

---

# Effective Exploration Based on the Structural Information Principles

---

Xianghua Zeng<sup>1</sup>, Hao Peng<sup>1</sup>, Angsheng Li<sup>1,2</sup>

<sup>1</sup> State Key Laboratory of Software Development Environment, Beihang University, Beijing, China

<sup>2</sup> Zhongguancun Laboratory, Beijing, China

{zengxianghua, penghao, angsheng}@buaa.edu.cn, liangsheng@gmail.zgclab.edu.cn

## Abstract

Traditional information theory provides a valuable foundation for Reinforcement Learning (RL), particularly through representation learning and entropy maximization for agent exploration. However, existing methods primarily concentrate on modeling the uncertainty associated with RL’s random variables, neglecting the inherent structure within the state and action spaces. In this paper, we propose a novel Structural Information principles-based Effective Exploration framework, namely **SI2E**. Structural mutual information between two variables is defined to address the single-variable limitation in structural information, and an innovative embedding principle is presented to capture dynamics-relevant state-action representations. The SI2E analyzes value differences in the agent’s policy between state-action pairs and minimizes structural entropy to derive the hierarchical state-action structure, referred to as the encoding tree. Under this tree structure, value-conditional structural entropy is defined and maximized to design an intrinsic reward mechanism that avoids redundant transitions and promotes enhanced coverage in the state-action space. Theoretical connections are established between SI2E and classical information-theoretic methodologies, highlighting our framework’s rationality and advantage. Comprehensive evaluations in the MiniGrid, MetaWorld, and DeepMind Control Suite benchmarks demonstrate that SI2E significantly outperforms state-of-the-art exploration baselines regarding final performance and sample efficiency, with maximum improvements of 37.63% and 60.25%, respectively.

## 1 Introduction

Reinforcement Learning (RL) has emerged as a pivotal technique for addressing sequential decision-making problems, including game intelligence [Vinyals et al., 2019, Badia et al., 2020], robotic control [Andrychowicz et al., 2017, Liu and Abbeel, 2021], and autonomous driving [Prathiba et al., 2021, Pérez-Gil et al., 2022]. In the realm of RL, striking a balance between exploration and exploitation is crucial for optimizing agent policies and mitigating the risk of suboptimal outcomes, especially in scenarios characterized by high dimensions and sparse rewards [Zhang et al., 2021b].

Recently, advancements in information-theoretic approaches have shown promise for exploration in self-supervised settings. The maximum entropy framework over the action space [Haarnoja et al., 2017] has led to the development of robust algorithms such as Soft Q-learning [Nachum et al., 2017], SAC [Haarnoja et al., 2018], and MPO [Abdolmaleki et al., 2018]. Additionally, various objectives focused on maximizing state entropy are utilized to ensure comprehensive state coverage [Hazan et al., 2019, Islam et al., 2019]. To facilitate the exploration of complex state-action pairs, MaxRenyi optimizes Rényi entropy across the state-action space [Zhang et al., 2021a]. However, a prevalent issue with entropy maximization strategies is their tendency to bias exploration towards low-value states, making them vulnerable to imbalanced state-value distributions in supervised settings. To mitigate

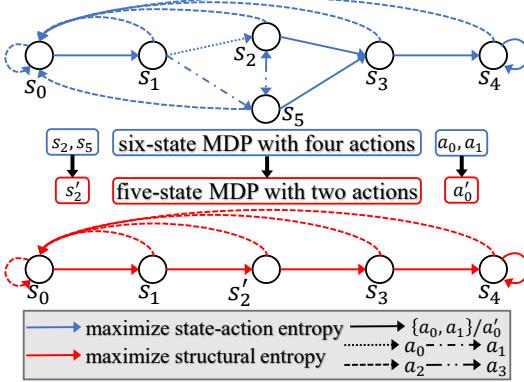


Figure 1: By incorporating the inherent state-action structure, we simplify the original six-state Markov Decision Process (MDP) with four actions to a five-state MDP with two actions, effectively reducing the size of state-action space from  $24(6 \times 4)$  to  $10(5 \times 2)$ . Here,  $s'_2$  and  $a'_0$  represent vertex communities  $\{s_2, s_5\}$  and  $\{a_0, a_1\}$ , respectively. In this scenario, a policy maximizing state-action Shannon entropy would encompass all possible transitions (blue color). In contrast, a policy incorporating the inherent state-action structure will divide these redundant state-action pairs into a vertex sub-community and minimize the entropy of this sub-community to avoid visiting it unnecessarily. Simultaneously, it maximizes state-action entropy, resulting in maximal coverage for transitions (red color) that are more likely to contribute to the desired outcome in the simplified five-state MDP.

