student policy and the teacher policy, the transition from the teacher policy to the student policy can also be facilitated through dropout of state information from the input [30–32]. Common issues include the imitation gap and suboptimal teacher, which can limit the potential capabilities of the student policy.

Using complete states to learn the belief representation. Belief state methods [1, 33] are classical in POMDPs. These methods often necessitate knowledge of the dynamics and reward functions, thus rendering them more applicable to scenarios with well-defined system dynamics. Consequently, they are typically constrained to simpler settings characterized by small state spaces, where the complexities of the environment are more tractable. Several recent studies [34–36] focus on learning belief representations by predicting future observations and rewards. VRM [34] utilizes a latent stochastic variable to model sequential observations. However, learning latent states that can capture long-term dependencies is challenging. Incorporating the complete states can help alleviate this issue. DouZero+ [37] introduces auxiliary tasks for predicting states while concurrently training policies conditioned on observations and the predicted states. Believer [13] decouples belief state modeling from policy optimization. They first capture state features through representation learning and then train the policy conditioned on observations and the predicted states.

3 Preliminaries

This section introduces the background topics required to understand our work: partially observable Markov decision processes, the recurrent Advantage Actor-Critic algorithm, and the Unbiased Asymmetric Actor-Critic algorithm.

3.1 Partially Observable Markov Decision Processes

A partially observable Markov decision process (POMDP) [1] is formally identified as a 7-tuple, namely $P = \langle \mathcal{S}, \mathcal{A}, T, \Omega, R, O, \gamma \rangle$, where $\mathcal{S}$, $\mathcal{A}$, and Ω represent the state space, action space, and observation space, respectively. The transition function $T : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$ is denoted as $T(s, a, s') = p(s'|s, a)$, describing the probability of transitioning from state s to state s' after taking action a. The agent receives observations rather than states, and the observation function $O : \mathcal{S} \times \mathcal{A} \to \Delta\Omega$ is defined as $O(s, a, o)$ representing the probability of observing o conditioned on state s and action a. The reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is expressed as $R(s, a)$, representing the reward received by acting a in state s, and $\gamma \in [0, 1]$ is the discount factor. In POMDPs, the optimal policy may depend on the entire history of observations and actions, denoted as $\mathcal{H} = \{h_t | h_t = (o_0, a_0, ...o_{t-1}, a_{t-1}, o_t)\}$. The belief-state $b : \mathcal{H} \to \Delta\mathcal{S}$ is the probability distribution over states s given history h. The history reward function is defined as $R(h, a) = \mathbb{E}_{s|h}[R(s, a)]$. The agent aims to maximize the expected cumulative discounted rewards $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$ through interactions with the environment.

This paper assumes that both states s_t and observations o_t are accessible during training, while only the observations o_t are accessible during deployment.

3.2 Recurrent Advantage Actor-Critic

Advantage Actor-Critic (A2C) [16] is primarily designed for fully observable control problems. It can be easily adapted to handle partially observable control problems by replacing all occurrences of the state s with the respective history h. To clearly distinguish between different methods, we denote this method as **AC**. It trains an actor model $\pi : \mathcal{H} \to \Delta\mathcal{A}$ parameterized by θ, and a critic model $V : \mathcal{H} \to \mathbb{R}$ parameterized by ϕ. The policy loss is defined as:

$$\mathcal{L}_{\text{policy}}^{\text{AC}}(\theta) = -\mathbb{E}\Big[\sum_{t=0}^{\infty} Q^{\pi}(h_t, a_t) \log \pi_\theta(a_t|h_t)\Big], \tag{1}$$

where $Q^{\pi}(h_t, a_t)$ denotes the history-action value of the policy π. In practice, $Q^{\pi}(h_t, a_t)$ is replaced by the n-step bootstrapped advantage to reduce variance:

$$\delta_{\phi}(h_t, a_t) = \sum_{d=0}^{n-1} \gamma^d R(h_{t+d}, a_{t+d}) + \gamma^n V_{\phi}(h_{t+n}) - V_{\phi}(h_t). \tag{2}$$

Correspondingly, the policy loss is $\mathcal{L}^{\text{AC}}_{\text{policy}}(\theta) = -\mathbb{E}\big[\sum_{t=0}^{\infty} \delta_{\phi}(h_t, a_t) \log \pi_{\theta}(a_t|h_t)\big]$. In this work, the value loss is defined as $\mathcal{L}^{\text{AC}}_{\text{value}}(\theta) = \mathbb{E}\big[\sum_{t=0}^{\infty} \delta_{\phi}(h_t, a_t)^2\big]$. The entropy loss is $\mathcal{L}_{\text{entropy}}(\theta) = -\mathbb{E}\Big[\sum_{t=0}^{\infty} \sum_{a\in\mathcal{A}} \pi_{\theta}(a_t|h_t) \log \pi_{\theta}(a_t|h_t)\Big]$. The total loss is $\mathcal{L}^{\text{AC}}_{\text{value}}(\phi) + \mathcal{L}^{\text{AC}}_{\text{policy}}(\theta) + \eta\mathcal{L}_{\text{entropy}}(\theta)$, where η is a weight for entropy regularization.

3.3 Unbiased Asymmetric Actor-Critic

Unbiased Asymmetric Actor-Critic [10] is a variant of recurrent A2C that can exploit state information during training. It considers an oracle critic model $V : \mathcal{H} \times \mathcal{S} \to \mathbb{R}$ parameterized by ψ instead of the standard critic model. To clearly distinguish between different methods, we denote this method as **UAAC**. The n-step bootstrapped advantage for UAAC is defined as:

$$\delta_{\psi}(h_t, s_t, a_t) = \sum_{d=0}^{n-1} \gamma^d R(s_{t+d}, a_{t+d}) + \gamma^n V_{\psi}(h_{t+n}, s_{t+n}) - V_{\psi}(h_t, s_t). \tag{3}$$

Correspondingly, the policy model is trained to minimize the following objective function:

$$\mathcal{L}^{\text{UAAC}}_{\text{policy}}(\theta) = -\mathbb{E}\Big[\sum_{t=0}^{\infty} \delta_{\psi}(h_t, s_t, a_t) \log \pi_{\theta}(a_t|h_t)\Big]. \tag{4}$$

The value loss is defined as $\mathcal{L}^{\text{UAAC}}_{\text{value}}(\psi) = \mathbb{E}\big[\sum_{t=0}^{\infty} \delta_{\psi}(h_t, s_t, a_t)^2\big]$. Compared with the standard actor-critic method, the state s provides additional context to determine the agent's true underlying situation, which may facilitate the learning of the oracle critic. Importantly, it has been proven that $V^{\pi}(h, s)$ provides Monte Carlo estimates of the respective $V^{\pi}(h)$ for arbitrary policies, which is theoretically unbiased, i.e.,

$$V^{\pi}(h) = \mathbb{E}_{s|h}[V^{\pi}(h, s)], \tag{5}$$

$$Q^{\pi}(h, a) = \mathbb{E}_{s|h}[Q^{\pi}(h, s, a)]. \tag{6}$$

Therefore, the equality $\nabla_{\theta}\mathcal{L}^{\text{AC}}_{\text{policy}}(\theta) = \nabla_{\theta}\mathcal{L}^{\text{UAAC}}_{\text{policy}}(\theta)$ holds, indicating that the policy gradient remains unchanged in expectation.

4 Dual Critic Reinforcement Learning

Critic models that learn accurate values slowly often become a bottleneck for policy performance in many scenarios. When state information is incorporated, it may be easier to learn $V^{\pi}(h, s)$ with a neural network compared with learning $V^{\pi}(h)$, leading to faster policy learning. However, $V^{\pi}(h, s)$ has nonzero variance. Taking into account the variance arising from $V^{\pi}(h, s)$: $\text{Var}_{s|h}[V^{\pi}(h, s)] = \mathbb{E}_{s|h}\big[\|V^{\pi}(h, s) - V^{\pi}(h)\|^2\big]$, with a practical batch size that is not too large, the variance tends to be high, especially in high-uncertainty belief states and large state spaces. In games like DouDizhu, when the initial information collection does not accurately infer the opponents' hands, training with $V^{\pi}(h, s)$ incorporating complete state information increases variance, thus limiting model generalization. Conversely, using $V^{\pi}(h)$, which relies solely on observations, enhances robustness in such cases. *This concept is akin to replacing the reward-to-go term with the Q-value to reduce variance in the vanilla policy gradient method, as discussed in [38, 39].*

4.1 Theories of the Dual Critic

Definition 1. Define the dual value function as $V^{\pi}_{\text{dual}}(h, s) = (1 - \beta)V^{\pi}(h, s) + \beta V^{\pi}(h)$, where β is a weight that balances the contributions of the $V^{\pi}(h, s)$ and $V^{\pi}(h)$, and $\beta \in [0, 1]$. Correspondingly, the dual Q function is defined as $Q^{\pi}_{\text{dual}}(h, s, a) = (1 - \beta)Q^{\pi}(h, s, a) + \beta Q^{\pi}(h, a)$.

