Robust and Scalable Autonomous Reinforcement Learning in Irreversible Environments

Sang-Hyun Lee
Department of Automotive Engineering
Ajou University
Gyeonggi-do, South Korea
sanghyunlee@ajou.ac.kr

Abstract

Reinforcement learning (RL) typically assumes repetitive resets to provide an agent with diverse and unbiased experiences. These resets require significant human intervention and result in poor training efficiency in real-world settings. Autonomous RL (ARL) addresses this challenge by jointly training forward and reset policies. While recent ARL algorithms have shown promise in reducing human intervention, they assume narrow support over the distributions of initial or goal states and rely on task-specific knowledge to identify irreversible states. In this paper, we propose a robust and scalable ARL algorithm, called RSA, that enables an agent to handle diverse initial and goal states and to avoid irreversible states without task-specific knowledge. RSA generates a curriculum by identifying informative states based on the learning progress of an agent. We hypothesize that informative states are neither overly difficult nor trivially easy for the agent being trained. To detect and avoid irreversible states without task-specific knowledge, RSA encodes the behaviors exhibited in those states rather than the states themselves. Experimental results demonstrate that RSA outperforms existing ARL algorithms with fewer manual resets in both reversible and irreversible environments.

1 Introduction

Reinforcement learning (RL) has demonstrated remarkable achievements in the field of robotics [16, 14, 8, 18]. However, most of these achievements rely on repeated resets between episodes to provide an agent with multiple attempts and unbiased experiences. While such resets are easily performed in simulated settings, they require substantial human intervention and lead to poor training efficiency in the real world [28, 6, 12]. Autonomous RL (ARL), which simultaneously learns how to solve a task and how to reset an environment, has gained significant attention as a promising alternative to reducing human intervention. The key idea behind ARL is to generate a curriculum that determines when to abort episodes and where to return an agent. For example, the curriculum encourages an agent to explore space near initial or goal states in the early stages of training and gradually move to more distant space as training progresses.

Recent ARL works have shown that their curricula enable an agent to learn diverse tasks with fewer manual resets [9, 31, 24, 26, 20]. However, these works require narrow support over the initial or goal state distributions to generate their curricula. These restricted distributions can lead to less robust performance. Furthermore, their curricula either do not consider irreversible states or rely on task-specific knowledge to identify them. Irreversible states are those from which an agent cannot recover without external intervention, such as when a vehicle is damaged in an accident or a manipulator pushes an object outside its workspace. Most real-world tasks involve such irreversible states, and the task-specific knowledge required to identify them is rarely available in practice.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

In this paper, we introduce RSA, a robust and scalable ARL algorithm that enables an agent to handle diverse initial and goal states and to detect and avoid irreversible states without task-specific knowledge. RSA generates a curriculum by identifying informative initial and goal states based on the agent's learning progress. We hypothesize that informative states are those that are neither overly difficult nor trivially easy for the agent under training. To detect and avoid irreversible states, RSA encodes the behaviors exhibited in such states rather than the states themselves. This is based on our observation that, while irreversible states may vary across tasks, the behaviors tend to follow common patterns.

The main contribution of this work is a robust and scalable ARL algorithm called RSA. We evaluate RSA against baselines on diverse navigation and manipulation tasks. Experimental results demonstrate that RSA generates a curriculum by identifying informative states based on the agent's learning progress, achieving better performance with fewer manual resets than the baselines in both reversible and irreversible environments.

2 Related Work

RL agents trained in simulation often struggle to perform well in the real world due to the fidelity gap between simulated and real-world environments. A straightforward approach to avoid this gap is to train agents directly in the real world. Chebotar et al. [6] combine model-based and model-free updates to learn manipulation tasks on a PR2 robot. Kendall et al. [12] demonstrate the first RL agent capable of driving a real-world vehicle along country roads. However, these works rely on manual resets after every episode, which require substantial human intervention. While several previous works employ additional instrumentation or scripted reset behaviors to automate environment resets, their reset strategies are task-specific and limited to particular scenarios [15, 30, 18, 29, 22].

ARL, which aims to learn without manual resets, has been actively studied in recent years [9, 27, 31, 23, 25, 24, 26, 13, 20]. Eysenbach et al. [9] train forward and reset policies simultaneously and generate a curriculum by early aborting forward episodes based on the reset value function. While their curriculum encourages agents to explore space they have already learned, Patil et al. [20] generate a curriculum that guides an agent to explore space they have not yet sufficiently learned. Both works assume an unimodal initial state distribution with narrow support, which can cause sub-optimal and non-robust performance. To address this limitation, Zhu et al. [31] introduce a random perturbation controller that discovers novel initial states, gradually broadening the support of the training distribution. Sharma et al. [24] encourage the agent to return to states from expert demonstrations, based on the hypothesis that such demonstrations provide a desired distribution over initial states. Although these works focus on ensuring diverse initial states, their curricula do not account for multiple goal settings, which makes their algorithms less scalable. To address this limitation, our work generates a curriculum that provides an agent with diverse initial and goal states by identifying informative states based on the agent's learning progress.

