CAUSAL INFERENCE Q-NETWORK: TOWARD RESILIENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning (DRL) has demonstrated impressive performance in various gaming simulators and real-world applications. In practice, however, a DRL agent may receive faulty observation by abrupt interferences such as black-out, frozen-screen, and adversarial perturbation. How to design a resilient DRL algorithm against these rare but mission-critical and safety-crucial scenarios is an important yet challenging task. In this paper, we consider a generative DRL framework training with an auxiliary task of observational interferences such as artificial noises. Under this framework, we discuss the importance of the causal relation and propose a causal inference based DRL algorithm called causal inference Q-network (CIQ). We evaluate the performance of CIQ in several benchmark DRL environments with different types of interferences as auxiliary labels. Our experimental results show that the proposed CIQ method could achieve higher performance and more resilience against observational interferences.

1 INTRODUCTION

Deep reinforcement learning (DRL) methods have shown enhanced performance, gained widespread applications (Mnih et al., 2015; 2016; Silver et al., 2017), and improved robot learning (Gu et al., 2017) in navigation systems (Tai et al., 2017; Nagabandi et al., 2018). However, most successful demonstrations of these DRL methods are usually trained and deployed under well-controlled situations. In contrast, real-world use cases often encounter inevitable observational uncertainty (Grigorescu et al., 2020; Hafner et al., 2018; Moreno et al., 2018) from an external attacker (Huang et al., 2017) or noisy sensor (Fortunato et al., 2018; Lee et al., 2018). For examples, playing online video games may experience sudden black-outs or frame-skippings due to network instabilities, and driving on the road may encounter temporary blindness when facing the sun. Such an abrupt interference on the observation could cause serious issues for DRL algorithms. Unlike other machine learning tasks that involve only a single mission at a time (e.g., image classification), an RL agent has to deal with a dynamic (Schmidhuber, 1992) or even learn from latent states with generative models (Schmidhuber, 1991; Jaderberg et al., 2017; Ha & Schmidhuber, 2018; Hafner et al., 2018; Lynch et al., 2020) to anticipate future rewards in complex environments. Therefore, DRL-based systems are likely to propagate and even enlarge risks (e.g., delay and noisy pulsed-signals on sensor-fusion (Yurtsever et al., 2020; Johansen et al., 2015)) induced from the uncertain interference.

In this paper, we investigate the *resilience* ability of an RL agent to withstand unforeseen, rare, adversarial and potentially catastrophic interferences, and to recover and adapt by improving itself in reaction to these events. We consider a resilient generative RL framework with observational interferences as an auxiliary task. At each time, the agent's observation is subjected to a type of sudden interference at a predefined possibility. Whether or not an observation has interfered is referred to as the interference label.

Specifically, to train a resilient agent, we provide the agent with the interference labels during training. For instance, the labels could be derived from some uncertain noise generators recording whether the agent observes an intervened state at the moment as a binary causation label. By applying the labels as an *intervention* into the environment, the RL agent is asked to learn a binary causation label and embed a latent state into its model. However, when the trained agent is deployed in the field (i.e., the testing phase), the agent only receives the interfered observations but is agnostic to interference labels and needs to act resiliently against the interference.

For an RL agent to be resilient against interference, the agent needs to diagnose observations to make the correct inference about the reward information. To achieve this, the RL agent has to reason about what leads to desired rewards despite the irrelevant intermittent interference. To equip an RL agent with this reasoning capability, we exploit the causal inference framework. Intuitively, a causal inference model for observation interference uses an unobserved confounder (Pearl, 2009; 2019; 1995b; Saunders et al., 2018; Bareinboim et al., 2015; Zhang et al., 2020b) to capture the effect of the interference on the rewards (outcomes) collected from the environment. In recent works, RL is also showing additional benefits incorporating generative causal modeling, such as providing interpretability (Madumal et al., 2020), treatment estimation (Zhang & Bareinboim, 2020), imitation learning (Zhang et al., 2020c), enhanced invariant prediction (Zhang et al., 2020a), and generative model for transfer learning Killian et al. (2020).

When such a confounder is available, the RL agent can focus on the confounder for relevant reward information and make the best decision. As illustrated in Figure 1, we propose a causal inference based DRL algorithm termed causal inference Q-network (CIQ). During training, when the interference labels are available, the CIQ agent will implicitly learn a causal inference model by embedding the confounder into a latent state. At the same time, the CIQ agent will also train a Q-network on the latent state for decision making. Then at testing, the CIQ agent will make use of the learned model to estimate the confounding latent state and the interference label. The history of latent states is combined into a causal inference state, which captures the relevant information for the Q-network to collect rewards in the environment despite of the observational interference.

In this paper, we evaluate the performance of our method in four environments: 1) Cartpole-v0 – the continuous control environment (Brockman et al., 2016); 2) the 3D graphical Banana Collector (Juliani et al., 2018)); 3) an Atari environment LunarLander-v2 (Brockman et al., 2016), and 4) pixel Cartpole – visual learning from the pixel inputs of Cartpole. For each of the environments, we consider four types of interference: (a) black-out, (b) Gaussian noise, (c) frozen screen, and (d) adversarial attack.

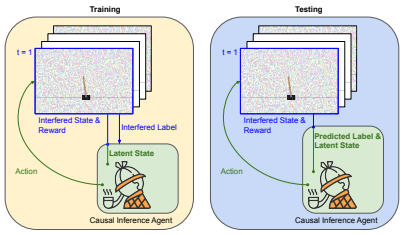


Figure 1: The proposed causal inference Q-network (CIQ) training and test framework, where the latent state is an unobserved (hidden) confounder variable. We refer the readers to Figure 2 for detailed descriptions.

In the testing phase mimicking the practical scenario that the agent may have interfered observations but is unaware of the true interference labels (i.e., happens or not), the results show that our CIQ method can perform better and more resilience against all the four types of interference. Furthermore, to benchmark the level of resilience of different RL models, we propose a new robustness measure, called CLEVER-Q, to evaluate the robustness of Q-network based RL algorithms. The idea is to compute a lower bound on the observation noise level such that the greedy action from the Q-network will remain the same against any noise below the lower bound. According to this robustness analysis, our CIQ algorithm indeed achieves higher CLEVER-Q scores compared with the baseline methods.

The main contributions of this paper include 1) a framework to evaluate the resilience of DRL methods under abrupt observational interferences; 2) the proposed CIQ architecture and algorithm towards training a resilient DRL agent, and 3) an extreme-value theory based robustness metric (CLEVER-Q) for quantifying the resilience of Q-network based RL algorithms.

2 Related Works

Causal Inference for Generative Reinforcement Learning: Causal inference (Greenland et al., 1999; Pearl, 2009; Pearl et al., 2016; Pearl, 2019; Robins et al., 1995) has been used to empower the learning process under noisy observation and have better interpretability on deep learning models (Shalit et al., 2017; Louizos et al., 2017), also with efforts (Jaber et al., 2019; Forney et al., 2017; Bareinboim et al., 2015) on causal online learning and bandit methods. Defining causation and applying causal inference framework to DRL still remains relatively unexplored. Recent works (Lu et al., 2018; Tennenholtz et al., 2019) study this problem by defining action as one kind of intervention

and calculating the treatment effects on the action. In contrast, we introduce causation into generative DRL by applying extra noisy and uncertain inventions. Different from the aforementioned approaches, we leverage the causal effect of observational interferences on states, and design an end-to-end structure for learning a *causal-observational* representation evaluating treatment effects on rewards.

Adversarial Perturbation: An intensifying challenge against deep neural network based systems is adversarial perturbation for making incorrect decisions. Many gradient-based noise-generating methods (Goodfellow et al., 2015; Huang et al., 2017) have been conducted for misclassification and mislead an agent's output action. As an example of using DRL model playing Atari games, an adversarial attacker (Lin et al., 2017; Yang et al., 2020c) could jam in a timely and barely detectable noise to maximize the prediction loss of a Q-network and cause massively degraded performance.

Partially Observable Markov Decision Processes (POMDPs): Our resilient RL framework can be viewed as a POMDP with interfered observations. Belief-state methods are available for simple POMDP problems (e.g., plan graph and the tiger problem (Kaelbling et al., 1998)), but no provably efficient algorithm is available for general POMDP settings (Papadimitriou & Tsitsiklis, 1987; Gregor et al., 2018). Recently, Igl et al. (2018) have proposed a DRL approach for POMDPs by combining variational autoencoder and policy-based learning, but this kind of methods do not consider the interference labels available during training in our resilient RL framework.

3 Resilient Reinforcement Learning

In this section, we formally introduce our resilient RL framework and provide an extreme-value theory based metric called CLEVER-Q for measuring the robustness of DQN-based methods.

We consider a sequential decision-making problem where an agent interacts with an environment. At each time t, the agent gets an observation x_t, e.g. a frame in a video environment. As in many RL domains (e.g., Atari games), we view $s_t = (x_{t-M+1}, \ldots, x_t)$ to be the state of the environment where M is a fixed number for the history of observations. Given a stochastic policy π, the agent chooses an action $a_t \sim \pi(s_t)$ from a discrete action space based on the observed state and receives a reward r_t from the environment. For a policy π, define the Q-function $Q^\pi(s,a) = \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s,\, a_0 = a, \pi\big]$ where $\gamma \in (0,1)$ is the discount factor. The agent's goal is to find the optimal policy π^* that achieves the optimal Q-function given by $Q^*(s,a) = \max_\pi Q^\pi(s,a)$.

3.1 Resilience base on an Interventional Perspective

To evaluate the resilience ability of RL agents, we introduce additional interference as auxiliary information (as illustrated in Fig 1) as an empirical process (Pearl, 2009; Louizos et al., 2017) for observation. Given a type of interference $\mathcal{I}$, the agent's observation becomes:

$$x'_t = F^{\mathcal{I}}(x_t, i_t) = i_t \times \mathcal{I}(x_t) + (1 - i_t) \times x_t \tag{1}$$

where $i_t \in \{0, 1\}$ is the label indicating whether the observation is interfered at time t or not, and $\mathcal{I}(x_t)$ is the interfered observation.

We assume that interference labels i_t follow an i.i.d. Bernoulli process with a fixed interference probability $p^{\mathcal{I}}$ as a noise level.[1] For example, when $p^{\mathcal{I}}$ equals to 10%, each observational state has a 10% chance to be intervened under a perturbation.

In this work, we consider the original observations, as illustrated in Figure 5 (a), under four types of interference as described below.

Gaussian Noise. Gaussian noise or white noise is a common interference to sensory data (Osband et al., 2019; Yurtsever et al., 2020). The interfered observation becomes $\mathcal{I}(x_t) = x_t + n_t$ with a zero-mean Gaussian noise n_t. The noise variance is set to be the variance of all recorded states as illustrated in Figure 5 (b).

Adversarial Observation. Following the standard adversarial RL attack setting, we use fast gradient sign method (FGSM) (Szegedy et al., 2014) to generate adversarial patterns against the DQN

[1] The i.i.d. assumption could be extended to a Markovian dynamic interference model. We show experiments with dynamic interference in Appendix C.7.

loss (Huang et al., 2017) as illustrated in Figure S5 (c). The observation is given by $\mathcal{I}(x_t) = x_t + \epsilon\,\text{sign}\left(\nabla_{x_t} Q(x_t, y; \theta)\right)$ where y is the optimal action by weighting over possible actions.

