Poisoning the Well: Can We Simultaneously Attack a Group of Learning Agents?

Ridhima Bector , **Hang Xu** , **Abhay Aradhya** , **Chai Quek** and **Zinovi Rabinovich**
Nanyang Technological University
{ridhima001, hang017}@e.ntu.edu.sg, {abhayaradhya, ashcquek, zinovi}@ntu.edu.sg

Abstract

Reinforcement Learning's (RL) ubiquity has instigated research on potential threats to its training and deployment. Many works study single-learner training-time attacks that "pre-programme" behavioral triggers into a strategy. However, attacks on collections of learning agents remain largely overlooked. We remedy the situation by developing a constructive training-time attack on a population of learning agents and additionally make the attack agnostic to the population's size. The attack constitutes a sequence of environment (re)parameterizations (poisonings), generated to overcome individual differences between agents and lead the entire population to the same target behavior while minimizing effective environment modulation. Our method is demonstrated on populations of independent learners in "ghost" environments (learners do not interact or perceive each other) as well as environments with mutual awareness, with or without individual learning. From the attack perspective, we pursue an ultra-blackbox setting, i.e., the attacker's training utilizes only across-policy traces of the victim learners for both attack conditioning *and* evaluation. The resulting uncertainty in population behavior is managed via a novel Wasserstein distance-based Gaussian embedding of behaviors detected within the victim population. To align with prior works on environment poisoning, our experiments are based on a 3D Grid World domain and show: a) feasibility, i.e., despite the uncertainty, the attack forces a population-wide adoption of target behavior; b) efficacy, i.e., the attack is size-agnostic and transferable. Code and Appendices are available at "bit.ly/github-rb-cep".

1 Introduction

Reinforcement learning (RL) has proliferated most AI applications that investigate unexplored spaces and has bestowed these applications with remarkable capabilities, superhuman at times [Mnih *et al.*, 2013; Mnih *et al.*, 2015; Lample and Chaplot, 2017]. Alas, there is no Superman without Kryptonite. RL methods are subject to a variety of attacks that can degrade a policy's performance during deployment; introduce behavior triggers into it or force an agent to learn an a priori non-optimal target strategy [Chen *et al.*, 2019; Ilahi *et al.*, 2021]. To achieve this, an adversarial system is constructed that encompasses an RL agent, its environment, and its task. In these systems, the RL agent is regarded as the victim, its RL environment, the victim environment, and the task, the victim task. In addition to the victim, the system includes an adversary, tasked with attacking the victim agent. Since the attacker's task is no easier than the victim's, machine learning solutions (and RL, in particular) have been deployed on the attacker's side as well. All attack solutions are commonly classified by 3 features: the form of attack (Train vs Test), the mode of attack (Reward vs Observation vs Environment), and the level of access (Whitebox vs Blackbox) to the victim's inner workings granted to the attacker.

In this paper, we focus on training-time, environment-poisoning attacks. That is we seek to influence the training/optimization of the victim agent's policy by means of altering the victim environment's dynamics akin to [Xu *et al.*, 2021; Behzadan and Munir, 2017; Rakhsha *et al.*, 2020]. The goal of the attack is to introduce "backdoors" or behavioral triggers into the victim's learned strategy by means encapsulated within the environment mechanics while avoiding access to the victim's inner workings. Furthermore, following this line of reasoning, we favor a blackbox setting, wherein an attacker estimates the victim's behavior policy by observing the victim's interaction with its environment and then conditions the attack on this estimated behavior. In fact, we expand this notion. Normally, during the attacker's training, the system would have access to a proxy victim's inner workings. Such a proxy victim's actual strategy would be used to generate an *extrinsic* reward signal for the attacker. In contrast, this paper adopts an Ultra-Blackbox (UB) setting, where the reward signal is *intrinsic*, i.e., generated based on the observed and perceived behavior of the (proxy) victim without any access to its inner workings.

There exists some progress in attacking Multi-agent RL (MARL) systems (e.g., [Chen *et al.*, 2022; Pham *et al.*, 2022; Liu *et al.*, 2023; Chelarescu, 2021; Lin *et al.*, 2020]). However, to the best of the authors' knowledge, none have yet studied the question of multiple RL agents being attacked simultaneously with an environment poisoning attack. Eyeing social and collective learning settings (e.g., [Parisotto *et*

al., 2019; Marthi *et al.*, 2005; Dimakopoulou *et al.*, 2018; Lupu and Precup, 2020; Yang *et al.*, 2020]), we seek to attack a *population* of learners without having the luxury of access to any individual's inner workings or to the inter-agent relationships. To begin, we adopt scenarios with simplified collectives: a) an *Implicit Collective*, where agents are unaware of each other (essentially inhabit copies of the same environment), and practice individual learning; b) a *Swarm Collective*, where agents are aware of each other's existence, but are anonymous to each other, and practice social learning; c) a *True Collective*, where agents are aware of each other's existence, and learn via individual as well as social learning. We describe these in greater detail in Appendix A.