Theorem 1. *For arbitrary control problems and policies, $V^{\pi}_{\text{dual}}(h,s)$ is an unbiased estimate of $V^{\pi}(h)$ and $Q^{\pi}_{\text{dual}}(h,s,a)$ is an unbiased estimate of $Q^{\pi}(h,a)$.*

Proof. Following from Equation (5), we have:

$$\begin{aligned}\mathbb{E}_{s|h}[V^{\pi}_{\text{dual}}(h,s)] &= \mathbb{E}_{s|h}\big[(1-\beta)V^{\pi}(h,s)+\beta V^{\pi}(h)\big]\\ &= (1-\beta)\mathbb{E}_{s|h}[V^{\pi}(h,s)]+\beta V^{\pi}(h)\\ &= V^{\pi}(h).\end{aligned} \tag{7}$$

Similarly, as inferred from Equation (6), $\mathbb{E}_{s|h}[Q^{\pi}_{\text{dual}}(h,s,a)] = Q^{\pi}(h,a)$ by replacing the V with Q using the same approach outlined above. □

Therefore, the variance of the dual value is defined as $\text{Var}_{s|h}[V^{\pi}_{\text{dual}}(h,s)] = \mathbb{E}_{s|h}\big[\|V^{\pi}_{\text{dual}}(h,s) - V^{\pi}(h)\|^2\big]$, and that of Unbiased Asymmetric Actor-Critic is defined as $\text{Var}_{s|h}[V^{\pi}(h,s)] = \mathbb{E}_{s|h}\big[\|V^{\pi}(h,s) - V^{\pi}(h)\|^2\big]$.

Theorem 2. *The variance of the dual value is smaller than that of the Unbiased Asymmetric Actor-Critic, i.e.,* $\text{Var}_{s|h}[V^{\pi}_{\text{dual}}(h,s)] \leq \text{Var}_{s|h}[V^{\pi}(h,s)]$.

Proof. According to the definitions and $\beta \in [0,1]$, it holds that

$$\begin{aligned}\text{Var}_{s|h}[V^{\pi}_{\text{dual}}(h,s)] &= \text{Var}_{s|h}\big[(1-\beta)V^{\pi}(h,s)+\beta V^{\pi}(h))\big]\\ &= (1-\beta)^2\text{Var}_{s|h}[V^{\pi}(h,s)]\\ &\leq \text{Var}_{s|h}[V^{\pi}(h,s)].\end{aligned} \tag{8}$$

Similarly, $\text{Var}_{s|h}[Q^{\pi}_{\text{dual}}(h,s,a)] \leq \text{Var}_{s|h}[Q^{\pi}(h,s,a)]$ by replacing the V with Q using the same approach outlined above. □

Considering the unbiasedness demonstrated above, we contemplate employing the dual Q function for policy updates. The policy gradient based on the dual Q function can be formulated as:

$$\nabla_{\theta}\mathcal{L}^{\text{dual}}_{\text{policy}}(\theta) = -\mathbb{E}\Big[\sum_t Q^{\pi}_{\text{dual}}(h_t,s_t,a_t)\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]. \tag{9}$$

This formulation maintains the policy gradient unchanged in expectation.

Theorem 3. *The dual Q function maintains the policy gradient unchanged in expectation.*

Proof. Following Theorem 1, therefore,

$$\begin{aligned}\nabla_{\theta}\mathcal{L}^{AC}_{\text{policy}}(\theta) &= -\mathbb{E}\Big[\sum_t Q^{\pi}(h_t,a_t)\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]\\ &= -\sum_t\ E_{h_t,a_t}\Big[Q^{\pi}(h_t,a_t)\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]\\ &= -\sum_t\mathbb{E}_{h_t,a_t}\Big[\mathbb{E}_{s_t|h_t}[Q^{\pi}_{\text{dual}}(h_t,s_t,a_t)]\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]\\ &= -\sum_t\mathbb{E}_{h_t,s_t,a_t}\Big[Q^{\pi}_{\text{dual}}(h_t,s_t,a_t)\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]\\ &= -\mathbb{E}\Big[\sum_t Q^{\pi}_{\text{dual}}(h_t,s_t,a_t)\nabla_{\theta}\log\pi_{\theta}(a_t|h_t)\Big]\\ &= \nabla_{\theta}\mathcal{L}^{\text{dual}}_{\text{policy}}(\theta).\end{aligned} \tag{10}$$

□

Similarly, the dual Q function decreases the variance of the policy gradient according to Theorem 2.

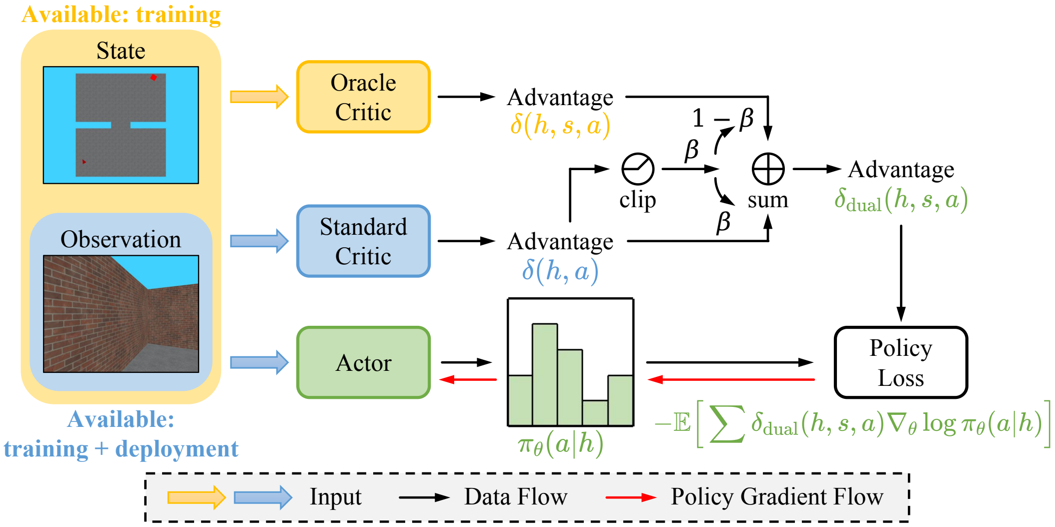

Figure 2: Implementation of DCRL framework. DCRL innovates a synergistic strategy to meld the strengths of the oracle critic for efficiency improvement and the standard critic for variance reduction.

4.2 Learning with Dual Critics

Building upon the abovementioned theory, we incorporate the dual Q function to balance variance and efficiency. This integration introduces the Dual Critic Reinforcement Learning framework (DCRL). As shown in Figure 2, the DCRL consists of three main components: 1) a *standard* critic network $V(h)$ parameterized by ϕ, 2) an *oracle* critic network $V(h, s)$ parameterized by ψ, and 3) an actor model $\pi(a|h)$ parameterized by θ. Inspired by the Equation (9) and Theorem 3, we propose the policy and value losses of DCRL as the following objective functions:

$$\mathcal{L}_{\text{policy}}^{\text{DCRL}}(\theta) = -\mathbb{E}\Big[\sum_{t=0}^{\infty}\big[(1-\beta)\delta_\psi(h_t, s_t, a_t)\text{log}\pi_\theta(a_t|h_t) + \beta\delta_\phi(h_t, a_t)\text{log}\pi_\theta(a_t|h_t)\big]\Big], \tag{11}$$

$$\mathcal{L}_{\text{value}}^{\text{DCRL}}(\phi, \psi) = \mathbb{E}\Big[\sum_{t=0}^{\infty}\delta_\phi(h_t, a_t)^2\Big] + \mathbb{E}\Big[\sum_{t=0}^{\infty}\delta_\psi(h_t, s_t, a_t)^2\Big], \tag{12}$$

where $\delta_\phi(h_t, a_t)$ and $\delta_\psi(h_t, s_t, a_t)$ follow the definitions from Equation (2) and Equation (3). The dual advantage is formulated as $\delta_{\text{dual}}(h_t, s_t, a_t) = (1-\beta)\delta_\psi(h_t, s_t, a_t) + \beta\delta_\phi(h_t, a_t)$.

We propose a novel weighting mechanism to synergistically meld the two critics and thoroughly exploit the complete state information. When the received return does not exceed the agent's value estimate, the advantage is non-positive ($\delta_\phi(h_t, a_t) \leq 0$), β is clipped to 0. In cases with $\delta_\phi(h_t, a_t) > 0$, β takes effect. β is a parameter associated with h but not with s. Hence, this mechanism maintains the theory mentioned above. This mechanism helps stabilize training and prevent large updates in cases where the advantage is non-positive. Similar to Self-Imitation Learning [40], this clipping mechanism can be viewed as a form of lower-bound-soft-Q-learning, where only samples with a positive advantage provide valuable insights into the optimal soft Q-value. From an empirical perspective, this approach embodies the notion that if there is still room for policy enhancement after being updated by the oracle critic $V(h, s)$, the standard critic $V(h)$ should be engaged. With full-state information from the oracle critic, DCRL avoids falling into a vicious cycle of poor policy and bad value estimates. The standard critic in DCRL can effectively mitigate variance, whereas recurrent Actor-Critic methods preclude such opportunities due to entanglement in a vicious cycle. These two critics establish a mutually supportive relationship, facilitating variance reduction while effectively using full-state information. The proof regarding lower-bound-soft-Q-learning is provided in Appendix C.