Observation Black-Out. Off-the-shelf hardware can affect the entire sensor networks as a sensing background (Yurtsever et al., 2020) over-shoot with $\mathcal{I}(x_t) = 0$ (Yan et al., 2016). This perturbation is realistic owing to overheat hardware and losing the observational information of sensors.

Frozen Frame. Lagging and frozen frame(s) (Kalashnikov et al., 2018) often come from limited data communication bottleneck bandwidth. A frozen frame is given by $\mathcal{I}(x_t) = x_{t-1}$. If the perturbation is constantly present, the frame will remain the first frozen frame since the perturbation happened.

3.2 Resilient Reinforcement Learning Framework

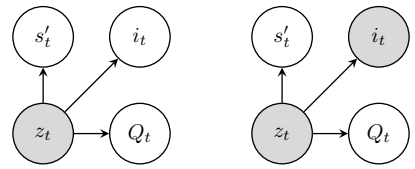
With observational interference, instead of the actual state s_t, the agent only observes $s'_t = (x'_{t-M+1}, \ldots, x'_t)$. The agent now needs to choose its actions $a_t \sim \pi(s'_t)$ based on the interfered observation. The resilient RL objective for the agent is to find a policy π to maximize rewards in this environment under observational interference. Under the resilient framework, the goal of a Q-learning based agent is to learn the relation between s'_t and Q_t where $Q_t(a) = \max_\pi \mathbb{E}\big[\sum_{\tau=t}^{\infty} \gamma^{(\tau-t)} r_\tau | s'_t,\, a_t = a, \pi\big]$ denotes the Q-values given the interfered observation s'_t at time t.

From the RL model and the observation model of Eq. (1), the relation among the observation s'_t, Q-values Q_t, and interference i_t can be described by a causal graphical model (CGM) in Figure 2. In the CGM, $z_t = (s_t, i_{t-M+1}, \ldots, i_t)$ includes the actual state s_t of the system together with the interference labels which casually affects all s'_t, Q_t, and i_t. Note that z_t is not observable to the agent due to the interference; z_t could be viewed as a hidden confounder in causal inference.

Since only the interfered observation s'_t is available, the interference label i_t is also non-observable in evaluating the resilience ability of an agent. However, the interference information is often accessible in the training phase, such as the use of a navigation simulator recorded with noisy augmentation (Grigorescu et al., 2020) for simulating interference in the training environment. We will discuss in the next subsection the benefit of utilizing the interference labels to improve learning efficiency.

3.3 Learning with Interference Labels

The goal of a resilient RL agent is to learn $P(Q_t|s'_t)$ to infer the Q-value Q_t based on the interfered observation s'_t. Note that one can compute $P(Q_t|s'_t)$ by determining the joint distribution $P(z_t, s'_t, i_t, Q_t)$ of all variables in the CGM in Figure 2. Despite the presence of the hidden variable z_t, similar to causal inference with hidden confounders (Louizos et al., 2017), estimating the joint distribution $P(z_t, s'_t, i_t, Q_t)$ could be done efficiently when the agent is provided the interference labels i_t during training. On the other hand, if only the observation s'_t is available, the agent can only directly estimate $P(Q_t|s'_t)$, which is less efficient in terms of training sample usage.



(a) Training phase (b) Testing phase

Figure 2: Causal graphical model (CGM) for the training phase (a) and the testing phase (b). White nodes s'_t and Q_t are observable. Node $z_t = (s_t, i_{t-M+1}, \ldots, i_t)$, colored by grey, is not observable. Node i_t, colored by grey in (b), is only observable during training.

We provide the interference type $\mathcal{I}$ and the interference labels i_t to efficiently train a resilient RL agent as shown in Figure 2(b); however, in the actual testing environment, the agent only has access to the interfered observations x'_t as in Figure 2(a).

3.4 Causal Inference Q-Network

With the observable variables (s'_t, i_t, Q_t) in Figure 2(a) during training, we aim to learn a model to infer the Q-values by estimating the joint distribution $P(z_t, s'_t, i_t, Q_t)$. Despite the underlying dynamics in the RL system, when we view the interference as a treatment, the CGM in Figure 2(a) resembles some common causal inference models with binary treatment information and hidden

confounders (Louizos et al., 2017). In this kind of causal inference problems, by leveraging on the binary property for treatment information, TARNet (Shalit et al., 2017) and CEVAE (Louizos et al., 2017) introduced a binary switching neural architecture to efficiently learn latent models for causal inference.

Inspired by the switching mechanism for causal inference, we propose the causal inference Q-network, referred as CIQ, that maps the interfered observation s'_t into a latent state z_t, makes proper inferences about the interference condition i_t, and adjusts its policy based on the estimated interference.

We approximate the latent state by a neural network $\tilde{z}_t = f_1(x'_t; \theta_1)$. From the latent state, we generate the estimated interference label $\tilde{i}_t \sim p(\tilde{i}_t|z_t) = f_I(z_t; \phi)$. We denote $s_t^{CI} = (\tilde{z}_{t-M+1}, \tilde{i}_{t-M+1}, \ldots, \tilde{z}_t, \tilde{i}_t)$ to be the causal inference state. As discussed in the previous subsection, the causal inference state acts as a confounder between the interference and the reward. Therefore, instead of using the interfered state s'_t, the causal inference state s_t^{CI} contains more relevant information for the agent to maximize rewards. Using the causal inference state helps focus on meaningful and informative details even under interference.

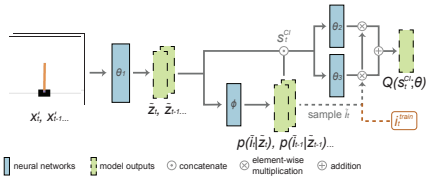


Figure 3: CIQ architecture. The notation i_t^{train} denotes the inference label available during training, whereas $\tilde{i}_t$ is sampled during inference as i_t is unknown.

With the causal inference state s_t^{CI}, the output of the Q-network $Q(s_t^{CI}; \theta)$ is set to be switched between two neural networks $f_2(s_t^{CI}; \theta_2)$ and $f_3(s_t^{CI}; \theta_3)$ by the interference label. Such a switching mechanism prevents our network from over-generalizing the causal inference state. During training, switching between the two neural networks is determined by the training interference label i_t^{train}. We assume that the true interference label is available in the training phase so $i_t^{\text{train}} = i_t$. In the testing, when i_t is not available, we use the predicted interference label $\tilde{i}_t$ as the switch to decide which of the two neural networks to use.

All the neural networks f_1, f_2, f_3, f_I have two fully connected layers[2] with each layer followed by the ReLU activation except for the last layer in f_2, f_3 and f_I. The overall CIQ model is shown in Figure 3 and $\theta = (\theta_1, \theta_2, \theta_3, \phi)$ denotes all its parameters. Note that, as common practice for discrete action spaces, the Q-network output $Q(s_t^{CI}; \theta)$ is an $\mathcal{A}$-dimensional vector where $\mathcal{A}$ is the size of the action space, and each dimension represents the value for taking the corresponding action.

Finally, we train the CIQ model $Q(s'_t; \theta)$ end-to-end by the DQN algorithm with an additional loss for predicting the interference label. The overall CIQ objective function is defined as:

$$
\begin{aligned}
L^{\text{CIQ}}(\theta_1, \theta_2, \theta_3, \phi) &= i_t^{\text{train}} \cdot L^{\text{DQN}}(\theta_1, \theta_2, \phi) \\
&+ (1 - i_t^{\text{train}}) \cdot L^{\text{DQN}}(\theta_1, \theta_3, \phi) + \lambda \cdot (i_t^{\text{train}} \log p(\tilde{i}_t|\tilde{z}_t; \theta_1, \phi) \\
&+ (1 - i_t^{\text{train}}) \log(1 - p(\tilde{i}_t|\tilde{z}_t; \theta_1, \phi))),
\end{aligned} \tag{2}
$$

where λ is a scaling constant and is set to 1 for simplicity. Due to the design of the causal inference state and the switching mechanism, we will show that CIQ can perform resilient behaviors against the observation interferences. Next, we introduce how to quantify the robustness of a Q-network under noisy observation in next subsection. The entire CIQ training procedure is described by Algorithm 1 in Appendix.

3.5 CLEVER-Q: A Robustness Evaluation Metric for Q-Networks

Here we provide a comprehensive score (CLEVER-Q) for evaluating the robustness of a Q-network model by extending the CLEVER robustness score (Weng et al., 2018) designed for classification tasks to Q-network based DRL tasks. Consider an ℓ_p-norm bounded ($p \geq 1$) perturbation δ to the state s_t. We first derive a lower bound β_L on the minimal perturbation to s_t for altering the action with the top Q-value, i.e., the greedy action. For a given s_t and a Q-network, this lower bound β_L

[2]Though such manner may lead to the myth of over-parameterization, our ablation study proves that we can achieve better results with almost the same amount of parameters.

provides a robustness guarantee that the greedy action at s_t will be the same as that of *any* perturbed state $s_t + \delta$, as long as the perturbation level $\|\delta\|_p \leq \beta_L$. Therefore, the larger the value β_L is, the more resilience of the Q-network against perturbations can be guaranteed. Our CLEVER-Q score uses the extreme value theory to evaluate the lower bound β_L as a robustness metric for benchmarking different Q-network models. The proof of Theorem 1. is available in appendix B.1.

Theorem 1. *Consider a Q-network $Q(s, a)$ and a state s_t. Let $\mathcal{A}^* = \arg\max_a Q(s_t, a)$ be the set of greedy (best) actions having the highest Q-value at s_t according to the Q-network. Define $g_a(s_t) = Q(s_t, \mathcal{A}^*) - Q(s_t, a)$ for every action a, where $Q(s_t, \mathcal{A}^*)$ denotes the best Q-value at s_t. Assume $g_a(s_t)$ is locally Lipschitz continuous*[3] *with its local Lipschitz constant denoted by L_q^a, where $1/p + 1/q = 1$ and $p \geq 1$. For any $p \geq 1$, define the lower bound*

$$\beta_L = min_{a \notin \mathcal{A}^*} g_a(s_t)/L_q^a. \tag{3}$$

Then for any δ such that $\|\delta\|_p \leq \beta_L$, we have $\arg\max_a Q(s_t, a) = \arg\max_a Q(s_t + \delta, a)$.

4 EXPERIMENTS

4.1 ENVIRONMENTS FOR DQNS

Our testing platforms were based on (a) OpenAI Gym (Brockman et al., 2016), (b) Unity-3D environments (Juliani et al., 2018), (c) a 2D gaming environment (Brockman et al., 2016), and (d) visual learning from pixel inputs of cart pole. Our test environments cover some major application scenarios and feature discrete actions for training DQN agents with the CLEVER-Q analysis.

Vector Cartpole: Cartpole (Sutton et al., 1998) is a classical continuous control problem. The defined environment is manipulated by adding a force of $+1$ or -1 to a moving cart. A pendulum starts upright, and the goal is to balance and prevent it from falling over. We use Cartpole-v0 from Gym (Brockman et al., 2016) with a targeted reward $= 195.0$ to solve the environment. The observational vector-state consist of four physical parameters of cart position and angle velocities.