While most previous works assume that environments are reversible, our work aims to reduce manual resets in both reversible and irreversible environments. Eysenbach et al. [9] and Xie et al. [26] are most closely related to ours, as they focus on reducing manual resets in irreversible environments. Eysenbach et al. [9] leverage the state-action value function, which is trained with reset reward functions, as a metric to determine whether an agent is in irreversible states. Xie et al. [26] propose a label-efficient binary search algorithm that detects irreversible states using a limited number of reversibility labels. Experimental results from both works demonstrate that their algorithms can identify irreversible states and prevent agents from entering them. However, in real-world scenarios, access to task-specific knowledge, such as reset reward functions or reversibility labels, is often unavailable. Our work enables an agent to identify and avoid irreversible states without task-specific knowledge. Table 1 summarizes the key features that distinguish our work from previous works.

3 Preliminaries

3.1 Goal-Conditioned Reinforcement Learning

We consider a goal-conditioned reinforcement learning (GCRL) task represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{T}, r_g, \rho_0, \gamma, \rho_g, m)$. $\mathcal{S}$ is the set of states s, $\mathcal{A}$ is the set of actions a, $\mathcal{G}$ is the set of goal states g, and $\mathcal{T} : S \times A \times S \to [0, 1]$ is the state transition model. $r_g : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the

Table 1: Comparison of Autonomous Reinforcement Learning (ARL) algorithms.

Property	LNT [9]	R3L [31]	MEDAL [24]	PAINT [26]	RSA (Ours)
Diverse Initial States	✗	✓	✓	✓	✓
Diverse Goal States	✗	✗	✗	✗	✓
No Extrinsic Reset Rewards	✗	✓	✓	✓	✓
No Demonstrations	✓	✓	✗	✓	✓
No Reversability Labels	✓	✓	✓	✗	✓
Irreversible Environments	✓	✗	✗	✓	✓

goal-conditioned reward function, ρ_0 is the initial state distribution, and γ is the discount factor. ρ_g is the goal state distribution, and $m : \mathcal{S} \to \mathcal{G}$ is a mapping function from states to corresponding goal states. The goal space may either be a subset of the state space or the state space itself. In the latter case, the mapping function is the identity function. An agent's behaviors are determined by a goal-conditioned policy and value function. The goal-conditioned policy $\pi(a \mid s, g)$ maps the current state and the current goal to a probability distribution over actions and the goal-conditioned value function $Q^\pi(s, a, g)$ represents the expected return when an agent takes action a in state s and goal g and follows the policy π. The goal of GCRL is to find the optimal policy that maximizes the expected return when the state transition model is unknown. Please refer to [11, 21, 1] for further details.

3.2 Successor Features

Assume the reward function is a linear combination of a feature vector $\phi(s, a, s') \in \mathbb{R}^d$ and a weight vector $\omega \in \mathbb{R}^d$, such that the reward is given by $r(s, a, s') = \phi(s, a, s')^\top \omega$. Following [2, 17, 10], the goal-conditioned reward function can be similarly decomposed as $r_g(s, a, s') = \phi(s, a, s')^\top \omega_g$, as the weight vector, also referred to as a task vector, encodes preferences over individual feature components. With this formulation, the goal-conditioned value function can then be written as follows:

$$
\begin{aligned}
Q^\pi(s, a, g) &= \mathbb{E}_\pi[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1}^T \omega_g \mid S_t = s, A_t = a] \\
&= \mathbb{E}_\pi[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1} \mid S_t = s, A_t = a]^\top \omega_g = \psi^\pi(s, a)^\top \omega_g.
\end{aligned}
$$

Barreto et al. [2] call $\psi^\pi(s, a)$ the Successor Features (SFs), which describe the expected discounted sum of features $\phi(s, a, s')$ under a policy π. The SFs $\psi^\pi(s, a)$ represent the expected discounted sum of features encountered when following policy π and can be regarded as a multi-dimensional value function, where a feature vector acts as a reward function. This implies that SFs can be trained with standard reinforcement learning algorithms.

4 Robust and Scalable Autonomous Reinforcement Learning

RSA is designed to enable an agent to achieve robust performance with fewer manual resets in both reversible and irreversible environments. Similar to previous ARL algorithms [9, 31, 24, 26], RSA defines a forward policy $\pi_f(a \mid s, g)$ and a reset policy $\pi_r(a \mid s, g)$, and alternates between them. The forward policy is trained to solve tasks and the reset policy is trained to reset the environment. Note that each policy takes the current goal state as an additional input.