In addition to the mechanism mentioned above, there is another natural way to leverage the dual-critic framework by interchanging the roles of the oracle and standard critics. Specifically, we shift the weighting factor to the forefront of the $\delta_\psi(h_t, s_t, a_t)$ term which denote as β', and the inside term

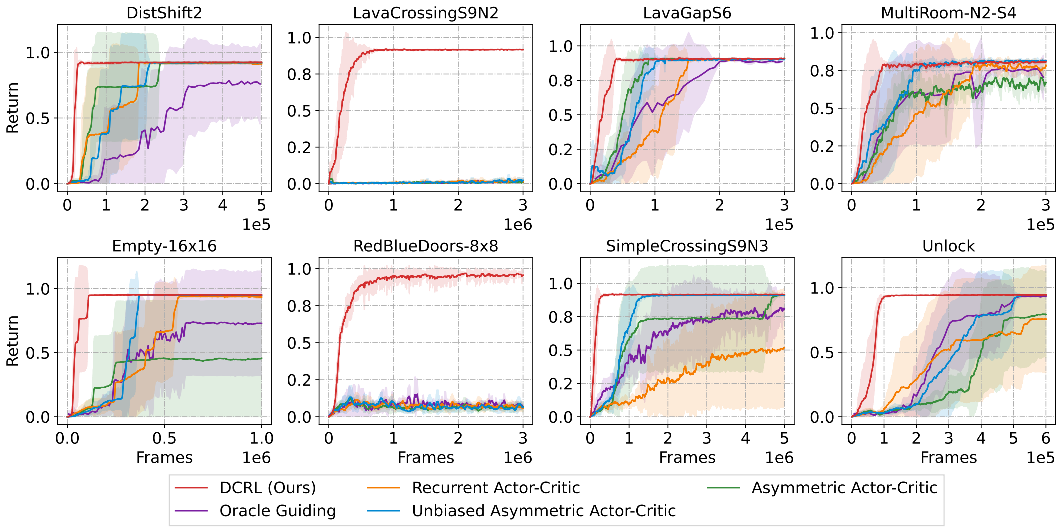

Figure 3: Learning curves on MiniGrid. The x-axis and y-axis represent the training frames and average returns, respectively. Shaded bars illustrate the standard error. All curves are trained based on A2C over 5 random seeds.

of the policy loss is defined as $(1-\beta')\delta_\phi(h_t, a_t)\log\pi_\theta(a_t|h_t) + \beta'\delta_\psi(h_t, s_t, a_t)\log\pi_\theta(a_t|h_t)$. β' is clipped to 0 when $\delta_\psi(h_t, s_t, a_t) \leq 0$ and otherwise takes a positive value. β' is a parameter associated with both h and s. Unlike DCRL, this alternative technique lacks the characteristic of unbiasedness and fails to thoroughly exploit state information while deviating from the lower-bound-soft-Q-learning framework. As a result, its performance exhibits instability compared with DCRL. The experimental section provides a detailed discussion of this technique's performance.

5 Experiments

We conduct a series of experiments to evaluate the effectiveness of DCRL in partially observable domains. Our research focuses on the model-free setting where complete states are available during training. We compare our DCRL approach with several baselines.

Baselines. Recurrent Actor-Critic is a recurrent version of the standard actor-critic algorithm, namely A2C [16] or PPO [17]. This baseline represents methods that do not utilize state information. Asymmetric Actor-Critic [8] is a representation method based on critic asymmetry. The critic uses the state as input, independent of the history. Oracle Guiding [30] is a representation method based on teacher policy. This approach gradually removes state information from the policy input, transitioning the teacher policy to a student policy under partial observability. The dropout probability linearly increases with training iterations until only observation remains. We adopt a 50% cutoff for all tasks. Unbiased Asymmetric Actor-Critic [10] is a state-of-the-art method for POMDPs. The critic uses the <history, state> tuple as input to exploit state information while introducing no bias into the training process, thus achieving superior performance. To ensure a fair comparison, we keep the main network architectures similar across all methods. We present the mean and standard deviation of results obtained from five different seeds for each environment. For a comprehensive overview of other implementation details and hyperparameters, please refer to Appendix E.

5.1 MiniGrid

To see how useful DCRL is across various POMDP tasks, we compare various methods in Box2D environments. MiniGrid [41] is a procedurally generated environment with goal-oriented tasks. As shown in Figure 4, the triangular agent with a discrete action space aims to reach the goal. These tasks involve interacting with different objects and solving diverse maze maps that vary in each episode. The observation comprises a first-person view representing a $7 \times 7 \times 3$ image. At the same time, the state offers a fully observable perspective, with the image size varying according to the tasks. The image is compact and efficient encoding, rather than raw pixels.

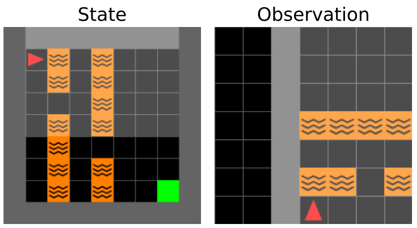

Figure 4: The LavaCrossingS9N2 environment in MiniGrid. The agent's goal (red triangle) is to reach the target (green box) within as few steps as possible under partial observability.

All experiments are conducted based on A2C, and the results are plotted in Figure 3. Recurrent Actor-Critic struggles to learn a robust policy across most environments, primarily due to the agent's need for extensive exploration without state information. Despite leveraging states, Oracle Guiding and Asymmetric Actor-Critic performance exhibits instability, attributed to their heuristic nature and lack of theoretical guarantees. In contrast, the Unbiased Asymmetric Actor-Critic maintains stability and outperforms the mentioned methods. Nonetheless, it faces challenges in scenarios such as *LavaCrossingS9N2* and *RedBlueDoors-8x8*. Our DCRL framework surpasses all other methods in eight environments, particularly excelling in challenging tasks. Taking *LavaCrossingS9N2* and *Empty-16x16* as examples, the map size of the former is smaller, but the layout is more complex. In the case of *Empty-16x16*, the states remain simple and relatively constant, making it easy to infer key information from history. Conversely, in *LavaCrossingS9N2*, objects change between episodes, leading to high uncertainty in the belief state. While leveraging states aids in precise value function learning, the variance poses challenges in policy optimization. DCRL effectively overcomes these limitations and emerges as a formidable competitor against other baseline methods.

Table 1: Performance of agents on MiniGrid after $1e5$ frames (27 games) and $1e6$ frames (13 games) of training. We report the mean and standard error of the performance.

Methods	A2C		PPO	
	$1e5$ frames	$1e6$ frames	$1e5$ frames	$1e6$ frames
Recurrent Actor-Critic	0.32±0.13	0.28±0.09	0.26±0.07	0.52±0.03
Asymmetric Actor-Critic	0.26±0.10	0.32±0.14	0.27±0.08	0.60±0.06
Oracle Guiding	0.18±0.13	0.27±0.06	0.25±0.10	0.48±0.05
Unbiased Asymmetric Actor-Critic	0.31±0.08	0.39±0.09	0.28±0.10	0.59±0.08
DCRL (Ours)	**0.63±0.07**	**0.89±0.01**	**0.47±0.08**	**0.84±0.03**

To further validate the effectiveness of DCRL, we provide the visualization of different critic values in Appendix D.1. The visualization demonstrates that the dual value function $V_{\text{dual}}(h, s)$ effectively reduces variance compared with the oracle critic. Besides, we evaluate our DCRL in 27 MiniGrid tasks based on A2C and PPO. The average score across all games is provided in Table 1 to demonstrate the overall performance for different methods. Our DCRL outperforms other baselines in 26 out of the 27 games. The detailed learning curves are available in Appendix D.2.

5.2 MiniWorld

This section investigates whether DCRL benefits more complex continuous state space tasks and whether it can be applied to other Actor-Critic algorithms, such as PPO. Unlike A2C, PPO lacks a strong theoretical connection to DCRL, but we claim they can still complement each other. To empirically verify this, we implement all methods based on PPO and evaluate them on the MiniWorld domain [41]. The observations and states are represented as $80 \times 60 \times 3$ images rendered as pixels. The results depicted in Figure 5 illustrate that DCRL enhances PPO performance across all four environments. In contrast to MiniGrid, the state format in this benchmark is not a concise encoding and tends to contain redundancy and noise. Although states assist the agent in effectively discerning information about the environment, the variance introduced by noise poses a challenge to the learning process. Furthermore, it is observed that the Unbiased Asymmetric Actor-Critic performs poorer than the Recurrent Actor-Critic across all tasks. Nevertheless, DCRL converges to superior policies by synergistically melding the two critics.

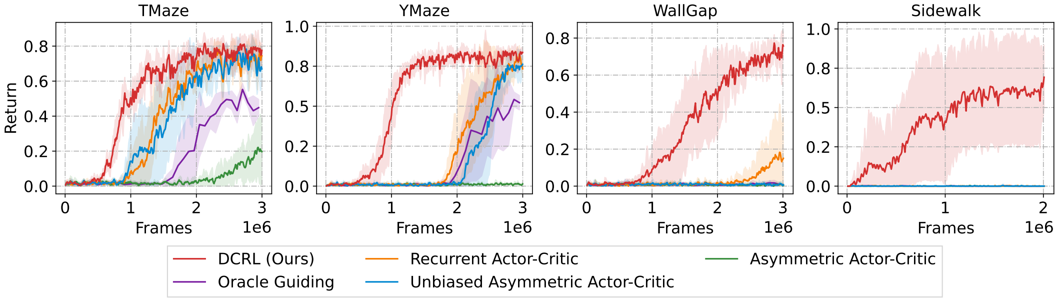

Figure 5: Learning curves on MiniWorld. The x-axis and y-axis represent the training frames and average returns, respectively. All curves are trained based on PPO over 5 random seeds.

