BELIEF-BASED RISK REDUCTION FOR ROBUST REINFORCEMENT LEARNING UNDER ADVERSARIAL OBSERVATION PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning (DRL) policies are highly vulnerable to adversarial noise in observations, posing significant risks in safety-critical scenarios. The primary challenge of adversarial perturbations lies in their ability to distort the information observed by an agent. Recently, state purification defenses have emerged as a promising solution. However, existing methods suffer from a distribution mismatch between clean states and adversarial observations, as well as uncertainty in state restoration and risks in decision-making. To address these challenges, we propose Belief-Based Risk Reduction (BBRR), a risk-aware state purification framework. BBRR uses a diffusion bridge to mitigate the distribution gap between adversarial observations and clean samples, while employing belief state approximation and Conditional Value-at-Risk (CVaR) to reduce restoration uncertainty and decision risks, respectively. Theoretical analysis and comprehensive evaluations across diverse environments demonstrate that BBRR consistently achieves state-of-the-art performance against a range of strong attacks, validating its robustness and practical applicability. Our code is available at `https://anonymous.4open.science/r/Robust_BBRR`.

1 INTRODUCTION

Deep reinforcement learning (DRL) has achieved increasing success in robotics Levine et al. (2016); Ibarz et al. (2021), creative generation OpenAI (2023); Wang et al. (2025), and locomotion control Rudin et al. (2022). However, the neural networks underlying DRL are often highly sensitive to input perturbations. Minor sensor noise, external disturbances, or carefully crafted adversarial perturbations can substantially alter policy outputs, reduce cumulative returns, and even induce arbitrarily poor or unsafe behavior Szegedy et al. (2013); Chakraborty et al. (2018); Huang et al. (2017); Zhang et al. (2021). This vulnerability poses significant performance and safety risks, limiting the real-world deployment of DRL in safety-critical applications Akhtar & Mian (2018); Pitropakis et al. (2019); Machado et al. (2021).

From a security perspective, DRL systems are exposed to a broad range of threats, including state or reward perturbations Zhang et al. (2020); Chen et al. (2025), action manipulation Kobayashi (2022), and model inference or poisoning attacks Ilahi et al. (2021). Among these threats, adversarial observation perturbations are particularly concerning because they can severely degrade policy performance while remaining difficult to detect Qiaoben et al. (2021); Sun et al. (2022); Ding et al. (2023). By subtly modifying the observations received by a trained agent, an attacker can mislead its decision-making process and substantially reduce its cumulative return. Therefore, it is essential to develop effective defense methods that enhance the robustness of DRL policies to observation perturbations.

Existing defenses against observation perturbations mainly follow several directions. Regularization-based methods Zhang et al. (2020); Shen et al. (2020); Oikarinen et al. (2021); Liang et al. (2022) encourage policy networks to produce consistent outputs for clean and perturbed observations. However, their effectiveness is sensitive to the regularization strength: overly strong regularization may oversmooth the policy and degrade its clean performance, whereas insufficient regularization may provide inadequate protection against strong attacks Kobayashi

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2022). Learned-adversary methods Gleave et al. (2020); Zhang et al. (2021); Sun et al. (2022) instead alternately optimize the agent and attacker policies, with the attacker trained to approximate worst-case attacks against the current policy. Nevertheless, training an RL-based attacker requires additional environment interactions and may be even more sample-intensive than learning the original policy Liang et al. (2022). Moreover, existing adversarial training methods may still rely on non-robust features, leaving the resulting policies vulnerable to black-box attacks Korkmaz (2022; 2023). Purification-based methods Yang & Xu (2024); Sun et al. (2024); Sun & Zheng (2024); Liu et al. (2025) mitigate the effects of observation perturbations at the input level, potentially avoiding the need to directly regularize the policy network or train an additional RL-based attacker. At the same time, existing methods still face a distribution mismatch between clean and adversarial observations. Besides, they often overlook uncertainty in the observation restoration process and the resulting decision-making risks.

In this work, we propose Belief-Based Risk Reduction (BBRR), a risk-aware state purification framework for robust reinforcement learning under adversarial observation perturbations. BBRR first employs a diffusion bridge to connect adversarial observations with the clean-state distribution, thereby reducing the distribution mismatch that limits conventional purification methods. It then maintains a belief state approximation over plausible clean states instead of relying on a single point estimate, allowing restoration uncertainty to be explicitly represented. Finally, BBRR incorporates Conditional Value-at-Risk (CVaR) into policy optimization to control the lower tail of the return distribution and avoid decisions that are successful only under optimistic restorations. We provide theoretical analyses of the purification and risk-sensitive objectives, and evaluate BBRR across diverse environments against adaptive and black-box attacks. The results show that BBRR substantially improves robustness while preserving clean task performance, establishing a practical and principled defense for adversarially perturbed observations.

2 Related Work

2.1 Adversarial Observation Attacks on DRL Agents

Early observation attacks on DRL agents adapted gradient-based methods, particularly FGSM, from image classification to perturb agents' state observations Huang et al. (2017); Kos & Song (2017). Subsequent studies modeled the interaction between the agent and adversary as a zero-sum Markov game Pinto et al. (2017); Gleave et al. (2020), or strategically selected attack timings and generated perturbations computed with the Carlini &Wagner adversarial formulation Lin et al. (2017); Sun et al. (2020). Other work revealed adversarial directions shared across states, environments, and learning algorithms, suggesting that DRL policies rely on persistent non-robust features that may remain even after adversarial training Korkmaz (2021; 2022).

2.2 Defenses Against Adversarial Observation Perturbations

(i) Regularization-based methods improve the intrinsic robustness of policy networks by encouraging consistent policy outputs. Zhang et al. (2020) first formalized the problem as a state-adversarial MDP (SA-MDP). Shen et al. (2020) employs an adversarial smoothness regularizer to enforce consistent policy outputs under small observation perturbations. Subsequent work further improved robustness by reducing the overlap between the clean and perturbed Q-values Oikarinen et al. (2021), or by combining worst-case value estimates with conventional temporal-difference (TD) targets Liang et al. (2022). (ii) Learned-adversary methods model the attacker as an RL agent and alternately optimize the policies of the agent and the attacker. Early studies Behzadan & Munir (2017); Kos & Song (2017); Mandlekar et al. (2017); Pattanaik et al. (2018) trained RL agents against gradient-based state-observation attacks. Moving beyond such fixed attacks, Zhang et al. (2021) proposed Alternating Training with Learned Adversaries (ATLA), which alternately optimizes an RL agent and an RL attacker to improve policy robustness. Sun et al. (2022) further developed PA-ATLA using the stronger PA-AD attacker. (iii) Purification-based methods mitigate the effects of observation perturbations at the input level. Diffusion-based methods such as Liu et al. (2025) and Sun & Zheng (2024) exploit the convergence of adversarial and clean samples after sufficient diffusion steps, using threshold or extended reverse steps to recover clean-like observations. Yang & Xu (2024) instead leverages historical information for robust reconstruction, while Sun et al. (2024) employs a dedicated CNN purifier for image observations. Despite their effectiveness, these

methods still face a distribution mismatch between clean and adversarial observations. Besides, they often overlook uncertainty in the observation restoration process and the resulting decision-making risks.

3 PRELIMINARIES

Reinforcement Learning: A reinforcement learning problem is commonly formulated as a Markov decision process (MDP), denoted as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively. The transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ describes the environment dynamics, while $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the immediate reward. The discount factor $\gamma \in [0, 1)$ controls the relative importance of future rewards. An agent interacts with the environment according to a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, which determines the action distribution at each state. The objective of a policy π is to maximize the expected discounted cumulative reward, given by $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$. For discrete action spaces, Deep Q Network (DQN) learns an action-value function $Q^\pi(s, a) = \mathbb{E}_{P,\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a \right]$. by minimizing the Bellman error and derive the policy $\pi(s) = \arg\max_a Q(s, a)$, where the optimal value satisfies $Q^*(s, a) = r + \gamma \mathbb{E}_{s'|s,a} \left[\max_{a'} Q^*(s', a') \right]$. For continuous action spaces, Proximal Policy Optimization (PPO) stabilizes on-policy updates by maximizing the clipped surrogate objective $\mathbb{E}_t \Big[\min \Big(r_t(\theta) \hat{A}_t, \mathrm{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \, \hat{A}_t \Big) \Big]$, where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and $\hat{A}_t$ denotes the estimated advantage.

Diffusion Models: Given $x_0 \sim q(x_0)$, DDPM Ho et al. (2020) defines a discrete-time Markov chain $\{x_0, \ldots, x_T\}$ that gradually adds Gaussian noise according to $q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{1 - \beta_t}\, x_{t-1}, \beta_t I)$. Letting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, the noisy sample can be obtained directly as $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, so that x_T approaches an isotropic Gaussian distribution. DDPM learns the reverse process using a noise predictor $\epsilon_\theta(x_t, t)$, from which the clean sample is estimated as $\hat{x}_0 = (x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)) / \sqrt{\bar{\alpha}_t}$. The model is trained by minimizing $\mathcal{L}_d = \mathbb{E}_{x_0,t,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2]$. Because DDPM sampling typically requires hundreds of reverse steps, DDIM Song et al. (2021a) introduces a non-Markovian formulation that enables substantially faster sampling with fewer steps, while continuous-time diffusion models further describe the process through stochastic differential equations Song et al. (2021b).

Diffusion Bridge Models: Diffusion bridges further generalize diffusion models by constructing a diffusion process between two prescribed endpoint distributions, including data-to-prior, Gaussian, and optimal transport bridges De Bortoli et al. (2021); Liu et al. (2023); Zhou et al. (2024); Zheng et al. (2025). Recent studies reveal that diffusion bridges can also establish a direct reverse process from the diffused adversarial data distribution to the clean data distribution Li et al. (2025), thereby progressively removing the adversarial perturbations from samples.

4 BELIEF-BASED RISK REDUCTION RL

4.1 MECHANISM 1: DIFFUSION BRIDGE PURIFICATION

Let s_t denote the clean state at environment step t, and let $\tilde{s}_t = s_t + \delta_t$ be its adversarially perturbed observation, where $\|\delta_t\|_\infty \le \epsilon_{\text{atk}}$. Although forward diffusion can reduce the structured perturbation, it does not necessarily remove it completely. In particular, at diffusion step ℓ,

$$x^a_{\ell,t} = \sqrt{\bar{\alpha}_\ell}\, \tilde{s}_t + \sqrt{1 - \bar{\alpha}_\ell}\, \varepsilon = \sqrt{\bar{\alpha}_\ell}\, s_t + \sqrt{\bar{\alpha}_\ell}\, \delta_t + \sqrt{1 - \bar{\alpha}_\ell}\, \varepsilon, \tag{1}$$

where $\varepsilon \sim \mathcal{N}(0, I)$, $\alpha_\ell = 1 - \beta_\ell$, and $\bar{\alpha}_\ell = \prod_{u=1}^{\ell} \alpha_u$. Thus, directly applying the reverse process of a clean-data diffusion model may preserve part of the adversarial component.

Proposition 1 (Diffusion bridge construction)**.** *Given the adversarial observation $\tilde{s}_t = s_t + \delta_t$, we construct the following bridge trajectory to connect its diffused counterpart to the corresponding clean state:*

$$x^d_{\ell,t} = x^a_{\ell,t} - k_\ell \delta_t, \qquad k_\ell = \frac{\bar{\alpha}_\ell - \bar{\alpha}_T}{\sqrt{\bar{\alpha}_\ell}(1 - \bar{\alpha}_T)}. \tag{2}$$

The coefficient satisfies $k_0 = 1$ and $k_T = 0$, yielding the boundary conditions $x^d_{T,t} = x^a_{T,t},\ x^d_{0,t} = \tilde{s}_t - \delta_t = s_t$. The reverse process therefore starts from the diffused adversarial distribution and has the clean-state distribution as its prescribed endpoint. This construction specifies the ideal bridge; reconstruction by a learned denoiser additionally depends on its prediction error. The proof is given in Appendix A. Substituting Equation equation 1 into Equation equation 2 gives

$$x^d_{\ell,t} = \sqrt{\bar{\alpha}_\ell}\, s_t + \sqrt{1-\bar{\alpha}_\ell}\, \varepsilon + \frac{\bar{\alpha}_T(1-\bar{\alpha}_\ell)}{\sqrt{\bar{\alpha}_\ell}(1-\bar{\alpha}_T)}\delta_t. \tag{3}$$

Accordingly, the denoising network is trained to predict both the Gaussian corruption and the scaled adversarial component:

$$\mathcal{L}_{\text{bridge}}(\psi) = \mathbb{E}_{s_t,\delta_t,\varepsilon,\ell}\left[\left\|\epsilon_\psi(x^d_{\ell,t},\ell) - (\varepsilon + \lambda_\ell \delta_t)\right\|_2^2\right], \qquad \lambda_\ell = \frac{\bar{\alpha}_T\sqrt{1-\bar{\alpha}_\ell}}{(1-\bar{\alpha}_T)\sqrt{\bar{\alpha}_\ell}}. \tag{4}$$

This objective explicitly teaches the reverse process to suppress the adversarial perturbation while recovering the clean state. At inference time, neither s_t nor δ_t is available. The observed state is first mapped to diffusion depth T as $x^a_{T,t} = \sqrt{\bar{\alpha}_T}\,\tilde{s}_t + \sqrt{1-\bar{\alpha}_T}\,\varepsilon, \varepsilon \sim \mathcal{N}(0, I)$, and the learned reverse bridge is then applied using a DDIM sampler to obtain the purified state $\hat{s}_t = \mathcal{P}_\psi\left(\tilde{s}_t; T, \varepsilon\right)$, where $\mathcal{P}_\psi$ denotes the complete forward and reverse diffusion process. The purifier is trained separately and remains frozen during the subsequent Q network fine-tuning.