In order to finalize our approach and define an optimal sequence of environment poisonings, we would need the capability to efficiently capture the distribution of policies used by the victim collective and measure the effect a poisoned environment has on such a distribution. Technically, our approach to these issues is *structurally* more related[1] to the Optimal Transport Kernel Embedding (OTKE) [Mialon *et al.*, 2020], than any other set representation method, such as [Qi *et al.*, 2017; Zaheer *et al.*, 2017; Skianis *et al.*, 2020]. Specifically, we use a two-step representation for the set of all policies found in a population: a) representing uncertainty in each agent's policy, given the agent's interaction trace across multiple policies (across-policy interaction trace); b) capturing the distribution of behaviors at the population level. The former is achieved by application of VAE, and the latter by construction of the Wasserstein barycenter in the resultant latent space. At this research stage we somewhat presume that environment poisoning is effective and thereby use a single barycenter; disregarding a potential behavior sub-structure in the population. In particular, we address two hypotheses: **H1)** Wasserstein distance-based Gaussian embedding is capable of capturing the behavior of different-sized victim populations; and **H2)** Attack strategy learned on a given population is transferable to other populations of different sizes.

To sum up, we introduce: a) **Collective Environment Poisoning (CEP)** framework, which we experimentally instantiate for three scenarios: *implicit collectives*, *swarm collectives* and *true collectives*; b) **Size-Agnostic Population Behavior Representation** based on a Wasserstein distance-based Gaussian embedding; c) **Ultra-Blackbox (UB) Adversarial Setting**, wherein across-policy behavior traces of the victim population are used to both condition and evaluate the attack.

2 Methodology

This section describes the developed methodology in increasing levels of detail. First, the overall interaction structure between the attacker and a population of victim learners is presented. Then, the specifics of encoding a distribution of behaviors within a population are described. Finer details of the attacker's intrinsic reward are delegated to Appendix C.

2.1 Bi-Level System Architecture

We formalize our method as a bi-level hierarchical framework wherein the attacker as well as each member of the victim

[1]An explicit related works section is delegated to Appendix B.

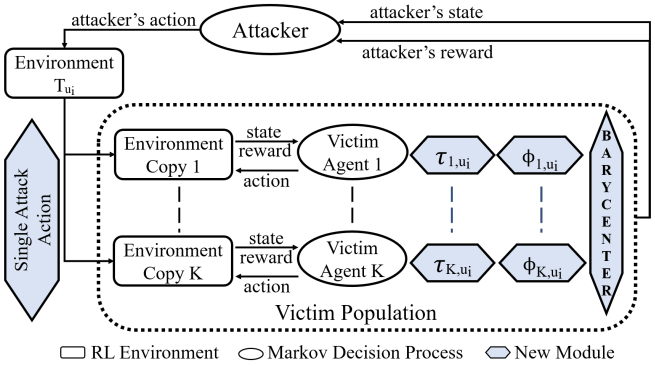

Figure 1: Bi-Level Attack Framework (Implicit Collective Scenario)

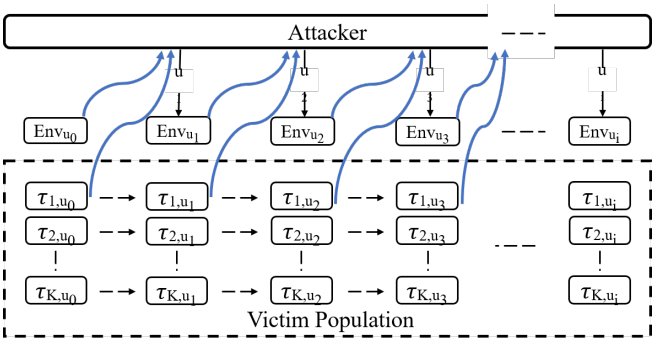

Figure 2: Attack Deployment

population is an independent reinforcement-learning agent with its individual learning algorithm, memory, and policy. The victim population is a collection of learners where each member trains to learn the given task in a common environment under the Swarm and True Collective scenarios; and in an environment copy that instantiates the common blueprint, in the Implicit Collective scenario. In order to learn the given task, each agent trains to maximize its individual cumulative discounted rewards, which correspond to its individual task. The attacker on the other hand observes the interaction of the population of victims with their environment and, based on the set of observed behaviors, takes an action that modifies the victim environment/blueprint. The goal of the attacker is to sequentially and minimally modify the victim environment's dynamics to drive the victim population to adopt the attacker-desired target behavior. Therefore, the overall system is formed by two nested closed-loop learning processes, wherein the attacker and members of the victim population are modeled as Markov Decision Processes (MDPs).

Victim Population MDP: The victim population's Markov process can be denoted by the tuple $< S, A, T_{u_i}, R_v, q_0, \gamma_v >$ where $S = s_1, s_2, ...$, and $A = a_1, a_2, ...$ are the victims' states and actions respectively; $R_v : S \times A \times S \to \mathbb{R}$ is the reward function which encodes each victim's task; $\gamma_v \in (0, 1)$ is the discount factor, $q_0(S)$ is the distribution over initial states; and, $T_{u_i} : S \times A \times S \to [0, 1]$ is the probabilistic transition function, where u_i denotes the environment parameterization that has resulted from the first i interventions on

the environment, by the attacker. In particular, T_{u_0} refers to the original, unaltered dynamics of the victim environment. A single attack action modifies the victim environment dynamics for the entire victim population. The objective of each member of the victim population is to find an optimal policy within the experienced environment.