Banana Collector: The Banana collector shown in Figure 7 (a) is one of the Unity baseline (Juliani et al., 2018) rendering by 3D engine. Different from the MuJoCo (Todorov et al., 2012) simulators with continuous actions, the Banana collector is controlled by four discrete actions corresponding to moving directions. The state-space has 37 dimensions included velocity and a ray-based perception of objects around the agent. The targeted reward is 12.0 points by accessing correct bananas $(+1)$.

Lunar Lander: Similar to the Atari gaming environments, Lunar Lander-v2 (Figure 7 (c)) is a discrete action environment from OpenAI Gym (Brockman et al., 2016). The state is an eight-dimensional vector that records the lander's position, velocity, angle, and angular velocities. The episode finishes if the lander crashes or comes to rest, receiving a reward -100 or $+100$ with a targeted reward of 200. Firing ejector costs -0.3 each frame with $+10$ for each ground contact.

Pixel Cartpole: To further evaluate our models, we conduct experiments from the pixel inputs in the cartpole environment as a visual learning task. The size of input state is 400×600. We use a max-pooling and a convolution layer to extract states as network inputs. The environment includes two discrete actions $\{left, right\}$, which is identical to the Cartpole-v0 of the vector version.

4.2 BASELINE METHODS

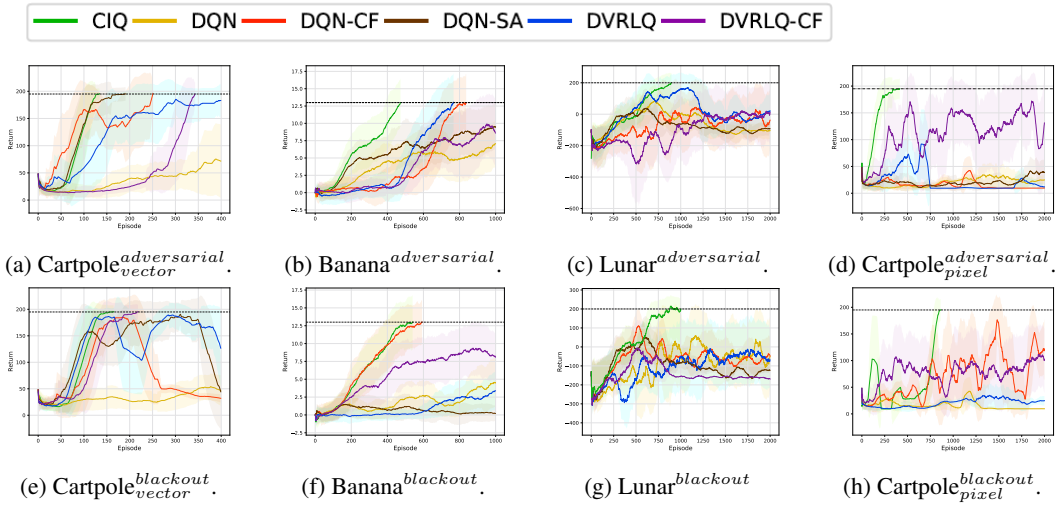
In the experiments, we compare our CIQ algorithm with two sets of DRL baselines to demonstrate the resilience capability of the proposed method. We ensure all the models have the **same number** of 9.7 millions **parameters** with careful fine-tuning to avoid model capacity issues.

Pure DQN: We use DQN as a baseline in our experiments. The DQN agent is trained and tested on interfered state s'_t. We also evaluate common DQN improvements in appendix C.1 and find the improvements have no significant effect against interference.

DQN with an interference classifier (DQN-CF): In the resilient reinforcement learning framework, the agent is given the true interference label i_t^{train} at training. Therefore, we would like to provide this

[3]Here locally Lipschitz continuous means $g_a(s_t)$ is Lipschitz continuous within the ℓ_p ball centered at s_t with radius R_p. We follow the same definition as in (Weng et al., 2018).

additional information to the DQN agent for a **fair comparision.** During training, the interfered state s'_t is concatenated with the true label i_t^{train} as the input for the DQN agent. Since the true label is not available at testing, we train an additional binary classifier (CF) for the DQN agent. The classifier is trained to predict the interference label, and this predicted label will be concatenated with the interfered state as the input for the DQN agent during testing.



(a) Cartpole$^{adversarial}_{vector}$. (b) Banana^{adversarial}. (c) Lunar^{adversarial}. (d) Cartpole$^{adversarial}_{pixel}$.

(e) Cartpole$^{blackout}_{vector}$. (f) Banana^{blackout}. (g) Lunar^{blackout} (h) Cartpole$^{blackout}_{pixel}$.

Figure 4: Performance of DQNs under potential (20%) adversarial and black-out interference.

DQN with safe actions (DQN-SA): Inspired by shielding-based safe RL (Alshiekh et al., 2018), we consider a DQN baseline with safe actions (SA). The DQN-SA agent will apply the DQN action if there is no interference. However, if the current observation is interfered, it will choose the action used for the last uninterfered observation as the safe action. This action-holding method is also a typical control approach when there are missing observations (Franklin et al., 1998). Similar to DQN-CF, a binary classifier for interference is trained to provide predicted labels at testing.

DVRLQ and DVRLQ-CF: Motivated by deep variational RL (DVRL) (Igl et al., 2018), we provide a version of DVRL as a POMDP baseline. We call this baseline DVRLQ because we replace the A2C-loss with the DQN loss. Similar to DQN-CF, we also consider another baseline of DVRLQ with a classifier, referred to as DVRLQ-CF, for a fair comparison using the interference labels.

Table 1: AC-Rate and CLEVER-Q robustness analysis under Gaussian (l_2-norm) and adversarial (l_∞-norm) perturbations in the vector Cartpole environment.

$\mathcal{I}$=L_2	AC-Rate		CLEVER-Q		$\mathcal{I}$=L_∞	AC-Rate		CLEVER-Q	
P%, $\mathcal{I}$	DQN	CIQ	DQN	CIQ	P%, $\mathcal{I}$	DQN	CIQ	DQN	CIQ
10%	82.10%	**99.61%**	0.176	**0.221**	10%	62.23%	**99.52%**	0.169	**0.248**
20%	72.15%	**98.52%**	0.130	**0.235**	20%	9.68%	**98.52%**	0.171	**0.236**
30%	69.74%	**98.12%**	0.109	**0.232**	30%	1.22%	**98.10%**	0.052	**0.230**

4.3 Resilient RL on Average Returns

We run performance evaluation with six different interference probabilities (p^I in Sec. 3.1), including $\{0\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$. We train each agent 50 times and highlight its standard deviation with lighter colors. Each agent is trained until the target score (shown as the dashed black line) is reached or until 400 episodes. We show the average returns for $p^I = 20\%$ under adversarial perturbation and black-out in Figure 4 and report the rest of the results in appendix C.1.

CIQ (green) clearly outperforms all the baselines under all types of interference, validating the effectiveness of our CIQ in learning to infer and gaining resilience against a wide range of observational interferences. Pure DQN (yellow) cannot handle the interference with 20% noise level. DQN-CF (orange) and DQN-SA (brown) have competitive performance in some environments against certain

interferences, but perform poorly in others. DVRLQ (blue) and DVRLQ-CF (purple) cannot achieve the target reward in most experiments and this might suggest the inefficiency of applying a general POMDP approach in a framework with a specific structure of observational interference.

4.4 Robustness Metrics based on Recording States

We evaluate the robustness of DQN and CIQ by the proposed CLEVER-Q metric. To make the test state environment consistent among different types and levels of interference, we record the interfered states, $S_N = \mathcal{I}(S_C)$, together with their clean states, S_C. We then calculate the average CLEVER-Q for DQN and CIQ based on the clean states S_C using Eq. 3 over 50 times experiments for each agent.

We also consider a retrospective robustness metric, the action correction rate (AC-Rate). Motivated by previous off-policy and error correction studies (Dulac-Arnold et al., 2012; Harutyunyan et al., 2016; Lin et al., 2017), AC-Rate is defined as the action matching rate $R_{Act} = \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{1}_{\{a_t = a_t^*\}}$ between a_t and a_t^* over an episode with length T. Here a_t denotes the action taken by the agent with interfered observations S_N, and a_t^* is the action of the agent if clean states S_C were observed instead.

The roles of CLEVER-Q and AC-Rate are complementary as robustness metrics. CLEVER-Q measures sensitivity in terms of the margin (minimum perturbation) required for a given state to change the original action. AC-rate measures the utility in terms of action consistency. Altogether, they provide a comprehensive resilience assessment.

Table 1 reports the two robustness metrics for DQN and CIQ under two types of interference. CIQ attains higher scores than DQN in both CLEVER-Q and AC-Rate, reflecting better resilience in CIQ evaluations. We provide more robustness measurements in appendix B.2 and E.

4.5 Additional analysis

We also conduct the following analysis to better understand our CIQ model. Environments with a dynamic noise level are evaluated. Due to the space limit, see their details in appendix C to E.

Neural saliency map: We apply the perturbation-based saliency map for DRL (Greydanus et al., 2018) as shown in appendix Figure S6 and E.4 to visualize the saliency centers of CIQ and others.

Treatment effect analysis: We provide treatment effect analysis on each kind of interference to statistically verify the CGM with lowest errors on average treatment effect refutation in appendix D.

Ablation studies: We conduct ablation studies by comparing several CIQ variants, each without certain CIQ component and verify the importance of the proposed CIQ architecture in appendix E.

Test on different noise levels: We train CIQ under one noise level and test on another level, which shows that the difference in noise level does not affect much on the performance in appendix C.6.

Transferability in robustness: Based on CIQ, we study how well can the robustness of different interference types transfer between training and testing environments . We evaluate two general settings (i) same interference type but different noise levels (appendix C.6) and (ii) different interference types (appendix E.5).

Multiple interference types: We also provide an generalized version of CIQ that deals with multiple interference types at training and testing environments. The generalized CIQ is equipped with a common encoder and individual interference decoders to study multi-module conditional inference, with some additional discussion in appendix E.6.

5 Conclusion

Our experiments suggest that, although some DRL-based algorithms can achieve high scores under the normal condition, their performance can be severely degraded in the presence of interference. In order to be resilient against interference, we propose CIQ, a novel causal-inference-driven DRL algorithm. Evaluated on a wide range of environments and multiple types of interferences, the CIQ results show consistently superior performance over several RL baseline methods. We also validate the improved resilience of CIQ by CLEVER-Q and AC-Rate metrics. We will provide CIQ implementations to the community in the future for further studies.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Prithviraj Ammanabrolu and Mark Riedl. Transfer in deep reinforcement learning using knowledge graphs. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pp. 1–10, 2019.

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems*, pp. 1342–1350, 2015.

Yoshua Bengio. Deep learning of representations: Looking forward. In *International Conference on Statistical Language and Speech Processing*, pp. 1–37. Springer, 2013.

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

Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Fast reinforcement learning with large action sets using error-correcting output codes for mdp factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 180–194. Springer, 2012.

Andrew Forney, Judea Pearl, and Elias Bareinboim. Counterfactual data-fusion for online reinforcement learners. In *International Conference on Machine Learning*, pp. 1156–1164, 2017.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. Noisy networks for exploration. *ICLR 2018, arXiv preprint arXiv:1706.10295*, 2018.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.