Theorem 1 (One-step purification error bound)**.** *Let $(s_t, \delta_t) \sim \mathcal{D}$, and let diffusion time ℓ be sampled independently with probabilities p_ℓ on $\{1, \ldots, T\}$, where $p_T > 0$. Suppose the expected training loss in Eq. equation 4 is at most η. For a deterministic one-step DDIM sampler initialized at $x^a_{T,t}$, the expected squared purification error under the same distribution $\mathcal{D}$ satisfies*

$$\mathbb{E}_{(s_t,\delta_t)\sim\mathcal{D},\varepsilon}\left[\|\hat{s}_t - s_t\|_2^2\right] \le C_T\,\eta, \tag{5}$$

where $C_T = (1-\bar{\alpha}_T)/(\bar{\alpha}_T p_T)$. In particular, uniform timestep sampling gives $C_T = T(1-\bar{\alpha}_T)/\bar{\alpha}_T$. The proof is given in Appendix A.2.

Theorem 1 implies that a sufficiently small bridge training loss enables accurate clean-state recovery with a one-step DDIM sampler, becoming perfect in expectation as the loss approaches zero.

For continuous-control tasks, policies are expected to produce semantically consistent actions across adjacent states. We therefore employ a conditional diffusion bridge conditioned on the temporal state-action context $c_t = (H_{t-1}, a_{t-1})$, where $H_{t-1} = [\hat{s}_{t-C}, \ldots, \hat{s}_{t-1}]$ contains the previously restored states, and a_{t-1} denotes the action executed at the previous timestep. This context provides a dynamical prior that helps suppress temporally inconsistent adversarial components and yields a more stable state representation. Accordingly, the denoising network in Eq. equation 4 is replaced by $\epsilon_\psi(x^d_{\ell,t},\ell) \longrightarrow \epsilon_\psi(x^d_{\ell,t},\ell,H_{t-1},a_{t-1})$. The continuous action is directly encoded without discretization and fused with the temporal representation extracted from the state history. Hence, the conditional bridge objective becomes

$$\mathcal{L}^{\text{cont}}_{\text{bridge}}(\psi) = \mathbb{E}\left[\left\|\epsilon_\psi(x^d_{\ell,t},\ell,H_{t-1},a_{t-1}) - (\varepsilon + \lambda_\ell\delta_t)\right\|_2^2\right], \tag{6}$$

where $x^d_{\ell,t}$ and λ_ℓ follow the bridge construction defined above. At inference time, the current observation is purified as $\hat{s}_t = \mathcal{P}_\psi\left(\tilde{s}_t \mid \hat{H}_{t-1}, a_{t-1}\right)$. When multiple belief states are generated, their coordinate-wise median is used as the restored state stored in $\hat{H}_t$ for conditioning the next purification step. The additional condition constrains the recovered state to be consistent with the transition induced by the preceding continuous action.

4.2 Mechanism 2: belief state approximation

We use a diffusion bridge model to purify the perturbed observation before performing action evaluation, thereby obtaining a reliable input state. However, direct purification implicitly assumes that a given perturbed observation $\tilde{s}_t$ can be mapped to a unique recovered state corresponding to the true state s_t. In practice, this assumption may not hold because diffusion-based purification introduces randomness through noise sampling and can produce multiple plausible recovered states from the same perturbed observation Fu et al. (2026).

Rather than assuming that a perturbed observation has a unique purified state, we consider two sources of uncertainty in the purification process. First, diffusion-based purification is inherently stochastic, as different noise sampling can lead to different reverse diffusion trajectories. Second, a perturbed observation may be compatible with multiple reliable purified state. Thus, committing to a single purification result may discard relevant state uncertainty and lead to unstable decisions.

Let $\tilde{s}_t$ denote the perturbed observation at time t. Under an ℓ_∞-bounded perturbation with budget ϵ_{attack}, the corresponding clean state is assumed to lie in the perturbation set $\mathcal{B}_{\epsilon_{\text{attack}}}(\tilde{s}_t) = \{s \in \mathcal{S} : \|s - \tilde{s}_t\|_\infty \leq \epsilon_{\text{attack}}\}$. Hence, from the defender's perspective, $\tilde{s}_t$ does not uniquely determine the underlying state. The defender only has partial information about the true state, which naturally induces state uncertainty in the decision-making process.

This observation uncertainty can be interpreted using the belief state view of partially observable Markov decision processes (POMDP) Belaire et al. (2024). In a POMDP, a belief state describes the defender's uncertainty over the underlying states given its available observations and past actions. An exact Bayesian belief generally requires knowledge of the transition model and the observation model induced by the adversary. Such a belief is difficult to compute explicitly for high-dimensional observations. We therefore use the diffusion bridge as a practical belief estimator to approximate the possible clean states associated with the perturbed observation.

Specifically, we apply the diffusion bridge to the same perturbed observation multiple times with different stochastic noise realizations. The resulting states form the belief set $M_t = \left\{m_t^1, m_t^2, \ldots, m_t^\kappa\right\}$, $m_t^j = N_{\text{DB}}(\tilde{s}_t; \xi_t^j)$, $j = 1, \ldots, \kappa$, where ξ_t^j denotes the randomness used in the j-th diffusion sampling process and $\kappa = |M_t|$ denotes the number of belief states. In practice, the perturbed observation is replicated κ times to form a batch, and the resulting batch is processed by the diffusion bridge. The belief set M_t provides a finite, sample-based representation of the defender's uncertainty about the underlying true state. Each $m_t^j \in M_t$ is a plausible clean state estimate, and M_t retains the uncertainty across purification for subsequent robust action selection.

4.3 Mechanism 3: Conditional Value-at-Risk (CVaR) for Risk Reduction

Given the purified belief set constructed in the previous subsection, we seek to derive a policy that accounts for decision uncertainty about the underlying true state. Such a policy must map multiple plausible belief states to a single executable action. To account for this uncertainty, we use the Conditional Value-at-Risk (CVaR) as a risk-sensitive value aggregation operator. CVaR is a coherent risk measure that characterizes the tail behavior of a random variable (Artzner et al., 1997; Rockafellar & Uryasev, 2000). Since lower Q-values correspond to less favorable outcomes in our setting, we consider the lower-tail CVaR. For a random variable X with probability measure $\mathbb{P}$ and a tail probability $\alpha \in (0, 1]$, the lower-tail CVaR is defined as

$$\text{CVaR}_\alpha^{\mathbb{P}}(X) = \sup_{\eta \in \mathbb{R}} \left\{ \eta - \frac{1}{\alpha} \mathbb{E}_{\mathbb{P}} \left[(\eta - X)_+ \right] \right\}, \tag{7}$$

where $(x)_+ = \max(x, 0)$. It can be interpreted as the expected value of the worst α fraction of the outcomes. A smaller α places greater emphasis on unfavorable outcomes, whereas $\alpha = 1$ reduces to the expectation. For value-based RL with discrete action spaces, such as DQN and its Q-learning variants, we use the following CVaR aggregation.

Definition 1. *At time t, let $\mathcal{M}_t = \{m_t^1, m_t^2, \ldots, m_t^K\}$ be the belief set of purified belief states. For each action a, define $q_t^j(a) = Q_\theta(m_t^j, a)$, and let $q_{t,(1)}(a) \leq \cdots \leq q_{t,(K)}(a)$ be these values in ascending order. For equally weighted belief states and an integer k satisfying $1 \leq k \leq K$, with $\alpha = k/K$, the lower-tail CVaR value and the induced action are defined by*

$$\bar{Q}_\theta^{\text{CVaR}}(\mathcal{M}_t, a) = \frac{1}{k} \sum_{i=1}^{k} q_{t,(i)}(a), \;\; 1 \leq k \leq K. \qquad a_t^{\text{CVaR}} = \arg\max_{a \in \mathcal{A}} \bar{Q}_\theta^{\text{CVaR}}(\mathcal{M}_t, a). \tag{8}$$

Here, k controls the fraction of unfavorable belief states considered in the aggregation. A smaller k places greater emphasis on low-value belief states, while a larger k incorporates more belief states into the estimate.

We analyze an idealized value iteration with exact updates, following the Bellman-operator approach used for pessimistic Q-learning. For this analysis, fix a stationary attack policy ω and a deterministic

purification map $\mathcal{M}$, and write $\pi_n(\tilde{s}) = \arg\max_a \bar{Q}_n^{\mathrm{CVaR}}(\mathcal{M}(\tilde{s}), a),\ \tilde{\pi}_n = \pi_n \circ \omega$. Here $\bar{Q}_n^{\mathrm{CVaR}}$ uses the preceding definition with Q_n in place of Q_θ, equal belief state weights, and $\alpha = k/K$. For a fixed pair of policies, the Bellman operator is

$$T^{\pi\circ\omega}Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a)Q(s', \pi(\omega(s'))). \tag{9}$$

Starting from $Q_0 = 0$, let $Q_{n+1} = T^{\tilde{\pi}_n}Q_n$. Let Q^* denote the optimal Q-function of the original MDP without attacks, and let $Q^{\tilde{\pi}_n}$ denote the ordinary discounted Q function of the joint policy.

Assumption 1. *The state and action spaces are finite, $0 \le \gamma < 1$, and $|R(s,a)| \le R_{\max}$. For all states and actions,*

$$|R(s_1,a) - R(s_2,a)| \le l_r\|s_1 - s_2\|, \qquad |P(s'|s_1,a) - P(s'|s_2,a)| \le l_p\|s_1 - s_2\|.$$

Every belief state $m \in \mathcal{M}(\omega(s))$ satisfies $\|m - s\| \le d$, for a constant $d \ge 0$.

Theorem 2 (Performance bound for CVaR-based value iteration)**.** *Under Assumption 1, the above value iteration satisfies*

$$\limsup_{n\to\infty} \|Q^* - Q^{\tilde{\pi}_n}\|_\infty \le \frac{(1+\gamma)\Delta}{(1-\gamma)^2}, \qquad \Delta = 2\gamma d\left(l_r + l_p|\mathcal{S}|\frac{R_{\max}}{1-\gamma}\right). \tag{10}$$

The complete proof is in Appendix B.1. The bound holds for every $1 \le k \le K$. During training, we use an ε-greedy strategy:

$$a_t = \begin{cases} \text{a uniformly sampled discrete action,} & \text{with probability } \varepsilon, \\ \arg\max_a \bar{Q}_\theta^{\mathrm{CVaR}}(\mathcal{M}_t, a), & \text{with probability } 1-\varepsilon. \end{cases} \tag{11}$$

Importantly, CVaR is applied to the Q-values of each action across the belief states. Thus, the complete belief set is retained and the uncertainty associated with the underlying state is incorporated at the action-value level.

For policy-based RL with continuous action spaces, such as PPO, we combine median policy smoothing with CVaR optimization. Median smoothing limits the influence of outlying belief states on action selection, while CVaR optimization focuses on trajectories with low returns. Let $\mathcal{M}_t = \{m_t^1, m_t^2, \ldots, m_t^K\}$ denote the belief set at time t, where each m_t^j is a purified belief state generated from the perturbed observation. We parameterize the base policy as a diagonal Gaussian: $\pi_\theta(a \mid m) = \mathcal{N}\big(a; \mu_\theta(m), \mathrm{diag}\big(\sigma_\theta^2(m)\big)\big)$. Instead of relying on a single purified state estimate, we aggregate the policy parameters over the belief distribution using coordinate-wise medians: $\widetilde{\mu}_{\theta,i}(\mathcal{M}_t) = Q_{median}^{m\sim b_t}[\mu_{\theta,i}(m)]\,, \widetilde{\sigma}_{\theta,i}(\mathcal{M}_t) = Q_{median}^{m\sim b_t}[\sigma_{\theta,i}(m)]$, where Q_{median} denotes the median and i indexes the action dimension. The smoothed behavior policy is

$$\widetilde{\pi}_\theta(a_t \mid \mathcal{M}_t) = \mathcal{N}\big(a_t; \widetilde{\mu}_\theta(\mathcal{M}_t), \mathrm{diag}\big(\widetilde{\sigma}_\theta^2(\mathcal{M}_t)\big)\big)\,. \tag{12}$$

Actions are sampled from this distribution. By aggregating policy parameters across plausible purified states, median smoothing limits the influence of extreme policy outputs while retaining a stochastic policy representation.

Although this aggregation addresses uncertainty in local action generation, it does not explicitly account for poor long-term outcomes. We therefore introduce a lower-tail CVaR objective for the trajectory return $G(\tau) = \sum_{t=0}^{T-1}\gamma^t r_t$. Let G_θ denote the total return induced by the smoothed policy. For a tail probability $\alpha \in (0,1)$, lower-tail CVaR measures the average return over the worst α fraction of outcomes:

$$\mathrm{CVaR}_\alpha^{\mathrm{lower}}(G_\theta) = \frac{1}{\alpha}\int_0^\alpha Q_u(G_\theta)\,\mathrm{d}u, \tag{13}$$

where Q_u denotes the u-quantile of the return distribution. To balance average and tail performance, we optimize $\max_\theta\ J(\theta) = (1-\lambda)\mathbb{E}[G_\theta] + \lambda\,\mathrm{CVaR}_\alpha^{\mathrm{lower}}(G_\theta),\ \lambda \in [0,1]$. Larger λ places greater emphasis on low-return trajectories, while smaller α focuses on more extreme adverse outcomes.