RSA considers two distinctive and challenging settings. First, RSA allows initial and goal states to be located anywhere in the state space $\mathcal{S}$. This contrasts with previous works that assume unimodal and narrow supports for the initial or goal state distributions to generate their curricula. Second, RSA assumes neither reversible environments nor access to task-specific knowledge for identifying irreversible states. This presents difficulties for previous works that rely on such task-specific knowledge, including reversibility labels or reset reward functions, to identify irreversible states.

To handle these settings, RSA 1) generates a curriculum by identifying informative initial and goal states based on the learning progress of an agent, and 2) identifies irreversible states by encoding the behaviors exhibited in those states without task-specific knowledge. In the remainder of this section, we describe how RSA generates the curriculum and identifies irreversible states in detail.

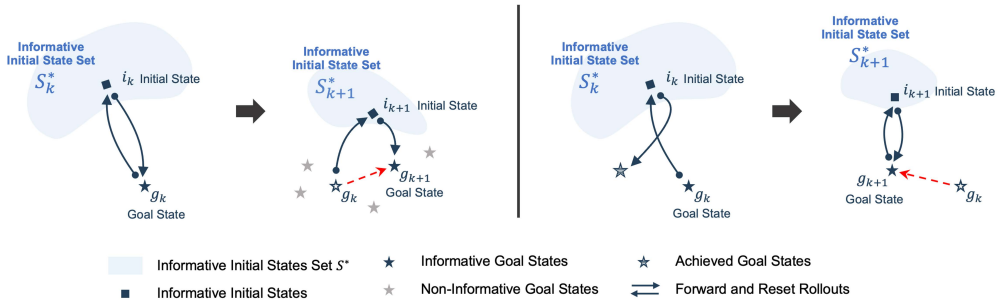

Figure 1: Curriculum generated by RSA. Given the current goal state g_k, the reset policy is activated until the agent reaches the informative initial state set S_k^* corresponding to that goal. **Left:** At the end of each iteration, RSA samples the next goal state from the informative goal state set. **Right:** If the agent fails to reach the current goal state g_k, RSA uses the final state s_T of the current iteration as the next goal state g_{k+1}.

4.1 Identifying Informative Initial and Goal States

Figure 1 illustrates how RSA generates a curriculum by identifying informative initial and goal states. The curriculum is built on our hypothesis that informative states are neither overly difficult nor trivially easy for the agent being trained. If the initial or goal states are too difficult, the agent may take unsafe actions and fail to reach the goal. On the other hand, the agent may struggle to obtain useful information even with sufficient time if the initial or goal states are too easy. Both cases can lead to suboptimal performance and poor sample efficiency. To identify informative states based on the agent's learning progress, RSA introduces the state information estimator, $I(s, g)$, which takes a pair of initial and goal states as input and estimates the probability of reaching the goal state from the initial state when the agent follows the forward policy being trained.

Here we describe how RSA utilizes the state information estimator to identify informative goal and initial states. RSA determines the informative goal state for the k th iteration, g_k, as follows:

$$g_k \sim Unif(\mathcal{G}_k^*), \quad \text{where } \mathcal{G}_k^* \triangleq \{g \in \mathcal{G}_{1:k-1} \mid \lambda_1 \leq \mathbb{E}_{s \sim B_r}[I(s, g)] \leq \lambda_2\}. \tag{1}$$

$\mathcal{G}_k^*$ is the set of informative goal states for the kth iteration, $\mathcal{G}_{1:k-1}$ is the set of goal states obtained until the previous iterations, B_r is the replay buffer for the reset policy, and λ_1 and λ_2 are the lower and upper thresholds of the reachability probability over the pairs of initial and goal states, respectively. This prevents an agent from being assigned goal states that are either too difficult or too easy, which allows it to obtain informative transitions. Specifically, a goal state is uniformly sampled from the set of informative goal states at the beginning of each iteration. If the agent reaches a goal state and it is still identified as part of the informative goal state set, RSA reuses it as the goal state for the next iteration rather than sampling a new one.

After the informative goal state g_k is determined, RSA attempts to discover the informative initial state i_k by activating the reset policy until the agent reaches the set of informative initial states $\mathcal{S}_k^* \subset S$ as follows:

$$\mathcal{S}_k^* \triangleq \{s \in S \mid \lambda_1 \leq I(s, g_k) \leq \lambda_2 \text{ for } g_k \sim Unif(\mathcal{G}_k^*)\}. \tag{2}$$

This prevents the agent from resetting with too challenging or too easy initial states. To continuously discover novel initial states, RSA trains the reset policy with an off-the-shelf exploration algorithm that guides an agent to under-explored states. While we implement the exploration algorithm using random network distillation (RND) [5] in our experiments due to its scalability and ease of implementation, RSA is compatible with any exploration algorithm [3, 19, 4].