this, value-conditional state entropy is introduced to compute intrinsic rewards based on the estimated values of visited states [Kim et al., 2023]. Due to their instability in noisy and high-dimensional environments, a Dynamic Bottleneck (DB) [Bai et al., 2021] is developed based on the Information Bottleneck (IB) principle [Tishby et al., 2000], thereby obtaining dynamics-relevant representations of state-action pairs. Despite their successes, existing information-theoretic exploration methods have a critical limitation: they often overlook the inherent structure within state and action spaces. This oversight necessitates new approaches to enhance exploration effectiveness.

Figure 1 illustrates a simple six-state Markov Decision Process (MDP) with four actions. The different densities of the blue and red lines represent different actions, as indicated in the legend, leading to state transitions aimed at optimizing the return to the initial state  $s_0$ . Solid lines specifically denote actions  $a_0$  and  $a_1$ . The transitions between states  $s_2$  and  $s_5$  are deemed redundant as they do not facilitate the primary objective of efficiently returning to  $s_0$ . Therefore, the state-action pairs  $(s_2, a_3)$  and  $(s_5, a_3)$  have lower policy values. A policy maximizing state-action Shannon entropy would encompass all possible transitions (blue color). In contrast, a policy incorporating the inherent state-action structure will divide these redundant state-action pairs into a vertex sub-community and minimize the entropy of this sub-community to avoid visiting it unnecessarily. Simultaneously, it maximizes state-action entropy, resulting in maximal coverage for transitions (red color) that are more likely to contribute to the desired outcome in the simplified five-state MDP.

Departing from traditional information theory applied to random variables, structural information [Li and Pan, 2016] has been devised to quantify dynamic uncertainty within complex graphs under a hierarchical partitioning structure known as an “encoding tree”. Structural entropy is conceptualized as the minimum number of bits required to encode a vertex accessible through a single-step random walk and is minimized to optimize the encoding tree. However, this definition is limited to single-variable graphs and cannot capture the structural relationship between two variables. While the underlying graph can encompass multiple variables, current structural information principles are limited in treating these variables as a single joint variable, measuring only its structural entropy. This limitation prevents the effective quantification of structural similarity between variables, inherently imposing a single-variable constraint. Prior research on reinforcement learning using structural information principles [Zeng et al., 2023b,c] has focused on independently modeling state or action variables without simultaneously considering state-action representations.

In this work, we propose SI2E, a novel and unified framework grounded in structural information principles for effective exploration within high-dimensional and sparse-reward environments. Initially, we embed state-action pairs into a low-dimensional space and present an innovative representation learning principle to capture dynamics-relevant information and compress dynamics-irrelevant information. Then, we strategically increase the state-action pairs’ structural mutual information with subsequent states while decreasing it with current states. We then analyze value differences among state-action representations to form a complete graph and minimize its structural entropy to derive the

optimal encoding tree, thereby unveiling the hierarchical community structure of state-action pairs. By leveraging this identified structure, we design an intrinsic reward mechanism tailored to avoid redundant transitions and enhance maximal coverage in exploring the state-action space. Furthermore, we establish theoretical connections between our framework and classical information-theoretic methodologies, highlighting the rationality and advantage of SI2E. Our thorough evaluations across diverse and challenging tasks in the MiniGrid, MetaWorld, and DeepMind Control Suite benchmarks have consistently shown SI2E’s superiority, with significant improvements in final performance and sample efficiency, surpassing state-of-the-art exploration baselines. For further research, the source code is available at <sup>1</sup>. Our contributions are summarized as follows:

- A novel framework based on structural information principles, SI2E, is proposed for effective exploration in high-dimensional RL environments with sparse rewards.
- An innovative principle of structural mutual information is introduced to overcome the single-variable constraint inherent in existing structural information and to enhance the acquisition of dynamics-relevant representations for state-action pairs.
- A unique intrinsic reward mechanism that maximizes the value-conditional structural entropy is designed to avoid redundant transitions and promote enhanced coverage in the state-action space.
- Our experiments on various challenging tasks demonstrate that SI2E significantly improves final performance and sample efficiency by up to 37.63% and 60.25%, respectively, compared to state-of-the-art baselines.