Proposition 2 (Robust representation of the mixed return objective)**.** *Let P_θ denote the trajectory distribution induced by the smoothed policy, and assume G_θ is integrable. Then*

$$J(\theta) = \inf_{Q\in\mathcal{Q}_{\alpha,\lambda}(P_\theta)} \mathbb{E}_Q[G(\tau)], \tag{14}$$

where $\mathcal{Q}_{\alpha,\lambda}(P_\theta)$ consists of probability distributions $Q \ll P_\theta$ satisfying the following bounds with probability one under P_θ:

$$1-\lambda \le \frac{\mathrm{d}Q}{\mathrm{d}P_\theta} \le 1-\lambda+\frac{\lambda}{\alpha} \tag{15}$$

Thus, $J(\theta)$ represents the worst expected return over a bounded family of trajectory-distribution shifts. The parameter α controls the allowable concentration on low-return trajectories, while λ controls the departure from the nominal trajectory distribution. The proof is given in Appendix B.2.

A commonly used equivalent formulation introduces an auxiliary return threshold η: $\max_{\theta,\eta}\ \mathcal{J}(\theta,\eta) = (1-\lambda)\mathbb{E}[G_\theta] + \lambda\left[\eta - \frac{1}{\alpha}\mathbb{E}[(\eta - G_\theta)_+]\right]$, where $(x)_+ = \max(x,0)$. This formulation penalizes return shortfalls below η, whose optimum corresponds to an α-quantile of the return distribution. For fixed η, define the risk-adjusted trajectory utility

$$U(G;\eta) = (1-\lambda)G - \frac{\lambda}{\alpha}(\eta - G)_+. \tag{16}$$

The resulting risk-sensitive clipped surrogate objective is

$$L^{\mathrm{risk}}_{\mathrm{clip}}(\theta) = \mathbb{E}_t\left[\min\left(\frac{\widetilde{\pi}_\theta(a_t \mid \mathcal{M}_t)}{\widetilde{\pi}_{\theta_{\mathrm{old}}}(a_t \mid \mathcal{M}_t)}\widehat{A}^{\mathrm{risk}}_t, \mathrm{clip}\left(\frac{\widetilde{\pi}_\theta(a_t \mid \mathcal{M}_t)}{\widetilde{\pi}_{\theta_{\mathrm{old}}}(a_t \mid \mathcal{M}_t)}, 1-\epsilon, 1+\epsilon\right)\widehat{A}^{\mathrm{risk}}_t\right)\right],$$

where the advantage function $\widehat{A}^{\mathrm{risk}}_t = U(G(\tau);\eta) - \mathbb{E}[U(G(\tau);\eta)]$. Together, belief-based median smoothing and CVaR optimization address residual state uncertainty and trajectory-level tail risk, respectively, improving robust performance in unfavorable scenarios.

5 Experiments

We evaluate BBRR on two representative reinforcement learning benchmarks: Atari Mnih (2013) games with discrete action spaces and MuJoCo Brockman et al. (2016) continuous control tasks. Following prior robust RL settings Shen et al. (2020); Oikarinen et al. (2021); Liang et al. (2022); Sun et al. (2024), we instantiate BBRR with DQN for high-dimensional pixel-based Atari environments, including *Pong*, *Freeway*, *BankHeist*, and *RoadRunner*. We instantiate BBRR with PPO on MuJoCo tasks, including *Hopper*, *Walker2d*, *HalfCheetah*, and *Ant*. We evaluate the trained agents under clean observations and multiple adversarial attacks with different perturbation budgets. Our comparison baselines include adversarially trained agents: Raidal-DQN, Radial-PPO Oikarinen et al. (2021), WocaR-DQN, WocaR-PPO Liang et al. (2022), SA-DQN Zhang et al. (2020), ATLA-PPO Zhang et al. (2021), PA-ATLAPPO Sun et al. (2022); purification-based agents: S-DQN Sun et al. (2024), DP-DQN Sun & Zheng (2024).

For Atari, we evaluate robustness by reporting the average episodic return under three commonly used adversarial attacks: (1) PGD Zhang et al. (2020), which iteratively perturbs observations within the allowed budget; (2) MinBest Huang et al. (2017), which reduces the probability assigned to the action preferred by the clean policy; and (3) PA-AD Sun et al. (2022), a learned RL-based adversary designed to minimize the agent's long-term return. For MuJoCo, we consider five representative attacks to evaluate robustness: (1) Random attack, which injects random bounded perturbations into state observations; (2) MaxDiff (MAD) Zhang et al. (2020), which searches for perturbations that maximize the discrepancy between the actions induced by clean and perturbed observations; (3) Robust Sarsa (RS) Zhang et al. (2020), which attacks the policy using a robust action-value function; (4) SA-RL Zhang et al. (2020), which learns an optimal state adversary; and (5) PA-AD Sun et al. (2022), a strong learned policy adversary based on reinforcement learning.

5.1 Robustness Evaluation

The evaluation results in Tables 1 and 4 show that our method achieves the strongest robustness across Atari and MuJoCo. BBRR-DQN maintains high returns across PGD, MinBest, and PA-AD, including the largest Atari budgets, while BBRR-PPO obtains the best attacked performance on all four MuJoCo tasks across MAD, RS, SA-RL, and PA-AD. These gains involve only a limited loss in clean performance.

The behavior of representative baselines explains this advantage. SA-DQN and WocaR-DQN perform reasonably under small perturbations, but their robustness deteriorates sharply on BankHeist

Table 1: Clean and adversarial episodic returns on four Atari environments. Results are mean ± standard deviation over six seeds, with 20 episodes per seed. Best results are bolded, and gray rows denote BBRR-DQN.

Env	Model	Natural Reward	PGD			MinBest			PA-AD		
			$\epsilon = 3/255$	$\epsilon = 5/255$	$\epsilon = 15/255$	$\epsilon = 3/255$	$\epsilon = 5/255$	$\epsilon = 15/255$	$\epsilon = 3/255$	$\epsilon = 5/255$	$\epsilon = 15/255$
Pong	Vanilla-DQN	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-20.2 ± 0.4	-21.0 ± 0.0	-21.0 ± 0.0
	SA-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	20.2 ± 1.1	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-20.7 ± 1.4	-21.0 ± 0.0
	WocaR-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-19.2 ± 0.6	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-19.2 ± 2.5	-21.0 ± 0.0
	Radial-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-20.1 ± 1.7	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-18.4 ± 1.4	-21.0 ± 0.0
	DP-DQN	19.9 ± 0.3	19.3 ± 0.4	12.5 ± 2.4	4.8 ± 1.2	20.5 ± 0.8	18.3 ± 1.6	14.4 ± 2.3	20.1 ± 1.0	13.6 ± 1.5	8.2 ± 2.3
	BBRR-DQN (Ours)	$\mathbf{21.0 \pm 0.0}$	20.8 ± 0.4	$\mathbf{20.8 \pm 0.6}$	$\mathbf{20.3 \pm 0.9}$	20.6 ± 0.9	20.4 ± 0.8	$\mathbf{21.0 \pm 0.0}$	20.9 ± 1.0	$\mathbf{20.0 \pm 1}$	$\mathbf{19.4 \pm 3.2}$
Freeway	Vanilla-DQN	33.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SA-DQN	30.0 ± 0.0	30.0 ± 0.0	28.3 ± 0.4	17.4 ± 1.2	28.0 ± 1.2	22.3 ± 0.8	18.7 ± 2.1	9.5 ± 3.8	3.4 ± 0.6	0 ± 0
	WocaR-DQN	32.6 ± 0.4	32.3 ± 0.6	31.5 ± 1.2	23.2 ± 1.9	32.6 ± 0.2	26.3 ± 1.0	20.4 ± 0.8	12.3 ± 3.2	8.7 ± 1.4	2.5 ± 1.0
	Radial-DQN	33.2 ± 0.8	33.1 ± 0.6	32.4 ± 0.6	21.8 ± 1.5	33.3 ± 0.4	27.3 ± 1.2	19.8 ± 1.1	10.8 ± 3.6	6.5 ± 1.8	0 ± 0
	DP-DQN	29.4 ± 0.5	29.2 ± 0.2	28.7 ± 0.1	26.6 ± 0.8	29.4 ± 0.9	28.4 ± 0.4	24.5 ± 1.2	28.3 ± 2.2	25.6 ± 1.9	20.3 ± 3.6
	BBRR-DQN (Ours)	$\mathbf{34.0 \pm 0.0}$	$\mathbf{34.0 \pm 0.0}$	$\mathbf{34.0 \pm 0.0}$	$\mathbf{33.5 \pm 0.2}$	$\mathbf{34.0 \pm 0.0}$	$\mathbf{34.0 \pm 0.0}$	$\mathbf{33.6 \pm 0.5}$	$\mathbf{33.9 \pm 0.2}$	$\mathbf{33.0 \pm 1.0}$	$\mathbf{31.4 \pm 0.6}$
BankHeist	Vanilla-DQN	$\mathbf{1302.2 \pm 3.4}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SA-DQN	1236.0 ± 1.4	1129.8 ± 52.7	925.4 ± 34.6	366.3 ± 21.2	1024.3 ± 37.1	902.4 ± 19.5	428.2 ± 26.0	489.2 ± 52.6	102.3 ± 32.6	0.0 ± 0.0
	WocaR-DQN	1220.3 ± 10.6	1193.2 ± 14.8	1037.5 ± 27.3	582.2 ± 19.3	1045.4 ± 17.3	915.0 ± 36.1	572.8 ± 42.5	714.6 ± 85.2	327.6 ± 102.4	25.3 ± 12.8
	Radial-DQN	1178.5 ± 9.5	1164.2 ± 10.6	1031.0 ± 22.4	516.9 ± 16.2	982.7 ± 38.3	864.7 ± 21.4	538.1 ± 62.3	549.6 ± 92.4	277.3 ± 132.4	0.0 ± 0.0
	DP-DQN	1021.5 ± 4.9	965.2 ± 23.8	849.6 ± 62.5	683.4 ± 72.1	930.6 ± 48.2	827.8 ± 55.4	693.5 ± 110.9	785.2 ± 126.8	619.5 ± 172.4	430.3 ± 98.7
	BBRR-DQN (Ours)	1244.5 ± 18.2	$\mathbf{1234.4 \pm 16.5}$	$\mathbf{1227.3 \pm 59.8}$	$\mathbf{939.6 \pm 44.2}$	$\mathbf{1227 \pm 18.3}$	$\mathbf{1225 \pm 23.7}$	$\mathbf{1037 \pm 163.2}$	$\mathbf{1234.3 \pm 42.4}$	$\mathbf{1218.2 \pm 156.8}$	$\mathbf{932.4 \pm 152.5}$
RoadRunner	Vanilla-DQN	45527 ± 2893	0 ± 0	0 ± 0	0 ± 0	2985 ± 1440	0 ± 0	0 ± 0	203 ± 65	0 ± 0	0 ± 0
	SA-DQN	44638 ± 2367	20910 ± 827	3074 ± 179	0 ± 0	25548 ± 737	12324 ± 529	0 ± 0	5420 ± 3685	2263 ± 825	0 ± 0
	WocaR-DQN	44156 ± 2279	38720 ± 1765	11872 ± 2973	0 ± 0	28476 ± 3623	17258 ± 1932	1087 ± 231	8239 ± 2766	3482 ± 759	0 ± 0
	Radial-DQN	44595 ± 1143	38485 ± 1068	18670 ± 839	685 ± 490	$\mathbf{43280 \pm 1358}$	25475 ± 603	670 ± 335	1366 ± 3354	294 ± 58	0 ± 0
	DP-DQN	42153 ± 1872	38630 ± 3223	11887 ± 2629	2165 ± 904	36227 ± 4184	7659 ± 2110	1335 ± 741	31873 ± 5247	5224 ± 1853	938 ± 545
	BBRR-DQN (Ours)	$\mathbf{49860 \pm 976}$	$\mathbf{41600 \pm 1798}$	$\mathbf{41110 \pm 1439}$	$\mathbf{19270 \pm 2412}$	39080 ± 2861	$\mathbf{44560 \pm 3252}$	$\mathbf{19650 \pm 1910}$	$\mathbf{38240 \pm 3713}$	$\mathbf{41390 \pm 2128}$	$\mathbf{18530 \pm 3730}$

and RoadRunner as the budget increases. Their IBP bounds and regularization objectives become less reliable over large regions, where enforcing one action for all possible states can select poor actions. In MuJoCo, ATLA-PPO and PA-ATLA-PPO improve robustness against some adversaries, but their performance varies across attack objectives, indicating limited transfer. Radial-PPO and WocaR-PPO are more stable in several cases but still suffer substantial losses under PA-AD. In contrast, BBRR uses the diffusion bridge to avoid relying on loose bounds over large perturbation regions, employing belief state approximation and Conditional Value-at-Risk (CVaR) to reduce restoration uncertainty and decision risks, respectively.

Table 2: Ablation study on four Atari environments. We compare our BBRR-DQN with variants that use only CVaR or the diffusion bridge.