Even when our curriculum provides an agent with informative initial and goal states, the agent being trained may still fail to reach the goal states. Repeated failures can cause the agent to be stuck in uninformative states. To address this challenge, when the agent fails to reach the goal state, RSA assigns the final state of the current iteration as the goal state for the next iteration. This idea is inspired by hindsight experience replay (HER) [1], which enables an agent to learn from undesired outcomes. The key distinction between RSA and HER is that, whereas HER treats the final state as the goal state within the same iteration, RSA uses the final state of the current iteration as the goal state for the next iteration.

Figure 2: Overview of the training procedure for the state information estimator. State-goal pairs from successful forward rollouts are treated as positive examples, while those from failed forward rollouts are treated as negative examples.

The state information estimator, which estimates the probability of reaching goal states from initial states, is a key component that enables RSA to provide an agent with diverse initial and goal states. The main challenge in training the state information estimator is the absence of explicit supervisory signals to estimate reachability based on the agent's learning progress. To address this, RSA extracts reachability labels e_t from forward rollouts and uses them to train the state information estimator in a self-supervised manner. Specifically, state-goal pairs from the rollouts in which the agent successfully reaches the goal state are treated as positive examples, while those from failed rollouts are treated as negative examples. The objective of the state information estimator can be written as follows:

$$\min_{I} -\mathbb{E}_{(s_t,g_t,e_t)\sim B_f}[e_t \log(I(s_t,g_t)) + (1-e_t)\log(1-I(s_t,g_t))], \tag{3}$$

where e_t is 1 for positive examples and 0 otherwise, and B_f is the replay buffer for the forward policy. Figure 2 illustrates how RSA extracts supervisory signals from forward rollouts to train the state information estimator.

4.2 Detecting and Avoiding Irreversible States

RSA enables an agent to identify irreversible states without task-specific knowledge by encoding the behaviors exhibited in those states, rather than the states themselves. This is based on our observation that, while irreversible states have task-specific features, the behaviors tend to follow a shared pattern: an agent in irreversible states loses control over goal-relevant features, regardless of the actions it takes. To identify this shared pattern, RSA leverages SFs to encode behaviors induced by the reset policy. The SFs are denoted by $\psi^{\pi_r}(s,a) = \mathbb{E}\pi_r[\sum_{k=t}^{\infty}\gamma^{k-t}\phi_{k+1}|S_t=s, A_t=a]$. Note that the reset policy continues to generate diverse behaviors and the feature vector is defined as the norm of difference in goal-relevant features, $\phi_k = \|m(s_{k+1}) - m(s_k)\|$, where m is the mapping function discussed in Section 3. RSA is agnostic to the form of the mapping function, and learning it with state-of-the-art representation learning is complementary to our work. However, we leave this for future work, as it is orthogonal to our main contribution.

We expect the SFs for irreversible state-action pairs to be much lower than those for reversible pairs. The set of irreversible states, $\mathcal{S}_{\text{irr}} \subset S$, can then be identified as follows:

$$S_{\text{irr}} \triangleq \{s \in S \mid \psi^{\pi_r}(s,a) \leq \lambda_3, \quad \text{for } a \sim \pi_r(a|s,g)\} \tag{4}$$

where λ_3 is the reversibility threshold. RSA leverages two techniques to enable an agent to detect and avoid the set of irreversible states S_{irr}. First, RSA employs a surrogate reward function that penalizes the agent when it enters the identified irreversible states. This is similar to Eysenbach et al. [9] and Xie et al. [26], in which the agent also receives penalties for visiting irreversible states. However, unlike both previous works, RSA does not depend on task-specific knowledge to identify such states. Second, RSA conservatively identifies the irreversible states, by setting the threshold parameter λ_3 higher than that for the actual set, and aborts forward episodes when the agent enters any state in the identified set. We empirically found that this conservative identification helps prevent the agent from entering irreversible states in our experiments.

Algorithm 1 Robust and Scalable Autonomous Reinforcement Learning

Initialize reset and forward policies $\pi_r(a|s,g), \pi_f(a|s,g)$
Initialize reset and forward replay buffers B_r, B_f
Initialize state information estimator $I(s,g)$ and successor features $\psi^{\pi_r}(s,a)$
for $k \leftarrow 1 \dots K$ **do**
  Sample goal state g_k with get_informative_goal$(B_r, \mathcal{G}_{1:k-1}, I(s,g))$
  for $t \leftarrow 1 \dots T_{\text{reset}}$ **do**
    Select reset action $a_t \sim \pi_r(a_t|s_t, g_k)$
    if $\lambda_1 \leq I(s_t, g_k) \leq \lambda_2$ **then**
      Set s_t as initial state and switch to forward policy
    end if
    Obtain reset transition $(s_t, a_t, r_t, s_{t+1}, g_k)$
    Compute surrogate reset reward $\hat{r}_t$
    Add transition to reset buffer $B_r \leftarrow B_r \cup \{(s_t, a_t, \hat{r}_t, s_{t+1}, g_k)\}$
    Update reset policy $\pi_r(a|s,g)$ and successor features $\psi^{\pi_r}(s,a)$
  end for
  for $t \leftarrow 1 \dots T_{\text{forward}}$ **do**
    Select forward action $a_t \sim \pi_f(a_t|s_t, g_k)$
    if $\psi^{\pi_r}(s_t, a_t) \leq \lambda_3$ **then**
      Abort and switch to reset policy
    end if
    Obtain forward transition $(s_t, a_t, r_t, s_{t+1}, g_k)$
    Compute surrogate forward reward $\hat{r}_t$
    Add transition to forward buffer $B_f \leftarrow B_f \cup \{(s_t, a_t, \hat{r}_t, s_{t+1}, g_k)\}$
    Update forward policy $\pi_f(a|s)$ and state information estimator $I(s,g)$
  end for
end for