## 2 Preliminaries

In this section, we formalize the definitions of fundamental concepts. The descriptions of primary notations are summarized in Appendix A.1 for ease of reference.

### 2.1 Traditional Information Principles

Consider the random variable pair  $Z = (X, Y)$  with a joint distribution probability denoted by  $p(x, y) \in (0, 1)$ . The marginal probabilities,  $p(x)$  and  $p(y)$ , are defined as  $p(x) = \sum_y p(x, y)$  and  $p(y) = \sum_x p(x, y)$ , respectively. The joint Shannon entropy [Shannon, 1953] of  $X$  and  $Y$  is  $H(X, Y) = -\sum_{(x,y)} [p(x, y) \cdot \log p(x, y)]$ , which quantifies the total uncertainty in  $Z$ . Conversely, the marginal entropies  $H(X) = -\sum_x [p(x) \cdot \log p(x)]$  and  $H(Y) = -\sum_y [p(y) \cdot \log p(y)]$  characterize the uncertainty in  $X$  and  $Y$  individually. The mutual information  $I(X; Y) = \sum_{x,y} \left[ p(x, y) \cdot \log \frac{p(x, y)}{p(x)p(y)} \right]$  quantifies the shared uncertainty between  $X$  and  $Y$ . It satisfies the following relationship:  $I(X; Y) = H(X) + H(Y) - H(X, Y)$ .

### 2.2 Reinforcement Learning

Within the context of RL, the sequential decision-making problem is formalized as a Markov Decision Process (MDP) [Bellman, 1957]. The MDP is characterized by a tuple  $(\mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}^e, \gamma)$ , where  $\mathcal{O}$  denotes the observation space,  $\mathcal{A}$  the action space,  $\mathcal{P}$  the environmental transition function,  $\mathcal{R}^e$  the extrinsic reward function, and  $\gamma \in [0, 1)$  the discount factor. At each discrete timestep  $t$ , the agent selects an action  $a_t \in \mathcal{A}$  upon observing  $o_t \in \mathcal{O}$ . This leads to a transition to a new observation  $o_{t+1} \sim \mathcal{P}(o_t, a_t)$  and a reward  $r_t^e \in \mathbb{R}$ . The policy network  $\pi$  is optimized to maximize the cumulative long-term expected discounted reward.

**Maximum State Entropy Exploration.** In environments with sparse rewards, agents are encouraged to explore the state space extensively, which can be incentivized by maximizing the Shannon entropy  $H(S)$  of state variable  $S$ . When the prior distribution  $p(s)$  is not available, the non-parametric  $k$ -nearest neighbors ( $k$ -NN) entropy estimator [Singh et al., 2003] is employed. For a given set of  $n$  independent and identically distributed samples from a  $d_x$ -dimensional space  $\{x_i\}_{i=0}^{n-1}$ , the entropy of variable  $X$  is estimated as follows:

$$\hat{H}_{KL}(X) = \frac{d_x}{n} \sum_{i=0}^{n-1} \log d(x_i) + C, \quad (1)$$

---

<sup>1</sup><https://github.com/SELGroup/SI2E>

where  $d(x_i)$  is twice the distance from  $x_i$  to its  $k$ -th nearest neighbor, and  $C$  is a constant term.

**Information Bottleneck Principle.** In the supervised learning paradigm, representation learning aims to transform an input source  $X$  into a representation  $Z$ , targeted towards an output source  $Y$ . The Information Bottleneck (IB) principle [Tishby et al., 2000] refines this process by maximizing the mutual information  $I(Z; Y)$  between  $Z$  and  $Y$ , capturing the relevant features of  $Y$  within  $Z$ . Concurrently, the IB principle imposes a complexity constraint by minimizing the mutual information  $I(Z; X)$  between  $Z$  and  $X$ , effectively discarding irrelevant features. To balance these objectives, the IB principle utilizes a Lagrangian multiplier, facilitating a balanced trade-off between the richness of the representation and its complexity.