Environment	Model	PGD $\epsilon = 15/255$	Environment	Model	PGD $\epsilon = 15/255$
Pong	CvaR Only	-21 ± 0	**Freeway**	CvaR Only	0 ± 0
	Diffusion Bridge Only	17.4 ± 2.5		Diffusion Bridge Only	30.6 ± 1.7
	BBRR-DQN (Ours)	$\mathbf{20.3 \pm 0.9}$		BBRR-DQN (Ours)	$\mathbf{33.5 \pm 0.2}$
BankHeist	CvaR Only	32 ± 2	**RoadRunner**	CvaR Only	0 ± 0
	Diffusion Bridge Only	820 ± 31		Diffusion Bridge Only	16280 ± 4371
	BBRR-DQN (Ours)	$\mathbf{939 \pm 44}$		BBRR-DQN (Ours)	$\mathbf{19270 \pm 2412}$

5.2 Ablation and Sensitivity Analysis

Component ablation on BBRR. In Tables 2 and 3, we compare BBRR-DQN and BBRR-PPO with several variants designed to isolate the contributions of the individual components. During testing, all variants operate on the perturbed state as follows. The CVaR variants use random samples from the ε ball around the perturbed state and apply the corresponding risk objective: lower tail aggregation of action values for DQN and lower tail optimization of trajectory returns for PPO. The median smoothing variants use the same random samples but aggregate the resulting policy outputs by their median. The diffusion bridge variant uses the corresponding vanilla DQN or PPO policy on a single state generated by the bridge. The results show that bridge-based restoration is critical,

Table 3: Ablation study on MuJoCo environments. We compare our BBRR-PPO with variants that independently use CVaR, median smoothing, or the diffusion bridge.

Environment	Model	PA-AD $\epsilon = 0.075$	Environment	Model	PA-AD $\epsilon = 0.075$
Walker2d	CvaR Only	684 ± 99	**Hopper**	CvaR Only	972 ± 185
	Median Smoothing Only	1077 ± 294		Median Smoothing Only	1190 ± 142
	Diffusion Bridge Only	3077 ± 143		Diffusion Bridge Only	2879 ± 343
	BBRR-PPO (Ours)	$\mathbf{3160 \pm 182}$		BBRR-PPO (Ours)	$\mathbf{2982 \pm 414}$

Table 4: Robustness and clean-performance comparison of BBRR-PPO with six baselines on four MuJoCo environments. Results are reported as the mean episodic return ± standard deviation over six random seeds, with 20 evaluation episodes per seed. The best result for each environment is shown in bold, and the most robust agent is highlighted in gray.

Env	epsilon ε	Model	Natural	Random	MAD	RS	SA-RL	PA-AD
HalfCheetah	**0.15**	Vanilla-PPO	**7117±98**	5486±1378	1836±866	489±758	-660±218	-356±407
		SA-PPO	3632±20	3619±18	3624±23	3283±20	3028±23	2512±16
		ATLA-PPO	6157±852	6164±603	5790±174	4806±392	5058±418	2576±548
		PA-ATLA-PPO	6289±342	**6215±346**	5961±253	5226±114	4872±379	3840±273
		WocaR-PPO	6032±68	5969±149	5850±228	5319±220	5365±54	4269±172
		Radial-PPO	4724±14	4731±42	3994±156	3864±232	3253±131	2674±168
		BBRR-PPO (Ours)	5959±81	5863±108	**6004±163**	**5412±89**	**5517±294**	**4571±372**
Walker2d	**0.05**	Vanilla-PPO	4472±635	3007±1200	2869±1271	1336±654	1086±516	804±130
		SA-PPO	4487±61	4465±39	3668±689	3808±138	2908±336	1042±353
		ATLA-PPO	3842±475	3927±368	3836±492	3239±294	3663±707	1224±770
		PA-ATLA-PPO	4178±529	4129±78	4024±272	3966±307	3450±178	2248±131
		WocaR-PPO	4156±495	4244±157	4177±176	4093±138	3770±196	2722±173
		Radial-PPO	**5251±12**	**5184±42**	4494±150	3572±239	3320±245	1395±194
		BBRR-PPO (Ours)	4884±682	4657±328	**4503±215**	**4238±224**	**4074±462**	**3302±283**
Hopper	**0.075**	Vanilla-PPO	3167±542	2101±793	1410±655	794±238	636±59	160±136
		SA-PPO	3705±28	2710±801	2652±835	1130±42	1076±791	856±21
		ATLA-PPO	3291±600	3165±576	2814±725	2244±618	1772±802	1232±350
		PA-ATLA-PPO	3449±237	3325±239	3145±546	3002±329	1529±284	2521±325
		WocaR-PPO	3616±99	3633±30	3541±207	3277±159	2390±145	2579±229
		Radial-PPO	**3740±44**	**3729±100**	3214±142	2141±232	1722±186	1439±204
		BBRR-PPO (Ours)	3641±11	3691±133	**3564±315**	**3462±173**	**3109±62**	**2982±414**
Ant	**0.15**	Vanilla-PPO	**5687±758**	5261±1005	1759±828	268±227	-872±436	-2580±872
		SA-PPO	4292±384	4986±452	4662±522	3412±1755	2511±1117	-1296±923
		ATLA-PPO	5359±153	5366±104	5240±170	4136±149	3765±101	220±338
		PA-ATLA-PPO	5469±106	5496±158	5328±196	4124±291	3694±188	2986±364
		WocaR-PPO	5596±225	5558±241	5284±182	4339±160	3822±185	3164±163
		Radial-PPO	5076±254	5031±142	4777±156	3731±177	3188±115	1544±194
		BBRR-PPO (Ours)	5613±174	**5589±236**	**5463±391**	**4927±283**	**4391±423**	**3822±462**

Table 5: Sensitivity to the number k of lowest-value belief states used in the lower-tail CVaR aggregation on BankHeist and RoadRunner under PGD attack with $\epsilon = 20/255$.

Environment	w/o Belief	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 10$
BankHeist	349 ± 48	$\mathbf{764 \pm 65}$	627 ± 198	504 ± 199	458 ± 169	422 ± 196
RoadRunner	12382 ± 3347	$\mathbf{18760 \pm 2174}$	17250 ± 2846	18560 ± 2111	14410 ± 2687	13410 ± 3471

while each additional component contributes to robustness and their integration yields the strongest overall performance.

Sensitivity to the CVaR Tail Parameter. The sensitivity analysis in Table 5 demonstrates the benefit of focusing CVaR aggregation on the lower tail of belief states. On both BankHeist and RoadRunner, $k = 2$ achieves the strongest performance, indicating that a limited number of low-value states captures restoration failures while avoiding excessive conservatism. In contrast, increasing k weakens the lower-tail signal and generally reduces robustness. Omitting belief states and CVaR aggregation reduces the policy to a single recovered state and leads to less robust decisions. These results show that lower-tail CVaR provides an effective balance between protecting against unfavorable restorations and maintaining policy performance.

6 Conclusion

We presented Belief Based Risk Reduction (BBRR), a unified framework for robust reinforcement learning under adversarial observation perturbations. BBRR uses a diffusion bridge to mitigate the distribution gap between adversarial observations and clean samples, while employing belief state approximation and Conditional Value-at-Risk (CVaR) to reduce restoration uncertainty and decision risks, respectively. Experiments on Atari and MuJoCo show that BBRR consistently improves robustness across diverse attacks while preserving strong clean performance. Future work includes extending BBRR to semantic perturbations, such as changes in contrast, brightness, and illumination, which are more representative of real world conditions for vision based RL agents, including autonomous vehicles and robotic systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing proofs. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, or interpret results. Dataset cleaning and reformatting and qualitative or thematic data analysis were not applicable to this work. We additionally used generative AI tools to search for information and edit the manuscript for readability. We reviewed all AI-assisted work and take responsibility for the final content of this paper, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human participants, personal data, or the release of new datasets. All experiments are conducted in standard simulated reinforcement learning environments. We do not identify any additional ethical concerns beyond those associated with the responsible deployment of reinforcement learning systems.

REPRODUCIBILITY STATEMENT

We provide an anonymous code repository at `https://anonymous.4open.science/r/Robust_BBRR`. Detailed training and testing procedures are provided in Appendix D.

REFERENCES

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Thinking coherently. *Risk*, 10(11):68–71, 1997.

Vahid Behzadan and Arslan Munir. Whatever does not kill deep reinforcement learning, makes it stronger. *arXiv preprint arXiv:1712.09344*, 2017.

Roman Belaire, Arunesh Sinha, and Pradeep Varakantham. On minimizing adversarial counterfactual error in adversarial rl. *arXiv preprint arXiv:2406.04724*, 2024.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Xi Chen, Zhihui Zhu, and Andrew Perrault. The distributional reward critic framework for reinforcement learning under perturbed rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15895–15903, 2025.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Wenhao Ding, Laixi Shi, Yuejie Chi, and DING ZHAO. Seeing is not believing: Robust reinforcement learning against spurious correlation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 66328–66363. Curran Associates, Inc., 2023. doi: 10.52202/075280-2895. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/d0c3841867db2c516454845a450ca885-Paper-Conference.pdf`.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Adversarial diffusion for robust reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Jia Fu, Yongtao Wu, Yihang Chen, Kunyu Peng, Xiao Zhang, Volkan Cevher, Sepideh Pashami, and Anders Holst. Diffusion-based cumulative adversarial purification for vision language models, 2026. URL https://openreview.net/forum?id=AYtAttCH5F.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgEMpVFwB.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 3(2):90–109, 2021.

Taisuke Kobayashi. L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4032–4039. IEEE, 2022.

Ezgi Korkmaz. Investigating vulnerabilities of deep neural policies. In *Uncertainty in Artificial Intelligence*, pp. 1661–1670. PMLR, 2021.

Ezgi Korkmaz. Deep reinforcement learning policies learn shared adversarial features across mdps. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7229–7238, 2022.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8369–8377, 2023.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*, 2017.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016. URL http://jmlr.org/papers/v17/15-522.html.

Xiao Li, Wenxuan Sun, Huanran Chen, Qiongxiu Li, Yingzhe He, Jie Shi, and Xiaolin Hu. Adbm: Adversarial diffusion bridge model for reliable adversarial purification. In *International Conference on Learning Representations*, volume 2025, pp. 78366–78388, 2025.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in neural information processing systems*, 35:22547–22561, 2022.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zeyuan Liu, Zhihe Yang, Jiawei Xu, Rui Yang, Jiafei Lyu, Baoxiang Wang, Yunjian Xu, and Xiu Li. ADG: Ambient diffusion-guided dataset recovery for corruption-robust offline reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8fECf5YbJY.

Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. Adversarial machine learning in image classification: A survey toward the defender's perspective. *ACM Computing Surveys (CSUR)*, 55(1):1–38, 2021.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3932–3939. IEEE, 2017.

Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. Robust deep reinforcement learning through adversarial loss. *Advances in Neural Information Processing Systems*, 34:26156–26167, 2021.

R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5):1, 2023.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2040–2042, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 34:100199, 2019.

You Qiaoben, Xinning Zhou, Chengyang Ying, and Jun Zhu. Strategically-timed state-observation attacks on deep reinforcement learning agents. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL https://openreview.net/forum?id=FSD_8Sglf_u.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*, pp. 8707–8718. PMLR, 2020.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Chung-En Sun, Sicun Gao, and Tsui-Wei Weng. Breaking the barrier: Enhanced utility and robustness in smoothed drl agents. In *International Conference on Machine Learning*, pp. 46957–46987. PMLR, 2024.

Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5883–5891, 2020.

Xiaolin Sun and Zizhan Zheng. Belief-enriched pessimistic q-learning against adversarial state perturbations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7gDENzTzw1.

Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep RL. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JM2kFbJvvI.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Du, and yelong shen. Reinforcement learning for reasoning in large language models with one training example. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 122721–122764. Curran Associates, Inc., 2025. doi: 10.52202/085713-4093. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b1ea3f93167c55f3fc2999b68a170ff3-Paper-Conference.pdf.

Zhihe Yang and Yunjian Xu. Dmbp: Diffusion model-based predictor for robust offline reinforcement learning against state observation perturbations. In *International Conference on Learning Representations*, volume 2024, pp. 38549–38578, 2024.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=sCZbhBvqQaU.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. In *International Conference on Learning Representations*, volume 2025, pp. 81857–81884, 2025.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *International Conference on Learning Representations*, volume 2024, pp. 8160–8171, 2024.

