Representation-Based Robustness in Goal-Conditioned Reinforcement Learning

Xiangyu Yin[1*], Sihao Wu[1*], Jiaxu Liu[1], Meng Fang[1], Xingyu Zhao[2], Xiaowei Huang[1], Wenjie Ruan[1†]

[1] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
[2] WMG, University of Warwick, Coventry, CV4 7AL, UK
{X.Yin22;sihao.wu;Meng.Fang;xiaowei.huang}@liverpool.ac.uk, xingyu.zhao@warwick.ac.uk, w.ruan@trustai.uk

Abstract

While Goal-Conditioned Reinforcement Learning (GCRL) has gained attention, its algorithmic robustness against adversarial perturbations remains unexplored. The attacks and robust representation training methods that are designed for traditional RL become less effective when applied to GCRL. To address this challenge, we first propose the *Semi-Contrastive Representation* attack, a novel approach inspired by the adversarial contrastive attack. Unlike existing attacks in RL, it only necessitates information from the policy function and can be seamlessly implemented during deployment. Then, to mitigate the vulnerability of existing GCRL algorithms, we introduce *Adversarial Representation Tactics*, which combines *Semi-Contrastive Adversarial Augmentation* with *Sensitivity-Aware Regularizer* to improve the adversarial robustness of the underlying RL agent against various types of perturbations. Extensive experiments validate the superior performance of our attack and defence methods across multiple state-of-the-art GCRL algorithms. Our code is available at https://github.com/TrustAI/ReRoGCRL.

Introduction

Goal-Conditioned Reinforcement Learning (GCRL) trains an agent to learn skills in the form of reaching distinct goals. Unlike conventional RL, GCRL necessitates the agent to make decisions that are aligned with goals. This attribute allows agents to learn and accomplish a variety of tasks with shared knowledge, better generalization, and improved exploration capabilities. Furthermore, it is binary-rewarded, which is easier to implement compared to hand-crafted complex reward functions. Recently, there has been a significant surge in research related to GCRL. Exemplary works include methods based on techniques such as hindsight experience replay (Andrychowicz et al. 2018; Fang et al. 2018, 2019), imitation learning (Ghosh et al. 2020; Yang et al. 2021b), or offline learning (Chebotar et al. 2021; Ma et al. 2022; Mezghani et al. 2023).

Generally, these studies tend to assume that the sensing and perception systems of agents are devoid of uncertainties. However, this presumption is hardly applicable in real-world scenarios because there are known gaps between simulated and real-world environments, such as measurement errors, motor noise, etc. The observations made by agents encompass unavoidable disturbances that originate from unforeseeable stochastic noises or errors in sensing (Huang et al. 2020; Zhang, Ruan, and Xu 2023; Wang et al. 2022; Huang, Jin, and Ruan 2023). For example, due to the existence of adversarial attacks in RL agents (Huang et al. 2017; Weng et al. 2019; Bai, Guan, and Wang 2019; Mu et al. 2023a,b), even minor perturbations can lead to unsafe outcomes in safety-critical GCRL control strategies (Gleave et al. 2019; Sun et al. 2020).

Recently, numerous robust techniques have been implemented in traditional RL, which can be broadly classified into two categories: *i)* Adversarial Training (AT), and *ii)* Robust Representation Training. Particularly, AT in RL trains agents with adversarial states or actions (Pinto et al. 2017; Kos and Song 2017; Zhang et al. 2020b), and then enhances their adversarial robustness. However, AT-trained RL agents cannot be utilized across downstream tasks, which limits their transferability. Compared to AT, robust representation training can deliver *low-dimensional*, *collapse-resistent*, and *perturbation-robust* representations of observations (Gelada et al. 2019; Zhang et al. 2020a; Zang, Li, and Wang 2022), which is the focus of recent studies. Specifically, it learns representations that capture only task-relevant information based on the bisimulation metric of states (Ferns, Panangaden, and Precup 2011). Technically, this is achieved by reducing the *behavioural difference* between representations of similar observation pairs in the latent space.

Although there are already many articles enhancing the robustness of RL, few works (He and Lv 2023) consider the robustness of GCRL against adversarial perturbations. Different from the vanilla one, GCRL employs reward functions characterized by sequences of unshaped binary signals. For example, (Andrychowicz et al. 2018) allocates the reward by determining whether the distance between the achieved goal $\hat{g}$ and the desired goal g is less than a threshold ϵ: $\mathbb{1}(\|\hat{g} - g\| \leq \epsilon)$. This makes the reward sequences in GCRL notably sparser than those in traditional RL. The sparsity of rewards in GCRL leads to inaccurate estimations of both Q-values and actions. This presents a significant challenge for directly implementing conventional RL attacks, particularly those that rely on pseudo labels such as Q-values or

*These authors contributed equally.
†Corresponding Author

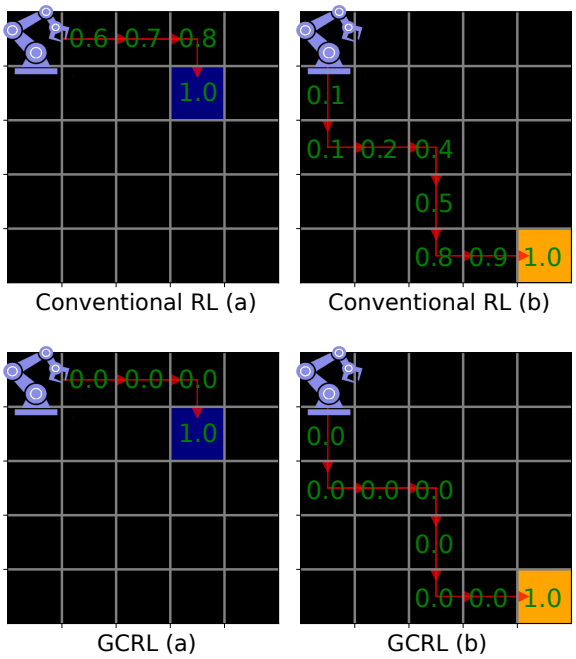

Figure 1: Trajectories of the agent at state s approaching blue and orange goals in conventional RL and GCRL, where the designated goals vary with different initialization. Rewards are indicated in each block along the trajectories.

action values.Consequently, this underscores the necessity to develop new attack methods tailored for GCRL to evaluate its robustness. Inspired by adversarial contrastive learning (Kim, Tack, and Hwang 2020; Jiang et al. 2020; Ho and Nvasconcelos 2020), we propose the Semi-Contrastive Representation (SCR) attack, with the aim of maximizing the distance between the representations of original states and their corresponding perturbed versions. This approach effectively bypasses the need for approximated labels. Notably, this method can operate independently of the critic function and is readily deployable.