Attacker MDP: The attacker's Markov process can be represented by the tuple $< \Theta, U, F, R_a, \tau^*, \gamma_a >$, where: $\Theta = [T_{u_i}, \overline{\phi}_{u_i}]$ is the attacker's state space comprising the victim environment's dynamics, T_{u_i} and the victim population's behavior, $\overline{\phi}_{u_i}$ that emerged in response to those dynamics; U is the attacker's action space, i.e., the set of all permissible changes that can be applied to the victim environment's dynamics, such that action u_i when applied on the environment with dynamics $T_{u_{i-1}}$ results in an environment with dynamics T_{u_i}. It is important to note here that environment dynamics at attack time step i are a result of accumulated changes caused by attack actions $u_1, u_2, ..., u_i$. $F : \Theta \times U \times \Theta \rightarrow [0, 1]$ is the probabilistic transition function that describes the response of the victim population to environmental experiences, i.e., how the distribution of behaviors within the population changes in response to changes in the environment dynamics; $R_a : \Theta \times U \times \Theta \rightarrow \mathbb{R}$ is the attacker's reward function that describes the combined accuracy (how concentrated is the victim population's behavior distribution around the ideal behavior, τ^*) and effort of (how small is) the accumulated environment modifications; where τ^* is the attacker-desired target victim policy. The attacker optimizes these dual objectives of accuracy (maximize) and effort (minimize) to learn the best attack generation strategy of the form $\sigma : \Theta \rightarrow U$, $\sigma(u_i|\Theta_{i-1})$. I.e., the attacker seeks the most *efficient* way to force all individual behaviors within the victim population to converge to the target policy τ^*.

2.2 Population Behavior Representation

Development of an attack generation function that is capable of pushing victim populations of different sizes towards a target behavior using a single, constrained attack action at every attack time step, under (Ultra-)Blackbox setting, entails two major challenges, as mentioned in Section 1. The first challenge is accurate approximation of individual behaviors present inside the victim population. Due to the (Ultra-)Blackbox nature of settings, individual victim behaviors can only be approximated (through observation of across-policy behavior traces) and never be captured completely with 100% certainty. The second challenge is to make the attack generation function agnostic to the size of the victim population.

The attacker observes the actions taken by the victims in different states, as they train to learn their tasks in the victim environment. As the victim population is under training, victims update their internal policies periodically. Depending on the frequency of these updates, each state-action pair of a behavior trace can potentially be generated by a different policy. In this paper, we work with a highly interactive setting wherein the victims update their policies after each interaction with their environment. In prior works, the attacker strives to capture a victim's behavior by noting its trajectories in the victim environment and conditioning each attack action on the last trajectory observed prior to the attack action [Xu *et al.*, 2021; Xu *et al.*, 2022]. The last trajectory however on one hand, does not capture information about frequently visited states that were not visited in the last trajectory; and on the other hand, does not retain any information about states/regions that are entirely unvisited and hence unimportant to the victim. Storing multiple recent trajectories can help add information regarding other frequently visited states as well as help capture the stochastic behaviors of a victim. However, in the Blackbox/Ultra-Blackbox settings, it is impossible for the attacker to discern between discarded (old) behaviors and stochastic (current) behaviors of a victim. Moreover, this uncertainty is further exacerbated in multi-agent victim settings wherein victim identities are not stored by the attacker. To do away with this uncertainty, reduce memory requirements, and retain information regarding unvisited (and hence unimportant) states; in this work, the attacker stores the last observed victim action corresponding to each victim state and uses a "no-action" symbol to demarcate unvisited states. Information regarding environment configurations that are unimportant with respect to a victim agent's objectives, can prove crucial to the attacker while deciding stealthy and efficient environment modifications that push the complete victim population towards the attacker-desired target behavior. This individual behavior information corresponding to a given victim k will hereafter be denoted as $\tau_{k,u_i} = \{s_1, a_1; s_2, a_2; ...; s_N, a_N\} \forall s_n \in S$, a_n is the latest action taken by victim k in state n or a no-action symbol in case state s_n was never visited by the victim, and N is the total number of states in the given environment with dynamics T_{u_i}. As τ_{k,u_i} contains the latest action / no-action symbol corresponding to all environment configurations, τ_{k,u_i}'s size can become extremely large in high-dimensional environments. Furthermore, in size-agnostic multi-victim attacks, the attacker needs a mechanism to generate a uniform-sized representation of all behaviors present inside the victim population. In this work, these two problems are solved by the attacker by learning a distributional low-dimensional latent space, Φ of individual behaviors using a variational autoencoder model. The latent behavior distribution corresponding to a given victim's individual behavior τ_{k,u_i} is denoted by ϕ_{k,u_i}. Herein the dimensionality of $\tau_{k,u_i} >> \phi_{k,u_i}$. The variational model consists of an encoder q_e that takes a given victim agent's τ_{k,u_i} as input and outputs parameters to it's latent behavior distribution ϕ_{k,u_i}; and a decoder q_d that takes two inputs, a sample z from the latent distribution ϕ_{k,u_i} and a victim environment state s_n, and outputs the probability with which victim k will take each available action in the given state s_n. The prior distribution $p(z)$ on the latent variables is the standard normal $N(z; 0, I)$ while the evidence lower bound, to be maximized over all k is:

$$\mathbb{E}_{z \sim \phi_{k,u_i}}[\, log\ q_d(a_n|z, s_n)] - D^{kl}\left(\phi_{k,u_i} \,||\, p(z)\right) \qquad (1)$$

The generative capability of the variational individual-behavior model is crucial in solving the second challenge of developing a size-agnostic attack strategy that is transferable across different victim populations of varied sizes. We exploit the regularity of the distributional latent space and utilize the