### 2.3 Structural Information Principles

The encoding tree  $T$  of an undirected and weighted graph  $G = (V, E)$  is characterized as a rooted tree with the following properties: 1) Each tree node  $\alpha$  in  $T$  corresponds to a subset of graph vertices  $V_\alpha \subseteq V$ . 2) The subset  $V_\lambda$  of tree root  $\lambda$  encompasses all vertices in  $V$ . 3) Each subset  $V_\nu$  of a leaf node  $\nu$  in  $T$  only contains a single vertex  $v$ , thus  $V_\nu = \{v\}$ . 4) For each non-leaf node  $\alpha$ , the number of its children is assumed as  $\bar{l}_\alpha$ , with the  $i$ -th child specified as  $\alpha_i$ . The collection of subsets  $V_{\alpha_1}, \dots, V_{\alpha_{\bar{l}_\alpha}}$  constitutes a sub-partition of  $V_\alpha$ .

Given an encoding tree  $T$  whose height is at most  $K$ , the  $K$ -dimensional structural entropy of graph  $G$  is defined as follows:

$$H^T(G) = - \sum_{\alpha \in T, \alpha \neq \lambda} \left[ \frac{g_\alpha}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha)}{\text{vol}(\alpha^-)} \right], \quad H^K(G) = \min_T H^T(G), \quad (2)$$

where  $g_\alpha$  is the weighted sum of all edges connecting vertices within the subset  $V_\alpha$  to vertices outside the subset  $V_\alpha$ .

## 3 Structural Mutual Information

In this section, we address the single-variable constraint prevalent in existing structural information principles and introduce the concept of structural mutual information for subsequent state-action representation learning within our SI2E framework.

Given the random variable pair  $(X, Y)$  with  $|X| = |Y| = n$ , we construct an undirected bipartite graph  $G_{xy}$  to represent the joint distribution of  $X$  and  $Y$ . In  $G_{xy}$ , each vertex  $x \in X$  connects to each vertex  $y \in Y$  via weighted edges, where the weight of each edge equals the joint probability  $p(x, y)$ . Notably, no edges connect vertices within the same set,  $X$  or  $Y$ , and the total sum of the edge weights is 1,  $\sum_{x,y} p(x, y) = 1$ . Each single-step random walk in  $G_{xy}$  accesses either a vertex from  $X$  or  $Y$ . The structural entropy of variable  $X$  in  $G_{xy}$  is defined as the number of bits required to encode all accessible vertices in the set  $X$ . It is calculated using the following formula:

$$H^{SI}(X) = - \sum_{x \in X} \left[ \frac{p(x)}{\text{vol}(G_{xy})} \cdot \log \frac{p(x)}{\text{vol}(G_{xy})} \right] = - \sum_{x \in X} \left[ \frac{p(x)}{2} \cdot \log \frac{p(x)}{2} \right], \quad (3)$$

where the sum of all vertex degrees is twice the total sum of edge weights, resulting in  $\text{vol}(G_{xy}) = 2$ .

The structural entropy  $H^{SI}(Y)$  is defined similarly. We restrict the partitioning structure of  $G_{xy}$  to 2-layer approximate binary trees, denoted as  $\mathcal{T}^2$ , to calculate the required bits to encode accessible vertices in  $X$  or  $Y$ , defined as the joint structural entropy. This tree structure mandates that each intermediate node (neither root nor leaf) has precisely two children. We begin by initializing a one-layer encoding tree,  $T_{xy}^0$ , designating each non-root node  $\alpha$ 's parent as the root  $\lambda$ , with  $\alpha^- = \lambda$ . By applying the stretch operator from the HCSSE algorithm [Pan et al., 2021], we pursue an iterative and greedy optimization of  $T_{xy}^0$ , further detailed in Appendix A.2. The optimal encoding tree,  $T_{xy}^*$ , for  $G_{xy}$  and the joint entropy under  $T_{xy}^*$  are achieved through:

$$T_{xy}^* = \arg \min_{T \in \mathcal{T}^2} H^T(G_{xy}), \quad H^{T_{xy}^*}(X, Y) = H^{T_{xy}^*}(G_{xy}). \quad (4)$$

Utilizing 2-layer approximate binary trees as the structural framework ensures computational tractability and more complex structures will increase the cost of increased computational complexity, which can be prohibitive for practical applications.