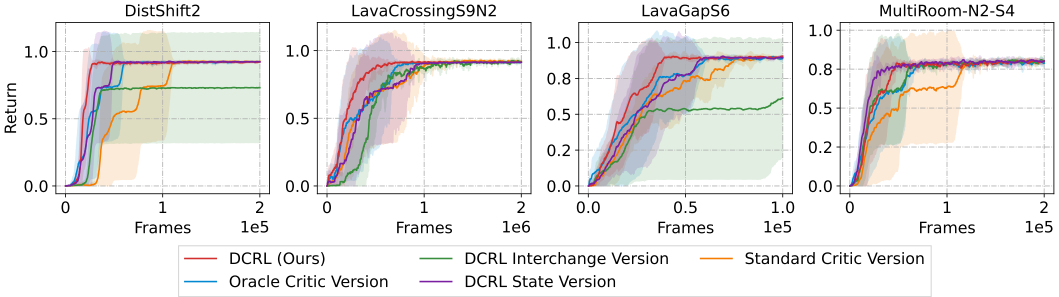

Figure 6: Ablation studies on MiniGrid to verify the two key factors of DCRL. All curves are trained based on A2C across 5 random seeds.

5.3 Ablation Studies

Ablation analysis of components in DCRL. We conduct detailed ablation studies to analyze how DCRL improves efficiency and enhances performance. DCRL contains two key factors: the dual-critic structure and the weighting mechanism. We replace the dual-critic structure with the single-critic structure to verify the structure's effectiveness. The Standard Critic Version and Oracle Critic Version are constructed with the following expressions for the inside terms of the policy loss: $\delta_\phi(h_t, a_t)\log\pi_\theta(a_t|h_t) + \beta\delta_\phi(h_t, a_t)\log\pi_\theta(a_t|h_t)$ and $\delta_\psi(h_t, s_t, a_t)\log\pi_\theta(a_t|h_t) + \beta'\delta_\psi(h_t, s_t, a_t)\log\pi_\theta(a_t|h_t)$. Next, we evaluate another weighting mechanism baseline (DCRL Interchange Version) discussed in Section 4.2, with the inside terms of the policy loss as $(1 - \beta')\delta_\phi(h_t, a_t)\log\pi_\theta(a_t|h_t) + \beta'\delta_\psi(h_t, s_t, a_t)\log\pi_\theta(a_t|h_t)$. The adjustment of β and β' follows the same way as discussed in Section 4.2. Additionally, we introduce an intuitive variant that modifies the *standard* critic's input from history to state, referred to as the DCRL State Version. This adjustment aligns with previous works that introduce biases; however, it has the potential to enhance efficiency gains. All methods utilize the same architectures and hyperparameters as DCRL.

We select four environments from the MiniGrid domain, where state information plays a crucial role. In these environments, methods that utilize states perform significantly better than those that do not. We compare our DCRL approach with four ablation baselines, and the results are presented in Figure 6. It can be observed that DCRL converges to optimal policies faster than other methods. Intuitively, incorporating the concept of positive advantage (δ_+) can optimize the utilization of beneficial samples and expedite the training process. Consequently, single-critic structure methods integrating a δ_+ term outperform those lacking it, as demonstrated by the superior performance of approaches like the Standard Critic Version compared with the Recurrent Actor-Critic. However, it is crucial that the two critics fulfill distinct and orthogonal roles within the dual-critic structure. DCRL incorporates an oracle critic that utilizes state information compared with the Standard Critic Version. The inclusion of state information provides the agents with additional context to accurately determine the environment's true state, facilitating learning an accurate value function and ultimately improving performance. In comparison to the Oracle Critic Version, DCRL reduces variance. These factors contribute to DCRL's efficiency. As for the DCRL Exchange Version, although it is also a dual-critic structure, its policy gradient is biased, and its weighting mechanism

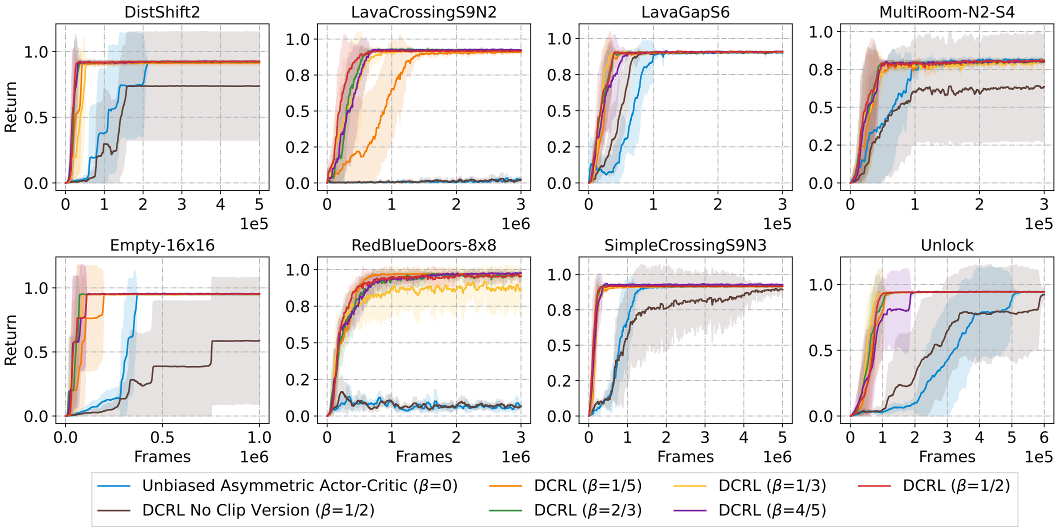

Figure 7: Ablation studies on MiniGrid to analyze the robustness of β in DCRL. All curves are trained based on A2C across 5 random seeds.

does not utilize state information as effectively as DCRL. Its performance only shows advantages in limited environments. It converges to suboptimal policies in *DistShift2* and *LavaGapS6*. The DCRL State Version demonstrates commendable performance among the various DCRL variants. However, the biased characteristics of the DCRL State Version contribute to instability in its effectiveness. Notably, our DCRL achieves superior training performance in three out of four environments. These findings further highlight the significance of our DCRL framework.

Ablation analysis of β in DCRL. Our DCRL incorporates a dynamic weighting mechanism, which can be interpreted as a form of lower-bound-soft-Q-learning, distinguishing it from conventional linear weighting methods. This mechanism substantially improves DCRL's performance while maintaining high stability concerning hyperparameters. To substantiate this claim, we implement a simplified weighting method with fixed ratios for weighting the two critics, referred to as the DCRL No Clip Version. Additionally, we conduct ablation experiments on the parameter $\beta \in \{1/5, 1/3, 1/2, 2/3, 4/5\}$. The results depicted in Figure 7 indicate that the DCRL No Clip version shows performance improvements in only some environments compared to the Unbiased Asymmetric Actor-Critic. In contrast, DCRL consistently enhances performance across all environments, underscoring the importance of the dynamic weighting mechanism. Furthermore, the stability of β is notably evident. We choose $\beta = 1/2$ for reporting results in the paper, given its consistently superior performance.

6 Conclusion

This paper introduces a novel reinforcement learning framework designed for partially observable tasks. The motivation behind this work is that complete state information may be accessible during training in many scenarios. However, introducing a state could potentially lead to bias or variance. The proposed DCRL method leverages two asymmetric critics to use the state to expedite training and mitigate variance in practice. We underscore the benefits of training with two critics and provide theoretical evidence supporting the unbiasedness and variance reduction achieved through our approach. Empirical results obtained across multiple tasks serve to validate the efficacy of DCRL.

Limitations. Our framework operates under the assumption that state information is accessible during training. However, this assumption may not always hold, as only part of the state information might be available within a broader context. It is crucial to discuss whether this limitation will impact the performance of our framework, necessitating further investigation. Furthermore, our observations suggest that the availability of complete state information can sometimes hinder performance in certain environments. This observation underscores the need for a more nuanced approach to the design and utilization of state information. Addressing these issues could lead to enhancements in the performance and applicability of our framework.

7 Acknowledgement

This work was supported in part by the Natural Science Foundation of China under Grant No. 62222606 and 62076238.

References

[1] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[5] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[6] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[7] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2151–2170, 2024.

[8] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems*, pages 1–10, 2018.

[9] Guan Yang, Minghuan Liu, Weijun Hong, Weinan Zhang, Fei Fang, Guangjun Zeng, and Yue Lin. PerfectDou: Dominating doudizhu with perfect information distillation. In *Advances in Neural Information Processing Systems*, pages 34954–34965, 2022.

[10] Andrea Baisero and Christopher Amato. Unbiased asymmetric reinforcement learning under partial observability. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 44–52, 2022.

[11] Dongqi Han, Tadashi Kozuno, Xufang Luo, Zhao-Yun Chen, Kenji Doya, Yuqing Yang, and Dongsheng Li. Variational oracle guiding for reinforcement learning. In *International Conference on Learning Representations*, pages 1–13, 2022.

[12] Gaspard Lambrechts, Adrien Bolland, and Damien Ernst. Informed pomdp: Leveraging additional information in model-based rl. *arXiv preprint arXiv:2306.11488*, 2023.

[13] Andrew Wang, Andrew C Li, Toryn Q Klassen, Rodrigo Toro Icarte, and Sheila A McIlraith. Learning belief representations for partially observable deep rl. In *International Conference on Machine Learning*, pages 35970–35988, 2023.

[14] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafa Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *International Journal of Robotics Research*, 39(1):3–20, 2020.