Contrary to conventional RL, where state rewards approach 1.0 as the agent nears the goal, in GCRL, rewards consistently remain at 0.0 until the goal is reached. This distinction is highlighted in Fig. 1: for a given state s, the rewards of subsequent states in Conventional RL (a) significantly differ from those in Conventional RL (b). However, such a disparity in rewards is not present in GCRL, where they invariably stay at 0.0. Consequently, bisimulation metric-based representation training methods cannot capture well the differences between state-goal tuples in GCRL. To deal with this, we propose a universal defensive framework named Adversarial Representation Tactics (ARTs). Firstly, ARTs enhance the robustness of vanilla GCRL algorithms with Semi-Contrastive Adversarial Augmentation (SCAA). Secondly, ARTs mitigate the performance degradation associated with bisimulation metric-based robust representation techniques using the Sensitivity-Aware Regularizer (SAR). In summary, our contributions are summarized as follows:

- We introduce a novel Semi-Contrastive Representation (SCR) attack. It can operate without the critic function and is ready for direct deployment.
- We propose a mixed defensive strategy, termed Adversarial Representation Tactics (ARTs), to dynamically enhance adversarial robustness tailored to specific GCRL algorithms.
- Extensive experiments validate that our proposed attack method and defence techniques outperform state-of-the-art algorithms in GCRL by a large margin.

Related Work

Goal-Conditioned Reinforcement Learning

Several methods have tackled GCRL. Hindsight experience replay (Andrychowicz et al. 2018) is often employed to enhance the effectiveness of policies by relabeling the goal. Goal-conditioned supervised learning approaches (Ghosh et al. 2020; Yang et al. 2021b) solve this problem by iteratively relabeling and imitating self-generated experiences. For long-horizon tasks, hierarchical RL (Chane-Sane, Schmid, and Laptev 2021) learns high-level policies to recursively estimate a sequence of intermediate subgoals. Moreover, model-based methods (Charlesworth and Montana 2020; Yang et al. 2021a), self-supervised learning (Mezghani et al. 2022; Eysenbach et al. 2022), and planning-based methods (Eysenbach, Salakhutdinov, and Levine 2019; Nasiriany et al. 2019) are also used to solve GCRL problems. Unlike these prior methods, our paper considers the representation training and robustness performance on GCRL.

Adversarial Contrastive Attack

Adversarial contrastive attacks have gained significant attention in recent studies. Particularly, (Kim, Tack, and Hwang 2020) proposes the instance-wise adversarial attack for unlabeled data, which makes the model confuse the instance-level identities of the perturbed data samples. (Jiang et al. 2020) directly exploits the Normalized Temperature-scaled Cross Entropy loss to generate PGD attacks. (Ho and Nvasconcelos 2020) introduces a new family of adversarial examples by backpropagating the gradients of the contrastive loss to the network input. The above methodologies consistently rely on the cosine similarity among the original sample, positive, and negative samples. In this paper, however, we devise a novel contrastive attack method, which is more suitable in GCRL and can be extended to other RL settings.

Bisimulation Metric

Bisimulation metrics offer a framework to gauge the similarity between states within MDPs. The traditional bisimulation metric, as defined by (Ferns, Panangaden, and Precup 2012), considers two states to be close if their immediate rewards and transition dynamics are similar (Larsen and Skou 1989; Givan, Dean, and Greig 2003). This self-referential concept has been mathematically formalized using the Kantorovich distance, leading to a unique fixed-point definition.

However, the conventional approach has been criticized for resulting in *pessimistic* outcomes, as it requires consideration of all actions, including those that may be suboptimal. To address this challenge, (Castro et al. 2022) introduced the strategy of Matching Under Independent Couplings (MICo). This approach focuses solely on actions induced by a specific policy π, offering a more optimistic and tailored measure of state equivalence. Whereas, in this work, we focus on their extended version, Simple State Representation (SimSR).

Background

In this section, we first present the fundamental concepts of goal-conditioned MDPs and delve into their extensions in adversarial state-goal scenarios. Then, we outline the architecture of our training backbone and describe SimSR, the cutting-edge robust representation training method.

Goal-Conditioned MDPs

In this section, we model our task as a goal-conditioned MDP. This can be formalized as a tuple $(\mathcal{S}, \mathcal{G}, \mathcal{Z}, \mathcal{A}, r, \gamma, \mathcal{T})$. Here, $\mathcal{S}$ represents the state space, $\mathcal{G}$ the goal space, $\mathcal{Z}$ the latent representation space, and $\mathcal{A}$ the action space. The reward function is denoted by r, γ stands for the discount factor, and $\mathcal{T}$ signifies the state-goal transition probability function.

We assume that the policy function π can be approximated using a combination of an encoder, $\psi(\cdot)$, and an actor-network, $\varphi(\cdot)$. The critic function is represented by $\varrho(\cdot)$. Specifically, for any given state $s \in \mathcal{S}$ and goal $g \in \mathcal{G}$, the encoder $\psi(\cdot)$ transforms the concatenated tuple $\langle s, g\rangle$ into the representation space $\mathcal{Z}$. The resulting feature, $z_{\langle s,g\rangle}$, is expressed as $\psi(\langle s, g\rangle)$. The actor network, $\varphi(\cdot)$, maps this feature, $z_{\langle s,g\rangle}$, to a specific action $a_{\langle s,g\rangle} \in \mathcal{A}$. The action distribution is defined as $a_{\langle s,g\rangle} \sim \pi(\cdot|\langle s, g\rangle)$. Based on the chosen action $a_{\langle s,g\rangle}$, the agent's subsequent state, s', can be sampled using $s' \sim \mathcal{T}(\cdot|\langle s, g\rangle, a_{\langle s,g\rangle})$ and succinctly represented as $\mathcal{T}^{\pi}_{\langle s,g\rangle}$.

In contrast to traditional RL algorithms, GCRL only provides rewards when the agent successfully achieves a predefined goal. Specifically, we define the reward for a state-goal tuple $\langle s, g\rangle$ as $r_{\langle s,g\rangle} = \mathbb{1}(\mathcal{D}(s, g) \leq \eta)$. Here, η is a predefined threshold, $\mathcal{D}$ acts as a distance metric, determining if s and g are sufficiently close. We utilize ℓ_∞-norm in this paper. Given an initial state s_0 and a goal g, our primary objective is to optimize the expected cumulative rewards across joint distributions following:

$$J_\pi(s_0, g) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot|\langle s_t, g\rangle), \\ s_{t+1} \sim \mathcal{T}^{\pi}_{\langle s_t, g\rangle}}} \left[\sum_{t=0}^{T-1} \gamma^t r_{\langle s_t, g\rangle} \right]. \quad (1)$$

In GCRL, the reward function yields only two possible outcomes: 0 or 1, which results in a notably sparser reward sequence $(r_{\langle s_0,g\rangle}, \cdots, r_{\langle s_{T-1},g\rangle})$ over a T-step trajectory compared to other RL algorithms. While these binary rewards may not offer granular insights into the agent-environment interactions, they do guide the agent in a strategy where it initially distances itself from the goal before converging towards it. To derive the optimal policy π^*, we refine Eq. (1) using the Q-value at the initial state s_0, then $Q_\pi(\langle s_0, g\rangle, a_0)$ can be estimated through the Bellman equations:

$$r_{\langle s_0,g\rangle} + \mathbb{E}_{s_1 \sim \mathcal{T}^{\pi}_{\langle s_0,g\rangle}} \max_{a_1 \sim \pi(\cdot|\langle s_1,g\rangle)} Q_\pi(\langle s_1, g\rangle, a_1). \quad (2)$$