Gene F Franklin, J David Powell, Michael L Workman, et al. *Digital control of dynamic systems*, volume 3. Addison-wesley Menlo Park, CA, 1998.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

Sander Greenland, Judea Pearl, and James M Robins. Causal diagrams for epidemiologic research. *Epidemiology*, pp. 37–48, 1999.

Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning*, pp. 1792–1801, 2018.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396. IEEE, 2017.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Anna Harutyunyan, Marc G Bellemare, Tom Stepleton, and Rémi Munos. Q lamda with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pp. 305–320. Springer, 2016.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

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

Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, pp. 2117–2126, 2018.

Guido W Imbens and Donald B Rubin. Rubin causal model. In *Microeconometrics*, pp. 229–241. Springer, 2010.

Amin Jaber, Jiji Zhang, and Elias Bareinboim. Causal identification under markov equivalence: Completeness results. In *International Conference on Machine Learning*, pp. 2981–2989, 2019.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *ICLR*, 2017.

Tor A Johansen, Andrea Cristofaro, Kim Sørensen, Jakob M Hansen, and Thor I Fossen. On estimation of wind velocity, angle-of-attack and sideslip angle of small uavs using standard sensors. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 510–519. IEEE, 2015.

Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

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

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Taylor W Killian, Marzyeh Ghassemi, and Shalmali Joshi. Counterfactually guided policy transfer in clinical settings. *arXiv preprint arXiv:2006.11654*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Gilwoo Lee, Brian Hou, Aditya Mandalika, Jeongseok Lee, Sanjiban Choudhury, and Siddhartha S Srinivasa. Bayesian policy optimization for model uncertainty. *arXiv preprint arXiv:1810.01014*, 2018.

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.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Deconfounding reinforcement learning in observational settings. *arXiv preprint arXiv:1812.10576*, 2018.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pp. 1113–1132. PMLR, 2020.

Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2493–2500, 2020.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

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

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Pol Moreno, Jan Humplik, George Papamakarios, Bernardo Avila Pires, Lars Buesing, Nicolas Heess, and Theophane Weber. Neural belief states for partially observed domains. In *NeurIPS 2018 workshop on Reinforcement Learning under Partial Observability*, 2018.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepezvari, Satinder Singh, et al. Behaviour suite for reinforcement learning. *arXiv preprint arXiv:1908.03568*, 2019.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995a.

Judea Pearl. On the testability of causal models with latent and instrumental variables. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 435–443. Morgan Kaufmann Publishers Inc., 1995b.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the american statistical association*, 90(429):106–121, 1995.

Kenneth J Rothman and Sander Greenland. Causation and causal inference in epidemiology. *American journal of public health*, 95(S1):S144–S150, 2005.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Jürgen Schmidhuber. Reinforcement learning in markovian and non-markovian environments. In *Advances in neural information processing systems*, pp. 500–506, 1991.

Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3076–3085. JMLR. org, 2017.

Amit Sharma, Emre Kiciman, et al. DoWhy a python package for causal inference. *KDD 2019 workshop*, 2019.

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

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, pp. 631–648, 2018.

Richard S Sutton, Andrew G Barto, Francis Bach, et al. *Reinforcement learning: An introduction*. MIT press, 1998.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014.

Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31–36. IEEE, 2017.

Guy Tennenholtz, Shie Mannor, and Uri Shalit. Off-policy evaluation in partially observable environments. *arXiv preprint arXiv:1909.03739*, 2019.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Herke Van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3928–3934. IEEE, 2016.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEFCON24*, 2016.

Chao-Han Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Chin-Hui Lee. Characterizing speech adversarial examples using self-attention u-net enhancement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3107–3111. IEEE, 2020a.

Chao-Han Huck Yang, Yi-Chieh Liu, Pin-Yu Chen, Xiaoli Ma, and Yi-Chang James Tsai. When causal intervention meets adversarial examples and image masking for deep neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3811–3815. IEEE, 2019.

Chao-Han Huck Yang, Linda Liu, Ankur Gandhe, Yile Gu, Anirudh Raju, Denis Filimonov, and Ivan Bulyko. Multi-task language modeling for improving speech recognition of rare words. *arXiv preprint arXiv:2011.11715*, 2020b.

Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Yi Ouyang, I-Te Danny Hung, Chin-Hui Lee, and Xiaoli Ma. Enhanced adversarial strategically-timed attacks against deep reinforcement learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3407–3411. IEEE, 2020c.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pp. 11214–11224. PMLR, 2020a.

Cheng Zhang, Kun Zhang, and Yingzhen Li. A causal view on robustness of neural networks. *Advances in Neural Information Processing Systems*, 33, 2020b.

Junzhe Zhang and Elias Bareinboim. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In *International Conference on Machine Learning*, pp. 11012–11022. PMLR, 2020.

Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. Causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems*, 33, 2020c.

A APPENDIX

INDEX

Our supplementary sections included:

- **B.** Proof of the CLEVER-Q Theorem and Additional Robustness Measurements
- **C.** Implementation Details and Additional Results
- **D.** Causal Relation Evaluation and Average Treatment Effects in CIQ Networks
- **E.** Ablation Studies

B PROOF OF THE CLEVER-Q THEOREM AND ADDITIONAL ROBUSTNESS MEASUREMENTS

B.1 PROOF OF THE CLEVER-Q THEOREM

Here we provide a comprehensive score (CLEVER-Q) for evaluating the robustness of a Q-network model by extending the CLEVER robustness score (Weng et al., 2018) designed for classification tasks to Q-network based DRL tasks. Consider an ℓ_p-norm bounded ($p \geq 1$) perturbation δ to the state s_t. We first derive a lower bound β_L on the minimal perturbation to s_t for altering the action with the top Q-value, i.e., the greedy action. For a given s_t and a Q-network, this lower bound β_L provides a robustness guarantee that the greedy action at s_t will be the same as that of *any* perturbed state $s_t + \delta$, as long as the perturbation level $\|\delta\|_p \leq \beta_L$. Therefore, the larger the value β_L is, the more resilience of the Q-network against perturbations can be guaranteed. Our CLEVER-Q score uses the extreme value theory to evaluate the lower bound β_L as a robustness metric for benchmarking different Q-network models.

Theorem 2. *Consider a Q-network $Q(s, a)$ and a state s_t. Let $\mathcal{A}^* = \arg\max_a Q(s_t, a)$ be the set of greedy (best) actions having the highest Q-value at s_t according to the Q-network. Define $g_a(s_t) = Q(s_t, \mathcal{A}^*) - Q(s_t, a)$ for every action a, where $Q(s_t, \mathcal{A}^*)$ denotes the best Q-value at s_t. Assume $g_a(s_t)$ is locally Lipschitz continuous[4] with its local Lipschitz constant denoted by L_q^a, where $1/p + 1/q = 1$ and $p \geq 1$. Then for any $p \geq 1$, define the lower bound*

$$\beta_L = min_{a \notin \mathcal{A}^*} g_a(s_t)/L_q^a.$$

Then for any δ such that $\|\delta\|_p \leq \beta_L$,

$$\arg\max_a Q(s_t, a) = \arg\max_a Q(s_t + \delta, a)$$

Proof. Because $g_a(s_t)$ is locally Lipschitz continuous, by Holder's inequality, we have

$$|g_a(x) - g_a(y)| \leq L_q^a ||x - y||_p, \tag{4}$$

for any x, y within the R_p-ball centered at s_t. Now let $x = s_t$ and $y = s_t + \delta$, where δ is some perturbation. Then

$$g_a(s_t) - L_q^a ||\delta||_p \leq g_a(s_t + \delta) \leq g_a(s_t) + L_q^a . ||\delta||_p \tag{5}$$

Note that if $g_a(s_t + \delta) \geq 0$, then A^* still remains as the top Q-value action set at state $s_t + \delta$. Moreover, $g_a(s_t) - L_q^a ||\delta||_p \geq 0$ implies $g_a(s_t + \delta) \geq 0$. Therefore,

$$||\delta||_p \leq g_a(s_t)/L_q^a, \tag{6}$$

provides a robustness guarantee that ensures $Q(s_t + \delta, \mathcal{A}^*) \geq Q(s_t + \delta, a)$ for any δ satisfying Eq. equation 4. Finally, to provide a robustness guarantee that $Q(s_t + \delta, \mathcal{A}^*) \geq Q(s_t + \delta, a)$ for any action $a \notin \mathcal{A}^*$, it suffices to take the minimum value of the bound (for each a) in Eq. equation 4 over all actions other than a^*, which gives the lower bound

$$\beta_L = min_{a \notin A^*} g_a(s_t)/L_q^a \tag{7}$$

□

[4]Here locally Lipschitz continuous means $g_a(s_t)$ is Lipschitz continuous within the ℓ_p ball centered at s_t with radius R_p. We follow the same definition as in Weng et al. (2018).

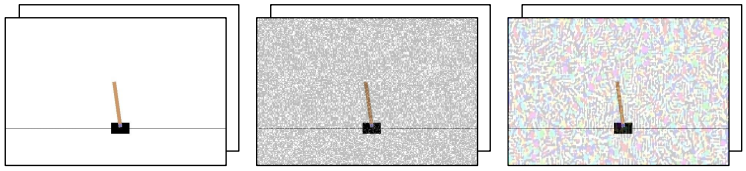


Figure 5: Visualization of perturbed observation (state) under uncertainty: (a) original state; (b) Gaussian perturbation and (c) adversarial perturbation Huang et al. (2017).

For computing β_L, while the numerator is easy to obtain, the local Lipschitz constant L_q^a cannot be directly computed. In our implementation, by using the fact that L_q^a is equivalent to the local maximum gradient norm (in ℓ_q norm), we use the same sampling technique from extreme value theory as proposed in (Weng et al., 2018) for estimating L_q^a.

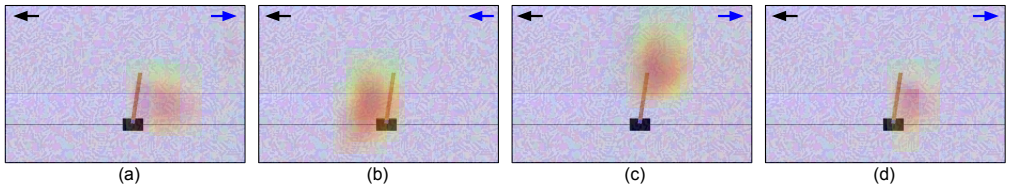


Figure 6: Perturbation-based saliency map on Pixel Cartpole under adversarial perturbation: (a) DQN, (b) CIQ, (c) DQN-CF, and (d) DVRLQ-CF. The black arrows are correct actions and blue arrows are agents' actions. The neural saliency of CIQ makes more correct actions responding to ground actions.

B.2 Additional Robustness Measurements

Following the discussion in Section 4.4 of the main content, we provide more experimental results related to CLEVER-Q measurement and use action correction rate (AC-Rate) mentioned in the main content as a reference.