4.3 Algorithm Summary

Algorithm 1 outlines the overall training procedure of RSA. At the beginning of each iteration, RSA utilizes the state information estimator to identify a set of informative goal states from all goal states encountered in previous forward rollouts. It then samples one informative goal state from this set for the current episode. Once the goal state is determined, RSA activates the reset policy until the agent reaches an informative initial state, such that the predicted probability of reaching the goal state lies between λ_1 and λ_2. Upon reaching such a state, RSA designates the current state as the initial state and switches to the forward policy. When the forward policy is activated, RSA aborts the episode early if the SFs for the reset policy are less than or equal to the reachability threshold λ_3. Please refer to Appendix A for additional details.

5 Experiments

We design our experiments to investigate the following questions: (1) Can RSA achieve more robust performance with fewer manual resets than previous algorithms in both reversible and irreversible environments? (2) Can RSA generate a curriculum by identifying informative initial and goal states based on the learning progress of an agent? (3) How does each main component of RSA contribute to its performance improvement?

To answer these questions, we evaluate RSA against the following baselines: (1) LNT, (2) LNT-MG, (3) R3L, and (4) R3L-MG. LNT [9] uses reset reward functions to identify irreversible states and assumes narrow initial state distributions to generate its curriculum. R3L [31] generates a curriculum with diverse initial states but cannot handle irreversible states. Since neither algorithm considers multi-goal settings, we implement LNT-MG and R3L-MG as variants of LNT and R3L that sample goals randomly and periodically from their buffers. Note that RSA generates a curriculum that provides an agent with diverse initial and goal states and does not use reset reward functions to detect irreversible states. While all baselines perform manual resets either when the agent fails to return to starting states within a fixed number of episodes or enters irreversible states, RSA triggers manual

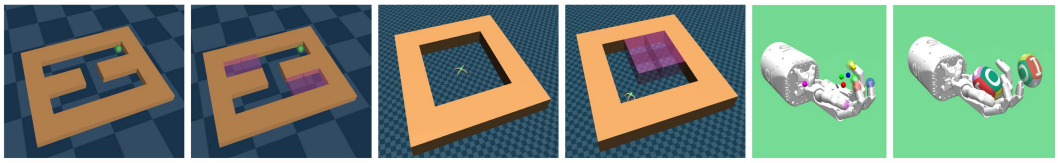

Figure 3: Navigation and manipulation tasks used in our experiments. PointIMaze and AntOpen are reversible, while PointIMaze-Trap and AntOpen-Trap include irreversible states, indicated by purple boxes. HandReach is reversible, whereas HandManipulate involves irreversible states, such as dropping the object from the hand.

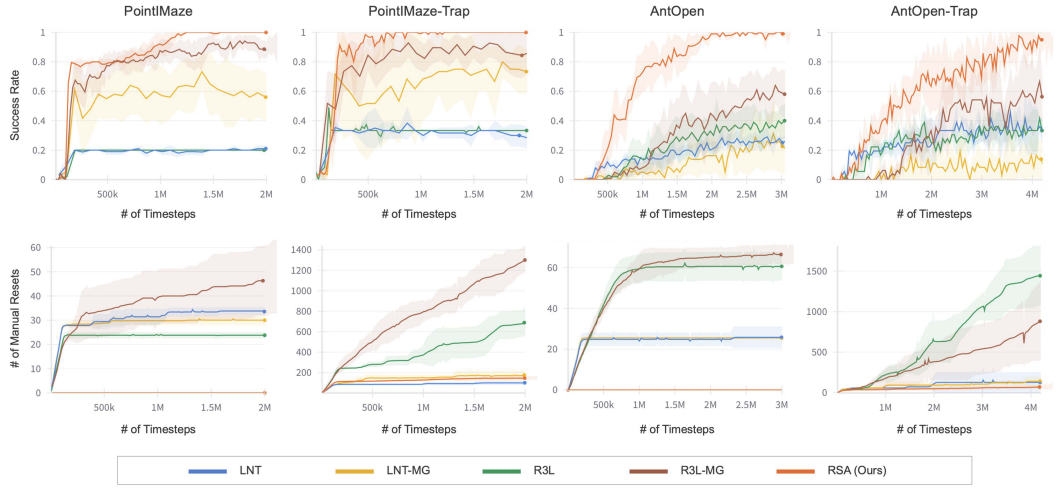

Figure 4: Learning curves for navigation tasks. The x-axis indicates the number of training steps, while the y-axis shows the success rate (top row) and the number of manual resets (bottom row). Darker-colored lines represent the means, and shaded regions denote the standard deviations across 5 random seeds.

resets only when the agent enters irreversible states. Appendix B provides additional details on our experimental setup and implementation.