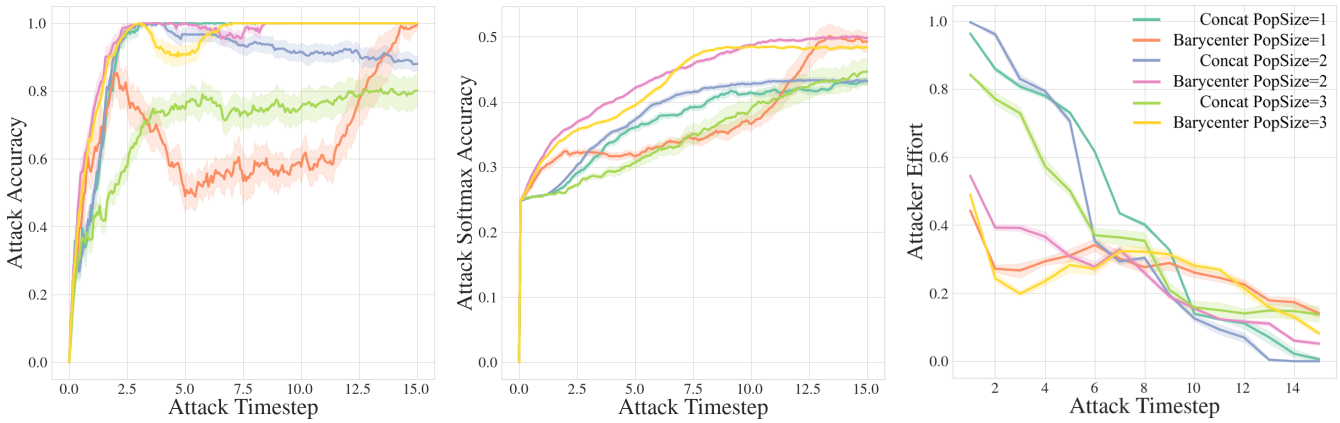

Figure 3: Experiment H1: Accuracies & Effort of attacks trained and tested on same-sized Implicit Collective victim populations

Wasserstein distance [Vaserstein, 1969] to generate a latent distribution that is representative of all individual behaviors approximated from the victim population. Wasserstein distance respects the underlying geometry of the metric space in which the distributions reside. Therefore, unlike other distances like Euclidean, Total Variation, Hellinger etc, Wasserstein distance provides an aggregation mechanism (Barycenter) that preserves the structure of individual behavior distributions; enabling the attacker to understand prevalent victim behaviors and thereby learn an efficient attack strategy for the population. Secondly, Wasserstein distance is insensitive to small changes in distributions which is a crucial property for this work as τ_{k,u_i}s are approximate representations of the actual policies of the victims and hence inherently possess a sizable margin of error. Lastly, Wasserstein distances can be computed between two discrete, two continuous as well as a discrete and a continuous distribution. This property supports scalability of the developed methodology to large, continuous/discrete environments as the developed approach remains agnostic to the nature (discrete/continuous) of ϕ_{k,u_i}. In this work we use a fixed-point approach for fast computation of the Wasserstein barycenter (Fréchet mean), $\overline{\phi}_{u_i}$ [Álvarez-Esteban *et al.*, 2016] of the individual latent behavior distributions ϕ_{k,u_i} corresponding to all K agents present in the victim population; $[\phi_{1,u_i}, \phi_{2,u_i}, ..., \phi_{K,u_i}]$. In formula 2 given below, W stands for L2-Wasserstein distance and λ corresponds to importance of a particular latent behavior distributions ϕ_{k,u_i}. Herein λ_k is assigned the value of 1/K for all k as the behavior of each victim is equally important for the attacker to consider.

$$\sum_{k=1}^{K} \lambda_k W^2(\phi_{k,u_i}, \overline{\phi}_{u_i}) = \min_{\phi \in \Phi} \left\{ \sum_{k=1}^{K} \lambda_k W^2(\phi_{k,u_i}, \phi) \right\} \quad (2)$$

3 Experiments

One of the primary cognitive capabilities is the ability to navigate in a new environment. This work tests and establishes the quality of the proposed methodology by training an attacker to learn to attack a population of navigational agents in a stochastic grid environment titled 3D Grid World [Rabinovich *et al.*, 2010]. This environment simulates an uneven terrain on a grid of 2 dimensional cells. The unevenness corresponds to the 3^{rd} dimension of the grid and is due to the elevation/altitude associated with each grid cell. The relative elevation between two cells decides the transition probability between them. A change in this relative elevation thus changes the manner in which the environment responds to a navigating agent's actions. The navigating agent's task is to find the shortest path from the start cell to the goal cell and its state is its position inside the grid world in Implicit Collective scenario and its position along with the position of all other members of the population, in Swarm and True Collective scenarios. At each time step, the navigating agent observes its state and takes one step in any of the four cardinal directions (N,S,E,W). The agent receives a reward of -1 for every action it takes until it reaches the goal state. A given victim training episode terminates once all agents in Implicit Collective scenario and at least one agent in Swarm and True Collective scenarios, reaches the goal state or maximum time has elapsed. This pushes the agents to find the shortest path(s) to the goal cell. This environment also allows the presence of an additional agent, the elevation expert who can view the altitude of each grid cell and take a constrained action to modify it. The elevation expert's state space comprises of the grid cells' altitudes along with the navigational population's behavior, while its action space is a vector of real numbers $[x^1, x^2, x^3, ..., x^M], x \in [-1.0, 1.0]$ where M is the total number of cells in the grid. In this work, each member of the victim population is a navigating agent while the attacker is the elevation expert. The attacker's objective is to efficiently force the victims to follow a target path to the attacker's desired destination. The target path is not an optimal path in the original environment and thus is not the optimal choice for the victims under default environment dynamics.