Definition 1 ((Zhang et al. 2020b)) *The adversarial version of a state s can be defined as $\mathcal{V}(s;\theta)$. It is considered stationary, deterministic, and Markovian if its behaviour is solely determined by s and the policy network, which is parameterized by θ. For simplicity, we denote $\mathcal{V}(s;\theta)$ as $\mathcal{V}(s)$.*

State-Goal Adversarial Observations

Building on Def. 1, we consider states and goals separately and individually construct the set of adversarial states $\mathcal{B}^\epsilon_p(s)$ and goals $\mathcal{B}^\epsilon_p(g)$ using the State-Adversarial MDP (SA-MDP) framework:

$$\begin{aligned} \mathcal{B}^\epsilon_p(s) &\coloneqq \{\mathcal{V}(s) \mid \|\mathcal{V}(s) - s\|_p \leq \epsilon_s\}, \\ \mathcal{B}^\epsilon_p(g) &\coloneqq \{\mathcal{V}(g) \mid \|\mathcal{V}(g) - g\|_p \leq \epsilon_g\}, \end{aligned} \quad (3)$$

where ϵ_s and ϵ_g are predefined thresholds of adversarial perturbations on states and goals respectively. $\mathcal{V}(\cdot)$ acts as a stationary and deterministic mapping, which transforms a clean input into its adversarial counterpart. Note that the adversary introduces perturbations solely to the state-goal observations. Consequently, even though the action is determined as $a \sim \pi(\cdot|\langle \mathcal{V}(s), \mathcal{V}(g)\rangle)$, transitions of the environment are still governed by the original state-goal pair $\langle s, g\rangle$, instead of its perturbed counterpart $\langle \mathcal{V}(s), \mathcal{V}(g)\rangle$. Due to uncertainties in the state-goal estimation, there is a potential for generating actions that are not optimal.

Neural Network-based Backbone

In this study, we utilize Multi-Layer Perceptrons (MLPs) of varying depths as the foundational architectures for both the policy function, represented as $\varphi(\psi(\cdot))$, and the critic function, $\varrho(\cdot)$. Specifically, for an encoder $\psi(\cdot)$ comprising L layers, its output, o_L, can be articulated as $o_L = \mathbf{W}_L\phi(\mathbf{W}_{L-1}\cdots\phi(\mathbf{W}_1 o_0))$, with the stipulation that $L \geq 2$. Here, o_0 refers to the input vector in its flattened form, $\phi(\cdot)$ represents the ReLU-based activation function, and $\mathbf{W}_i$ is the weight matrix associated with the i-th layer. To be simplified, we denote the parameters and the depths of $\psi(\cdot)$, $\varphi(\cdot)$, and $\varrho(\cdot)$ as θ_ψ, θ_φ, θ_ϱ, and L_ψ, L_φ, L_ϱ, respectively. For clarity, we omit the bias vector present at each layer.

Simple State Representation (SimSR)

In this paper, SimSR functions as a metric within the representation space, enhancing our base methods in GCRL. Unlike the bisimulation metric (Ferns, Panangaden, and Precup 2012) or MICo (Castro et al. 2022), which employ the Wasserstein or diffuse metric to gauge the distance between two distributions, SimSR offers a more relaxed approach. Take input tuples $\langle s_i, g_1\rangle$ and $\langle s_j, g_2\rangle$ from Fig. 1 for instance, the measure $\mathcal{M}(\cdot,\cdot)$ used to quantify the relationship between their representations, is derived using the co-

sine distance. This metric is fundamentally based on the Euclidean normalized dot product distance, formulated as:

$$\mathcal{M}(\langle s_i, g_1\rangle, \langle s_j, g_2\rangle) = 1 - \frac{\psi(\langle s_i, g_1\rangle)^\top \psi(\langle s_j, g_2\rangle)}{\|\psi(\langle s_i, g_1\rangle)\|_2 \|\psi(\langle s_j, g_2\rangle)\|_2} \quad (4)$$

Note that SimSR shares the same fixed point as MICo. Its update operator can be defined in a manner analogous to MICo.

Theorem 1 ((Zang, Li, and Wang 2022)) *Given a policy function π, the SimSR operator T^π which defines the state-goal tuple similarity between $\langle s_i, g_1\rangle$ and $\langle s_j, g_2\rangle$ can be updated as:*

$$(T^\pi\mathcal{M})(\langle s_i, g_1\rangle, \langle s_j, g_2\rangle) = |r_{\langle s_i,g_1\rangle} - r_{\langle s_j,g_2\rangle}| + \gamma\mathbb{E}_{s_{i+1}\sim\mathcal{T}^\pi_{\langle s_i,g_1\rangle}, s_{j+1}\sim\mathcal{T}^\pi_{\langle s_j,g_2\rangle}}\left[\mathcal{M}(\langle s_{i+1}, g_1\rangle, \langle s_{j+1}, g_2\rangle)\right], \quad (5)$$

for all $\mathcal{M} : \langle\mathcal{S}\times\mathcal{G}\rangle \times \langle\mathcal{S}\times\mathcal{G}\rangle \to \mathbb{R}$.

The second term in Eq. (5) is derived through sample-based approximation. Using the SimSR operator, one can iteratively refine the representation by applying it to an arbitrarily initialized $\psi(\cdot)$.

Note that by using cosine distance as the foundational metric in SimSR, all derived state features are normalized to unit length. Furthermore, in contrast to the bisimulation metric which employs the Wasserstein distance, the SimSR operator achieves a computational complexity on par with the MICo operator.

Semi-Contrastive Representation Attack

We present a representation-based adversarial attack for GCRL, which is independent of the critic function and can be seamlessly integrated in the deployment phase.

Limitations of Traditional Attacks in GCRL

Adversarial attacks on states lack the gradient information from labeled examples. To navigate these constraints, existing methods lean on various pseudo-labels, such as Q-values (Zhang et al. 2020b; Kos and Song 2017) or actions (Gleave et al. 2019; Sun et al. 2020) to generate adversarial states. However, as illustrated in Eq. (2), Q-value function is heavily influenced by the reward sequence. Moreover, many of these attacks necessitate access to value networks, making them unsuitable for direct deployment in the inference stage. To address these challenges, we introduce a novel attack method tailored for GCRL algorithms. This approach neither depends on specific pseudo-labels nor requires access to the critic network. Although crafted with GCRL's unique characteristics, it is versatile enough for broader applications of RL algorithms.