A PROOFS FOR DIFFUSION BRIDGE PURIFICATION

In this section, we present the proofs of Proposition 1 on diffusion bridge construction and Theorem 1 on one-step reconstruction error. The proof is based on the proof proposed by Li et al. (2025)

A.1 DIFFUSION BRIDGE CONSTRUCTION

Proof of Proposition 1. Write $a_\ell = \bar{\alpha}_\ell$, $s = s_t$, $d = \delta_t$, and suppress the environment-time subscript. Condition on the pair (s, d). To specify the joint law across diffusion times, start with the usual forward Markov chain

$$x_0^a = s + d, \qquad x_\ell^a = \sqrt{\alpha_\ell} x_{\ell-1}^a + \sqrt{\beta_\ell} z_\ell, \quad z_\ell \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I). \tag{17}$$

Its marginal at time ℓ is Eq. equation 1. That marginal representation does not require the same Gaussian noise to be shared across all steps of this Markov chain. For a deterministic sequence k_ℓ, let $h_\ell = \sqrt{a_\ell} - k_\ell$ and $x_\ell^d = x_\ell^a - k_\ell d$. Translation of the Gaussian transition and marginal gives

$$q(x_\ell^d \mid x_{\ell-1}^d, s, d) = \mathcal{N}\big(\sqrt{\alpha_\ell} x_{\ell-1}^d + (\sqrt{\alpha_\ell} k_{\ell-1} - k_\ell) d, \beta_\ell I\big)\,, \tag{18}$$

$$q(x_\ell^d \mid s, d) = \mathcal{N}(\sqrt{a_\ell} s + h_\ell d, (1 - a_\ell) I)\,. \tag{19}$$

For $\ell \geq 2$, Bayes' rule yields a Gaussian reverse posterior. To exhibit its dependence on the attack explicitly, put $y = x_{\ell-1}^d$, $x = x_\ell^d$, $b_\ell = \sqrt{\alpha_\ell} k_{\ell-1} - k_\ell$, and $m_{\ell-1} = \sqrt{a_{\ell-1}} s + h_{\ell-1} d$. Then

$$\log q(y \mid x, s, d) = -\frac{\|x - \sqrt{\alpha_\ell} y - b_\ell d\|_2^2}{2\beta_\ell} - \frac{\|y - m_{\ell-1}\|_2^2}{2(1 - a_{\ell-1})} + \text{const}, \tag{20}$$

where the last term is independent of y. Completing the square gives

$$\widetilde{\beta}_\ell = \left(\frac{\alpha_\ell}{\beta_\ell} + \frac{1}{1 - a_{\ell-1}}\right)^{-1} = \frac{\beta_\ell (1 - a_{\ell-1})}{1 - a_\ell}, \tag{21}$$

$$\mu_\ell = \widetilde{\beta}_\ell \left[\frac{\sqrt{\alpha_\ell}}{\beta_\ell}(x - b_\ell d) + \frac{m_{\ell-1}}{1 - a_{\ell-1}}\right] = A_\ell x + B_\ell s + D_\ell d, \tag{22}$$

$$A_\ell = \frac{\sqrt{\alpha_\ell}(1 - a_{\ell-1})}{1 - a_\ell}, \qquad B_\ell = \frac{\sqrt{a_{\ell-1}} \beta_\ell}{1 - a_\ell}, \qquad D_\ell = h_{\ell-1} - A_\ell h_\ell. \tag{23}$$

Here the expression for D_ℓ uses $\sqrt{a_{\ell-1}} - A_\ell \sqrt{a_\ell} = B_\ell$. The covariance already depends only on the schedule. Consequently, eliminating the explicit attack term from the normalized posterior is equivalent to $D_\ell = 0$, or

$$h_{\ell-1} = \frac{\sqrt{\alpha_\ell}(1 - a_{\ell-1})}{1 - a_\ell} h_\ell, \qquad 2 \leq \ell \leq T. \tag{24}$$

Terms independent of y in Eq. equation 20 are absorbed by normalization and impose no additional condition.

Preserving the terminal input requires $k_T = 0$, hence $h_T = \sqrt{a_T}$. The recurrence makes $h_\ell \sqrt{a_\ell}/(1 - a_\ell)$ constant for $1 \leq \ell \leq T$. Its terminal value is $a_T/(1 - a_T)$, so

$$h_\ell = \frac{a_T (1 - a_\ell)}{\sqrt{a_\ell}(1 - a_T)}, \qquad k_\ell = \sqrt{a_\ell} - h_\ell = \frac{a_\ell - a_T}{\sqrt{a_\ell}(1 - a_T)}. \tag{25}$$

At $\ell = 0$, define $h_0 = 0$ and $k_0 = 1$, also consistent with this formula since $a_0 = 1$. Thus $x_0^d = s$ and $x_T^d = x_T^a$. The $\ell = 1$ posterior is the point mass at s; it is handled directly instead of dividing by $1 - a_0 = 0$. For $\ell \geq 2$, the resulting posterior is

$$q(x_{\ell-1}^d \mid x_\ell^d, s, d) = \mathcal{N}(A_\ell x_\ell^d + B_\ell s, \widetilde{\beta}_\ell I). \tag{26}$$

This is an attack-independent kernel conditional on the clean state; a practical reverse model must still estimate that state.

Finally, substituting Eq. equation 25 into the marginal representation gives

$$x_\ell^d = \sqrt{a_\ell} s + \sqrt{1 - a_\ell}\, \varepsilon + h_\ell d = \sqrt{a_\ell} s + \sqrt{1 - a_\ell}\, (\varepsilon + \lambda_\ell d), \quad \lambda_\ell = \frac{a_T \sqrt{1 - a_\ell}}{\sqrt{a_\ell}(1 - a_T)}, \quad \ell \geq 1. \tag{27}$$

This proves Eq. equation 3 and explains the composite target in Eq. equation 4. In particular, predicting this target exactly in the clean-state reconstruction formula recovers s. The endpoint identities hold for every pair (s, d) and hence also for their induced distributions, completing the proof. □

A.2 One-step purification error bound

Proof of Theorem 1. For a deterministic DDIM step from diffusion time ℓ to $r < \ell$, the update is

$$\widehat{x}_{r,t} = \sqrt{\bar{\alpha}_r}\frac{\widehat{x}_{\ell,t} - \sqrt{1-\bar{\alpha}_\ell}\epsilon_\psi(\widehat{x}_{\ell,t}, \ell)}{\sqrt{\bar{\alpha}_\ell}} + \sqrt{1-\bar{\alpha}_r}\,\epsilon_\psi(\widehat{x}_{\ell,t}, \ell). \tag{28}$$

Set $\ell = T$, $r = 0$, and $\widehat{x}_{T,t} = x^a_{T,t} = x^d_{T,t}$. Because $\bar{\alpha}_0 = 1$, the second term vanishes. Writing $a = \bar{\alpha}_T \in (0, 1)$ gives

$$\hat{s}_t = \frac{x^a_{T,t} - \sqrt{1-a}\,\epsilon_\psi(x^d_{T,t}, T)}{\sqrt{a}} \tag{29}$$

$$= s_t + \delta_t + \sqrt{\frac{1-a}{a}}[\varepsilon - \epsilon_\psi(x^d_{T,t}, T)]. \tag{30}$$

At the terminal time, the training target satisfies

$$\lambda_T = \frac{a\sqrt{1-a}}{(1-a)\sqrt{a}} = \sqrt{\frac{a}{1-a}}. \tag{31}$$

Thus the reconstruction error is exactly a rescaled prediction residual:

$$\hat{s}_t - s_t = \sqrt{\frac{1-a}{a}}\left[\varepsilon + \lambda_T\delta_t - \epsilon_\psi(x^d_{T,t}, T)\right]. \tag{32}$$

For each diffusion time define the expected residual

$$R_\ell = \mathbb{E}_{(s_t,\delta_t)\sim\mathcal{D},\varepsilon}\left\|\epsilon_\psi(x^d_{\ell,t}, \ell) - (\varepsilon + \lambda_\ell\delta_t)\right\|_2^2. \tag{33}$$

Taking squared norms and expectations in Eq. equation 32 gives the identity

$$\mathbb{E}_{\mathcal{D},\varepsilon}\|\hat{s}_t - s_t\|_2^2 = \frac{1-a}{a}R_T. \tag{34}$$

Independence of the sampled diffusion time implies $\mathcal{L}_{\text{bridge}}(\psi) = \sum_{\ell=1}^T p_\ell R_\ell$. Since each term is nonnegative and $p_T > 0$,

$$p_T R_T \le \mathcal{L}_{\text{bridge}}(\psi) \le \eta. \tag{35}$$

Combining the last two displays proves

$$\mathbb{E}_{\mathcal{D},\varepsilon}\|\hat{s}_t - s_t\|_2^2 \le \frac{1-\bar{\alpha}_T}{\bar{\alpha}_T p_T}\eta. \tag{36}$$

For uniform sampling on $\{1, \dots, T\}$, $p_T = 1/T$, giving the stated constant. □

For completeness, for any fixed pair (s_t, δ_t) define $R_\ell(s_t, \delta_t)$ by taking only the Gaussian expectation in R_ℓ above. The same algebra gives the exact conditional identity

$$\mathbb{E}_\varepsilon\|\hat{s}_t - s_t\|_2^2 = \frac{1-\bar{\alpha}_T}{\bar{\alpha}_T}R_T(s_t, \delta_t). \tag{37}$$

If the conditional loss $\sum_\ell p_\ell R_\ell(s_t, \delta_t) \le \eta$ is assumed for this pair, the conditional bound is $C_T\eta$. Alternatively, a direct terminal assumption $R_T(s_t, \delta_t) \le \eta_T$ gives $(1-\bar{\alpha}_T)\eta_T/\bar{\alpha}_T$. A distribution-level loss bound alone does not imply either conditional assumption for every state–attack pair. Likewise, an empirical training loss needs a generalization argument before it can replace the distribution-level loss used here. The theorem concerns a single reverse step; it does not bound error accumulation in a multistep sampler or under a different test attack distribution. For a context-conditioned denoiser, the same identities hold with the context included in the conditioning and in the expectation under the data distribution.

B Proofs for CVaR-Based Risk Reduction

In this section, we present the proofs of Theorem 2 on CVaR performance bound and Proposition 2 on mixed CVaR robust representation. The proof is based on the proof proposed by Sun & Zheng (2024) and Foffano et al. (2025).

B.1 Proof of the CVaR Performance Bound

Proof of Theorem 2. We first invoke Assumption 1. In particular, its reward bound gives a uniform bound on all discounted action values, while its Lipschitz conditions on the reward and transition kernel give the state regularity needed below. The candidate-distance condition in the same assumption then transfers this regularity from the true state to every belief state used by the CVaR aggregation. Set $V_{\max} = R_{\max}/(1-\gamma)$ and $L = l_r + l_p|\mathcal{S}|V_{\max}$. Since $Q_0 = 0$, the exact update implies by induction that $\|Q_n\|_\infty \le V_{\max}$. For any $n \ge 0$, the common continuation function $Q_n(s', \tilde{\pi}_n(s'))$ in the two Bellman updates gives

$$\begin{aligned}|Q_{n+1}(s_1,a) - Q_{n+1}(s_2,a)| &\le l_r\|s_1 - s_2\| \\ &\quad + \gamma V_{\max}\sum_{s'}|P(s'|s_1,a) - P(s'|s_2,a)| \\ &\le L\|s_1 - s_2\|.\end{aligned}$$

The same Lipschitz bound holds for Q_0.

All belief state values therefore lie between $Q_n(s,a) - Ld$ and $Q_n(s,a) + Ld$. Their lowest-k average satisfies

$$|\bar{Q}_n^{\mathrm{CVaR}}(\mathcal{M}(\omega(s)), a) - Q_n(s,a)| \le Ld.$$

Let $a^* \in \arg\max_a Q_n(s,a)$ and $\hat{a} = \tilde{\pi}_n(s)$. The maximizing property of $\hat{a}$ yields

$$\begin{aligned}0 \le Q_n(s,a^*) - Q_n(s,\hat{a}) &\le \bar{Q}_n^{\mathrm{CVaR}}(\mathcal{M}(\omega(s)), a^*) + Ld \\ &\quad - \bar{Q}_n^{\mathrm{CVaR}}(\mathcal{M}(\omega(s)), \hat{a}) + Ld \\ &\le 2Ld.\end{aligned}$$

Consequently,

$$\|T^*Q_n - T^{\tilde{\pi}_n}Q_n\|_\infty = \|T^*Q_n - Q_{n+1}\|_\infty \le 2\gamma Ld = \Delta.$$

Write $e_n = \|Q^* - Q_n\|_\infty$. Contraction of T^* gives $e_{n+1} \le \gamma e_n + \Delta$, and hence

$$\limsup_{n\to\infty} e_n \le \frac{\Delta}{1-\gamma}.$$

For each fixed n, $T^{\tilde{\pi}_n}$ is a γ-contraction with fixed point $Q^{\tilde{\pi}_n}$. Thus, writing $D_n = \|Q^* - Q^{\tilde{\pi}_n}\|_\infty$,

$$\begin{aligned}D_n &\le \|T^*Q^* - T^{\tilde{\pi}_n}Q^*\|_\infty + \gamma D_n \\ &\le 2\gamma e_n + \Delta + \gamma D_n.\end{aligned}$$

Taking the limit superior and rearranging yields

$$\limsup_n D_n \le \frac{(1+\gamma)\Delta}{(1-\gamma)^2},$$

which proves the stated bound. No convergence of the changing-policy sequence Q_n is required. □

B.2 Proof of the Mixed CVaR Robust Representation

Proof of Proposition 2. Fix θ and abbreviate $P = P_\theta$ and $G = G(\tau)$. All expectations in this proof are under P unless specified otherwise, and all bounds on density ratios hold P-almost surely. The trajectory law includes the randomness of the environment, observation perturbations, purification, and action sampling under the specified rollout protocol. We assume $\mathbb{E}[|G|] < \infty$, $0 < \alpha < 1$, and $0 \le \lambda \le 1$, as in the proposition.