[15] Wangyang Yue, Yuan Zhou, Xiaochuan Zhang, Yuchen Hua, Zhiyuan Wang, and Guang Kou. Aacc: Asymmetric actor-critic in contextual reinforcement learning. *arXiv preprint arXiv:2208.02376*, 2022.

[16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[18] Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[19] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv:1507.06527*, 2015.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[21] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[22] Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. In *International Conference on Machine Learning*, pages 16691–16723, 2022.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[24] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, pages 7487–7498, 2020.

[25] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

[26] Fabian Paischer, Thomas Adler, Vihang Patil, Angela Bitto-Nemling, Markus Holzleitner, Sebastian Lehner, Hamid Eghbal-Zadeh, and Sepp Hochreiter. History compression via language models in reinforcement learning. In *International Conference on Machine Learning*, pages 17156–17185, 2022.

[27] Jinqiu Li, Shuang Wu, Haobo Fu, Qiang Fu, Enmin Zhao, and Junliang Xing. Speedup training artificial intelligence for mahjong via reward variance reduction. In *IEEE Conference on Games*, pages 345–352, 2022.

[28] Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou, Weinan Zhang, Jiang Bian, Yong Yu, and Tie-Yan Liu. Universal trading for order execution with oracle policy distillation. In *AAAI Conference on Artificial Intelligence*, pages 107–115, 2021.

[29] Hai Nguyen, Andrea Baisero, Dian Wang, Christopher Amato, and Robert Platt. Leveraging fully observable policies for learning under partial observability. In *Conference on Robot Learning*, pages 1673–1683, 2023.

[30] Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, and Hsiao-Wuen Hon. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020.

[31] Pierre-Alexandre Kamienny, Kai Arulkumaran, Feryal Behbahani, Wendelin Boehmer, and Shimon Whiteson. Privileged information dropout in reinforcement learning. *arXiv preprint arXiv:2005.09220*, 2020.

[32] Stephan Weigand, Pascal Klink, Jan Peters, and Joni Pajarinen. Reinforcement learning using guided observability. *arXiv preprint arXiv:2104.10986*, 2021.

[33] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, pages 2164–2172, 2010.

[34] Dongqi Han, Kenji Doya, and Jun Tani. Variational recurrent models for solving partially observable control tasks. In *International Conference on Learning Representations*, pages 1–12, 2020.

[35] Yuhui Wang and Xiaoyang Tan. Deep recurrent belief propagation network for pomdps. In *AAAI Conference on Artificial Intelligence*, pages 10236–10244, 2021.

[36] Xiaoyu Chen, Yao Mark Mu, Ping Luo, Shengbo Li, and Jianyu Chen. Flow-based recurrent belief state learning for pomdps. In *International Conference on Machine Learning*, pages 3444–3468, 2022.

[37] Youpeng Zhao, Jian Zhao, Xunhan Hu, Wengang Zhou, and Houqiang Li. Full douzero+: Improving doudizhu ai by opponent modeling, coach-guided training and bidding learning. *IEEE Transactions on Games*, pages 1–13, 2023.

[38] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[39] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 1999.

[40] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International Conference on Machine Learning*, pages 3878–3887, 2018.

[41] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *arXiv preprint arXiv:2306.13831*, 2023.

[42] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI conference on artificial intelligence*, pages 2974 – 2982, 2018.

[43] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. pages 6382 – 6393, 2017.

[44] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

[45] David Henry Mguni, Haojun Chen, Taher Jafferjee, Jianhong Wang, Longfei Yue, Xidong Feng, Stephen Marcus Mcaleer, Feifei Tong, Jun Wang, and Yaodong Yang. Mansa: Learning fast and slow in multi-agent systems. In *International Conference on Machine Learning*, pages 24631–24658. PMLR, 2023.

[46] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

[47] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596, 2018.

[48] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[49] Dongming Wu, Xingping Dong, Jianbing Shen, and Steven CH Hoi. Reducing estimation bias via triplet-average deep deterministic policy gradient. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4933–4945, 2020.

[50] Yen-Chen Wu, Bo-Hsiang Tseng, and Milica Gasic. Actor-double-critic: Incorporating model-based critic for task-oriented dialogue systems. In *Conference on Empirical Methods in Natural Language Processing: Findings*, pages 854–863, 2020.

[51] Jiafei Lyu, Xiaoteng Ma, Jiangpeng Yan, and Xiu Li. Efficient continuous control with double actors and regularized critics. In *AAAI Conference on Artificial Intelligence*, pages 7655–7663, 2022.

[52] Shangtong Zhang and Shimon Whiteson. Dac: The double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems*, pages 1–11, 2019.

[53] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

A Discussions

Incorporating state information can offer significant advantages for training, as it often encapsulates high-level features that improve the learning of $V(h,s)$ compared to $V(h)$ [10]. For example, in an extreme scenario where the state information s represents the true value of $V(h)$, the agent can effortlessly infer this true value through $V(h,s)$. Specifically, in the context of the Empty Navigation task, when the state information indicates the agent's relative distance to the goal, the value function learning process is significantly streamlined. Besides, in most POMDPs, rewards are directly defined based on state information rather than on observations. Consequently, learning the value function $V(h,s)$ with the reward as the target is typically simpler than learning $V(h)$. This characteristic makes state information effective in most practical applications, a conclusion that numerous previous studies have empirically validated.

However, state information does not always enhance training. When the state information contains significant noise, introducing excessive noisy features to the input can hinder learning the value function. In unfamiliar environments, whether state information genuinely supports the learning process remains to be determined. The motivation behind this work is that state information may be accessible during training in many scenarios. However, introducing a state could potentially lead to bias or variance. In most cases, state information offers higher-level features that assist agents in decision-making. In these situations, DCRL enhances the process by further reducing variance. Conversely, when state information and partial observations correspond one-to-one, the oracle critic demonstrates no variance; DCRL may become ineffective under these circumstances.

B Additional Related Work

Using complete states to train the world model. Traditional POMDP solutions rely on accurate environment models. Informed Dreamer [12] improves world modeling by incorporating state information. However, the dynamics models of POMDP environments are complex, posing challenges to modeling even with state information. Model inaccuracies will inevitably limit the performance. Our focus in this paper is on model-free methods.

Using complete states in multi-agent reinforcement learning. Centralized Training Decentralized Execution (CTDE) [42–44] is a prevalent paradigm for addressing challenges in multi-agent reinforcement learning (MARL). The optimization objective in MARL is to maximize the rewards under the global states, optimizing each agent accordingly. These methods mostly presuppose that the combination of optimized actions under partial observations leads to optimality under global states. Building on CTDE, MANSA[45] introduces a Global agent to decide when to activate global information usage. However, the policy that maximizes rewards under partial observations does not necessarily align with the policy that maximizes rewards under global states, i.e., $Q(h,a) \neq Q(s,a)$ [19]. Contrary to CTDE's assumptions, the learning objective of DCRL is to derive the optimal policy under partial observations.

Double structure in the actor-critic framework. In our work, we employ two asymmetric critics to address partially observable tasks. Previous studies have explored various double structures. In the context of double critics, Double DQN [46] utilizes a replicated Q-network as the target network to handle overestimation issues, a method later extended to subsequent works such as TD3 [47] and SAC [48]. Considering the opposite properties of single-critic and double-critic methods, TADD [49] incorporates a novel triplet critics mechanism to reduce the estimation bias. Additionally, ADC [50] takes advantage of environmental models by training a model-based critic, which is then combined with the model-free critic. This approach enables learning from the model-free path despite model noise, enhancing the robustness to environmental models. In the context of double actors, DARC [51] introduces a double actor structure to boost the agent's exploration ability by providing two paths for policy optimization, thus reducing the risk of the agent getting stuck locally. Moreover, DAC [52] utilizes a parallel actor-critic architecture to learn better options, indicating that the two critics can be effectively consolidated into one.

C Additional Proofs

Theorem 4. *In the DCRL Interchange Version,* $\mathbb{E}_{s|h}\big[(1-\beta')V^\pi(h) + \beta' V^\pi(h,s)\big] \neq V^\pi(h)$.