5.1 Environments

Figure 3 shows the navigation and manipulation tasks used in our experiments. All tasks are provided by Gymnasium-Robotics [7]. The common objective across these tasks is to move an agent, such as a mobile robot, the fingertips of the hand, or an object, to target locations as quickly as possible. PointIMaze, AntOpen, and HandReach do not involve irreversible states, whereas PointIMaze-Trap, AntOpen-Trap, and HandManipulate contain such states. For example, when an agent in a navigation task enters a trap shown as a purple box, it struggles to move and cannot escape without external intervention. This is analogous to a vehicle damaged in an accident that is unable to move on its own. In the manipulation tasks, once an object is dropped, an agent can no longer pick it up or control it. For a fair comparison, we evaluate both the baselines and our algorithm using the same set of episodes with diverse initial and goal states. Appendix C provides additional details on the environments.

5.2 Experimental Results and Analysis

Figure 4 illustrates the learning curves computed over 5 random seeds for both reversible and irreversible navigation tasks. R3L and LNT achieve low success rates due to their narrow initial or goal state distributions. While R3L-MG and LNT-MG obtain robust performance in PointIMaze and PointIMaze-Trap, both suffer performance drops in AntOpen and AntOpen-Trap. We empirically observed that in AntOpen and AntOpen-Trap, agents tend to collect more biased experiences compared

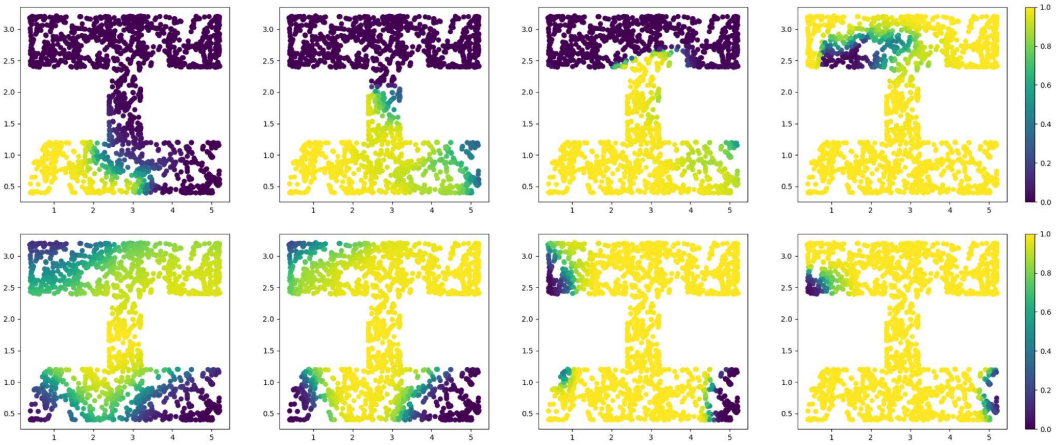

Figure 5: Changes in informative initial state sets over training. Each column shows how the informative initial states change as training progresses from left to right. The color of each initial state indicates its predicted reachability probability for a fixed goal state. **Top:** The goal state is located at (0.8, 0.8). **Bottom:** The goal state is located at (2.8, 1.8).

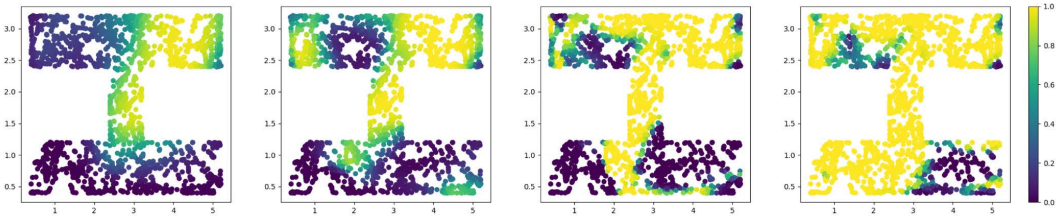

Figure 6: Changes in informative goal state sets over training. Each column shows how the informative goal states for a fixed initial state change as training progresses from left to right. The initial state is fixed at (4.8, 2.8), and the color of each goal state indicates its predicted reachability probability from that initial state.

to PointIMaze and PointIMaze-Trap. This suggests that randomly sampling initial and goal states from the buffer fails to provide an agent with informative transitions when it struggles to explore diverse states. RSA achieves the best performance across all tasks and converges faster and more stably than the baselines.