Define the CVaR risk envelope

$$\mathcal{U}_\alpha(P) = \{\xi : \ 0 \le \xi(\tau) \le 1/\alpha, \quad \mathbb{E}_P[\xi(\tau)] = 1\}, \tag{38}$$

where ξ is measurable and the inequalities hold P-almost surely. Define the threshold objective

$$F_\alpha(\eta) = \eta - \frac{1}{\alpha}\mathbb{E}[(\eta - G)_+].$$

For any $\xi \in \mathcal{U}_\alpha(P)$, the normalization $\mathbb{E}[\xi] = 1$ implies

$$\begin{aligned}\mathbb{E}[\xi G] &= \eta + \mathbb{E}[\xi(G-\eta)] \\ &\geq \eta - \frac{1}{\alpha}\mathbb{E}[(\eta - G)_+] = F_\alpha(\eta).\end{aligned}$$

Indeed, on $\{G < \eta\}$, multiplying $\xi \leq 1/\alpha$ by $G - \eta < 0$ reverses the inequality; on $\{G \geq \eta\}$, $\xi(G-\eta) \geq 0$. All these expectations are finite because ξ is bounded and G is integrable. Taking the supremum over η and then the infimum over feasible ξ gives

$$\sup_{\eta \in \mathbb{R}} F_\alpha(\eta) \leq \inf_{\xi \in \mathcal{U}_\alpha(P)} \mathbb{E}[\xi G]. \tag{39}$$

Let $q_\alpha = \inf\{x : P(G \leq x) \geq \alpha\}$ be the lower α-quantile. Since $0 < \alpha < 1$, this quantile is finite and

$$p_- := P(G < q_\alpha) \leq \alpha \leq P(G \leq q_\alpha) = p_- + p_0, \qquad p_0 := P(G = q_\alpha).$$

Set $\beta = (\alpha - p_-)/p_0$ if $p_0 > 0$, and $\beta = 0$ otherwise. When $p_0 = 0$, the preceding inequalities imply $p_- = \alpha$; in either case, $0 \leq \beta \leq 1$ and $p_- + \beta p_0 = \alpha$. Define the measurable trajectory weight

$$\xi^*(\tau) = \frac{1}{\alpha}\left(\mathbf{1}_{\{G(\tau) < q_\alpha\}} + \beta \mathbf{1}_{\{G(\tau) = q_\alpha\}}\right). \tag{40}$$

It satisfies $0 \leq \xi^* \leq 1/\alpha$ and $\mathbb{E}[\xi^*] = (p_- + \beta p_0)/\alpha = 1$, so it belongs to $\mathcal{U}_\alpha(P)$. Moreover,

$$\begin{aligned}\mathbb{E}[\xi^* G] &= \frac{1}{\alpha}\left(\mathbb{E}[G\mathbf{1}_{\{G<q_\alpha\}}] + q_\alpha(\alpha - p_-)\right) \\ &= q_\alpha - \frac{1}{\alpha}\mathbb{E}[(q_\alpha - G)\mathbf{1}_{\{G<q_\alpha\}}] \\ &= F_\alpha(q_\alpha).\end{aligned}$$

Together with Equation equation 39, this yields

$$F_\alpha(q_\alpha) \leq \sup_\eta F_\alpha(\eta) \leq \inf_{\xi \in \mathcal{U}_\alpha(P)} \mathbb{E}[\xi G] \leq \mathbb{E}[\xi^* G] = F_\alpha(q_\alpha).$$

Thus all inequalities are equalities. The common value is also the quantile-integral definition used in the main text, since

$$\int_0^\alpha Q_u(G)\,\mathrm{d}u = \mathbb{E}[G\mathbf{1}_{\{G<q_\alpha\}}] + q_\alpha(\alpha - p_-).$$

The second term includes exactly the required fraction of the probability mass at the quantile. This establishes the dual identity even for discrete return distributions or distributions with atoms:

$$\mathrm{CVaR}_\alpha^{\mathrm{lower}}(G_\theta) = \inf_{\xi \in \mathcal{U}_\alpha(P_\theta)} \mathbb{E}_{P_\theta}[\xi(\tau)G(\tau)]. \tag{41}$$

For $\lambda > 0$, Equation equation 41 gives

$$\begin{aligned}J(\theta) &= (1-\lambda)\mathbb{E}[G] + \lambda \inf_{\xi \in \mathcal{U}_\alpha(P)} \mathbb{E}[\xi G] \\ &= \inf_{\xi \in \mathcal{U}_\alpha(P)} \{(1-\lambda)\mathbb{E}[G] + \lambda \mathbb{E}[\xi G]\} \\ &= \inf_{\xi \in \mathcal{U}_\alpha(P)} \mathbb{E}[((1-\lambda) + \lambda\xi)G].\end{aligned}$$

The nominal expectation is independent of ξ, and multiplication by the positive scalar λ preserves the infimum. For $\lambda = 0$, both sides equal $\mathbb{E}[G]$; the envelope is nonempty because $\xi \equiv 1$ is feasible. Thus,

$$J(\theta) = \inf_{\xi \in \mathcal{U}_\alpha(P)} \mathbb{E}[((1-\lambda) + \lambda\xi)G]. \tag{42}$$

For any $\xi \in \mathcal{U}_\alpha(P_\theta)$, set $w(\tau) = (1-\lambda) + \lambda\xi(\tau)$. Then $1 - \lambda \leq w(\tau) \leq 1 - \lambda + \lambda/\alpha$ and $\mathbb{E}_{P_\theta}[w] = (1-\lambda) + \lambda\mathbb{E}[\xi] = 1$. For every measurable trajectory event A, define

$$Q(A) = \int_A w(\tau)\, P(\mathrm{d}\tau).$$

Nonnegativity and normalization of w make Q a probability measure, and $P(A) = 0$ implies $Q(A) = 0$. Thus $Q \ll P$, with $\mathrm{d}Q/\mathrm{d}P = w$, and the displayed bounds place Q in $\mathcal{Q}_{\alpha,\lambda}(P)$. The change-of-measure formula gives

$$\mathbb{E}_Q[G] = \int Gw \, \mathrm{d}P = \mathbb{E}_P[wG].$$

Conversely, when $\lambda > 0$, every Q in this set defines $\xi = (\mathrm{d}Q/\mathrm{d}P_\theta - (1-\lambda))/\lambda$. The bounds on $\mathrm{d}Q/\mathrm{d}P_\theta$ imply $0 \le \xi \le 1/\alpha$, and

$$\mathbb{E}_P[\xi] = \frac{\mathbb{E}_P[\mathrm{d}Q/\mathrm{d}P] - (1-\lambda)}{\lambda} = \frac{1-(1-\lambda)}{\lambda} = 1.$$

Hence $\xi \in \mathcal{U}_\alpha(P)$, and the two constructions are inverse to one another up to P-null sets. For $\lambda = 0$, the set reduces to $Q = P_\theta$. Therefore the two infima are equal, proving Equation equation 14. □

The minimizing measure is explicitly given by $\mathrm{d}Q^*/\mathrm{d}P_\theta = (1-\lambda)+\lambda\xi^*$. For $\lambda = 1$ it concentrates on the lower return tail, including a fractional weight at the quantile when necessary; for $\lambda = 0$ it is the original trajectory law. The result also extends to $\alpha = 1$: $0 \le \xi \le 1$ and $\mathbb{E}[\xi] = 1$ force $\xi = 1$ almost surely, and both the mixed objective and its robust representation reduce to $\mathbb{E}[G]$.

This representation is a consequence of standard CVaR duality specialized to the mixed objective. It concerns distributions absolutely continuous with respect to the specified rollout law, with the stated density bounds; it does not establish coverage of every observation attack or convergence of the PPO updates.

C ALGORITHMS

Algorithm 1: CVaR Q-Iteration

Result: Risk-sensitive Q-function Q

Initialize $Q_0(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$;

for $n = 0, 1, 2, \ldots$ **do**

  For every $\tilde{s} \in \omega(\mathcal{S})$ and $a \in \mathcal{A}$, sort $\{Q_n(m,a) : m \in \mathcal{M}(\tilde{s})\}$ as $q_{n,(1)}(\tilde{s},a) \le \cdots \le q_{n,(K)}(\tilde{s},a)$;

  Update the CVaR policy: $\pi_n(\tilde{s}) = \arg\max_{a \in \mathcal{A}} \frac{1}{k} \sum_{i=1}^{k} q_{n,(i)}(\tilde{s},a)$;

  for $s \in \mathcal{S}$ **do**

    for $a \in \mathcal{A}$ **do**

      $Q_{n+1}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s' \mid s,a) Q_n(s', \pi_n(\omega(s')))$;

Algorithm 2: Belief-Based Risk Reduction(BBRR)-DQN Training

Data: Pretrained Q-network Q_{V}; frozen purifier $\mathcal{P}_\psi$; belief-set size K; tail size k; discount factor γ; total environment steps T_{train}; replay buffer $\mathcal{D}$; target-update period C

Result: Risk-sensitive Q-network Q_θ

Set the empirical CVaR level $\alpha = k/K$

Initialize $Q_\theta \leftarrow Q_{\mathrm{V}}$, $Q_{\theta^-} \leftarrow Q_\theta$, and $\mathcal{D} \leftarrow \emptyset$

Reset the environment and receive the initial perturbed observation $\tilde{s}_0$

Construct $\mathcal{M}_0 = \{m_0^i\}_{i=1}^K$, where $m_0^i = \mathcal{P}_\psi(\tilde{s}_0; T, \xi_0^i)$

for $t = 0, 1, 2, \ldots, T_{\mathrm{train}} - 1$ **do**

  For each $a \in \mathcal{A}$, sort $\{Q_\theta(m_t^i, a)\}_{i=1}^K$ as $q_{t,(1)}(a) \le \cdots \le q_{t,(K)}(a)$ and set $\bar{Q}_\theta^{\mathrm{CVaR}}(\mathcal{M}_t, a) \leftarrow k^{-1} \sum_{i=1}^k q_{t,(i)}(a)$

  Select an action using CVaR-based ε-greedy exploration:

$$a_t = \begin{cases} \text{a uniformly sampled action}, & \text{with probability } \varepsilon_t, \\ \arg\max_{a \in \mathcal{A}} \bar{Q}_\theta^{\mathrm{CVaR}}(\mathcal{M}_t, a), & \text{with probability } 1 - \varepsilon_t. \end{cases}$$

  Execute a_t and receive $(r_t, \tilde{s}_{t+1}, d_t)$

  if $d_t = 0$ **then**

    Construct $\mathcal{M}_{t+1} = \{m_{t+1}^i\}_{i=1}^K$, where $m_{t+1}^i = \mathcal{P}_\psi(\tilde{s}_{t+1}; T, \xi_{t+1}^i)$

  else

    Set $\mathcal{M}_{t+1} = \emptyset$

  Store $(\mathcal{M}_t, a_t, r_t, \mathcal{M}_{t+1}, d_t)$ in $\mathcal{D}$

  if $\mathcal{D}$ *contains enough transitions* **then**

    Sample $\{(\mathcal{M}_j, a_j, r_j, \mathcal{M}_{j+1}, d_j)\}_{j=1}^B$ from $\mathcal{D}$

    for $j = 1, \ldots, B$ **do**

      Sort $\{Q_\theta(m, a_j) : m \in \mathcal{M}_j\}$ and let $\widehat{Q}_j = k^{-1} \sum_{i=1}^k q_{j,(i)}(a_j)$

      if $d_j = 0$ **then**

        For every a, sort $\{Q_\theta(m, a) : m \in \mathcal{M}_{j+1}\}$ and compute $\bar{Q}_\theta^{\mathrm{CVaR}}(\mathcal{M}_{j+1}, a)$

        Set $a_j^* = \arg\max_a \bar{Q}_\theta^{\mathrm{CVaR}}(\mathcal{M}_{j+1}, a)$

        Sort $\{Q_{\theta^-}(m, a_j^*) : m \in \mathcal{M}_{j+1}\}$ and set $\bar{Q}_{\theta^-}^{\mathrm{CVaR}}(\mathcal{M}_{j+1}, a_j^*) = k^{-1} \sum_{i=1}^k q_{j+1,(i)}^-(a_j^*)$

        $y_j \leftarrow r_j + \gamma \bar{Q}_{\theta^-}^{\mathrm{CVaR}}(\mathcal{M}_{j+1}, a_j^*)$

      else

        $y_j \leftarrow r_j$

    Update θ by minimizing $\mathcal{L}_{\mathrm{TD}} = B^{-1} \sum_{j=1}^B \mathrm{Huber}(\widehat{Q}_j, y_j)$

  if $t \bmod C = 0$ **then**

    Set $\theta^- \leftarrow \theta$

  if $d_t = 1$ **then**

    Reset the environment, receive $\tilde{s}_{t+1}$, and construct $\mathcal{M}_{t+1}$ with the frozen purifier

return Q_θ

Algorithm 3: BBRR-DQN Testing

Data: Fine-tuned Q-network $Q_{\theta^\star}$; frozen purifier $\mathcal{P}_\psi$; test environment $\mathcal{E}$; belief-set size K; tail size k; number of episodes L

Result: Mean and standard deviation of episodic returns

Set $Q_{\theta^\star}$ and $\mathcal{P}_\psi$ to evaluation mode, set $\alpha = k/K$, and initialize $\mathcal{R} \leftarrow \emptyset$

for $\ell = 1, 2, \ldots, L$ **do**

  Reset $\mathcal{E}$ to obtain $s_0 \sim \rho_0$ and set $R_\ell \leftarrow 0$

  for $t = 0, 1, 2, \ldots, T_{\max} - 1$ **do**

    Observe the perturbed state $\tilde{s}_t$

    for $i = 1, 2, \ldots, K$ **do**

      Independently sample purifier randomness ξ_t^i and set $m_t^i \leftarrow \mathcal{P}_\psi(\tilde{s}_t; T, \xi_t^i)$

    Set $\mathcal{M}_t \leftarrow \{m_t^i\}_{i=1}^K$

    for $a \in \mathcal{A}$ **do**

      Sort $\{Q_{\theta^\star}(m_t^i, a)\}_{i=1}^K$ as $q_{t,(1)}(a) \le \cdots \le q_{t,(K)}(a)$