Negative Tuple-Based Adversary

To craft adversaries without relying on labels or pseudo-labels, several studies (Ho and Nvasconcelos 2020; Jiang et al. 2020; Kim, Tack, and Hwang 2020) have introduced the concept of adversarial contrastive attack in the realm of unsupervised learning. Viewed through the lens of representation, an adversary $\mathcal{V}(x)$ for a clean input x is designed to ensure that the feature diverges significantly from the one of a positive sample while converging towards the feature of a negative sample. Typically, these methods employ the NT-Xent loss to gauge the similarity among $f(\mathcal{V}(x))$, $f(x^+)$, and $f(x^-)$. This loss can be articulated as:

$$-\log \frac{\sum_{\{x^+\}} \exp\left(\mathbb{S}(\mathcal{V}(x), x^+)\right)}{\sum_{\{x^+\}} \exp\left(\mathbb{S}(\mathcal{V}(x), x^+)\right) + \sum_{\{x^-\}} \exp\left(\mathbb{S}(\mathcal{V}(x), x^-)\right)}, \quad (6)$$

where $\mathbb{S}(a, b) = \frac{f(a)^\top f(b)}{\tau}$, τ refers to the temperature hyperparameter. Normally, the positive set $\{x^+\}$ can be constructed using different types of data augmentation, such as rotation, color jittering, or scaling, while $\{x^-\}$ indicates the set of samples collected from other classes. In RL contexts, it is challenging to directly quantify positively correlated samples. Thus, in this section, we present an adversarial contrastive attack approach that exclusively focuses on the negative tuple, termed the Semi-Contrastive Representation (SCR) attack in Def. 2.

Definition 2 (SCR attack) *Given a feature extraction function $f(\cdot)$ and the input tuple $\langle s, g\rangle$, the semi-contrastive representation attack can be defined as follows:*

$$\underset{\mathcal{V}(s)\in\mathcal{B}^\epsilon_p(s), \mathcal{V}(g)\in\mathcal{B}^\epsilon_p(g)}{\arg\sup} \mathbb{E}_{\langle s,g\rangle^-}\left[\mathcal{L}_{lg}\left(-f(\langle\mathcal{V}(s), \mathcal{V}(g)\rangle)^\top f(\langle s, g\rangle^-)\right)\right], \quad (7)$$

where the logistic function $\mathcal{L}_{lg}(v) = \log(1 + \exp(-v))$ for any $v \in \mathbb{R}$, $\langle s, g\rangle^-$ indicates the negative tuple.

It is easy to derive that $\frac{\mathcal{L}_{lg}(v_1)+\mathcal{L}_{lg}(v_2)}{2} \geq \mathcal{L}_{lg}(\frac{v_1+v_2}{2})$, so we can apply the sub-additivity to the supremum of the values of $\mathcal{L}_{lg}$ by:

$$\begin{aligned}&\sup_{\mathcal{V}(s),\mathcal{V}(g)} \mathbb{E}_{\langle s,g\rangle^-}\left[\mathcal{L}_{lg}\left(-f(\langle\mathcal{V}(s), \mathcal{V}(g)\rangle)^\top f(\langle s, g\rangle^-)\right)\right] \\ &\leq \mathbb{E}_{\langle s,g\rangle^-} \sup_{\mathcal{V}(s),\mathcal{V}(g)}\left[\mathcal{L}_{lg}\left(-f(\langle\mathcal{V}(s), \mathcal{V}(g)\rangle)^\top f(\langle s, g\rangle^-)\right)\right].\end{aligned} \quad (8)$$

Consequently, we can reframe the challenge of computing the supremum over expectations into determining the expectation over the supremum. This means our goal is to optimize $\mathcal{L}_{lg}$ for a given negative tuple $\langle s, g\rangle^-$. Moreover, leveraging the non-increasing nature of $\mathcal{L}_{lg}(\cdot)$, we can approximate our attack objective by directly minimizing the similarity-based loss $-f(\langle\mathcal{V}(s), \mathcal{V}(g)\rangle)^\top f(\langle s, g\rangle^-)$. In this paper, $f(\cdot)$ is equivalent to $\psi(\cdot)$. Building on this, we present a Projected Gradient Descent (PGD)-based approximation for both $\mathcal{V}(s)$ and $\mathcal{V}(g)$.

Definition 3 (PGD-based Approximation) *Given an encoder $\psi(\cdot)$, an original input tuple $\langle s, g\rangle$, a pre-defined negative tuple $\langle s, g\rangle^-$, and the step size α, then the semi-contrastive representation attack $\mathcal{V}_{scr}(s)$ and $\mathcal{V}_{scr}(g)$ at the iterative step $i+1$ can be individually defined as:*

$$\mathcal{V}^{i+1}_{scr}(s) = \mathcal{V}^i_{scr}(s) - \alpha\nabla_{\mathcal{V}^i_{scr}(s)}\mathcal{L}_{sim}(s, g, i), \quad (9)$$

$$\mathcal{V}^{i+1}_{scr}(g) = \mathcal{V}^i_{scr}(g) - \alpha\nabla_{\mathcal{V}^i_{scr}(g)}\mathcal{L}_{sim}(s, g, i), \quad (10)$$

Particularly, we define:

$$\mathcal{L}_{sim}(s, g, i) = -\psi\left(\langle\mathcal{V}^i_{scr}(s), \mathcal{V}^i_{scr}(g)\rangle\right)^\top \psi(\langle s, g\rangle^-). \quad (11)$$

If the number of iteration steps is $\mathcal{I}(\geq 1)$, then the finally generated adversarial tuple can be defined as $\langle\mathrm{proj}(\mathcal{V}^I_{scr}(s)), \mathrm{proj}(\mathcal{V}^I_{scr}(g))\rangle$, where $\mathrm{proj}(\cdot)$ is the projection head.

In this work, We denote $proj(\cdot)$ in Def. 3 as an ϵ-bounded ℓ_∞-norm ball. In a straightforward yet effective manner, we construct the negative tuple $\langle s, g\rangle^-$ by performing negation operation, including $\langle -s, -g\rangle$, $\langle -s, g\rangle$ and $\langle s, -g\rangle$. By incorporating the SCR attack at each timestep within an episode of T steps in GCRL, our primary aim is to divert the agent, ensuring that it remains distant from the goal. This approach ensures that the reward sequence $(r_0, \cdots, r_{T-1})$ remains as sparse as possible.

To verify the efficacy of our approach, we train multiple agents using diverse GCRL algorithms and evaluate their adversarial robustness against our introduced attack. Comprehensive results are presented in the *Experiments* section.

Adversarial Representation Tactics

As illustrated in Tab. 1 and 2, GCRL agents trained using both base methods and SimSR-strengthened ones are vulnerable to our SCR attack, highlighting a significant security concern. To address this, we introduce a composite defensive strategy termed ARTs in this section, which strategically combine the Semi-Contrastive Adversarial Augmentation (SCAA) and the Sensitivity-Aware Regularizer (SAR), catering to diverse GCRL algorithms.