The performance of the attacker is measured in terms of the accuracy (Attack Accuracy) and degree of adherence (Attack SoftMax Accuracy) with which the victim population (unknowingly) adopts the target behavior; as well as the changes brought about in the victim environment by the attacker (At-

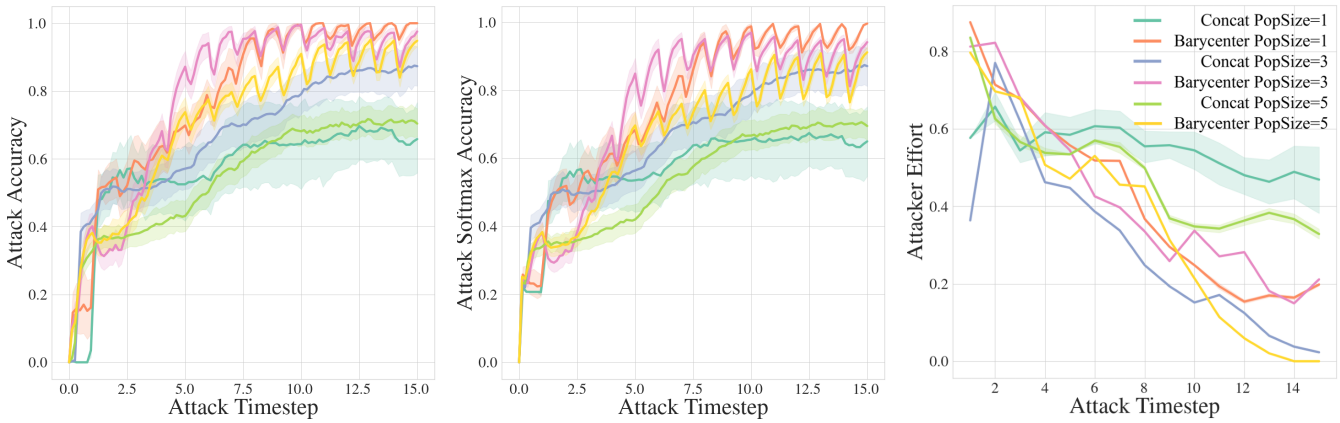

Figure 4: Experiment H1: Accuracies & Effort of attacks trained and tested on same-sized Swarm Collective victim populations

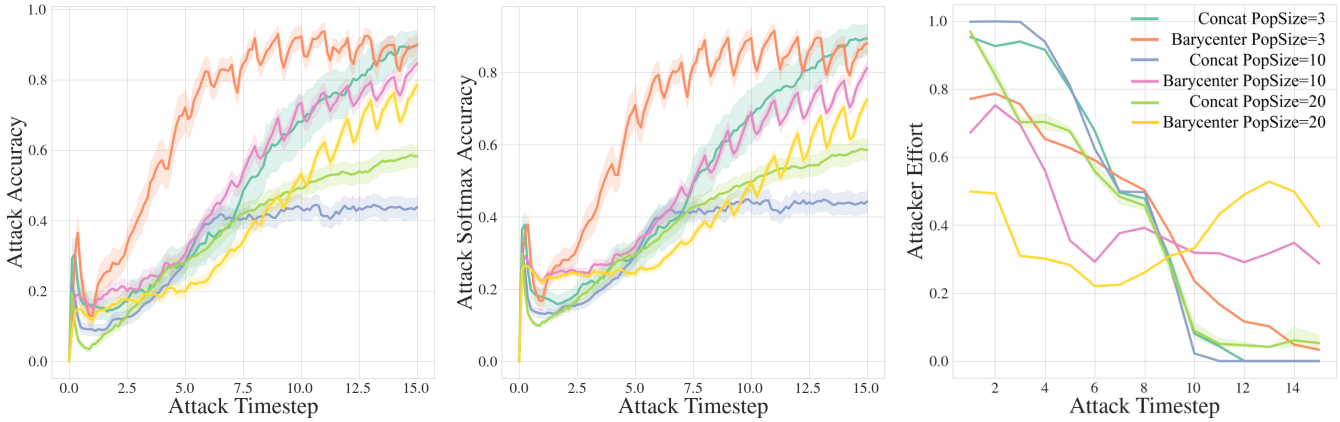

Figure 5: Experiment H1: Accuracies & Effort of attacks trained and tested on same-sized True Collective victim populations

tacker Effort). **Attack Accuracy** (abbreviated as **@Acc**) represents the level of adoption of the target behavior by the victim population i.e. the extent to which the victim population assigns the highest probability to target actions (attacker-desired victim-actions) in target states (states included in the target path). The rate of convergence of @Acc reflects the speed with which the attack results in target behavior's adoption in the victim population. **Attack SoftMax Accuracy (@SoftAcc)** constitutes the probabilities assigned to the target actions by the victim population and encapsulates the population's degree of adherence to the target behavior. **Attacker Effort (@Effort)** represents the magnitude to which the attacker modifies the victim environment and is computed as the mean of the absolute difference between the previous and current attack time step's grid cells' altitudes. These performance metrics are mathematically described in Appendix F.