Table S1. reports the two robustness metrics for DQN, CIQ, DQN-CF (a DQN joint-training with an interference classifier, denoted as Q-CF), DQN-SA (a DQN joint-training with safe-action replay, denoted as Q-SA), DVRLQ (a deep variational reinforcement learning framework (Igl et al., 2018) with a DQN-loss, denoted as V-Q), and DVRLQ-CF (a V-Q joint-training with an interference classifier, denoted as V-CF) under two types of L_n-norm (Weng et al., 2018) interference. CIQ attains higher scores than DQN in both CLEVER-Q and AC-Rate, reflecting better resilience in CIQ evaluations. The performance of the returns of each agent is shown in Table S2. We observe that variational auto-encoding methods included V-Q and V-CF attaining a lower CLEVER-Q, Act-Rate, and average returns from Table S1 and S2. From previous studies (Van Hoof et al., 2016), reasons would be difficulties (Van Hasselt et al., 2016) of estimation considering temporal information (Yang et al., 2020a) and the various state is hard to disentangle by a single network from counterfactual learning (Shalit et al., 2017; Louizos et al., 2017). We also conduct an experiment on a DQN extension of TD-VAE (Gregor et al., 2018), but the performance of this extension becomes even lower in all metrics than the V-CF after carefully fine-tuning. We also find that the DQNs would be a benefit on the performance with a joint-trained interference classifier shown in Table S1 and S2.

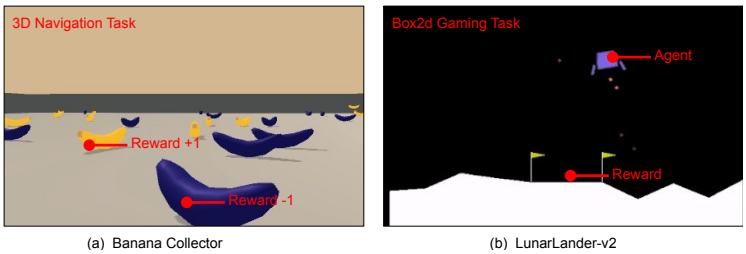


Figure 7: Illustration of our environments on: (a) a 3D navigation task, banana collector (Juliani et al., 2018), and (b) a video game, LunarLander (Brockman et al., 2016).

Table 2: AC-Rate and CLEVER-Q robustness analysis under Gaussian (Lipschitz l_2-norm) and adversarial (Lipschitz l_∞-norm) Huang et al. (2017) perturbations in the vector Cartpole environment.

$\mathcal{I}$=L_2	AC-Rate		CLEVER-Q		$\mathcal{I}$=L_∞	AC-Rate		CLEVER-Q	
P%, $\mathcal{I}$	DQN	CIQ	DQN	CIQ	P%, $\mathcal{I}$	DQN	CIQ	DQN	CIQ
10%	82.10%	**99.61%**	0.176	**0.221**	10%	62.23%	**99.52%**	0.169	**0.248**
20%	72.15%	**98.52%**	0.130	**0.235**	20%	9.68%	**98.52%**	0.171	**0.236**
30%	69.74%	**98.12%**	0.109	**0.232**	30%	1.22%	**98.10%**	0.052	**0.230**
P%, $\mathcal{I}$	Q-CF	Q-SA	Q-CF	Q-SA	P%, $\mathcal{I}$	Q-CF	Q-SA	Q-CF	Q-SA
10%	85.10%	83.58%	0.185	0.182	10%	71.82%	67.56%	0.181	0.174
20%	75.23%	74.94%	0.145	0.152	20%	55.28%	51.20%	0.136	0.124
30%	72.25%	71.47%	0.127	0.125	30%	46.45%	43.27%	0.109	0.102
P%, $\mathcal{I}$	V-Q	V-CF	V-Q	V-CF	P%, $\mathcal{I}$	V-Q	V-CF	V-Q	V-CF
10%	83.65%	84.32%	0.184	0.188	10%	61.92%	65.87%	0.173	0.179
20%	71.79%	72.83%	0.123	0.138	20%	51.06%	53.48%	0.117	0.121
30%	69.70%	70.85%	0.094	0.108	30%	36.99%	37.89%	0.087	0.092

C Implementation Details and Additional Results

Algorithm 1 CIQ Training

Inputs: $Agent$, $NoisyEnv$, $Oracle$, max_step, $NoisyEnv_test$, $target$, $eval_steps$
Initialize: $t = 0$, $score = 0$, $s'_t = NoisyEnv$.reset()
while $t < max_step$ **and** $score < target$ **do**
  i_t = $oracle(NoisyEnv, t)$
  a_t = $Agent$.act(s'_t, i_t)
  s'_{t+1}, r_t, $done$ = $NoisyEnv$.step(a_t)
  $Agent$.learn(s'_t, a_t, r_t, s'_{t+1}, i_t)
  if $t \in eval_steps$ **then**
    $score$ = $Agent$.evaluate($NoisyEnv_test$)
  end if
  if $done$ **then**
    s'_t = $NoisyEnv$.reset()
  else
    s'_t = s'_{t+1}
  end if
  t = t + 1
end while
Return $Agent$

C.1 Background and Training Setting

To scale to high-dimensional problems, one can use a parameterized deep neural network $Q(s, a; \theta)$ to approximate the Q-function, and the network $Q(s, a; \theta)$ is referred to as the deep Q-network (DQN). The DQN algorithm Mnih et al. (2015) updates parameter θ according to the loss function:

$$L^{\text{DQN}}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim D}\left[(y_t^{\text{DQN}} - Q(s_t, a_t; \theta))^2\right]$$

where the transitions (s_t, a_t, r_t, s_{t+1}) are uniformly sampled from the replay buffer D of previously observed transitions, and $y_t^{\text{DQN}} = r_t + \gamma \max_a Q(s_{t+1}, a; \theta^-)$ is the DQN target with θ^- being the target network parameter periodically updated by θ.

Double DQN (DDQN) Van Hasselt et al. (2016) further improves the performance by modifying the target to $y_t^{\text{DDQN}} = r_t + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta); \theta^-)$. Prioritized replay is another DQN

improvement which samples transitions (s_t, a_t, r_t, s_{t+1}) from the replay buffer according to the probabilities p_t proportional to their temporal difference (TD) error: $p_t \propto |y_t^{\text{DDQN}} - Q(s_t, a_t; \theta)|^{\alpha}$ where α is a hyperparameter.

We use Pytorch 1.2 to design both DQN and causal inference Q (CIQ) networks in our experiments. Our code can be found in the supplementary material. We use Nvidia GeForce RTX 2080 Ti GPUs with CUDA 10.0 for our experiments. We use the Quantile Huber loss (Dabney et al., 2018) $\mathcal{L}_\kappa$ for DQN models with $\kappa = 1$ in Sup-Eq. 9, which allows less dramatic changes from Huber loss:

$$\mathcal{L}_\kappa(u) = \begin{cases} \frac{1}{2}u^2, & \text{if } |u| \leq \kappa \\ \kappa\left(|u| - \frac{1}{2}\kappa\right), & \text{otherwise} \end{cases} \tag{8}$$

The quantile Huber loss (Dabney et al., 2018) is the asymmetric variant of the Huber loss for quantile $\tau \in [0, 1]$ from Sup-Eq. 8:

$$\rho_\tau^\kappa(u) = \left|\tau - \delta_{\{u<0\}}\right| \mathcal{L}_\kappa(u). \tag{9}$$

After the a maximum update step in the temporal loss u in Sup-Eq. 8, we synchronize θ_i^- with θ_i follow the implementation from the OpenAI baseline Dhariwal et al. (2017) in Sup-Eq 10:

$$u_i\left(\theta_i\right) = \mathbb{E}\left(\underbrace{y^{\text{DDQN}}}_{\theta_{\text{target}}} - \underbrace{Q\left(s, a; \theta_i\right)}_{\theta_{\text{local}}}\right)^2. \tag{10}$$

We use the soft-update Fox et al. (2015) to update the DQN target network as in Sup-Eq 11:

$$\theta_{\text{local}} = \tau \times \theta_{\text{local}} + (1 - \tau) \times \theta_{\text{target}}, \tag{11}$$

where θ_{target} and θ_{local} represent the two neural networks in DQN and τ is the soft update parameter depending on the task.

For each environment, in additional to the 5 baselines described in Section 4.2, we also evaluate the performance of common DQN improvements such as deep double Q-networks (DDQN) for DDQN with dueling (DDQN$_d$), DDQN with a prioritized replay (DDQN$_p$), DDQN with a joint-training interference classifier (DDQN-CF), and DDQN with a safe action reply (DDQN-SA). We test each model against four types of interference, Gaussian, Adversarial, Blackout, and Frozen Frame, with $p^I \in [10\%, 20\%, 30\%, 40\%, 50\%]$. We also consider a non-stationary noise-level sampling from a cosine-wave in a range of [0%, 30%] for every ten steps. Results in Table S3-S8 show a return averaged from the four types of noise in the Env$_1$ to Env$_4$. CIQ shows a better and continuous performance to solve the environments before the noise level attaining 40% and under the cosine-noise. Compared to variational based DQNs methods, joint-trained DDQN-CF show a much obvious advantages when the noise levels are in the range of 40% to 50%.

C.2 ENV$_1$: CARTPOLE ENVIRONMENT.

We use a four-layer neural network, which included an input layer, two 32-unit wide ReLU hidden layers, and an output layer (2 dimensions). The observation dimension of Cartpole-v1 Brockman et al. (2016) is 4 and the input stacks 4 consecutive observations. The dimension of the input layer is $[4 \times 4]$. We design a replay buffer with a memory of 100,000, with a mini-batch size of 32, the discount factor γ is set to 0.99, the τ for a soft update of target parameter is 5×10^{-3}, a learning rate for Adam Kingma & Ba (2014) optimization is 5×10^{-4}, a regularization term for weight decay is 1×10^{-4}, the coefficient α for importance sampling exponent is 0.6, the coefficient of prioritization exponent is 0.4. We train each model 1,000 times for each case and report the mean of the average final performance (average over all types of interference) in Table S3. Env$_1$ is often recognized as a simplest environment for DQN training. However, we observe an stability issue of attaining reward over 190.0, when most DQN models attain an over 100.0 score in a 10% noise level. CIQ perform best and competitive results without internal affects from over-parameterization

C.3 ENV$_2$: 3D BANANA COLLECTOR ENVIRONMENT.

We utilize the Unity Machine Learning Agents Toolkit (Juliani et al., 2018), which is an open-source[5] and reproducible 3D rendering environment for the task of Banana Collector. We use

[5]Source: https://github.com/Unity-Technologies/ml-agents

Table 3: Performance on return in clean and five different noise level in Env_1 evaluated by an average of under uncertain perturbation included Gaussian, adversarial, blackout, and frozen frame. All DQN models solve the environment with over 195.0 average returns in a clean state (a.k.a. no noise).