Semi-Contrastive Adversarial Augmentation

In Algorithm 1, we investigate the influence of data augmentations by subjecting input tuples to the SCR attack. Specifically, during each training epoch for the base GCRL agent, we retrieve a mini-batch of tuples from the replay buffer and generate semi-contrastive augmented samples, $\mathcal{V}_{scr}(s_t^i)$ and $\mathcal{V}_{scr}(g^i)$, using the constructed negative tuples. We then utilize the augmented critic value, denoted as $\hat{\varrho}$, and the actor value, represented as $\hat{\varphi}$, to refine the weights of the encoder (θ_ψ), actor network (θ_φ), and critic network (θ_ϱ). Notably, $\mathcal{L}_\psi$, $\mathcal{L}_\varphi$, and $\mathcal{L}_\varrho$ correspond to the loss functions for the encoder, actor network, and critic network, respectively.

Sensitivity-Aware Regularizer

Given two pairs of observations $\left(s_i, g_1, r_{\langle s_i, g_1\rangle}, a_i, s_{i+1}\right)$ and $\left(s_j, g_2, r_{\langle s_j, g_2\rangle}, a_j, s_{j+1}\right)$ shown in Fig. 1, it is common that $r_{\langle s_i, g_1\rangle} = r_{\langle s_j, g_2\rangle} = 0$, due to the sparsity of reward sequences. Therefore, given the above pair of transitions, we can draw from Thm. 1, and iteratively update the encoder $\psi(\cdot)$ in the policy function using the mean square loss:

$$\left(\mathcal{M}(\langle s_i, g_1\rangle, \langle s_j, g_2\rangle) - \gamma \mathbb{E}_{s_{i+1}, s_{j+1}}\left[\mathcal{M}(\langle s_{i+1}, g_1\rangle, \langle s_{j+1}, g_2\rangle)\right]\right)^2, \quad (12)$$

where $\mathcal{M}(\langle s_i, g_1\rangle, \langle s_j, g_2\rangle)$ is highly dependent on the next states s_{i+1} and s_{j+1}, and rarely captures information from the absolute reward difference $\left|r_{\langle s_i, g_1\rangle} - r_{\langle s_j, g_2\rangle}\right|$. This cannot provide any evaluations of the difference between state-goal tuples at the current step and prevents the policy and critic functions from gaining insight from the interaction between the agent and the environment. To compensate for this deficiency of SimSR utilized in GCRL, we utilize the Sensitivity-Aware Regularizer as a substitute for the absolute value of the reward difference.

Algorithm 1: Semi-Contrastive Adversarial Augmentation

Require: Offline dataset $\mathcal{D}$, learning rate α, training steps T
Initialisation: Critic function $\varrho(\cdot)$, policy function $\varphi(\psi(\cdot))$

for training timestep $1 \dots T$ **do**
- Sample a mini-batch from the offline dataset: $(s_t^i, a_t^i, s_{t+1}^i, g^i) \sim \mathcal{D}$
- Construct negative tuples: $\left\langle s_t^i, g^i\right\rangle^-$ and $\left\langle s_{t+1}^i, g^i\right\rangle^-$
- Compute the semi-contrastive augmented samples: $\mathcal{V}_{scr}(s_t^i)$ and $\mathcal{V}_{scr}(g^i)$ by Eq. (9-10)
- Augment critic value: $\hat{\varrho} \leftarrow \mathbb{E}[\varrho(s_t^i, g^i, a_t^i) + \varrho(\mathcal{V}_{scr}(s_t^i), \mathcal{V}_{scr}(g^i), a_t^i)]$
- Augment actor value: $\hat{\varphi} \leftarrow \mathbb{E}[\varphi(\psi(\left\langle s_t^i, g^i\right\rangle)) + \varphi(\psi(\left\langle \mathcal{V}_{scr}(s_t^i), \mathcal{V}_{scr}(g^i)\right\rangle))]$
- Update encoder weights: $\theta_\psi \leftarrow \theta_\psi - \alpha \nabla \mathcal{L}_\psi(\left\langle s_t^i, g^i\right\rangle)$
- Update actor weights: $\theta_\varphi \leftarrow \theta_\varphi - \alpha \nabla \mathcal{L}_\varphi(\hat{\varphi})$
- Update critic weights: $\theta_\varrho \leftarrow \theta_\varrho - \alpha \nabla \mathcal{L}_\varrho(\hat{\varrho})$

Definition 4 (Sensitivity-Aware Regularizer) *Given perturbations δ_{s_i}, δ_{g_1}, δ_{s_j}, δ_{g_2} for tuples $\langle s_i, g_1\rangle$ and $\langle s_j, g_2\rangle$, and a trade-off factor β, the sensitivity-aware regularizer for the encoder $\psi(\cdot)$ can be defined as:*

$$\left|\frac{\mathcal{M}(\langle s_i, g_1\rangle, \langle s_i + \delta_{s_i}, g_1\rangle)}{\|\delta_{s_i}\|_2} - \frac{\mathcal{M}(\langle s_j, g_2\rangle, \langle s_j + \delta_{s_j}, g_2\rangle)}{\|\delta_{s_j}\|_2}\right| + \beta\left|\frac{\mathcal{M}(\langle s_i, g_1\rangle, \langle s_i, g_1 + \delta_{g_1}\rangle)}{\|\delta_{g_1}\|_2} - \frac{\mathcal{M}(\langle s_j, g_2\rangle, \langle s_j, g_2 + \delta_{g_2}\rangle)}{\|\delta_{g_2}\|_2}\right|, \quad (13)$$

then we can empirically optimize the encoder parameters θ_ψ using the loss function combining Eq. (12) and Eq. (13).

As shown in Def. 4, we explore the Lipschitz constant-based robustness term $\mathcal{M}(\cdot)/\|\delta\|_2$ to indicate the implicit difference. Generally, on the one hand, physically close tuples should have similar Lipschitz constants both on states and goals with a high probability, physically distant tuples. On the other hand, physically distant tuples tend to have dissimilar Lipschitz constants with a high probability. This reflects the similar characteristic as the absolute reward difference utilized in Eq. (5).

Experiments

Our experimental design proceeds as follows: Initially, we identify the optimal representation layer by executing SCR attacks across various layers. Next, we benchmark the SCR attack against other adversarial attacks, using state-of-the-art algorithms as base methods. Finally, we achieve enhanced robustness of different base methods using ARTs.