      Set $\bar{Q}_{\theta^\star}^{\mathrm{CVaR}}(\mathcal{M}_t, a) \leftarrow k^{-1} \sum_{i=1}^k q_{t,(i)}(a)$

    Set $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \bar{Q}_{\theta^\star}^{\mathrm{CVaR}}(\mathcal{M}_t, a)$

    Execute a_t to obtain (s_{t+1}, r_t, d_t) and set $R_\ell \leftarrow R_\ell + r_t$

    if $d_t = 1$ **then**

      break

  Set $\mathcal{R} \leftarrow \mathcal{R} \cup \{R_\ell\}$

Compute $\overline{R} = L^{-1} \sum_{\ell=1}^L R_\ell$ and $\sigma_R = \sqrt{L^{-1} \sum_{\ell=1}^L (R_\ell - \overline{R})^2}$

return $\overline{R}, \sigma_R$

Algorithm 4: BBRR-PPO Training

Data: Pretrained policy π_{θ_0}; frozen conditional purifier $\mathcal{P}_\psi$; belief-set size K; CVaR level α; risk weight λ; PPO clipping threshold ϵ_{clip}; discount factor γ; number of iterations U

Result: Risk-sensitive policy π_θ

Initialize $\theta \leftarrow \theta_0$ and keep $\mathcal{P}_\psi$ frozen

for $u = 1, 2, \ldots, U$ **do**

  Set $\theta_{\mathrm{old}} \leftarrow \theta$

  Collect a batch of trajectories using the current policy

  for *each environment step* t **do**

    Observe the perturbed state $\tilde{s}_t$

    Construct the belief set $\mathcal{M}_t = \{m_t^i\}_{i=1}^K$, where $m_t^i = \mathcal{P}_\psi(\tilde{s}_t \mid \hat{H}_{t-1}, a_{t-1}; \xi_t^i)$ with independent purifier randomness ξ_t^i

    Evaluate $(\mu_\theta(m_t^i), \sigma_\theta(m_t^i))$ for all $m_t^i \in \mathcal{M}_t$

    Compute the coordinate-wise medians $\widetilde{\mu}_\theta(\mathcal{M}_t)$ and $\widetilde{\sigma}_\theta(\mathcal{M}_t)$

    Form the smoothed policy $\widetilde{\pi}_\theta(\cdot \mid \mathcal{M}_t) = \mathcal{N}(\widetilde{\mu}_\theta(\mathcal{M}_t), \mathrm{diag}(\widetilde{\sigma}_\theta^2(\mathcal{M}_t)))$

    Sample $a_t \sim \widetilde{\pi}_\theta(\cdot \mid \mathcal{M}_t)$, execute a_t, and record r_t, $\mathcal{M}_t$, a_t, and $\log \widetilde{\pi}_{\theta_{\mathrm{old}}}(a_t \mid \mathcal{M}_t)$

    Update the conditional history $\hat{H}_t$ with the restored state used by the conditional purifier

  For each trajectory τ_e, compute $G_e = \sum_t \gamma^t r_t$

  Estimate the lower-tail threshold $\eta \leftarrow \mathrm{Quantile}_\alpha(\{G_e\})$

  Compute $U_e = (1 - \lambda) G_e - (\lambda/\alpha)(\eta - G_e)_+$ and $\widehat{A}_e^{\mathrm{risk}} \leftarrow U_e - E^{-1} \sum_{e'=1}^E U_{e'}$, where E is the number of collected trajectories

  Assign $\widehat{A}_e^{\mathrm{risk}}$ to the transitions in τ_e

  for *each PPO epoch and mini-batch* **do**

    Recompute $\widetilde{\pi}_\theta(a_t \mid \mathcal{M}_t)$ and set $r_t(\theta) = \widetilde{\pi}_\theta(a_t \mid \mathcal{M}_t) / \widetilde{\pi}_{\theta_{\mathrm{old}}}(a_t \mid \mathcal{M}_t)$

    Update θ by maximizing $\mathbb{E}_t \Big[\min\Big(r_t(\theta) \widehat{A}_t^{\mathrm{risk}}, \mathrm{clip}(r_t(\theta), 1 - \epsilon_{\mathrm{clip}}, 1 + \epsilon_{\mathrm{clip}}) \widehat{A}_t^{\mathrm{risk}} \Big) \Big]$.

return π_θ

Algorithm 5: Testing BBRR-PPO with a Frozen Conditional Diffusion Bridge

Data: Fine-tuned policy $\pi_{\theta^\star}$; frozen conditional purifier $\mathcal{P}_\psi$; test environment $\mathcal{E}$; belief-set size K; number of episodes L; maximum episode length $T_{\max}$

Result: Mean and standard deviation of episodic returns

Set $\pi_{\theta^\star}$ and $\mathcal{P}_\psi$ to evaluation mode and initialize $\mathcal{R} \leftarrow \emptyset$

for $e = 1, 2, \ldots, L$ **do**

  Reset $\mathcal{E}$, set $\hat{H}_{-1} \leftarrow \emptyset$, $a_{-1} \leftarrow \mathbf{0}$, and $R_e \leftarrow 0$

  for $t = 0, 1, \ldots, T_{\max} - 1$ **do**

    Observe the perturbed state $\tilde{s}_t$

    for $i = 1, 2, \ldots, K$ **do**

      Independently sample purifier randomness ξ_t^i and set $m_t^i \leftarrow \mathcal{P}_\psi(\tilde{s}_t \mid \hat{H}_{t-1}, a_{t-1}; \xi_t^i)$

    Set $\mathcal{M}_t \leftarrow \{m_t^i\}_{i=1}^K$ and evaluate $(\mu_{\theta^\star}(m_t^i), \sigma_{\theta^\star}(m_t^i))$ for all $m_t^i \in \mathcal{M}_t$

    Compute the coordinate-wise medians $\widetilde{\mu}_{\theta^\star}(\mathcal{M}_t)$ and $\widetilde{\sigma}_{\theta^\star}(\mathcal{M}_t)$

    Form $\widetilde{\pi}_{\theta^\star}(\cdot \mid \mathcal{M}_t) = \mathcal{N}(\widetilde{\mu}_{\theta^\star}(\mathcal{M}_t), \mathrm{diag}(\widetilde{\sigma}^2_{\theta^\star}(\mathcal{M}_t)))$ and sample $a_t \sim \widetilde{\pi}_{\theta^\star}(\cdot \mid \mathcal{M}_t)$

    Execute a_t to obtain (r_t, d_t) and set $R_e \leftarrow R_e + r_t$

    Set $\hat{s}_t$ to the coordinate-wise median of $\mathcal{M}_t$, update $\hat{H}_t$ by appending $\hat{s}_t$, and set the previous action to a_t

    if $d_t = 1$ **then**

      break

  Set $\mathcal{R} \leftarrow \mathcal{R} \cup \{R_e\}$

Compute $\overline{R} = L^{-1} \sum_{e=1}^{L} R_e$ and $\sigma_R = \sqrt{L^{-1} \sum_{e=1}^{L} (R_e - \overline{R})^2}$

return $\overline{R}, \sigma_R$

D EXPERIMENT DETAILS AND ADDITIONAL RESULTS

D.1 EXPERIMENT SETUP JUSTIFICATION

The choice between BBRR-DQN and BBRR-PPO, as well as the corresponding purification method, is determined by the characteristics of the target environment. Both methods instantiate the same framework of diffusion bridge purification, belief-set approximation, and risk-aware decision-making. In Atari, the environment provides high-dimensional pixel observations and a discrete action space; we therefore combine DQN with a diffusion bridge that purifies the current observation. In MuJoCo, the environment provides low-dimensional state vectors and continuous actions; we therefore combine PPO with a conditional diffusion bridge that incorporates temporal context. More specifically, the action space motivates the choice of the underlying RL algorithm, while the observation representation, state semantics, attack mechanism, and computational cost determine which purification method is appropriate. Thus, the difference between the two implementations is not solely an algorithmic choice, but also an adaptation to their respective environments.

For the Atari environments, BBRR-DQN purifies the current perturbed observation without an additional history-conditioning mechanism. Under the bounded pixel-wise attacks considered here, the observed image typically retains substantial spatial structure and information about the underlying scene, even when the perturbation misleads the policy. This information provides a basis for learning to restore plausible clean observations from the current input. Explicitly modeling a history of high-dimensional images would add computational and modeling costs. We therefore use the current observation to construct the purified belief set and account for residual restoration uncertainty through risk-aware action evaluation. This choice concerns explicit conditioning of the purifier; it does not imply that temporal information is generally unhelpful in Atari or exclude temporal information already encoded in the observation representation.

For the MuJoCo environments, BBRR-PPO addresses perturbations that directly alter state-vector components such as joint positions and velocities. A perturbed vector may remain a plausible observation while indicating a different physical configuration or motion. Consequently, plausibility under the clean-state distribution alone may not distinguish the actual state from other possible states. The conditional diffusion bridge therefore uses the previously restored states and the pre-

ceding executed action, namely $c_t = (\hat{H}_{t-1}, a_{t-1})$, as additional context. This context provides a learned dynamical prior that helps resolve ambiguities in the current observation and favors restorations consistent with the preceding trajectory and action. Conditioning on a history of compact state vectors also makes temporal modeling more tractable than directly modeling a history of pixel observations. At inference time, the history is updated with the coordinate-wise median of the purified belief set, so the method does not require access to past clean states.

Accordingly, the recommended configuration depends on the environment rather than on the RL algorithm alone. Pixel-based environments with discrete actions and attacks that largely preserve scene semantics are suited to the BBRR-DQN configuration with current-observation purification. Continuous-control environments with compact state vectors, where perturbations can change the apparent physical state, are suited to the BBRR-PPO configuration with history-conditioned purification. These are design choices for the settings studied here, rather than inherent restrictions of DQN or PPO. A history-conditioned bridge may also benefit a visual environment when the current observation is insufficient to identify the underlying state, while a bridge without historical conditioning remains applicable to state-vector observations when the current state is sufficiently informative.

D.2 EXPERIMENT SETUP

Atari. We evaluate all methods on four Atari games: *Pong*, *Freeway*, *BankHeist*, and *RoadRunner*. These environments exhibit diverse visual dynamics, discrete action spaces, and reward scales, and therefore provide a broad evaluation of robustness in high-dimensional visual control. For a fair comparison, all methods use the same `NoFrameskip-v4` environments, environment wrappers, game-specific action restrictions, and observation preprocessing pipeline. Each resulting observation is centrally cropped using the game-specific crop offset, converted to grayscale, resized to 84×84. The agent receives a single observation frame of shape $1 \times 84 \times 84$; no temporal frame stacking is used. Before being passed to the Q-network and diffusion bridge, pixel values are converted from their original integer range $[0, 255]$ to floating-point values in $[0, 1]$. During adversarial evaluation, the attack is applied only to the normalized observation received by the agent. The perturbed observation is projected onto the ℓ_∞-bounded neighborhood of the clean observation and clipped to the valid pixel range $[0, 1]$. The underlying Atari state, transition dynamics, action semantics, and reward function are not modified. All methods are evaluated under the same episode and random seed protocol, and the reported episodic returns are computed using the original unclipped game rewards.

MuJoCo. We evaluate all methods on four MuJoCo continuous-control tasks: *HalfCheetah*, *Hopper*, *Walker2d*, and *Ant*. In *HalfCheetah*, a planar robot with six actuated joints must run forward as quickly as possible. *Hopper* requires a planar one-legged robot to hop forward using three actuated joints, while *Walker2d* extends the task to six-joint bipedal locomotion. *Ant* requires a three-dimensional quadruped to coordinate eight actuated joints for forward movement. These environments cover diverse robot morphologies, locomotion dynamics, and continuous action dimensions, providing a broad test of robustness in continuous-control tasks. For a fair comparison, every method uses the same MuJoCo environments, wrappers, state preprocessing, and episode protocol. At each time step, the agent receives a vector observation containing the robot's proprioceptive state, including body and joint positions and velocities, and outputs continuous joint torques. During adversarial evaluation, perturbations are applied only to the observation received by the agent and are constrained by the environment-specific l_∞ budget; the underlying simulator state, transition dynamics, and reward function remain unchanged. The reported episodic returns are computed from the original environment rewards.

Training and Testing Details for BBRR-DQN. We initialize the Q network using a pretrained vanilla dueling DQN. The pretrained diffusion bridge is fixed throughout BBRR-DQN fine-tuning. The formal experiments fine-tune the Q network for $300{,}000$ environment frames using Adam with a learning rate of 5×10^{-6} and $\epsilon_{\text{Adam}} = 1.5 \times 10^{-4}$. We use a discount factor of $\gamma = 0.99$, a mini-batch size of 32, gradient clipping with maximum norm 10, and a replay buffer containing up to $200{,}000$ transitions. During fine-tuning, we generate each perturbed observation using a 10-step projected-gradient attack with random initialization. The training attack radius is linearly increased from $1/255$ to $10/255$. The frozen diffusion bridge performs one reverse step and produces $K = 4$

independently sampled belief states. The belief states are quantized to 8-bit observations before replay storage and Q-value computation, so action selection and replay optimization use the same belief-state representation. For each action, BBRR-DQN sorts the four belief-state Q-values and averages the lowest $k = 2$, corresponding to an empirical lower-tail CVaR level of $\alpha = k/K = 0.5$. During data collection, we use ϵ-greedy exploration, with the exploration probability scheduled from 0.05 to 0.01 over the fine-tuning period. At test time, all network and diffusion-bridge parameters are frozen. For every attacked observation, the fixed bridge generates $K = 4$ purified belief states using one reverse step, and the executed action maximizes the mean of the lowest $k = 2$ belief-state Q-values. We evaluate six random seeds with 20 episodes per seed and report the mean and standard deviation of episodic returns.