Model	0%	10%	20%	30%	40%	50%	Cosine	Para.
DQN	195.1	115.0	68.9	32.3	22.8	19.1	42.1	6.9 M
DDQN_d	195.1	130.3	85.6	57.2	29.6	23.4	68.2	9.7 M
DDQN_p	195.1	143.6	79.3	56.1	31.2	28.6	60.1	9.7 M
CIQ	195.1	**195.1**	**195.1**	**195.0**	**168.2**	**113.1**	**195.0**	9.7 M
DQN-CF	195.1	192.4	143.6	128.7	68.2	57.8	138.3	9.7 M
DDQN-CF	195.1	190.2	153.2	138.9	71.4	54.6	141.2	10.7 M
DVRLQ	195.1	152.6	107.8	76.12	58.12	19.2	72.9	9.7 M
DVRLQ-CF	195.1	163.1	119.3	91.3	70.9	25.1	89.2	10.7 M
DQN-SA	195.1	143.2	132.4	112.1	82.9	23.2	107.1	10.7 M
DDQN-SA	195.1	123.4	73.2	59.4	28.1	22.8	62.8	9.7 M

Table 4: Ablation study on parameter of different DQN models using in our experiments training under four different noise type of noisy environments ($P = 20\%$), which included Blackout, Adversarial, Gaussian, and Frozen frame for Env_1 reported in the main content. The minimal parameters of each model denote as Para. in the Tab. 14.

Model	Para.	Gaussian	Adversarial	Blackout	Frozen
DQN	6.9M	67.4	42.5	85.7	62.1
CIQ	9.7M	**195.1**	**195.0**	**195.1**	**195.2**
DQN-CF	9.7M	149.2	129.1	161.3	167.2
DQN-SA	9.7M	128.9	144.5	109.1	165.8
DVRLQ	9.7M	107.1	87.3	127.56	142.4
DVRLQ-CF	10.7M	112.3	97.82	131.3	152.3

open-source graphic rendering version [6] with Unity backbone (Juliani et al., 2018) for reproducible DQN experiments, which is designed to render the collector agent for both Linux and Windows systems. A reward of $+1$ is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. We use a six-layer deep network, which includes an input layer, three 64-unit fully-connected ReLU hidden layers, and an output layer (2 dimensions). We use $[37 \times 4]$ for our input layer, which composes from the observation dimension (37) and the stacked input of 4 consecutive observations. We design a replay buffer with a memory of 100,000, with a mini-batch size of 32, the discount factor γ is equal to 0.99, the τ for a soft update of target parameter is 10^{-3}, a learning rate for Adam Kingma & Ba (2014) optimization is 5×10^{-4}, a regularization term for weight decay is 1×10^{-4}, the coefficient α for importance sampling exponent is 0.6, the coefficient of prioritization exponent is 0.4.

We train each model 1,000 times for each case and report the mean of the average final performance (average over all types of interference) in Table S5. We report the DVRLQ-CF, which attains a higher performance among DVRLQ and DVRLQ-SA, to compare with DQN-CF, DDQN-CF. CIQ still performs an overall best performance compared to the other baselines. Interestingly, learning an interference improves general performance combined with the joint-training frameworks.

C.4 Env_3: Lunar Lander Environment.

The lunar lander-v2 Brockman et al. (2016) is one of the most challenging environments with discrete actions. The observation dimension of Lunar Lander-v2 (Brockman et al., 2016) is 8 and the input stacks 10 consecutive observations. The objective of the game is to navigate the lunar lander spaceship to a targeted landing spot without a collision. A collection of six discrete actions controls two real-valued vectors ranging from -1 to +1. The first dimension controls the main engine on and off numerically, and the second dimension throttles from 50% to 100% power. The following two actions represent for firing left, and the last two actions represent for firing the right engine. The

[6] Source: https://github.com/udacity/deep-reinforcement-learning/tree/master/p1_navigation

Table 5: Performance on return in clean and five different noise level in Env_2 evaluated by an average of under uncertain perturbation included Gaussian, adversarial, blackout, and frozen frame. All DQN models solve the environment with over 12.0 average returns in a clean state (a.k.a. no noise).

Model	0 %	10%	20%	30%	40%	50%	Cosine	Para.
DQN	12.0	9.5	5.1	3.4	2.0	2.1	3.2	7.6 M
DDQN	12.0	10.1	5.8	6.8	4.8	2.6	6.9	9.7 M
DDQN_d	12.0	11.8	6.7	7.4	5.4	2.8	6.6	9.7 M
DDQN_p	12.0	11.9	9.2	7.2	5.4	3.2	8.3	9.7M
CIQ	12.0	**12.0**	**12.0**	**12.0**	**11.8**	**7.8**	**12.0**	9.7 M
DQN-CF	12.0	**12.0**	11.5	10.1	9.0	6.9	11.8	9.7 M
DDQN-CF	12.0	**12.0**	11.8	10.8	10.0	7.1	11.9	10.7 M
DVRLQ-CF	12.0	**12.0**	11.2	10.5	9.2	6.8	11.6	10.7 M

Table 6: Ablation study on parameter of different DQN models using in our experiments training under four different noise type of noisy environments ($P = 20\%$), which included Blackout, Adversarial, Gaussian, and Frozen frame for Env_2 reported in the main content. The minimal parameters of each model denote as Para. in the Tab. 14.

Model	Para.	Gaussian	Adversarial	Blackout	Frozen
DQN	6.9M	6.1	4.5	5.2	5.7
CIQ	9.7M	**12.0**	**12.0**	**12.0**	**12.0**
DQN-CF	9.7M	**12.0**	10.9	11.3	11.8
DQN-SA	9.7M	**12.0**	11.1	11.1	11.2
DVRLQ	9.7M	**12.0**	11.3	10.3	11.6
DVRLQ-CF	10.7M	**12.0**	11.5	10.5	11.6

dimension of the input layer is $[8 \times 10]$. We design a 7-layers neural network for this task, which includes 1 input layer, 2 layer of 32 unit wide fully-connected ReLU network, 2 layers deep 64-unit wide ReLU networks, (for all DQNs), 1 layer of 16 unit wide fully-connected ReLU network, and 1 output layer (4 dimensions). The replay buffer size is 500,000; the minimum batch size is 64, the discount factor is 0.99, the τ for a soft update of target parameters is 10^{-3}, the learning rate is 5×10^{-4}, the minimal step for reset memory buffer is 50. We train each model 1,000 times for each case and report the mean of the average final performance (average over all types of interference) in Table S7. Env_3 is a challenging task owing to often receive negative reward during the training. We thus consider a non-stationary noise-level sampling from a cosine-wave in a narrow range of [0%, 20%] for every ten steps. Results suggest CIQ could still solve the environment before the noise-level going over to 30%. For the various noisy test, CIQ attains a best performance over 200.0 the other DQNs algorithms (skipping the table since only CIQ and DQN-CF have solved the environment over 200.0 training with adversarial and blackout interference.)

Table 7: Performance on average return in clean and five different noise level in Env_3 evaluated by an average of under uncertain perturbation included Gaussian, adversarial, blackout, and frozen frame. All DQN models solve the environment with over 200.0 average returns in a clean state input (a.k.a. no noise).

Model	0%	10%	20%	30%	40%	50%	Cosine	Para.
DQN	200.0	54.2	-102.1	-134.6	-200.2	-298.7	12.4	12.3M
DDQN	200.0	70.1	15.8	-6.8	-124.8	-243.2	32.5	15.5M
DDQN_d	200.0	92.2	26.7	-7.4	-167.4	-100.1	43.3	15.5M
DDQN_p	200.0	102.1	39.2	-17.2	-189	-107.8	65.0	15.5M
CIQ	200.0	**200.0**	**200.0**	**107.8**	**50.1**	**17.2**	**200.0**	15.5M
DQN-CF	200.0	98.2	81.5	40.1	9.0	-59.9	85.6	15.5M
DDQN-CF	200.0	188.2	91.8	58.2	10.0	-20.1	128.9	15.5M
DVRLQ-CF	200.0	198.2	121.1	80.2	29.2	11.2	165.4	15.5M

C.5 ENV$_4$: PIXEL CARTPOLE ENVIRONMENT

To observe pixel inputs of Cartpole-v1 as states, we use a screen-wrapper with an original size of [400, 600, 3]. We first resize the original frame into a single gray-scale channel, [100, 150] from the RGN2GRAY function in the OpenCV. The implementation details are shown in the "pixel_tool.py" and "cartpole_pixel.py", which could be refereed to the submitted supplementary code. Then we stack 4 consecutive gray-scale frames as the input. We design a 7-layer DQN model, which included input layer, the first hidden layer convolves 32 filters of a $[8 \times 8]$ kernel with stride 4, the second hidden layer convolves 64 filters of a $[4 \times 4]$ kernel with stride 2, the third layer is a fully-connected layer with 128 units, from fourth to fifth layers are fully-connected layer with 64 units, and the output layer (2 dimensions). The replay buffer size is 500,000; the minimum batch size is 32, the discount factor is 0.99, the τ for a soft update of target parameters is 10^{-3}, the learning rate is 5×10^{-4}, the minimal step for reset memory buffer is 1000.

We train each model 1,000 times for each case and report the mean of the average final performance (average over all types of interference) in Table S8.

Table 8: Performance on average return in clean and five different noise level in Env$_4$ evaluated by an average of under uncertain perturbation included Gaussian, adversarial, blackout, and frozen frame. Only selected DQN models below solve the environment with over 195.0 average returns in a clean state input (a.k.a. no noise).

Model	0%	10%	20%	30%	40%	50%	Cosine	Para.
DDQN$_d$	195.0	182.7	99.7	75.5	47.4	13.2	90.4	20.5M
DDQN$_p$	195.0	187.9	139.2	88.3	51.6	15.9	120.3	20.5M
CIQ	195.0	**195.0**	**195.0**	**157.8**	**90.1**	**87.2**	195.0	20.5M
DQN-CF	195.0	175.0	112.5	90.0	52.0	39.2	92.8	20.5M
DDQN-CF	195.0	187.2	162.8	117.9	59.8	49.1	102.6	20.5M
DVRLQ-CF	195.0	193.6	121.1	80.2	63.2	41.4	91.2	20.5M

Table 9: Ablation study on parameter of different DQN models using in our experiments training under four different noise type of noisy environments ($P = 20\%$), which included Blackout, Adversarial, Gaussian, and Frozen frame for Env$_4$ reported in the main content. The minimal parameters of each model denote as Para. in the Tab. 14.

Model	Para.	Gaussian	Adversarial	Blackout	Frozen
DQN	6.9M	52.6	37.5	70.5	47.6
CIQ	9.7M	**195.0**	**195.0**	**195.0**	**195.0**
DQN-CF	9.7M	119.2	99.1	131.3	137.2
DQN-SA	9.7M	98.0	114.6	79.2	135.2
DVRLQ	9.7M	107.1	87.3	127.56	142.4
DVRLQ-CF	10.7M	121.3	97.82	131.3	152.3

C.6 TRAIN AND TEST ON DIFFERENT NOISE LEVEL

We consider settings with different training and testing noise levels for CIQ evaluation. The (train, test)% case trains with train% noise then tests with test% noise. Their results shown in Table S11 are similar to the cases with the same training and testing noise level. We observe that CIQ have the capability of learning transformable q-value estimation, which attain a succeed score of 195.00 in the noise level $30 \pm 10\%$. Meanwhile, other DQNs methods included DDQN-CF, DVRLQ-CF, DDQN-SA perform a general performance decay in the test on different noise level. This result would be limited to the generalization of power and challenges (Bengio, 2013; Higgins et al., 2018) in as disentangle unseen state of a single parameterized deep network.