Proof. Since β' is a parameter associated with both h and s, it cannot shift to the forefront of the expected symbol. Therefore, the dual value of the DCRL Interchange Version is biased. Also, the policy gradient in the DCRL Interchange Version is biased. □

Theorem 5. *The learning objective* $\left(\delta_\phi(h,a)\right)_+ \log\pi(a|h)$ *in DCRL can be viewed as a form of lower-bound-soft-Q-learning.*

Proof. According to Self-Imitation Learning [40], the Lower bound soft Q-learning updates $Q(h,a)$ as follows:

$$L^{lb} = \mathbb{E}_{h,a,R\sim\mu}[\|\left(R - Q(h,a)\right)_+\|^2], \tag{13}$$

where μ is a behavior policy, (h_t, a_t, R_t) is the trajectory triples from μ, $R_t = r_t + \sum_{k=t+1}^{\infty} \gamma^{k-t}(r_k + \alpha\mathcal{H}_k^{\mu})$ is the entropy-regularized return, α represents the weight of entropy bonus, and $\mathcal{H}_k^{\mu} = -\log\pi(a_k|h_k)$ is the entropy of the policy. According to the form of optimal soft value function and optimal policy in the entropy-regularized RL, it is natural to consider the following forms of a value function V and a policy π:

$$V(h) = \alpha \log \textstyle\sum_a \exp\left(Q(h,a)/\alpha\right), \tag{14}$$

$$\pi(a \mid h) = \exp\left(\left(Q(h,a) - V(h)\right)/\alpha\right), \tag{15}$$

$$Q(h,a) = V(h) + \alpha \log \pi(a \mid h). \tag{16}$$

For convenience, we can define the following:

$$\hat{R} = R - \alpha \log \pi(a \mid h), \tag{17}$$

$$\Delta = R - Q(h,a) = \hat{R} - V(h). \tag{18}$$

We can derive the gradient estimator of lower-bound-soft-Q-learning for the actor-critic architecture as follows:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}_{h,a,R\sim\mu}\left[\frac{1}{2}\left\|\left(R - Q(h,a)\right)_+\right\|^2\right] \\
&= \mathbb{E}\left[-\nabla_\theta Q(h,a)\Delta_+\right] \\
&= \mathbb{E}\left[-\nabla_\theta\left(\alpha \log \pi(a \mid h) + V(h)\right)\Delta_+\right] \\
&= \mathbb{E}\left[-\alpha\nabla_\theta \log \pi(a \mid h)\Delta_+ - \nabla_\theta V(h)\Delta_+\right] \\
&= \mathbb{E}\left[\alpha\nabla_\theta \mathcal{L}_{\text{policy}}^{\text{lb}} - \nabla_\theta V(h)\Delta_+\right] \\
&= \mathbb{E}\left[\alpha\nabla_\theta \mathcal{L}_{\text{policy}}^{\text{lb}} - \nabla_\theta V(h)\left(R - Q(h,a)\right)_+\right] \\
&= \mathbb{E}\left[\alpha\nabla_\theta \mathcal{L}_{\text{policy}}^{\text{lb}} - \nabla_\theta V(h)\left(\hat{R} - V(h)\right)_+\right] \\
&= \mathbb{E}\left[\alpha\nabla_\theta \mathcal{L}_{\text{policy}}^{\text{lb}} + \nabla_\theta \frac{1}{2}\left\|\left(\hat{R} - V(h)\right)_+\right\|^2\right] \\
&= \mathbb{E}\left[\alpha\nabla_\theta \mathcal{L}_{\text{policy}}^{\text{lb}} + \nabla_\theta \mathcal{L}_{\text{value}}^{\text{lb}}\right],
\end{aligned} \tag{19}$$

where $\mathcal{L}_{\text{policy}}^{\text{lb}} = -\log \pi(a|h)\left(\hat{R} - V(h)\right)_+$ and $\mathcal{L}_{\text{value}}^{\text{lb}} = \frac{1}{2}\left\|(\hat{R} - V(h))_+\right\|^2$.

As $n \to \infty$, the learning objective in DCRL is $\left(\sum_{d=0}^{\infty}\gamma^d r_t - V(h)\right)_+ \log\pi(a|h)$. It can be viewed as a form of lower-bound-soft-Q-learning when $\alpha \to 0$. □

In the DCRL Interchange Version, the equation $\mathbb{E}_{s|h}[\hat{R} - V(h,s)] = \hat{R} - V(h)$ holds according to Equation (5). However, the equation $\mathbb{E}_{s|h}[(\hat{R} - V(h,s))_+] = (\hat{R} - V(h))_+$ does not hold due to the presence of the $+$ operation. The inconsistency in the equation arises from the fact that the $+$ operation does not commute with the expectation operator, leading to a difference in the results. Consequently, the learning objective in the DCRL Interchange Version cannot be considered a form of lower-bound soft Q-learning.

D Additional Experimental Results

D.1 Visualization of Critic Values

To illustrate the effectiveness of DCRL in variance reduction, we visualize the values of different critics in the *Empty-Random-6x6* environment with a first-person view size of 3×3, as depicted in Figure 8. In this configuration, the same history is linked to different states. We collect 8 history samples, each associated with over 300 state samples, meaning that each $V^{\pi}(h)$ is connected to multiple $V^{\pi}(h, s)$ and multiple $V^{\pi}_{dual}(h, s)$. The converged critic models are used to predict $V^{\pi}(h)$, $V^{\pi}(h, s)$, and $V^{\pi}_{dual}(h, s)$, respectively. The 8 violin plots in Figure 9 depict the details of each history sample. These plots show density estimation of the values, with the center line indicating the mean value. It is evident that DCRL (shown in orange) demonstrates lower variance compared with the oracle critic (shown in blue).

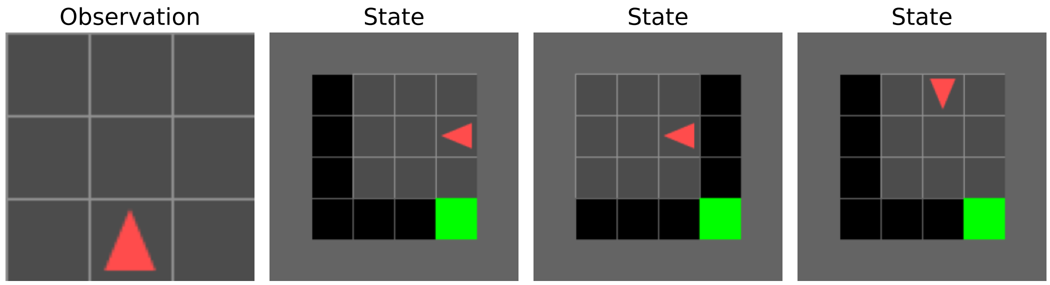

Figure 8: MiniGrid-Empty-Random-6x6. The dimensions of the first-person view are 3×3. The left figure illustrates the observation, while the right displays 3 different states that share the same observation.

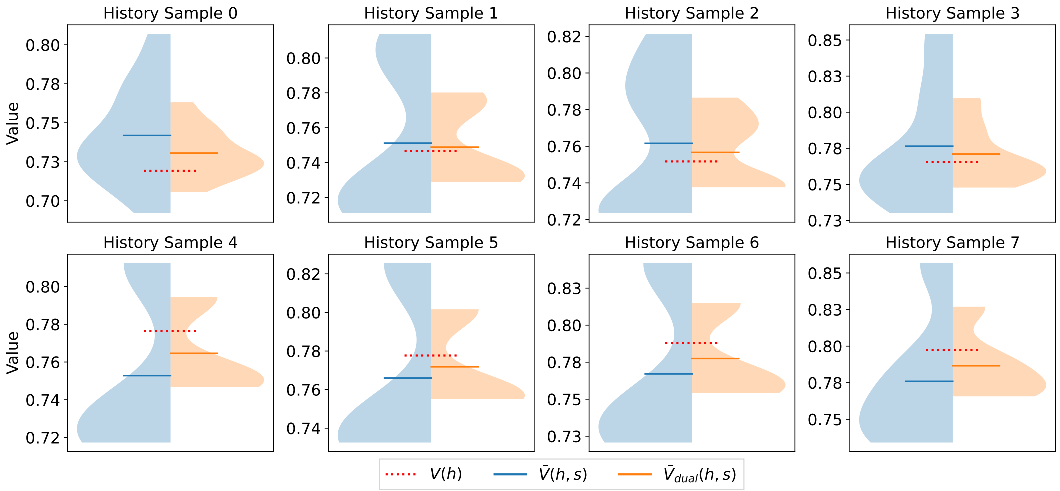

Figure 9: Visualization of 8 history samples. The violin plots show density estimation of the values, with the center line indicating the mean value.

D.2 Learning Curves of All the Compared Methods

Figures 11 and 12 present the learning curves of all compared methods across 27 MiniGrid tasks, utilizing A2C and PPO, respectively.

D.3 Learning Curves of Different β

Figure 13 presents the learning curves for different values of β across 27 MiniGrid tasks.

D.4 Comparison with DreamerV3

Although our research focuses on the model-free setting, we also compare DCRL with DreamerV3 [53], a widely used model-based method in POMDPs. DreamerV3 constructs a model of the environment and enhances its behavior through the imagination of future scenarios. Figure 10 presents the average scores over $1e^5$ steps across 8 tasks in the MiniGrid environment. The results demonstrate that DCRL consistently outperforms DreamerV3 across all tasks. Notably, DCRL requires fewer computational resources and less training time than DreamerV3.

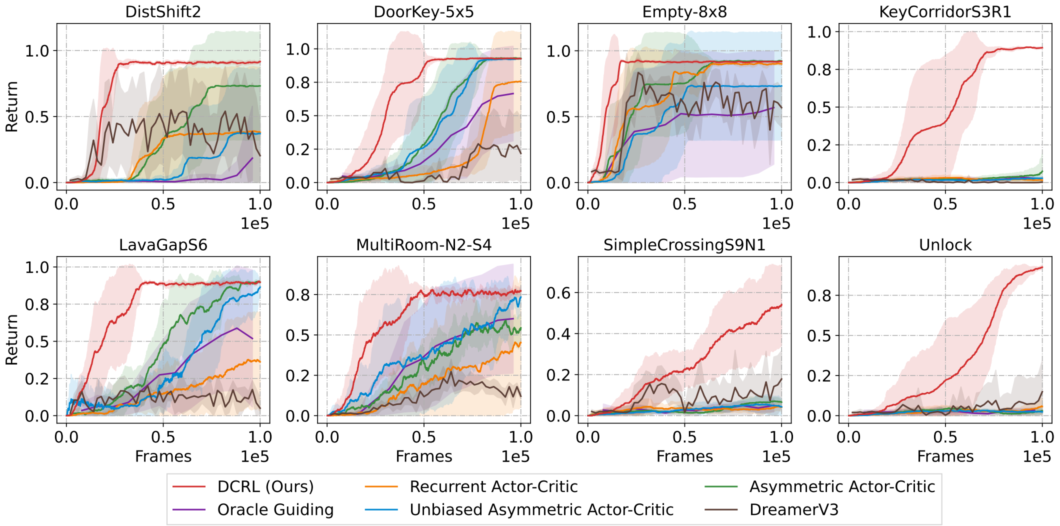

Figure 10: Learning curves on MiniGrid. All curves are trained based on A2C over 5 random seeds.