Baselines

We evaluate our method[1] using four robot manipulation tasks, namely FetchPush, FetchReach, FetchSlide, and FetchPick, as described in (Plappert et al. 2018). The offline dataset for each task is collected through either a purely random policy or a combination of 90% random policies

[1]Our code is available at github.com/TrustAI/ReRoGCRL

Task	Method	Nature	Attack Return								
			Uniform	SA-FGSM	SA-PGD	SCR-F(S)	SCR-F(G)	SCR-F(S+G)	SCR-P(S)	SCR-P(G)	SCR-P(S+G)
Pick	DDPG	14.82	16.68±4.33	19.61±2.84	19.92±2.32	11.18±2.74	15.00±1.88	**7.83±3.06**	10.97±4.12	16.38±2.72	10.49±2.86
	GCSL	11.39	9.95±1.65	9.94±1.80	10.07±1.15	9.52±2.13	10.41±1.95	8.99±1.56	8.21±3.12	11.72±1.81	**8.11±2.04**
	GoFar	21.01	19.91±2.39	20.07±2.13	18.92±1.80	17.14±2.89	20.22±3.25	16.71±3.26	17.93±2.48	19.84±1.88	**15.65±1.83**
Push	DDPG	12.81	13.61±4.81	17.24±2.12	15.42±3.38	8.71±3.22	10.27±3.85	6.43±2.33	9.79±2.15	13.61±2.56	**5.42±3.00**
	GCSL	12.74	12.57±1.67	9.64±3.28	9.71±3.37	9.64±0.51	12.97±3.09	8.99±1.56	**6.20±3.66**	13.56±1.49	8.11±2.04
	GoFar	18.38	16.11±2.39	15.45±2.71	13.66±3.97	11.66±3.93	17.47±2.07	**8.19**±3.99	12.14±3.78	18.38±2.91	10.57±3.93
Reach	DDPG	29.92	28.86±0.46	24.21±2.21	20.14±5.04	6.10±7.33	5.14±7.19	2.02±3.32	7.02±2.20	11.03±8.47	**1.86±3.31**
	GCSL	22.04	22.75±0.82	22.01±1.47	21.95±1.51	**20.97±1.31**	22.24±0.78	23.04±1.62	21.74±0.81	22.22±0.79	23.46±1.14
	GoFar	27.84	27.57±0.92	27.22±0.49	27.22±0.49	27.51±0.45	27.45±1.11	15.71±4.48	27.96±0.38	27.87±0.33	**11.24±6.76**
Slide	DDPG	0.58	1.89±0.89	0.59±1.09	0.50±1.12	**0.00±0.00**	0.36±0.72	**0.00±0.00**	**0.00±0.00**	0.38±0.60	**0.00±0.00**
	GCSL	1.55	1.84±1.19	0.60±0.42	0.57±0.59	0.28±0.63	0.97±0.63	0.37±0.51	0.37±0.84	0.96±1.26	**0.00±0.00**
	GoFar	2.55	1.23±1.14	0.77±0.92	1.13±0.72	0.09±0.21	1.25±0.55	**0.00±0.00**	**0.00±0.00**	1.86±0.73	0.09±0.22

Table 1: Discounted returns against various attack methods. SCR-F and SCR-P indicate SCR-FGSM and SCR-PGD respectively. S, G, and S+G individually illustrate the perturbations added on state, goal, and state+goal. In each row, values marked in bold signify the lowest returns. The experiments are averaged over 5 seeds. For simplicity, we omit Fetch before each task.

and 10% expert policies, depending on whether the random data sufficiently represents the desired goal distribution. We select 3 algorithms as our baselines for GCRL: Deep Deterministic Policy Gradient (DDPG) (Andrychowicz et al. 2018), Goal-Conditioned Supervised Learning (GCSL) (Ghosh et al. 2020), and Goal-Conditioned F-Advantage Regression (GoFar) (Ma et al. 2022). To circumvent risky environment interactions, we learn general goal-reaching policies from offline interaction datasets.

Network Architectures

We use 3-layer MLPs as the backbones of $\varphi(\psi(\cdot))$ and $\varrho(\cdot)$. To determine L_ψ and L_φ, we conduct preliminary experiments based on $\varphi(\psi(\cdot))$, trained via DDPG and GoFar to ascertain the optimal layer for representation. As depicted in Fig. 3, we compute the discounted returns across three distinct layers (L-1 denotes the first layer of $\varphi(\psi(\cdot))$, and so forth), with each layer acting as the representation layer. Comprehensive results indicate that the first layer is the most susceptible. Therefore, all attacks and training algorithms in this paper are computed on the first layer.

In particular, the encoder $\psi(\cdot)$ processes the input state-goal tuple o_0^π, of which the dimension is D_{sg}. Thus, we define the latent representation given by $\psi(o_0^\pi) = \phi(\mathbf{W}_1 o_0^\pi)$, $\mathbf{W}_1^\pi \in \mathbb{R}^{256\times D_{sg}}$. Subsequently, the actor network is constructed as $\varphi(\psi(o_0^\pi)) = \mathbf{W}_4^\pi \phi(\mathbf{W}_3^\pi \phi(\mathbf{W}_2^\pi \psi(o_0^\pi)))$, with $\mathbf{W}_2^\pi, \mathbf{W}_3^\pi \in \mathbb{R}^{256\times 256}$, $\mathbf{W}_4^\pi \in \mathbb{R}^{D_a\times 256}$, where D_a is the dimension of the action. We construct the output of critic function as $o_4^c = \mathbf{W}_4^c \phi(\mathbf{W}_3^c \phi(\mathbf{W}_2^c \phi(\mathbf{W}_1^c o_0^c)))$, where o_0^c is the concatenation of the state-goal tuple and the action, which has $(D_{sg} + D_a)$ dimensions. Specifically, $\mathbf{W}_1^c \in \mathbb{R}^{256\times(D_{sg}+D_a)}$,$\mathbf{W}_2^c$, $\mathbf{W}_3^c \in \mathbb{R}^{256\times 256}$, and $\mathbf{W}_4^c \in \mathbb{R}^{1\times 256}$.

Comparison of SCR and Other Attacks

Following Def. 3, we evaluate the robustness of multiple GCRL algorithms using 5 different attack methods, including Uniform, SA-FGSM, SA-PGD, SCR-FGSM, and SCR-PGD. Specifically, we employ a 10-step PGD with a designated step size of 0.01. For both FGSM and PGD attacks, the attack radius is set to 0.1.

Our SCR attack introduces perturbations in the form of $\langle \mathcal{V}(s), g\rangle$, $\langle s, \mathcal{V}(g)\rangle$, and $\langle \mathcal{V}(s), \mathcal{V}(g)\rangle$, respectively. Specifically, Tab. 1-2 presents the results when noise is added solely to the states. We provide the remaining attack results in the Appendix. As outlined in Def. 3, The columns *state*, *goal*, and *state+goal* denote negative tuples as $\langle -s, g\rangle$, $\langle s, -g\rangle$, and $\langle -s, -g\rangle$ individually.

As shown in Tab. 1, for DDPG-based GCRL, we achieve 47.17%, 57.69%, 93.78%, and 100.00% decrease of discounted returns in FetchPick, FetchPush, FetchReach, and FetchSlide respectively, outperforming SA-based attacks by 60.07%, 64.85%, 90.76%, and 100.00%. In detail, all of them are derived from the negative tuple $\langle -s, -g\rangle$. The results are similar in GCSL, compared with the nature return, our method degrades the performance in FetchPick, FetchPush, and FetchSlide each by 18.41%, 35.68%, and 100.00%. As the state-of-the-art algorithm in GCRL, GoFar achieves better nature returns than DDPG and GCSL, thus all attacks exhibit a degradation in their attack capabilities. Particularly, our method achieves a reduction in overall returns of 17.28%, 40.04%, 58.71%, and 100.00% across the four tasks, respectively. Similar to DDPG, the best-performing attack for each of these 4 tasks originates from $\langle -s, -g\rangle$.