Training and Testing Details for BBRR-PPO. We initialize the BBRR-PPO policy from a pretrained vanilla PPO checkpoint. The conditional diffusion bridge is pretrained separately and remains frozen throughout policy fine-tuning and evaluation. For each perturbed observation, the bridge generates $K = 5$ purified belief states conditioned on the restored state history and the previously executed action. The policy is evaluated on all belief states, and coordinate-wise medians are computed for both the action means and standard deviations. Actions are sampled from the resulting belief-smoothed Gaussian policy. Risk sensitivity is introduced at the trajectory level. Within each PPO update, we compute the empirical lower-tail quantile of the discounted episode returns using $\alpha = 0.2$. The risk-adjusted trajectory utility is $U(G; \eta_\alpha) = (1 - \lambda)G - \frac{\lambda}{\alpha}(\eta_\alpha - G)_+$, where η_α is the empirical lower-α quantile and $\lambda = 0.3$. The centered trajectory utilities provide the risk-adjusted advantages used in the clipped PPO objective. At test time, both the fine-tuned policy and the conditional diffusion bridge are frozen. We use $K = 5$ belief states and sample actions from the belief-smoothed Gaussian policy. Each episode is limited to 1000 steps, and returns are computed from the original MuJoCo rewards.

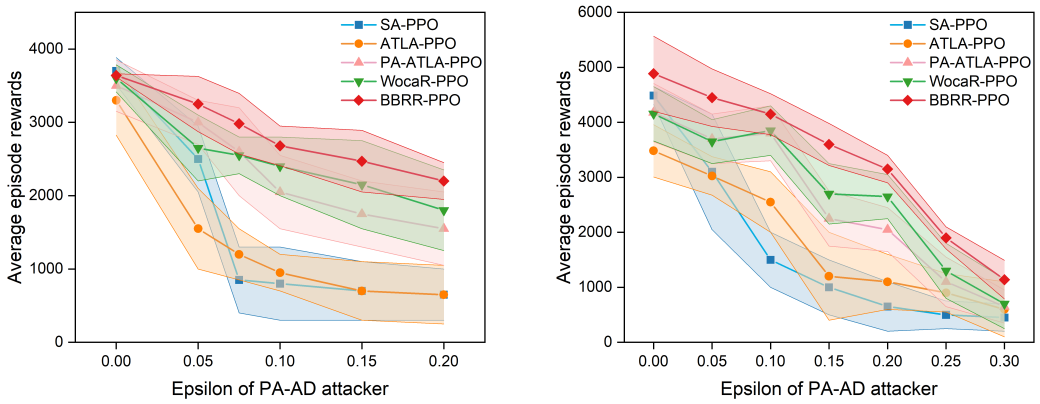
Hardware configuration. All training and testing are conducted on a server equipped with two AMD EPYC 7402 CPU sockets, providing 48 physical cores and 96 logical threads, together with four NVIDIA GeForce RTX 4090 GPUs and one NVIDIA A100-SXM4 GPU. Each RTX 4090 provides approximately 24 GB of GPU memory, while the A100 provides 80 GB. The server has approximately 503 GiB of system memory.

D.3 MORE EXPERIMENT RESULTS

D.3.1 MORE BASELINE RESULTS

Table 6: Clean and adversarial episodic returns on four Atari environments. Results are mean ± standard deviation over six seeds, with 20 episodes per seed. Best results are bolded, and gray rows denote BBRR-DQN.

Env	Model	Natural Reward	PGD			MinBest			PA-AD		
			$\epsilon = 1/255$	$\epsilon = 10/255$	$\epsilon = 20/255$	$\epsilon = 1/255$	$\epsilon = 10/255$	$\epsilon = 20/255$	$\epsilon = 1/255$	$\epsilon = 10/255$	$\epsilon = 20/255$
Pong	DQN	21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-18.2 ± 2.3	-21.0 ± 0.0	-21.0 ± 0.0
	SA-DQN	21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0
	WocaR-DQN	21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-19.5 ± 1.4	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0
	Radial-DQN	21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-20.3 ± 0.2	-21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0
	DP-DQN	19.9 ± 0.3	20.2 ± 0.2	8.4 ± 1.6	-0.7 ± 3.0	20.3 ± 0.4	16.9 ± 2.0	7.6 ± 1.1	19.9 ± 1.4	11.7 ± 1.2	2.3 ± 1.8
	BBRR-DQN (Ours)	21.0 ± 0.0	$\mathbf{21.0 \pm 0.0}$	$\mathbf{20.6 \pm 0.8}$	$\mathbf{20.8 \pm 0.5}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{20.6 \pm 0.3}$	$\mathbf{20.6 \pm 0.5}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{20.3 \pm 1.2}$	$\mathbf{18.2 \pm 1.8}$
Freeway	DQN	33.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SA-DQN	30.0 ± 0.0	30.0 ± 0.0	18.4 ± 1.6	17.2 ± 2.5	29.1 ± 0.8	20.9 ± 1.6	18.0 ± 2.2	9.5 ± 3.8	0.0 ± 0.0	0 ± 0
	WocaR-DQN	32.6 ± 0.4	33.3 ± 0.6	24.3 ± 3.8	22.8 ± 1.6	33.0 ± 0.1	21.4 ± 2.9	20.0 ± 1.2	23.3 ± 2.9	7.2 ± 3.6	0.0 ± 0.0
	Radial-DQN	33.2 ± 0.8	33.1 ± 0.6	22.0 ± 1.7	21.7 ± 1.4	33.1 ± 0.4	20.3 ± 2.0	19.8 ± 1.2	$\mathbf{29.2 \pm 0.6}$	5.9 ± 1.3	0 ± 0
	DP-DQN	29.4 ± 0.5	29.4 ± 0.7	26.8 ± 2.5	24.9 ± 2.4	29.3 ± 0.5	25.9 ± 1.4	23.9 ± 2.67	28.8 ± 1.7	$\mathbf{25.0 \pm 2.8}$	$\mathbf{20.0 \pm 4.2}$
	BBRR-DQN (Ours)	34.0 ± 0.0	$\mathbf{34.0 \pm 0.0}$	$\mathbf{33.8 \pm 0.4}$	$\mathbf{33.5 \pm 0.2}$	$\mathbf{34.0 \pm 0.0}$	$\mathbf{33.8 \pm 0.4}$	$\mathbf{33.6 \pm 0.2}$	0 ± 0	0 ± 0	0 ± 0
BankHeist	DQN	1302.2 ± 3.4	54.2 ± 20.0	0.0 ± 0.0	0.0 ± 0.0	210.8 ± 79.3	0.0 ± 0.0	0.0 ± 0.0	213.8 ± 111.4	0.0 ± 0.0	0.0 ± 0.0
	SA-DQN	1236.0 ± 1.4	$\mathbf{1245.1 \pm 10.7}$	637.4 ± 49.3	372.9 ± 87.3	1148.2 ± 64.3	725.3 ± 41.3	417.2 ± 52.8	$\mathbf{1054.2 \pm 11.6}$	0.0 ± 0.0	0.0 ± 0.0
	WocaR-DQN	1220.3 ± 10.6	1216.2 ± 20.6	719.3 ± 43.8	492.4 ± 63.2	1192.3 ± 28.4	692.3 ± 76.2	502.4 ± 98.2	1096.5 ± 39.4	256.1 ± 97.4	0.0 ± 0.0
	Radial-DQN	1178.5 ± 9.5	1164.2 ± 10.6	1031.0 ± 22.4	516.9 ± 16.2	982.7 ± 38.3	864.7 ± 21.4	538.1 ± 62.3	549.6 ± 92.4	277.3 ± 132.4	0.0 ± 0.0
	DP-DQN	1021.5 ± 4.9	982.6 ± 9.5	857.4 ± 102.7	628.3 ± 143.4	1032.6 ± 42.3	792.4 ± 75.4	652.8 ± 162.3	862.5 ± 62.3	$\mathbf{602.3 \pm 78.2}$	$\mathbf{402.7 \pm 147.2}$
	BBRR-DQN (Ours)	1244.5 ± 18.2	1229.4 ± 4.2	$\mathbf{1135.3 \pm 25.8}$	$\mathbf{764.6 \pm 65.2}$	$\mathbf{1219.3 \pm 14.3}$	$\mathbf{1157.3 \pm 25.7}$	$\mathbf{836 \pm 163.2}$	0 ± 0	0 ± 0	0 ± 0
RoadRunner	DQN	45527 ± 2893	0 ± 0	0 ± 0	0 ± 0	14962 ± 6431	0 ± 0	0 ± 0	842 ± 41	0 ± 0	0 ± 0
	SA-DQN	44638 ± 2367	43970 ± 975	1273 ± 224	0 ± 0	39736 ± 2315	2521 ± 926	0 ± 0	38432 ± 3574	1382 ± 427	0 ± 0
	WocaR-DQN	44156 ± 2279	45298 ± 2834	1529 ± 492	893 ± 325	41783 ± 2834	3826 ± 632	427 ± 63	42891 ± 3621	1238 ± 522	624 ± 382
	Radial-DQN	44595 ± 1143	44365 ± 4537	890 ± 336	960 ± 499	$\mathbf{45395 \pm 5131}$	3095 ± 552	235 ± 85	$\mathbf{43712 \pm 6964}$	1429 ± 614	917 ± 263
	DP-DQN	42100 ± 1800	38600 ± 3200	11800 ± 2600	2100 ± 900	36200 ± 4100	7600 ± 2100	1300 ± 700	31800 ± 5200	5200 ± 1800	900 ± 500
	BBRR-DQN (Ours)	49860 ± 976	$\mathbf{50380 \pm 5872}$	$\mathbf{23260 \pm 2656}$	$\mathbf{18340 \pm 1121}$	42140 ± 2050	$\mathbf{20510 \pm 2859}$	$\mathbf{18620 \pm 1530}$	43680 ± 1830	$\mathbf{25500 \pm 4496}$	$\mathbf{17710 \pm 2731}$



(a) Hopper (b) Walker2d

Figure 1: Performance of BBRR-PPO and baseline methods under PAAD attacks across different perturbation budgets (ϵ).

D.3.2 Training and Testing Overhead.

Table 7 summarizes the computational overhead of BBRR relative to the evaluated baselines. On Atari, BBRR-DQN takes 18 hours to train, which is substantially less than DP-DQN (49 hours) and close to WocaR-DQN (18 hours). The training time of BBRR-DQN includes pretraining the vanilla DQN policy (12 hours), training the diffusion bridge (3 hours), and fine-tuning DQN (3 hours). Its testing throughput is 79 FPS, lower than the other standard DQN-based defenses (502–641 FPS) but considerably higher than DP-DQN (6.6 FPS). By employing a single-step DDIM sampler for fast diffusion process, the time cost of BBRR-DQN remains practical and is markedly lower than that of the diffusion purification baseline DP-DQN. On the MuJoCo benchmark, BBRR-PPO requires 7.3 training hours in total, including the pretrained vanilla PPO policy (2.3 hours), the diffusion bridge (3 hours) and fine-tuned PPO (2 hours). This is higher than Vanilla-PPO, SA-PPO, WocaR-PPO, and Radial-PPO, but remains comparable to the more expensive ATLA-PPO (5.6 hours). The main cost appears during inference: BBRR-PPO achieves 134 FPS, whereas the baselines run at 607-1335 FPS. This reduction is expected because BBRR-PPO uses the diffusion bridge model to generate multiple plausible belief states before selecting an action. Overall, BBRR trades additional training and inference computation for robust decision-making under observation perturbations.

Environment	Model	Training (hours)	Testing (FPS)
Hopper	Vanilla-PPO	2.3	607
	SA-PPO	3.0	763
	ATLA-PPO	5.6	982
	WocaR-PPO	2.5	1335
	Radial-PPO	3.2	826
	BBRR-PPO (Ours)	2.3+3+2	134

(a) MuJoCo

Environment	Model	Training (hours)	Testing (FPS)
Pong	Vanilla-DQN	12	641
	SA-DQN	35	502
	Radial-DQN	17	584
	WocaR-DQN	18	635
	DP-DQN	49	6.6
	BBRR-DQN (Ours)	12+3+3	79

(b) Atari

Table 7: Training and testing overhead of the evaluated methods.

D.4 Limitations

Our framework has several limitations. First, the diffusion bridge and belief state construction introduce additional computational overhead because multiple purified states may need to be generated and evaluated at each test step. This overhead can limit the applicability of BBRR in latency sensitive or resource constrained settings. Nevertheless, our implementation uses a single step DDIM sampler for diffusion bridge generation, which keeps the additional overhead within an acceptable range. Second, although we evaluate BBRR against a diverse set of established attacks, our experiments do not fully cover adaptive adversaries that jointly optimize their perturbations against the

diffusion bridge, belief state construction, and policy. The robustness of BBRR under such attacks therefore remains an open question. Third, our evaluation mainly considers numerical observation perturbations, whereas real world vision based agents may encounter semantic changes such as variations in brightness, contrast, illumination, occlusion, or weather. Extending BBRR to these more realistic perturbations is an important direction for future work.