E Experiment Details

E.1 Environment Descriptions

We describe each task environment used in our experiments. More detailed introductions can be found on the official project homepage of each benchmark. The source code is available in the supplementary.

MiniGrid

MiniGrid [41] comprises Box2D environments featuring various room layouts, interactive objects, and goals. A red triangular agent operates within a discrete action space, with actions such as “Move forward”, “Turn left”, “Turn right”, “Pick up an object”, “Toggle/activate an object”, and others. Interactive objects include keys, doors, balls, and more. The agent must learn specific action sequences to achieve the final goal within its limited forward field of view, with rewards given only upon goal attainment. Each episode samples a different grid layout, and the agent has a first-person partially observable view of the environment.

We selected different experimental settings. For instance, as depicted in Figure 4, the agent in the *LavaCrossingS9N2* environment aims to reach the green goal square in the room’s opposite corner while avoiding deadly lava rivers that terminate the episode in failure. Each lava stream runs either horizontally or vertically across the room, with a single safe crossing point. Fortunately, a path to the goal is guaranteed to exist.

The agent receives a $7 \times 7 \times 3$ dimensional image of the environment as observation. The three channels represent object ID, color ID, and state. The agent is positioned at the observation’s bottom center, and the state is encoded similarly to the observation, consistently displaying all grids without adjusting to the agent’s orientation. The final channel may include 1, 2, 3, or 4 values at the agent’s current coordinates, indicating the agent’s facing direction (east, south, west, and north).

MiniWorld
Similar to MiniGrid, MiniWorld [41] is a minimalistic Box3D interior environment simulator consisting of connected rooms with objects inside. In MiniWorld, the agent has a 3D partial view of the environment and must navigate to the red box. For simplicity, actions are discrete, including “Turn left”, “Turn right”, “Move forward”, “Move back” and others. Turn and move actions rotate or move the agent by a small fixed interval, simulating the behavior of a differential drive robot. Rewards are based on the number of time steps required to successfully complete the task, with a small penalty incurred. If the task is not completed within the maximum steps, the current episode is terminated, and a reward of 0 is assigned.

We selected four experimental settings. For instance, as depicted in Figure 1, the *WallGap* environment features two rooms connected by a gap in a wall. The agent must navigate toward the red box, randomly placed in one of the rooms.

Observations consist of RGB images sized $80 \times 60 \times 3$ captured from the agents perspective, while states are RGB images of size $80 \times 60 \times 3$, representing an aerial view of the environment. In our experiments, RGB images are converted into grayscale images.

E.2 Hyperparameters

We run all experiments on a single server with 64 Intel(R) Xeon(R) Gold 5218 CPU processors @ 2.30GHz and 1 Tesla V100 GPU. The hyperparameters for training each method are summarized in Table 2.

Table 2: Hyperparameters used for training each method.

Hyperparameter	**MiniGrid**	**MiniGrid**	**MiniWorld**
Algorithm	A2C	PPO	PPO
Seeds in experiments	5	5	5
Discount factor γ	0.99	0.99	0.99
λ for GAE	1	0.95	0.95
Rollout steps	5	512	512
Number of workers	16	16	16
Entropy loss coef	0.01	0.01	0.01
Optimizer	RMSprop	Adam	Adam
learning rate	1e-3	3e-4	3e-4
max grad norm	0.5	0.5	0.5
PPO clip range	-	0.2	0.2
PPO training epochs	-	4	4
PPO mini-batch size	-	512	512
dual update per iteration	16	4	4
dual training epochs	4	8	8
dual batch size	640	2048	2048
β	0.5	0.5	Best chosen from $\{0.1, 0.5\}$

E.3 Network Structures

In this section, we describe the structures of the policy network, the standard critic network, and the oracle critic network. The general architecture for all environments is the same, except for the representation of input images. We utilize the CNN model for representation, and the structures are contingent upon the dimensions (n, m) of the input images. Detailed information for each environment is provided below.

Representation Network
MiniGrid
(1) $n \leq 3$ or $m \leq 3$
Conv2d(in=3, out=32, kernel=2, stride=1, pad=1),
Relu,
Conv2d(in=32, out=64, kernel=2, stride=1, pad=0),

Relu,
Conv2d(in=64, out=64, kernel=2, stride=1, pad=0),
Relu,
Flatten.
(2) $n \leq 15$ or $m \leq 15$
Conv2d(in=3, out=32, kernel=2, stride=1, pad=0),
Relu,
Conv2d(in=32, out=64, kernel=2, stride=1, pad=0),
Relu,
Conv2d(in=64, out=64, kernel=2, stride=1, pad=0),
Relu,
Flatten.
(3) $n > 15$ or $m > 15$
Conv2d(in=3, out=32, kernel=2, stride=1, pad=0),
Relu,
MaxPool2d(kernel=2),
Conv2d(in=32, out=64, kernel=2, stride=1, pad=0),
Relu,
Conv2d(in=64, out=64, kernel=2, stride=1, pad=0),
Relu,
Flatten.

MiniWorld
Conv2d(in=1, out=32, kernel=8, stride=4, pad=0),
Relu,
Conv2d(in=32, out=64, kernel=4, stride=2, pad=0),
Relu,
Conv2d(in=64, out=64, kernel=3, stride=1, pad=0),
Relu,
Flatten.

Policy Network
Representation Network,
FC(in=input_dim, out=64),
Relu,
LSTM(in=64, out=64).
FC(in=128, out=128),
Relu,
FC(in=128, out=number of actions).

Standrad Critic Network
The structure is the same as that of the policy network, except for the dimension of the final MLP output head, which is set to 1.

Oracle Critic Network
The structure is the same as that of the policy network, except for the dimension of the final MLP output head, which is set to 1.

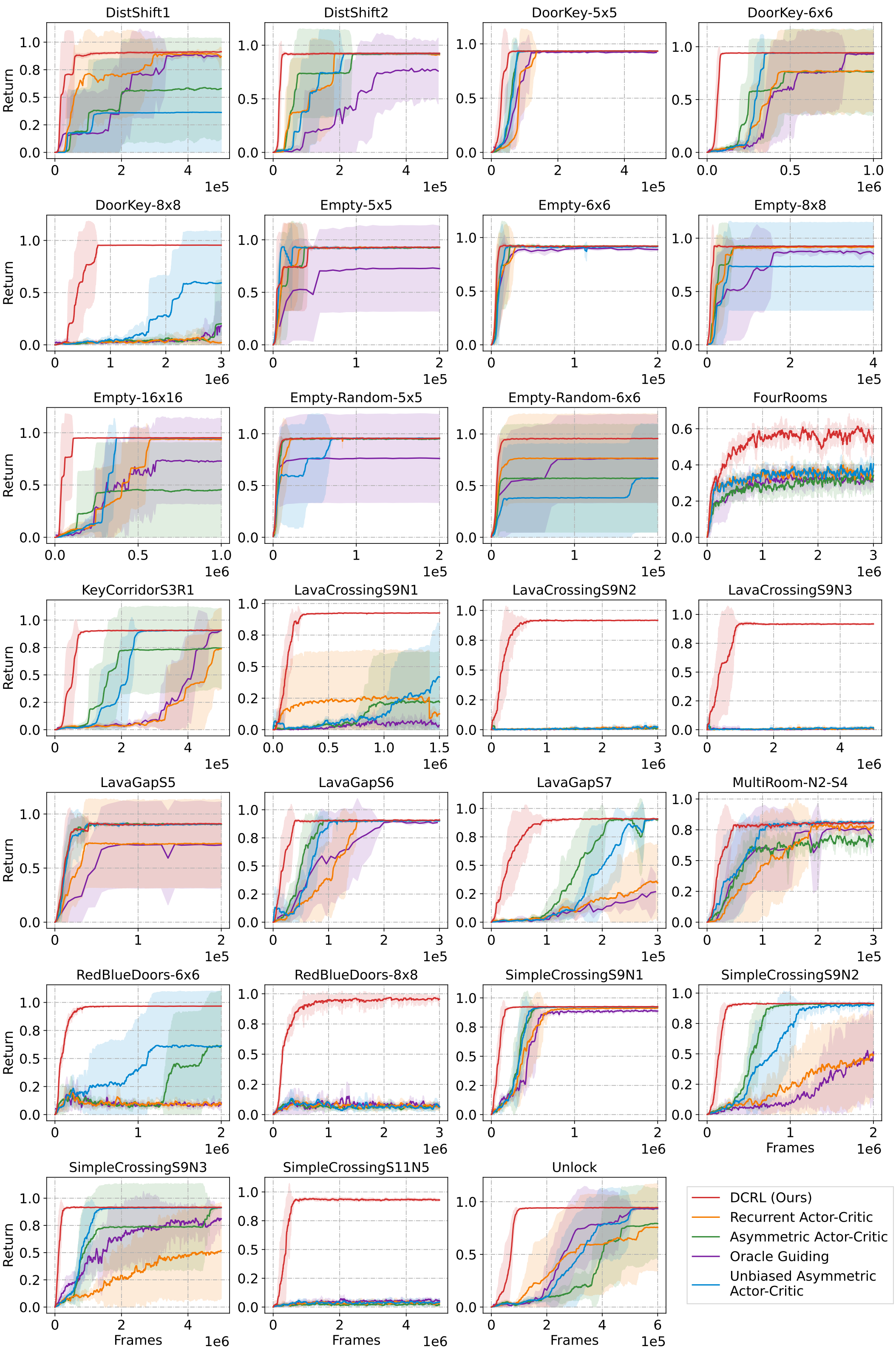

Figure 11: Learning curves on MiniGrid. All curves are trained based on A2C over 5 random seeds.