We further evaluate the adversarial robustness of SimSR-enhanced algorithms in Tab. 2. All attack strategies employ the negative tuple $\langle -s, g\rangle$.

Effectiveness of ARTs

To test the efficacy of ARTs, we perform attacks on ARTs-enhanced versions of different GCRL algorithms. Following Alg. 1, we train the agent using the SCR attack. For SimSR-boosted representations, we adaptively employ SAR during the optimization. Tab. 3 showcases the best attack re-

Task	DDPG (SimSR)						GoFar (SimSR)					
	Nature	Uniform	SA-FGSM	SA-PGD	SCR-FGSM	SCR-PGD	Nature	Uniform	SA-FGSM	SA-PGD	SCR-FGSM	SCR-PGD
FetchPick	16.78	16.00	19.51	17.92	**15.74**	16.15	17.44	16.73	17.63	**13.80**	16.27	16.11
FetchPush	12.63	15.43	16.24	14.61	10.70	**10.51**	14.19	13.13	15.52	13.91	**12.44**	13.08
FetchReach	29.90	29.46	28.68	**26.76**	27.72	27.16	27.93	**27.91**	27.95	27.96	27.96	27.98
FetchSlide	0.62	0.83	0.80	0.43	**0.37**	0.84	1.66	1.78	2.31	1.41	1.07	**0.59**

Table 2: Comparison of discounted returns for DDPG (SimSR) and GoFar (SimSR) against various attack methods in GCRL, averaged over 5 seeds. Values highlighted in bold indicate the lowest return for each row.

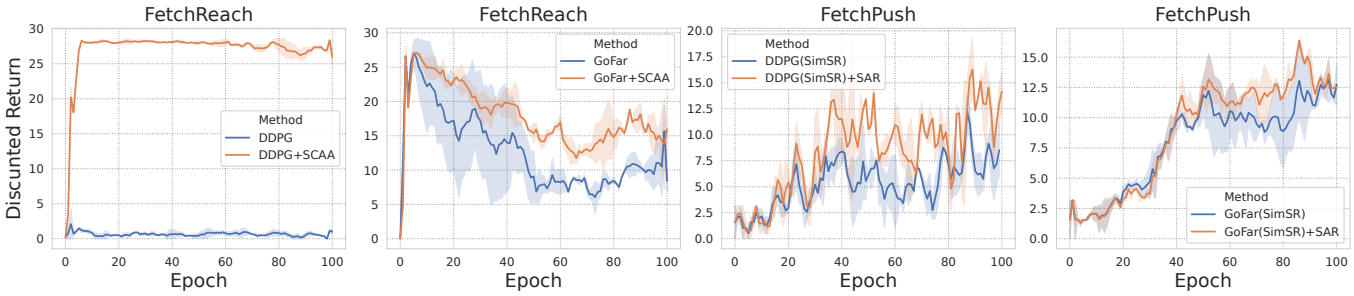

Figure 2: Epoch-wise evaluations on the SCR attack of ARTs-defended DDPG, DDPG (SimSR), GoFar, and GoFar (SimSR).

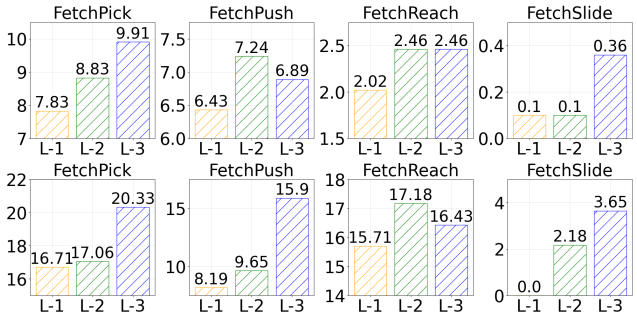

Figure 3: Evaluating the SCR attack in MLP-based architectures across different layers (where 'L' denotes 'Layer'). The top row of charts presents results using the DDPG method, while the bottom row employs the GoFar method.

Methods	FetchPick	FetchPush	FetchReach	FetchSlide
DDPG	7.83±3.06	5.42±3.00	1.86±3.31	0.00±0.00
DDPG+SCAA	**14.53±3.08**	**10.79±4.55**	**27.68±1.30**	**0.00±0.00**
DDPG(S)	15.74±0.65	10.51±5.79	27.16±2.75	0.37±0.60
DDPG(S)+SAR	**16.78±1.39**	**12.63±3.89**	**29.91±0.20**	**0.55±0.20**
GoFar	**15.65±1.83**	**8.19±3.99**	11.24±6.76	0.00±0.00
GoFar+SCAA	14.72±4.40	8.00±3.76	**13.99±6.46**	**0.00±0.00**
GoFar(S)	13.80±4.40	12.44±3.80	**27.91±0.34**	0.59±0.66
GoFar(S)+SAR	**14.82±3.04**	**12.44±3.80**	27.84±0.32	**1.47±0.79**

Table 3: Evaluations of ARTs on DDPG, GoFar, SimSR-strengthened DDPG (DDPG(S)) and SimSR-strengthened GoFar (GoFar(S)) against best attacks, averaged over 5 seeds. The Bold value indicates the better result in a block.

sults from Uniform, SA-FGSM, SA-PGD, SCR-FGSM, and SCR-PGD. All attacks and training procedures in this section are based on $\langle -s, g \rangle$.

As illustrated in Tab. 3, our defensive strategy significantly bolsters the robust performance of DDPG. Specifically, we register performance enhancements of 85.57%, 99.08%, and 1388.17% in FetchPick, FetchPush and FetchReach, respectively. In a similar vein, for DDPG (SimSR), the implementation of ARTs leads to robustness improvements of 6.61%, 20.17%, 10.13%, and 48.65% across FetchPick, FetchPush, FetchReach, and FetchSlide, respectively.

For GoFar, our results are comparable and even superior in FetchPush, FetchReach, and FetchSlide. Additionally, for GoFar (SimSR), we observe a performance boost of 7.39% and 149.15%, respectively, in FetchPick and FetchSlide. Complete results are available in the Appendix.

Qualitatively, Fig. 2 provides a visual representation of the epoch-wise performance of ARTs-enhanced GCRL algorithms in comparison to several base methods.

Conclusion

Due to the sparse rewards in GCRL, we introduce the Semi-Contrastive Representation attack to probe its vulnerability, which only requires access to the policy function. Accordingly, we propose Adversarial Representation Tactics to bolster the robust performance of the underlying agent. In particular, we devise the Semi-Contrastive Adversarial Augmentation for baseline methods in GCRL and introduce the Sensitivity-Aware Regularizer for their robust counterparts that are guided by the bisimulation metric. We utilize SimSR to learn invariant representations. The effectiveness of our methods is validated by intensive empirical experiments.