Plots visualizing these performance metrics are presented for each experiment conducted in this study. The attacker training episodes are 15-step sequential attacks on freshly initialized victim populations wherein attack step 0 corresponds to the original environment with default dynamics. After each episode, the attack strategy employed in that episode is saved if it is better or equal to the best attack strategy found so far, with respect to last-timestep, mean or cumulative value of at least one strategy quality criterion. A given strategy's quality is approximated using 3 internal and 5 external quality criteria which are described in detail in Appendix G. Herein criteria that are approximated by the attacker are referred to as internal while criteria computed by the external system for the purpose of training the attacker are termed external.

The experiment graphs demonstrate performance of the best attack strategies found by the different models. These best strategies are selected by prioritizing @Acc, as the main goal of this work is to find strategies that push victim populations to adopt the target behavior, while, adherence to the target behavior (@SoftAcc) and magnitude of changes made to the environment (@Effort) in order to achieve this adoption demonstrate additional/secondary capabilities of the attack strategies. @Acc and @SoftAcc are measured along the victim timescale to observe how the accuracies change (and thereby understand how the victim population behaves) in-between attack actions. @Effort on the other hand can only be measured corresponding to each attack action and hence is measured along the attacker timescale. Due to this difference,

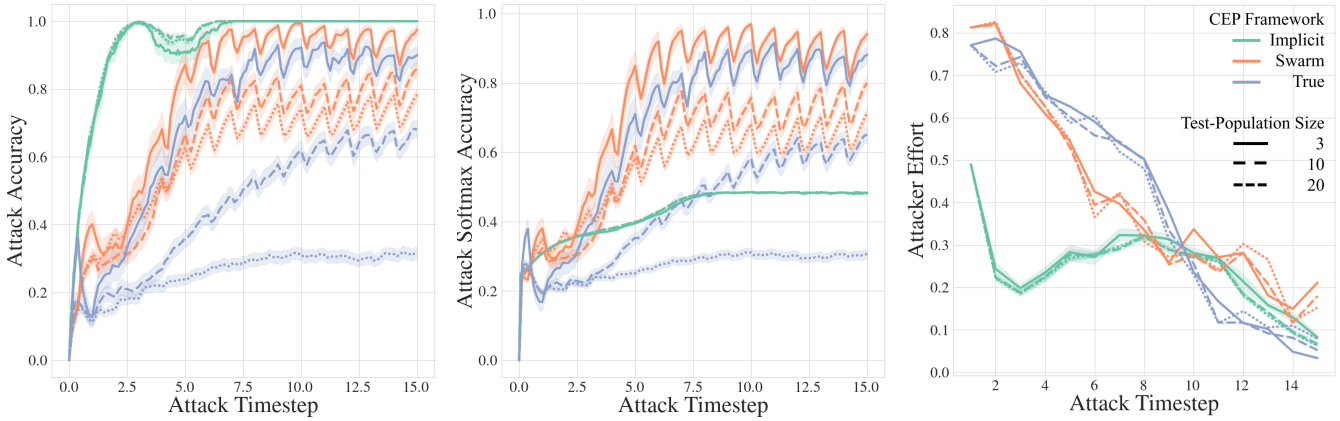

Figure 6: Experiment H2: Accuracies & Effort of Barycenter attacks tested on victim collectives of sizes 3, 10, and 20

accuracy plots begin from attack step 0 while effort plots begin from attack step 1. Each graph corresponds to attacks carried out on 20 separate victim populations.

Given that this is the first work that studies a multi-victim attack via a common attack strategy, we have constructed a high-performance artificial baseline for comparison by extending the single-agent SOTA environment-poisoning model, TEPA [Xu *et al.*, 2021] to the multi-agent setting. TEPA is an auto-encoder-based model that has been shown capable of extracting the behavior of a single victim agent in whitebox and proxy-blackbox adversarial settings. TEPA uses an agent's state, action trajectory to capture its behavior. In this work, TEPA becomes capable of attacking a population by utilizing the concatenation of latent representations of individual behavior trajectories as the population's behavior representation. This population behavior representation includes all available information from individual behavior representations, in its entirety, which on one hand empowers the attacker with more information but on the other hand, does not support size-agnostic attacks and/or attacks transferable to different-sized populations (size-transferable attacks). The method of concatenation is utilized for comparison in order to demonstrate the performance of an approach that makes use of maximum available information compared to our approach that utilizes less information but has size-agnostic capabilities. Two experiments, based on the two hypotheses outlined in Section 1 are conducted under each multi-victim collective scenario. **Experiment H1 - Concatenation vs Barycenter** demonstrates the capability of Wasserstein distance-based Gaussian embedding in capturing the behavior of different-sized victim populations in contrast to the concatenation-based baseline. In this experiment, attack strategies are trained and tested on populations of same size. Each strategy is tested on 20 populations. In Implicit Collective scenario with Q-learning victim agents, 10 test populations use the same seed as the one used by the victim populations during training, while each agent in each of the remaining 10 populations uses a different seed. In Swarm and True Collective scenarios with DQN victim agents, neural networks corresponding to 10 test populations are initialized using random numbers from the same range as used during training, while the remaining 10 test populations are initialized using a different range. **Experiment H2 - Size Agnosticity of Behavior Barycenter** demonstrates the size-agnostic and size-transferable capabilities of the developed barycenter-based approach. The attacker learns attack strategies on populations of sizes 3,5,10, and 20. Each of these strategies is tested on 4 sets of 20 populations of sizes 3,5,10, and 20 respectively. Each set of 20 populations is generated in the same manner as under Experiment H1.