C.7 MARKOVIAN NOISE

We also provide a dynamic noise study for CIQ training with i.i.d. Gaussian interference, $p^{\mathcal{I}} = 0.2$; testing with Markov distribution $P(i_t = 1|i_{t-1} = 1) = 0.55$, $P(i_t = 1|i_{t-1} = 0) = 0.05$, stationary

$p^{\mathcal{I}} = 0.1$ testing in Env_1 and Env_2. This experiment shows the learning power against unseen Markovian interference in Table 10, which further confirms CIQ's ability against unseen interference distribution and dynamics.

Table 10: CIQ training against unseen interference distribution and dynamics

Model	DQN	CIQ	DQN-CF	DQN-SA	DVRLQ	DQN-VAE	DQN-CEVAE
Env_1 Markov	112.3	**195.0**	181.4	131.4	112.1	163.7	155.6
Env_2 Markov	9.4	**12.1**	11.7	9.1	11.5	11.2	11.2

Table 11: Stability test of proposed CIQ (*Train* Noise-Level, *Test* Noise-Level)

Metrics	(10, 30)%	(30, 10)%	(30, 20)%	(30, 30)%	(30, 40)%	(30, 50)%
Performance	182.8	**195.0**	**195.0**	**195.0**	**195.0**	185.7
CLEVER-Q	0.195	0.239	0.232	0.230	0.224	0.215
AC-Rate	91.45 %	98.54%	98.62%	99.45%	98.45%	92.45%

C.8 Advantages of Training with Interference Labels

We provide an example to analytically demonstrate the learning advantage of having the interference labels during training. Consider an environment of i.i.d. Bernoulli states $s_t = x_t$ with $P(x_t = 1) = P(x_t = 0) = 0.5$ and two actions 0 and 1. There is no reward taking action $a_t = 0$. When $a_t = 1$, the agent pays one unit to have a chance to win a two unit reward with probability q_x at state $x_t = x \in \{0, 1\}$. Therefore, $P(r_t = 1|x_t = x, a_t = 1) = q_x$ and $P(r_t = -1|x_t = x, a_t = 1) = 1 - q_x$. This simple environment is a contextual bandit problem where the optimal policy is to pick $a_t = 1$ at state $x_t = x$ if $q_x > 0.5$, and $a_t = 0$ if $q_x \leq 0.5$. If the goal is to find an approximately optimal policy, the agent should take action $a_t = 1$ during training to learn the probabilities q_0 and q_1. Suppose the environment is subjected to observation black-out $x'_t = 0$ with $p^{\mathcal{I}} = 0.2$ when $x_t = 1$, and no interference when $x_t = 0$. Assume $q_0 = (3 - q_1)/5$. Then we have $P(r_t = 1|x'_t = 1, a_t = 1) = q_1$, and $P(r_t = 1|x'_t = 0, a_t = 1) = q_0 P(x_t = 0|x'_t = 0) + q_1 P(x_t = 1|x'_t = 0) = 0.5$. If the agent only has the interfered observation x'_t, the samples for $x'_t = 0$ are irrelevant to learning q_1 because rewards just randomly occur with probability half given $x'_t = 0$. Therefore, the sample complexity bound is proportional to $1/P(x'_t = 1)$ because only samples with $x'_t = 1$ are relevant. On the other hand, if the agent has access to the labels i_t during training, even when observed $x'_t = 0$, the agent can infer whether $x_t = 1$ by checking $i_t = 1$ or not. Therefore, the causal relation allows the agent to learn q_1 by utilizing all samples with $x_t = 1$, and the sample complexity bound is proportional to $1/P(x_t = 1) = 2$ which is a 20% reduction from $1/P(x'_t = 1) = 2.5$ when the labels are not available.

Note that $z_t = (x_t, i_t)$ is a latent state for this example, and the latent state and its causal relation is very important to improving learning performance.

D Causal Effects

In a causal learning setting, evaluating treatment effects and conducting statistical refuting experiments are essential to support the underlying causal graphical model. Through resilient reinforcement learning framework, we could interpret DQN by estimating the average treatment effect (ATE) of each noisy and adversarial observation. We first define how to calculate a treatment effect in the resilient RL settings and conduct statistical refuting tests including random common cause variable test (T_c), replacing treatment with a random (placebo) variable (T_p), and removing a random subset of data (T_s). The open-source causal inference package Dowhy (Sharma et al., 2019) is used for analysis.

D.1 Average Treatment Effect under Intervention

We refine a Q-network with discrete actions for estimating treatment effects based on Theorem 1 in (Louizos et al., 2017). In particular, individual treatment effect (ITE) can be defined as the

difference between the two potential outcomes of a Q-network; and the average treatment effect (ATE) is the expected value of the potential outcomes over the subjects. In a binary treatment setting, for a Q-value function $Q_t(s_t)$ and the interfered state $\mathcal{I}(s_t)$, the ITE and ATE are calculated by:

$$Q_t^{ITE} = Q_t(s_t)\,(1 - p_t) + Q_t(\mathcal{I}(s_t))p_t \tag{12}$$

$$ATE = \sum_{t=1}^{\mathcal{T}} \frac{\mathbb{E}\left[Q_t^{ITE}(\mathcal{I}(s_t))\right] - \mathbb{E}\left[Q_t^{ITE}(s_t)\right]}{\mathcal{T}} \tag{13}$$

where p_t is the estimated inference label by the agent and $\mathcal{T}$ is the total time steps of each episode. As expected, we find that CIQ indeed attains a better ATE and its significance can be informed by the refuting tests based on T_c, T_p and T_s.

To evaluate the causal effect, we follow a standard refuting setting (Rothman & Greenland, 2005; Pearl et al., 2016; Pearl, 1995b) with the causal graphical model in Fig. 3 of the main context to run three major tests, as reported in Tab. 13. The code for the statistical test was conducted by Dowhy Sharma et al. (2019), which has been submitted as supplementary material. (We intend to open source as a reproducible result.)

Pearl (Pearl, 1995a) introduces a "do-operator" to study this problem under intervention. The do symbol removes the treatment $\mathbf{tr}$, which is equal to interference $\mathcal{I}$ in the Eq. (1) of the main content , from the given mechanism and sets it to a specific value by some external intervention. The notation $P(r_t|do(tr))$ denotes the probability of reward r_t with possible interventions on treatment at time t. Following Pearl's back-door adjustment formula (Pearl, 2009) and the causal graphical model in Figure 2 of the main content., it is proved in (Louizos et al., 2017) that the causal effect for a given binary treatment $\mathbf{tr}$ (denoted as a binary interference label i_t in Eq. (1) of the main content), a series of proxy variables $\mathbf{X} = (\sum_{t=1}^{\mathcal{T}} x_t) \equiv \mathbf{S}' = (\sum_{t=1}^{\mathcal{T}} s'_t)$, as s'_t in Eq. (1) of the main content, a summation of accumulated reward $\mathbf{R} = (\sum_{t=1}^{\mathcal{T}} \mathbf{r_t})$ and a confounding variable $\mathbf{Z}$ can be evaluated by (similarly for $\mathbf{tr} = 0$):

$$\begin{aligned} &p(\mathbf{R}|\mathbf{S}', do(\mathbf{tr}=1)) = \textstyle\int_{\mathbf{Z}} p(\mathbf{R}|\mathbf{S}, do(\mathbf{tr}=1), \mathbf{Z})p(\mathbf{Z}|\mathbf{S}, do(\mathbf{tr}=1))d\mathbf{Z} \overset{(i)}{=} \\ &\textstyle\int_{\mathbf{Z}} p(\mathbf{R}|\mathbf{S}', \mathbf{tr}=1, \mathbf{Z})p(\mathbf{Z}|\mathbf{S}')d\mathbf{Z}, \end{aligned} \tag{14}$$

where equality (i) is by the rules of do-calculus (Pearl, 1995a; Pearl et al., 2016) applied to the causal graph applied on Figure 3 (a) of the main content. We extend to Eq. 14 on individual outcome study with DQNs, which is known by the Theorem 1. from Louizos et. al. (Louizos et al., 2017) and Chapter 3.2 of Pearl (Pearl, 2009).

D.2 Refutation Test:

A sampling plan for collecting samples refer to as subgroups (i=1, ..., k). Common cause variation (T-c) is denoted as σ_c, which is an estimate of common cause variation within the subgroups in terms of the standard deviation of the within subgroup variation:

$$\sigma_c \cong \sum_{i=1}^{k} s_i/k, \tag{15}$$

where k denotes as the number of sample size. We introduce intervention a error rate n, which is a probability to feed error interference (e.g., feed i_t = 0 even under interference with a probability of n) and results shown in Table 12.

The test (T-p) of replacing treatment with a random (placebo) variable is conducted by modifying the graphical relationship in the proposed probabilistic model in Fig. 3 of the main context. The new assign variable will follow the placebo note but with a value sampling from a random Gaussian distribution. The test of removing a random subset of data (T-r) is to randomly split and sampling the subset value to calculate an average treatment value in the proposed graphical model. We use the official dowhy[7] implementation, which includes: (1) confounders effect on treatment: how the simulated confounder affects the value of treatment; (2) confounders effect on outcome: how the

[7]Source:github.com/microsoft/dowhy/causal_refuters

simulated confounder affects the value of outcome; (3) effect strength on treatment: parameter for the strength of the effect of simulated confounder on treatment, and (4) effect strength on outcome: parameter for the strength of the effect of simulated confounder on outcome. Following the refutation experiment in the CEVAE paper, we conduct experiments shown in Tab. S12 and S13 with 10 % to 50 % intervention noise on the binary treatment labels. The results in Tab. S12 show that proposed CIQ maintains a lower rate compared with the benchmark methods included logistic regression and CEVAE (refer to Fig. 4 (b) in (Louizos et al., 2017)).

Through Eq. (9) to (10) and the corresponding correct action rate in the main context, we could interpret deep q-network by estimating the average treatment effect (ATE) of each noisy and adversarial observation. ATE (Louizos et al., 2017; Shalit et al., 2017) is defined as the expected value of the potential outcomes (e.g., disease) over the subjects (e.g., clinical features.) For example, in navigation environments, we could rank the harmfulness of each noisy observation (Yang et al., 2019) against q-network from the autonomous driving agent.

Table 12: Absolute error ATE estimate; lower value indicates a much stable causal inference under perturbation on logic direction with $P^I = 10\%$ and n=error rate of intervention on the binary label.

Model	n=0.1	n=0.2	n=0.3	n=0.4	n=0.5
LR	0.062	0.084	0.128	0.151	0.164
CEVAE	0.021	0.042	0.062	0.072	0.081
CIQ	**0.019**	**0.020**	**0.015**	**0.018**	**0.023**

Table 13: Validation of causal effect by three causal refuting tests. The causal effect estimate is tested by random common cause variable test (T-c), replacing treatment with a random (placebo) variable (T-p – lower is better), and removing a random subset of data (T-r). Adversarial attack outperforms in most tests.