Acknowledgements

This work is supported by the UK EPSRC under project EnCORE [EP/T026995/1], the University of Liverpool and the China Scholarship Council.

References

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, P.; and Zaremba, W. 2018. Hindsight Experience Replay. arXiv:1707.01495.

Bai, X.; Guan, J.; and Wang, H. 2019. A model-based reinforcement learning with adversarial training for online recommendation. *Advances in Neural Information Processing Systems*, 32.

Castro, P. S.; Kastner, T.; Panangaden, P.; and Rowland, M. 2022. MICo: Improved representations via sampling-based state similarity for Markov decision processes. arXiv:2106.08229.

Chane-Sane, E.; Schmid, C.; and Laptev, I. 2021. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, 1430–1440. PMLR.

Charlesworth, H.; and Montana, G. 2020. Plangan: Model-based planning with sparse rewards and multiple goals. *Advances in Neural Information Processing Systems*, 33: 8532–8542.

Chebotar, Y.; Hausman, K.; Lu, Y.; Xiao, T.; Kalashnikov, D.; Varley, J.; Irpan, A.; Eysenbach, B.; Julian, R.; Finn, C.; et al. 2021. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*.

Eysenbach, B.; Salakhutdinov, R. R.; and Levine, S. 2019. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Eysenbach, B.; Zhang, T.; Levine, S.; and Salakhutdinov, R. R. 2022. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 35603–35620.

Fang, M.; Zhou, C.; Shi, B.; Gong, B.; Xu, J.; and Zhang, T. 2018. DHER: Hindsight experience replay for dynamic goals. In *International Conference on Learning Representations*.

Fang, M.; Zhou, T.; Du, Y.; Han, L.; and Zhang, Z. 2019. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32.

Ferns, N.; Panangaden, P.; and Precup, D. 2011. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6): 1662–1714.

Ferns, N.; Panangaden, P.; and Precup, D. 2012. Metrics for Finite Markov Decision Processes. arXiv:1207.4114.

Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Bellemare, M. G. 2019. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. arXiv:1906.02736.

Ghosh, D.; Gupta, A.; Reddy, A.; Fu, J.; Devin, C. M.; Eysenbach, B.; and Levine, S. 2020. Learning to Reach Goals via Iterated Supervised Learning. In *International Conference on Learning Representations*.

Givan, R.; Dean, T.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1-2): 163–223.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2019. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*.

He, X.; and Lv, C. 2023. Robotic Control in Adversarial and Sparse Reward Environments: A Robust Goal-Conditioned Reinforcement Learning Approach. *IEEE Transactions on Artificial Intelligence*, 1–10.

Ho, C.-H.; and Nvasconcelos, N. 2020. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33: 17081–17093.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

Huang, X.; Jin, G.; and Ruan, W. 2023. Deep reinforcement learning. In *Machine Learning Safety*, 219–235. Springer.

Huang, X.; Kroening, D.; Ruan, W.; Sharp, J.; Sun, Y.; Thamo, E.; Wu, M.; and Yi, X. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37: 100270.

Jiang, Z.; Chen, T.; Chen, T.; and Wang, Z. 2020. Robust pre-training by adversarial contrastive learning. *Advances in neural information processing systems*, 33: 16199–16210.

Kim, M.; Tack, J.; and Hwang, S. J. 2020. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 2983–2994.

Kos, J.; and Song, D. 2017. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*.

Larsen, K. G.; and Skou, A. 1989. Bisimulation through probabilistic testing (preliminary report). In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 344–352.

Ma, J. Y.; Yan, J.; Jayaraman, D.; and Bastani, O. 2022. Offline goal-conditioned reinforcement learning via f-advantage regression. *Advances in Neural Information Processing Systems*, 35: 310–323.

Mezghani, L.; Sukhbaatar, S.; Bojanowski, P.; Lazaric, A.; and Alahari, K. 2022. Learning Goal-Conditioned Policies Offline with Self-Supervised Reward Shaping. In *6th Annual Conference on Robot Learning*.

Mezghani, L.; Sukhbaatar, S.; Bojanowski, P.; Lazaric, A.; and Alahari, K. 2023. Learning Goal-Conditioned Policies Offline with Self-Supervised Reward Shaping. In *Conference on Robot Learning*, 1401–1410. PMLR.

Mu, R.; Marcolino, L. S.; Zhang, T.; Zhang, Y.; Huang, X.; and Ruan, W. 2023a. Reward Certification for Policy Smoothed Reinforcement Learning. *arXiv preprint arXiv:2312.06436*.

Mu, R.; Ruan, W.; Marcolino, L. S.; Jin, G.; and Ni, Q. 2023b. Certified Policy Smoothing for Cooperative Multi-Agent Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*.

Nasiriany, S.; Pong, V.; Lin, S.; and Levine, S. 2019. Planning with goal-conditioned policies. *Advances in Neural Information Processing Systems*, 32.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2817–2826. PMLR.

Plappert, M.; Andrychowicz, M.; Ray, A.; McGrew, B.; Baker, B.; Powell, G.; Schneider, J.; Tobin, J.; Chociej, M.; Welinder, P.; et al. 2018. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*.

Sun, J.; Zhang, T.; Xie, X.; Ma, L.; Zheng, Y.; Chen, K.; and Liu, Y. 2020. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5883–5891.

Wang, F.; Zhang, C.; Xu, P.; and Ruan, W. 2022. Deep learning and its adversarial robustness: A brief introduction. In *HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation*, 547–584.

Weng, T.-W.; Dvijotham, K. D.; Uesato, J.; Xiao, K.; Gowal, S.; Stanforth, R.; and Kohli, P. 2019. Toward evaluating robustness of deep reinforcement learning with continuous control. In *International Conference on Learning Representations*.

Yang, R.; Fang, M.; Han, L.; Du, Y.; Luo, F.; and Li, X. 2021a. MHER: Model-based hindsight experience replay. *arXiv preprint arXiv:2107.00306*.

Yang, R.; Lu, Y.; Li, W.; Sun, H.; Fang, M.; Du, Y.; Li, X.; Han, L.; and Zhang, C. 2021b. Rethinking Goal-Conditioned Supervised Learning and Its Connection to Offline RL. In *International Conference on Learning Representations*.

Zang, H.; Li, X.; and Wang, M. 2022. SimSR: Simple Distance-based State Representation for Deep Reinforcement Learning. arXiv:2112.15303.

Zhang, A.; McAllister, R.; Calandra, R.; Gal, Y.; and Levine, S. 2020a. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*.

Zhang, C.; Ruan, W.; and Xu, P. 2023. Reachability Analysis of Neural Network Control Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D.; and Hsieh, C.-J. 2020b. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33: 21024–21037.