Experiment H1 under Implicit Collective scenario presented in Figure 3 encapsulates the feasibility study of this work wherein the proposed and artificially-constructed baseline methods are used to attack populations of sizes 1,2, and 3. The concatenation-based baseline performs perfectly in terms of @Acc while attacking populations of size 1 (Concat Popsize=1) with convergence to 1.0 @Acc within 3 attack actions. However, as the size of the victim population increases, the @Acc decreases while its variance increases. Moreover, the @Acc of attack on size-3 populations initialized using different seeds is much lower than that on populations initialized using the same seed (see Appendix H). This shows that the baseline attack strategy finds it harder to attack larger populations as well as populations with greater differences from the ones used during training, suggesting that complete information regarding multiple victim trajectories confuses the attacker, especially when those trajectories correspond to very differently initialized victim agents. Population behavior barycenter based attack on the other hand displays the opposite trend. Attack strategy trained on populations of size 1 performs worse than those trained on populations of sizes 2 and 3; in terms of variance as well as rate of convergence of @Acc. Also, interestingly, attacks on populations initialized with different seeds converge slightly faster than those initialized with the same seed (Appendix H). @SoftAcc graph shows that barycenter-based attacks result in stronger adherence to target behavior as the victim populations end up assigning higher probabilities to attacker-desired actions. The @Effort graph shows that at the beginning of the attack, concatenation-based strategies modify nearly all cells by al-

most maximum permissible amount exerting @Effort between 0.8 and 1.0. Barycenter-based approaches on the other hand begin the attack with almost half the level of modification, between 0.4 and 0.6. By the end of the attack, all strategies are able to bound environment modifications between 0.0 and 0.2. This experiment, therefore, demonstrates the feasibility of multi-victim attacks under Blackbox setting by carrying out successful and efficient attacks on different-sized Implicit Collectives using two population behavior representation methods. Experiment H1 under Swarm Collective scenario presented in Figure 4 expands the feasibility study by including attacks on size-5 populations. As seen in Implicit Collective scenario, here too the barycenter-based approach performs better than the baseline and is able to drive @Acc and @SoftAcc to above 0.9 by the end of the attack. This performance degrades only slightly with increase in victim population size. Moreover, barycenter-based approach bounds @Effort to below 0.2 in all strategies, unlike concatenation-based approach where certain strategies exert higher @Effort (in spite of achieving lower @Acc and @SoftAcc). Experiment H1 under True Collective scenario presented in Figure 5 demonstrates the capabilities of the proposed and artificially-constructed baseline methods by testing them on large populations of sizes 10 and 20. Under this scenario, the baseline achieves high @Acc only while attacking small populations of size 3. On the other hand, all barycenter-based strategies achieve high @Acc by the end of the attack. The final @Acc achieved by the last attack time step decreases with increasing population size but remains roughly above 0.8. Similar trends are visible for @SoftAcc. Therefore barycenter-based approach (under all three scenarios) not only pushes the victim populations to assign the maximum probability to attacker-desired target actions but also ensures that this probability itself is large. Under True Collective scenario, @Effort of most concatenation-based strategies begins near 1.0 and ends below 0.1 while that of barycenter-based strategies begins between 0.5 and 0.8 and ends between 0.0 and 0.1 for smaller populations and between 0.3 and 0.4 for larger populations. This observation implies that with increasing population size, concatenation-based approach minimizes effort without achieving high accuracy while barycenter-based approach finds strategies that continue to exert certain effort until the end of the attack in order to achieve high accuracy. It is interesting to note that @Acc and @SoftAcc plots of all barycenter-based attacks exhibit a sawtooth shape as a function of victim learning epochs. This is due to the fact that an attack action causes a re-parameterization of the victim environment but persists over multiple victim training epochs. When victims first experience this re-parameterization, their current policy cannot affect a behavior as well as it used to. As victims' learning continues, their experiences shape the policy, and behavioral accuracy grows. This cycle repeats every time a new attack action is actuated, and is more prominent for more attack-resilient populations. Attack actions are learned in a way that maximizes the sawtooth global incline.

Experiment H2 presented in Figure 6 demonstrates size-agnosticity of the barycenter-based approach. Strategies trained on size-3 populations under all three collective scenarios are tested on larger populations of sizes 10 and 20. Under Implicit Collective scenario performance of barycenter-based approach is agnostic to the testing population size across all three metrics and converges to the perfect @Acc of 1.0, high @SoftAcc of 0.5, and low @Effort of 0.1, by the end of the attack. Under Swarm and True Collective scenarios, for each strategy, accuracy decreases with increasing test population size. However, this decrease reduces with increasing training population size as demonstrated in Appendix H. Therefore, barycenter-based approach is not entirely size-agnostic under Swarm and True Collective scenarios but its performance degrades gracefully when strategies trained on very small populations are tested on large populations. Furthermore, strategies trained under Swarm Collective achieve higher accuracy and lower degradation compared to True Collective scenario. This implies that barycenter-based strategies are more effective at attacking collectives that practice individual (Implicit Collective) or social (Swarm Collective) learning than at attacking collectives that exploit both (True Collective).