Noise : $do(\mathcal{I})$	n = 0.1				n = 0.2			
Method	ATE	w/ T-c	w/ T-p	w/ T-s	ATE	w/ T-c	w/ T-p	w/ T-s
Adversarial	0.2432	0.2431	0.0294	0.2488	0.0868	0.0868	0.0109	0.0865
Black-out	0.2354	0.2212	0.0244	0.2351	0.0873	0.0870	0.0140	0.0781
Gaussian	0.1792	0.1763	0.0120	0.1751	0.0590	0.0610	0.0130	0.0571
Frozen Frame	0.1614	0.1614	0.0168	0.1435	0.0868	0.0868	0.0140	0.0573

E ABLATION STUDIES

E.1 THE NUMBER OF MODEL PARAMETERS

We also spend efforts on a parameter-study on the results of average returns between different DQN-based models, which included DQN, Double DQN (DDQN), DDQN with dueling, CIQ, DQN with a classifier (DQN-CF), DDQN with a classifier (DDQN-CF), DQN with a variational autoencoder (Kingma & Welling, 2013) (DQN-VAE), NoisyNet, and using the latent input of causal effect variational autoencoder for Q network (CEVAE-Q) prediction. Overall, CEVAE-Q is with minimal-requested parameters with 14.4 M (in Env_1) as the largest model used in our experiments in Tab. 14. CIQ remains roughly similar parameters as 9.7M compared with DDQN, DDQN_d, and Noisy Net. Our ablation study in Tab. 14 indicates the advantages of CIQ are not owing to extensive features using in the model according to the size of parameters. CIQ attains benchmark results in our resilient reinforcement learning setting compared to the other DQN models.

Noisy Nets (Fortunato et al., 2018) has been introduced as a benchmark whose parameters are perturbed by a parametric noise function. We select Noisy Net in a DQN format as a noisy training baseline with interfered state s'_t concated with a interference label i_t from a classifier.

E.2 LATENT REPRESENTATIONS

We conduct an ablation study by comparing other latent representation methods to the proposed CIQ model.

Table 14: Ablation study on parameter of different DQN models using in our experiments in Env_1, Env_2, Env_3, and Env_4. The minimal parameters of each model denote as Para. in the Table 9.

Model	Para.	Env_1	Env_2	Env_3	Env_4
DQN	6.9M	20.2	3.1	-113.6	10.8
DDQN	9.7M	41.1	3.5	-123.4	57.9
DDQN_d	9.7M	82.9	4.7	-136.3	67.2
CIQ	9.7M	**195.1**	**12.5**	**200.1**	**195.2**
DQN-CF	9.7M	140.5	**12.5**	-78.3	120.2
DDQN-CF	12.1M	161.3	**12.5**	-10.1	128.2
DQN-VAE	9.7M	151.1	7.6	-92.9	24.1
NoisyNet	9.7M	158.6	5.5	50.1	100.1
CEVAE-Q	12.5M	39.8	11.5	-156.5	45.8
DVRLQ-CF	10.7M	107.11	9.2	-34.9	42.5

DQN with an variational autoencoder (DQN-VAE): To learn important features from observations, many recent works leverage deep variational inference for accessing latent states for feeding into DQN. We provide a baseline on training a variational autoencoder (VAE) built upon the DQN baseline, denoted as DQN-VAE. The DQN-VAE baseline is targeted to recover a potential noisy state and feed the bottleneck latent features into the Q-network.

CEVAE-Q Network: TARNet (Shalit et al., 2017; Louizos et al., 2017) is a major class of neural network architectures for estimating outcomes of a binary treatment on linear data (e.g., clinical reports). Our proposed CIQ uses an end-to-end approach to learn the interventional (causal) features. We provide another baseline on using the latent features from a causal variational autoencoder (Louizos et al., 2017) (CEVAE) as latent features as state inputs followed the loss function in (Louizos et al., 2017). To get the causal latent model in Q-network, we approximate the posterior distribution by a neural network $z_t \sim p(z_t|\tilde{x}_t) = \phi(\tilde{x}_t; \theta_1)$. Then we train this neural network, CEVAE-Q, by variational inference using the generative model.

We conduct 10,000 times experiments and fine-tuning on DQN-VAE and CEVAE-Q. The results in Table 15 shows that the latent representation learned by CIQ provides better resilience than other representations.

Table 15: Performance on average return in clean and five different noise level in Env_1 evaluated by an average of under uncertain perturbation included Gaussian, adversarial, blackout, and frozen frame. All DQN models solve the environment with over 195.0 average returns in a clean state input (a.k.a. no noise).

Model	0%	10%	20%	30%	40%	50%	Cosine	Para.
DQN	195.1	115.0	68.9	32.3	22.8	19.1	42.1	6.9 M
DDQN	195.1	123.4	73.2	59.4	28.1	22.8	62.8	9.7 M
CIQ	195.1	**195.1**	**195.1**	**195.0**	**168.2**	**113.1**	**195.0**	9.7 M
DQN-VAE	195.1	173.5	141.3	124.8	86.5	33.3	101.2	9.7 M
DQN-CEVAE	195.1	154.4	111.9	94.8	75.5	48.3	82.1	12.5 M

Table 16: Structure-wise ablation studies of CIQ in Env_1 (noise level $P = 20\%$).

Model	Return	CLEVER-Q	AC-Rate
CIQ	195.1	0.241	97.3
B1: CIQ w/o the concatenation	152.1	0.196	78.2
B2: CIQ w/o the θ_3 network	150.1	0.182	65.6
B3: CIQ w/o providing grounded i_t for training	135.1	0.142	53.6

E.3 Architecture Ablation Study on CIQ

To study the importance of specific components in CIQ, we conducted additional ablation studies and constructed two new baseline models shown in Table 16 tested in Env_1 (Cartpole). Baseline

1 (B1) - CIQ w/o the concatenation of $\tilde{i}_t$ in S_I^C. This comparison shows the importance of using both the predicted confounder $\tilde{z}_t$ and the predicted label $\tilde{i}_t$. B1 uses label prediction to help latent representation but not using the predicted labels in decision-making. The structure is motivated by a task-specific (depth-only information from a maze environment) DQN network from a previous study (Mirowski et al., 2016). Baseline 2 (B2) - CIQ w/o the θ_3 network (for testing θ_3's importance) The structure (Humplik et al., 2019) is motivated by a meta-inference reinforcement learning proposed by.Baseline 3 (B3) - CIQ w/o providing grounded i_t for training, for testing the importance of the inference loss and joint loss propagation. The superior performance of CIQ validates the proposed model is indeed crucial from the previous discussion in Section 3 of the main content.The setting used for Table 16 is the same as the setting for the third column (noise level = 20%) in Table 5 and the third column (noise level = 20%) in Table 15, tested in Env_1 (Cartpole).

E.4 Perturbation-based Neural Saliency for DQN Agents

To better understand our CIQ model, we use the benchmark saliency method on DQN agent, perturbation-based saliency map, (Greydanus et al., 2018) to visualize the salient pixels, which are sensitive to the loss function of the trained DQNs. We made a case study of an input frame under an adversarial perturbation, as shown in Fig. 6. We evaluate DQN agents included DQN, CIQ, DQN-CF, DVRLQ-CF and record its weighted center from the neural saliency map, where saliency pixels of CIQ respond to ground true actions more frequent (96.2%) than the other DQN methods.

E.5 Robustness Transferability among Different Interference Types

We conduct additional experiments to study robustness transferability of DQN and CIQ when training and testing under different kinds of interference types in Env_1. Note that both architectures would solve a clean environment successfully (over 195.0). The reported numbers are averaged over 20 independent runs for each condition. As shown in Table 17 and Table 18, CIQ agents consistently attain significant performance improvement when compared with DQN agents, especially between Gaussian and adversarial perturbation. For example, CIQ succeeded to solve the environment 12 times out of 20 independent runs, with an average score of 165.2 in Gaussian (train)-Adversarial (test) adaptation. n particular, for CIQ, 12 times out of 20 independent runs are successfully transfered from Gaussian to Adversarial perturbation. Interestingly, augmenting adversarial perturbation does not always guarantee the best policy transfer when testing in the Blackout and Frozen conditions, which shows a slightly lower performance compared with training on Gaussian interference. The reason could be attributed to the recent findings that adversarial training can undermine model generalization (Raghunathan et al., 2019; Su et al., 2018).

Table 17: DQN adaptation: train and test on different interference (noise level $P = 20\%$) in Env_1.

Train / **Test**	**Gaussian**	**Adversarial**	**Blackout**	**Frozen**
Gaussian	67.4	38.4	43.7	52.1
Adversarial	53.2	42.5	35.3	44.2
Blackout	46.2	27.4	85.7	50.3
Frozen	62.3	26.2	45.9	62.1

Table 18: CIQ adaptation: train and test on different interference (noise level $P = 20\%$) in Env_1.

Train / **Test**	**Gaussian**	**Adversarial**	**Blackout**	**Frozen**
Gaussian	**195.1**	165.2	158.2	167.8
Adversarial	162.8	**195.0**	152.4	162.5
Blackout	131.3	121.1	**195.3**	145.7
Frozen	161.6	135.8	147.1	**195.2**

E.6 CIQ with Multi-Interference.

Here we show how the proposed CIQ model can be extended from the architecture shown in Figure 3 to the multi-interference (MI) setting. The design intuition is based on two-step inference by a

common encoder, to infer a clean or noisy observation, followed by an individual decoder tied to an interference type, to infer noisy types and activate the corresponding Q-network (named θ_4).
Note that the two-step inference mechanism follows the RCM as two sequential potential outcome estimation models (Rubin, 1974; Imbens & Rubin, 2010), where interfered observation x'_t is determined by two labels $i_{1,t}$ and $i_{2,t}$ according to $x'_t = i_{1,t}(i_{2,t}\mathcal{I}_1(x_t)+(1-i_{2,t})\mathcal{I}_2(x_t))+(1-i_{1,t})x_t$ extended from Eq.(1), where $i_{1,t}$ indicates the presence of interference and $i_{2,t}$ indicates which interference type (here we show the case of two types). As a proof of concept, we consider two interference types together, Gaussian noise and adversarial perturbation. In this setting every observation (state) can possibly undergo an interference with either Gaussian noise or Adversarial perturbation. From the results shown in Table. 19, we find that the extended version of CIQ, CIQ-MI, is capable of making correct action to solve (over 195.0) the environment when training with mixed interference types (last row). Another finding is that robustness tranferability (153.9/154.2) in CIQ-MI is slightly degraded compared to the results (162.8/165.2) in Table 18 with the same training episodes (500) and runs (20), which could be caused by the increased requirement of model capacity (Ammanabrolu & Riedl, 2019; Yang et al., 2020b) of CIQ-MI.

Table 19: CIQ-MI: CIQ agent with an extended multi-interference (MI) architecture testing in Env_1 (noise level $P = 20\%$).

Train / **Test**	**Gaussian**	**Adversarial**	**Gaussian + Adversarial**
Gaussian	**195.1**	154.2	96.3
Adversarial	153.9	**195.0**	105.1
Gaussian + Adversarial	**195.0**	**195.0**	**195.0**

E.7 Hardware Setup and Energy Cost

We use Nvidia GPUs (2080-Ti and V100) for our experiments with Compute Unified Device Architecture (CUDA) version 10.1. To conduct the results shown in the paper, it takes around 20 min to run 1,000 epochs (maximum) with a batch size 32 for each environment considering the hyperparameters tuning described in Section C.1 of the main paper. In total, the experiments presented (four environment with four different types of noise and its ablation studies) in this paper took around 343 computing hours with a 300W power supplier.