LNT and LNT-MG require significantly fewer manual resets than R3L and R3L-MG. This result is expected, as LNT and LNT-MG assume narrow initial state distributions and leverage privileged information about irreversible states in the form of reset reward functions. However, constrained initial states make it difficult for an agent to collect diverse transitions, leading to suboptimal performance. Moreover, privileged information about irreversible states is rarely available in practice. RSA does not trigger manual resets in reversible navigation tasks, PointIMaze and AntOpen. This suggests that RSA can generate a curriculum that enables an agent to collect informative transitions even when it fails to return to initial states or reach to goal states. In the irreversible tasks, PointIMaze-Trap and AntOpen-Trap, RSA requires fewer manual resets than RSL and RSL-MG, and triggers a comparable number of resets to LNT and LNT-MG. These results indicate that, unlike LNT and LNT-MG, RSA can identify irreversible states without task-specific knowledge.

To investigate whether RSA identifies informative state sets based on the agent's learning progress, we visualize how the informative initial and goal state sets evolve during training on PointIMaze. Note that, in our work, informative initial and goal states are those whose estimated reachability probabilities fall between the hyperparameters λ_1 and λ_2. Figure 5 illustrates the changes in the informative initial state set for two different goal states: $(0.8, 0.8)$ and $(2.8, 1.8)$. In the early stages of training, the identified initial state sets are close to the respective goal states and gradually move farther away as training progresses. Figure 6 shows the changes in the informative goal state set for an initial state $(4.8, 2.8)$. We observed that similar to the informative initial state sets shown in Figure

5, the informative goal states are identified near the given initial state at the beginning of training and gradually spread out in diverse directions over time. These results demonstrate that RSA identifies informative initial and goal states based on the agent's learning progress. They also suggest that the curriculum generated by RSA enables an agent to efficiently bootstrap from success on easier initial and goal states to tackle more challenging ones. Appendix C provides additional results on the identification of irreversible states.

Finally, we conducted an ablation study to examine the benefits of two key components of RSA: (1) the state information estimator for identifying informative states, and (2) the successor features (SFs) for detecting and avoiding irreversible states. The left plot of Figure 7 shows the performance of RSA and its variant, RSA w/o SIE, which does not identify informative states, on HandReach. Note that in this reversible task, neither RSA nor RSA w/o SIE triggers manual resets, as both algorithms reuse the final state of the current iteration as the next goal state instead of resetting the environment. The performance gap between RSA and RSA w/o SIE indicates that identifying informative initial and goal states contributes to better asymptotic performance and sample efficiency. The right plot of Figure 7 illustrates the performance of RSA and another variant, RSA w/o SFs, which does not identify irreversible states, on HandManipulate. Although both RSA and RSA w/o SFs can identify informative states using the state information estimator, we observed that RSA achieves better performance with the same number of manual resets. This result confirms that SFs of RSA play an important role in reducing manual resets.

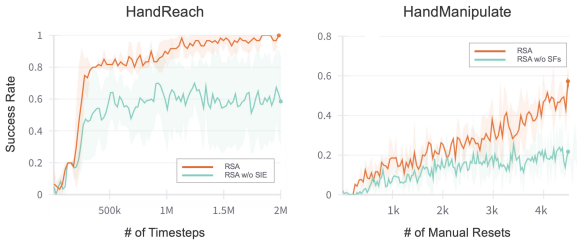

Figure 7: Ablation results of RSA components. The darker lines and shaded areas represent the mean and standard deviation over 5 random seeds. **Left:** Effect of identifying informative states on performance in HandReach. **Right:** Effect of identifying irreversible states on performance in HandManipulate.

6 Conclusion

We introduce a robust and scalable autonomous reinforcement learning (ARL) algorithm, referred to as RSA, that reduces manual resets in both reversible and irreversible environments. RSA identifies informative states to generate a curriculum that provides an agent with diverse initial and goal states. Furthermore, RSA enables an agent to detect and avoid irreversible states without task-specific knowledge. Experimental results demonstrate that RSA allows an agent to identify informative states based on the agent's learning progress and to avoid irreversible states, achieving better performance with fewer manual resets than existing ARL algorithms. RSA has several limitations that should be addressed in future work. First, RSA uses randomized reset behaviors, which are inefficient for discovering initial states in complex environments. We plan to investigate whether leveraging structured and consistent exploration behaviors can mitigate this limitation. Second, RSA uses predefined features to encode behaviors via successor features. We expect that combining RSA with state-of-the-art representation learning algorithms will be a promising approach to alleviating this limitation. Finally, RSA cannot prevent an agent from entering irreversible states before the associated behaviors have been encoded. We believe that the common-sense reasoning abilities of foundation models will make them promising for identifying irreversible states across diverse tasks, even in the early stages of training.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00255968) grant funded by the Korea government(MSIT), Korea Ministry of Trade Industry and Energy, Korea Planning & Evaluation Institute of Industrial Technology(KEIT) under Grant RS-2024-00448797, and Convergence and Open Sharing System Project granted by the Ministry of Education and National Research Foundation of Korea.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[2] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[4] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *ICLR*, 2019.