4 Conclusion, Limitations and Future Work

This paper develops an extension of environmental poisoning attacks to populations of reinforcement (RL) agents and introduces the Collective Environment Poisoning (CEP) framework that constitutes; a) Implicit Collective (Blackbox); b) Swarm Collective (Ultra-Blackbox); and c) True Collective (Ultra-Blackbox) scenarios. The authors show that concatenation-based population behavior representation, not only creates attack strategies that are non-transferable to different-sized populations but also overloads the attacker with information inhibiting it from finding strategies that achieve high attack accuracy with low attacker effort. In contrast, barycenter-based population behavior representation achieves both of the aforementioned feats in Implicit Collective scenario wherein population members practice individual learning. In Swarm and True Collective scenarios wherein victim agents learn via social and individual+social learning respectively, barycenter-based attack strategies achieve high accuracies with bounded attacker effort when trained and tested on same-sized populations. These strategies are transferable to different-sized populations except for strategies trained on very small populations ($\sim$3) and tested on very large populations ($\sim$20). However, even in such cases, the performance degrades gracefully.

The current methodology approximates each individual victim's policy using the last action taken by the victim in each environment state. This data can however only be captured for victims training in a discrete environment. Similarly, KLR that is used to create the extrinsic (Blackbox) and intrinsic (Ultra-Blackbox) attacker reward signals requires the underlying MDP to be based on environments with discrete state and action spaces. Our next step entails expansion of the proposed methodology to continuous environments, for e.g., by utilizing discrete latent space encoders. Furthermore, the developed CEP framework does not include provisions for attacking heterogeneous victim populations or open multi-victim systems wherein victims can freely enter/exit the system. Future work constitutes expanding CEP to study attacks on heterogeneous, open multi-victim systems.

Acknowledgements

This research was partly supported by the NTU SUG "Choice Manipulation and Security Games".

References

[Álvarez-Esteban *et al.*, 2016] Pedro C Álvarez-Esteban, E Del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.

[Behzadan and Munir, 2017] Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)*, pages 262–275, 2017.

[Chelarescu, 2021] Paul Chelarescu. Deception in social learning: A multi-agent reinforcement learning perspective. *arXiv preprint arXiv:2106.05402*, 2021.

[Chen *et al.*, 2019] Tong Chen, Jiqiang Liu, Yingxiao Xiang, Wenjia Niu, Endong Tong, and Zhen Han. Adversarial attack and defense in reinforcement learning-from ai security view. *Cybersecurity*, 2(1):1–22, 2019.

[Chen *et al.*, 2022] Yanjiao Chen, Zhicong Zheng, and Xueluan Gong. Marnet: Backdoor attacks against cooperative multi-agent reinforcement learning. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[Dimakopoulou *et al.*, 2018] Maria Dimakopoulou, Ian Osband, and Benjamin Van Roy. Scalable coordinated exploration in concurrent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

[Lample and Chaplot, 2017] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *Thirty-First Association for the Advancement of Artificial Intelligence Conference (AAAI)*, 2017.

[Lin *et al.*, 2020] Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. On the robustness of cooperative multi-agent reinforcement learning. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 62–68, 2020.

[Liu *et al.*, 2023] Tongtong Liu, Joe McCalmon, Md Asifur Rahman, Cameron Lischke, Talal Halabi, and Sarra Alqahtani. Weaponizing actions in multi-agent reinforcement learning: Theoretical and empirical study on security and robustness. In *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 347–363, 2023.

[Lupu and Precup, 2020] Andrei Lupu and Doina Precup. Gifting in multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 789–797, 2020.

[Marthi *et al.*, 2005] Bhaskara Marthi, Stuart Russell, David Latham, and Carlos Guestrin. Concurrent hierarchical reinforcement learning. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 779–785, 2005.

[Mialon *et al.*, 2020] Grégoire Mialon, Dexiong Chen, Alexandre d'Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation. In *International Conference on Learning Representations (ICLR)*, 2020.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Parisotto *et al.*, 2019] Emilio Parisotto, Soham Ghosh, Sai Bhargav Yalamanchi, Varsha Chinnaobireddy, Yuhuai Wu, and Ruslan Salakhutdinov. Concurrent meta reinforcement learning. *arXiv preprint arXiv:1903.02710*, 2019.

[Pham *et al.*, 2022] Nhan Pham, Lam Nguyen, Jie Chen, Lam Thanh Hoang, Subhro Das, and Lily Weng. c-mba: Adversarial attack for cooperative marl using learned dynamics model. In *Machine Learning Safety Workshop at NeurIPS*, 2022.

[Qi *et al.*, 2017] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 652–660, 2017.

[Rabinovich *et al.*, 2010] Zinovi Rabinovich, Lachlan Dufton, Kate Larson, and Nicholas R. Jennings. Cultivating desired behaviour: Policy teaching via environment-dynamics tweaks. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1097–1104, 2010.

[Rakhsha *et al.*, 2020] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 7974–7984. PMLR, 2020.

[Skianis *et al.*, 2020] Konstantinos Skianis, Giannis Nikolentzos, Stratis Limnios, and Michalis Vazirgiannis. Rep the set: Neural networks for learning set representations. In *International conference on artificial intelligence and statistics (AISTATS)*, pages 1410–1420, 2020.

[Vaserstein, 1969] Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

[Xu *et al.*, 2021] Hang Xu, Rundong Wang, Lev Raizman, and Zinovi Rabinovich. Transferable environment poisoning: Training-time attack on reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1398–1406, 2021.

[Xu *et al.*, 2022] Hang Xu, Xinghua Qu, and Zinovi Rabinovich. Spiking pitch black: Poisoning an unknown environment to attack unknown reinforcement learners. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1409–1417, 2022.

[Yang *et al.*, 2020] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15208–15219, 2020.

[Zaheer *et al.*, 2017] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.