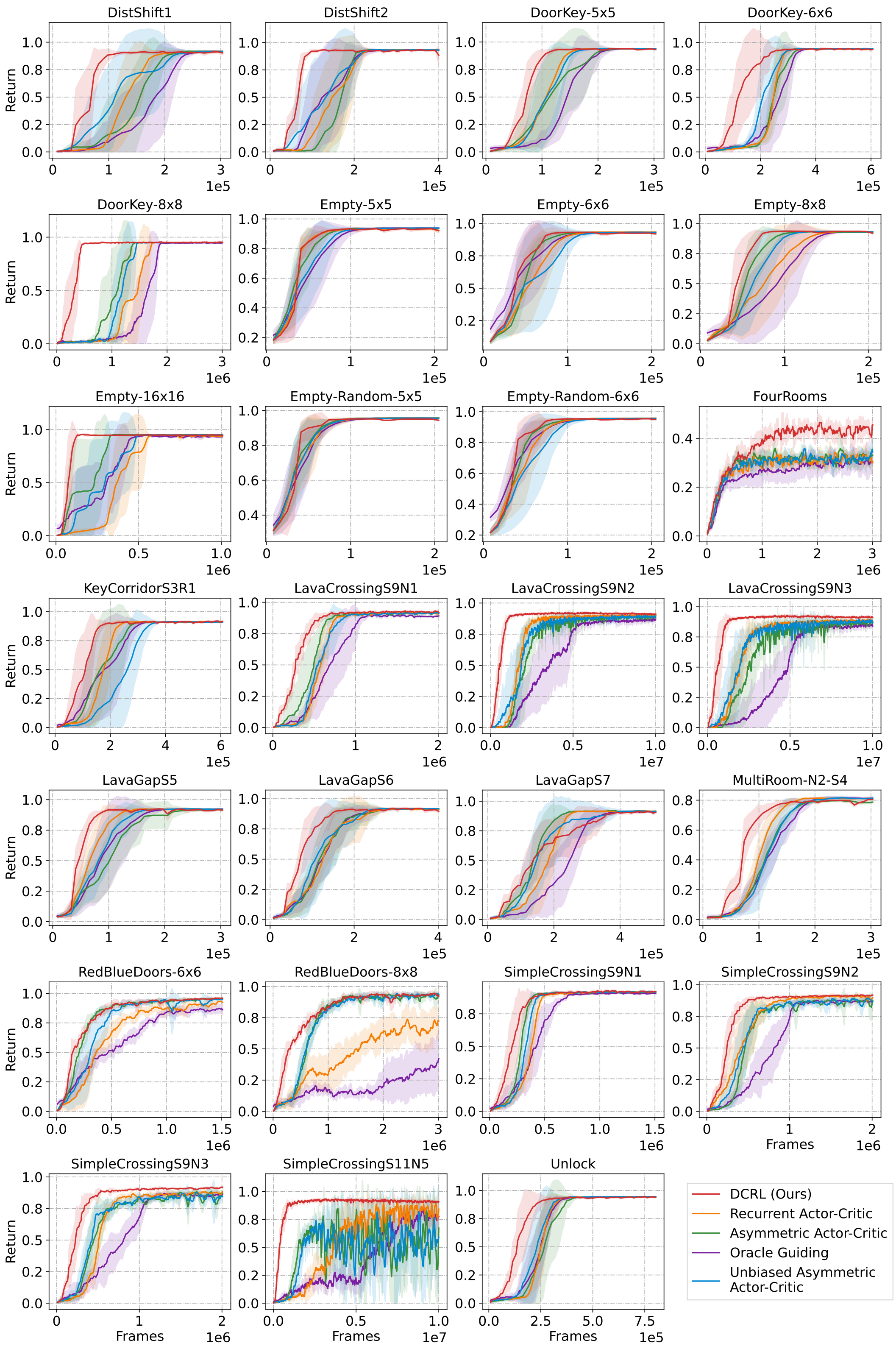

Figure 12: Learning curves on MiniGrid. All curves are trained based on PPO over 5 random seeds.

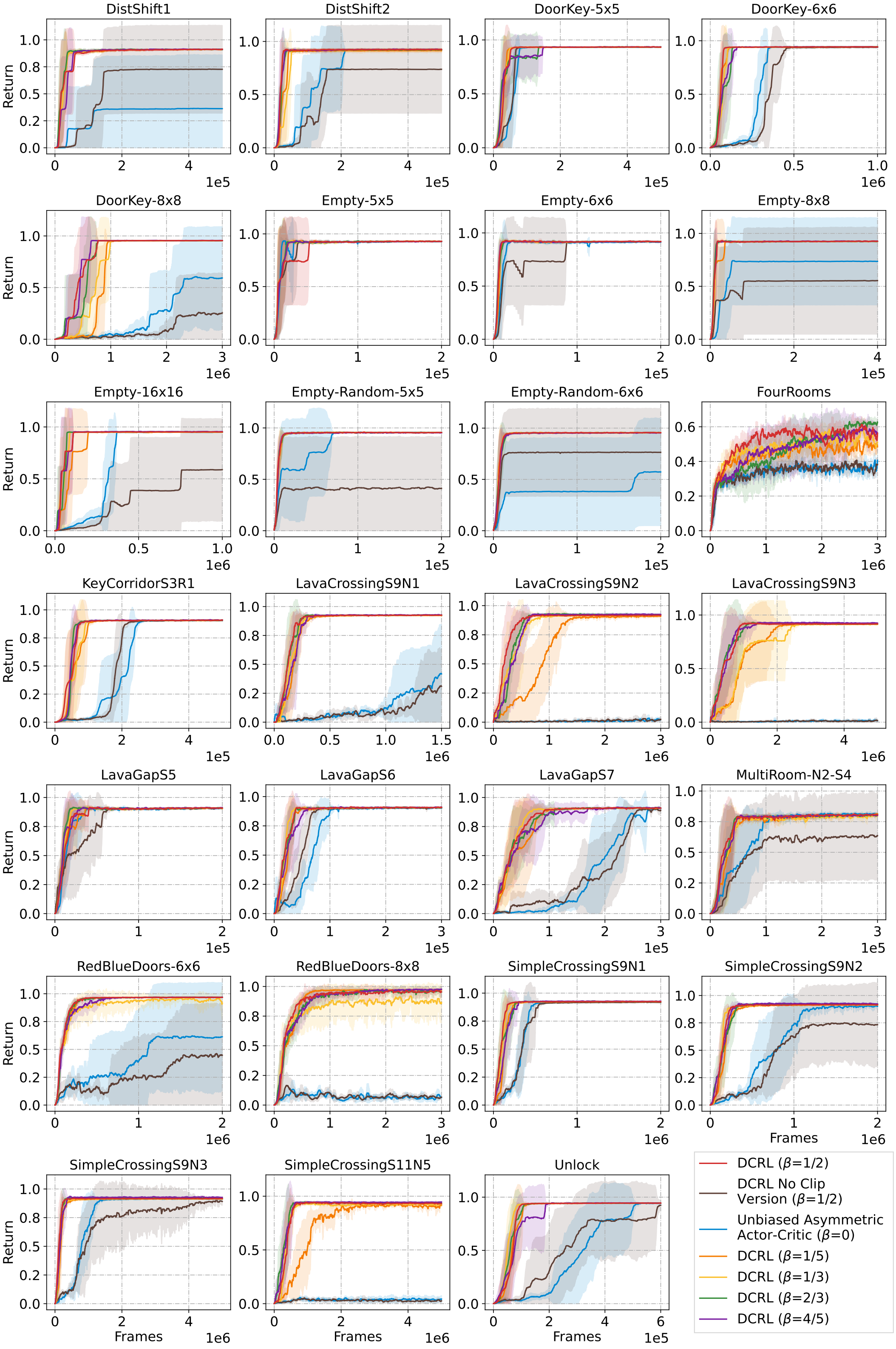

Figure 13: Ablation studies on different values of β and DCRL No Clip Version. All curves are trained based on A2C over 5 random seeds.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide a detailed introduction to the proposed method and provide relevant theoretical proof in Section 4. In Section 5, we present experimental results to demonstrate the effectiveness of the proposed method.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide a complete proof in Section 4 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and theorems that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer:[Yes]

Justification: We provide experimental details in Appendix E, and the source code is available in the supplementary.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code is available in the supplementary.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer:[Yes]

Justification: We provide experimental details in Appendix E, and the source code is available in the supplementary.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer:[Yes]

Justification: We present the mean and standard deviation of results obtained from five different seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:[Yes]

   Justification: We provide experimental details in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the original paper that produced the code package.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the source code and instructions in the supplementary.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.