[5] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, pages 4003–4019, 2019.

[6] Yevgen Chebotar, Karol Hausman, Marvin Zhang, Gaurav Sukhatme, Stefan Schaal, and Sergey Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In *International conference on machine learning*, pages 703–711. PMLR, 2017.

[7] Rodrigo de Lazcano, Kallinteris Andreas, Jun Jet Tai, Seungjae Ryan Lee, and Jordan Terry. Gymnasium robotics, 2024. URL `http://github.com/Farama-Foundation/Gymnasium-Robotics`.

[8] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016.

[9] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, pages 1799–1816, 2018.

[10] Jiaocheng Hu, Yuexin Ma, Haiyun Jiang, Shaofeng He, Gelu Liu, Qizhen Weng, and Xiangwei Zhu. A new representation of universal successor features for enhancing the generalization of target-driven visual navigation. *IEEE Robotics and Automation Letters*, 2024.

[11] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–1098. Citeseer, 1993.

[12] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pages 8248–8254. IEEE, 2019.

[13] Sang-Hyun Lee and Seung-Woo Seo. Self-supervised curriculum generation for autonomous reinforcement learning without task-specific knowledge. *IEEE Robotics and Automation Letters*, 2024.

[14] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[15] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.

[16] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[17] Chen Ma, Dylan R Ashley, Junfeng Wen, and Yoshua Bengio. Universal successor features for transfer reinforcement learning. *arXiv preprint arXiv:2001.04025*, 2020.

[18] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on robot learning*, pages 1101–1112. PMLR, 2020.

[19] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[20] Darshan Patil, Janarthanan Rajendran, Glen Berseth, and Sarath Chandar. Intelligent switching for reset-free rl. *arXiv preprint arXiv:2405.01684*, 2024.

[21] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[22] Archit Sharma, Michael Ahn, Sergey Levine, Vikash Kumar, Karol Hausman, and Shixiang Gu. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. *arXiv preprint arXiv:2004.12974*, 2020.

[23] Archit Sharma, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Autonomous reinforcement learning via subgoal curricula. *Advances in Neural Information Processing Systems*, 34:18474–18486, 2021.

[24] Archit Sharma, Rehaan Ahmad, and Chelsea Finn. A state-distribution matching approach to non-episodic reinforcement learning. In *International Conference on Machine Learning*, pages 19645–19657. PMLR, 2022.

[25] Archit Sharma, Kelvin Xu, Nikhil Sardana, Abhishek Gupta, Karol Hausman, Sergey Levine, and Chelsea Finn. Autonomous reinforcement learning: Formalism and benchmarking. In *International Conference on Learning Representations*, 2022.

[26] Annie Xie, Fahim Tajwar, Archit Sharma, and Chelsea Finn. When to ask for help: Proactive interventions in autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 35:16918–16930, 2022.

[27] Kelvin Xu, Siddharth Verma, Chelsea Finn, and Sergey Levine. Continual learning of control primitives: Skill discovery via reset-games. *Advances in Neural Information Processing Systems*, 33:4999–5010, 2020.

[28] Ali Yahya, Adrian Li, Mrinal Kalakrishnan, Yevgen Chebotar, and Sergey Levine. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 79–86. IEEE, 2017.

[29] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36 (4):1307–1319, 2020.

[30] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.

[31] Henry Zhu, Justin Yu, Abhishek Gupta, Dhruv Shah, Kristian Hartikainen, Avi Singh, Vikash Kumar, and Sergey Levine. The ingredients of real-world robotic reinforcement learning. In *International Conference on Learning Representations*, pages 10896–10916, 2020.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main contribution of this paper is a robust and scalable autonomous reinforcement learning (ARL) algorithm. We believe that this contribution and its scope are clearly reflected in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of our work are discussed in the conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides sufficient information to reproduce our experimental results, as detailed in the experiments and Appendices B and C.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The supplementary materials include all of our code as well as a README.md file with instructions for reproducing the main experimental results.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The paper provides all training and evaluation details, including hyperparameters and the type of optimizer, in Appendices B and C.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The quantitative results in the experiments report the means and standard deviations computed over five random seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources to reproduce our experimental results in Appendix C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and confirm that our work fully complies with it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both the potential positive and negative societal impacts in Appendix D.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We used Gymnasium, a set of benchmark tasks, to evaluate our work and properly cited it in the experiments and Appendix C.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: While no new assets are released with this submission, the code and its instructions in the supplementary materials will be made publicly available if the paper is accepted.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

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

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

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

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.