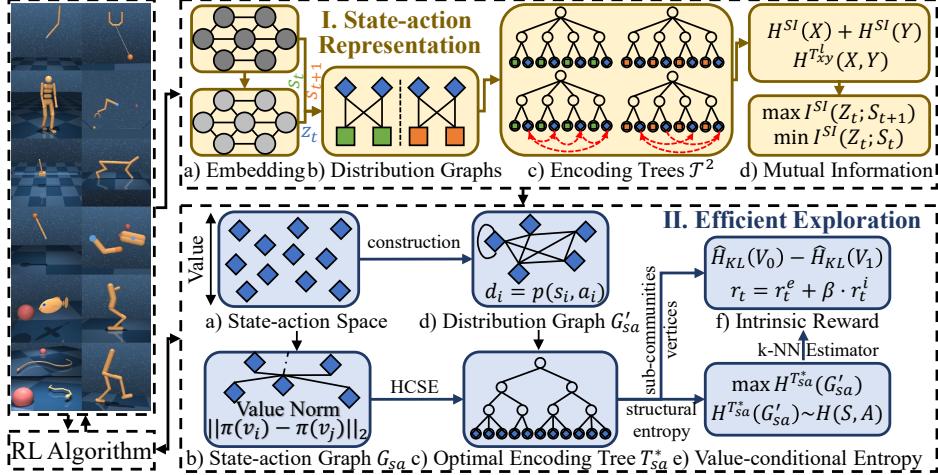


Figure 2: The SI2E’s overview architecture, including state-action representation learning and maximum structural entropy exploration.

We derive the following proposition regarding  $\mathcal{T}^2$ , with the detailed proof provided in Appendix B.1.

**Proposition 3.1.** Consider an undirected graph  $G = (V, E)$  with vertices  $v_i$  and  $v_j$  in  $V$ . If the edge  $(v_i, v_j)$  is absent from  $E$ , then in the 2-layer approximate binary optimal encoding tree  $T^* \in \mathcal{T}^2$ , there does not exist any non-root node  $\alpha$  such that both  $v_i$  and  $v_j$  are included in its subset  $V_\alpha$ .

Each intermediate node  $\alpha \in T^*_{xy}$  corresponds to a subset  $T^*_\alpha$  comprising exactly one  $x$  vertex and one  $y$  vertex, thus establishing a one-to-one matching structure between variables  $X$  and  $Y$ . The  $i$ -th intermediate node in  $T^*_{xy}$ , ordered from left to right, is denoted as  $\alpha_i$ . Within this subset  $T^*_{\alpha_i}$ , the  $x$  and  $y$  vertices are labeled as  $x_i$  and  $y_i$ , respectively.

To define structural mutual information accurately, it is essential to consider the joint entropy of two variables under various partition structures. We introduce an  $l$ -transformation applied to  $T^*_{xy}$  to systematically traverse all potential one-to-one matching of these variables, providing a comprehensive measure of their structural similarity. Given an integer parameter  $l > 0$ , this transformation generates a new 2-layer approximate binary tree,  $T^l_{xy}$ , representing an alternative one-to-one matching structure.

**Definition 3.2.** For each intermediate node  $\alpha_i$  in  $T^l_{xy}$ , the  $x$  and  $y$  vertices in  $T^l_{\alpha_i}$  are specified as  $T^l_{\alpha_i} = \{x_{i'}, y_i\}$ , where  $i' = (i + l) \bmod n$ .

The resulting tree  $T^l_{xy}$  is equivalent to the optimal tree  $T^*_{xy}$  when  $l = 0$ . We provide an example in Appendix A.3 with  $n = 4$  for an intuitive understanding of the described process.

**Definition 3.3.** Leveraging the relationship  $I(X; Y) = H(X) + H(Y) - H(X, Y)$  in traditional information theory, we formally define the structural mutual information,  $I^{SI}(X; Y)$ , as follows:

$$I^{SI}(X; Y) = \sum_{l=0}^{n-1} \left[ H^{SI}(X) + H^{SI}(Y) - H^{T^l_{xy}}(X, Y) \right] = \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{2}{p(x_i) + p(y_j)} \right]. \quad (5)$$

The detailed derivation is provided in Appendix C.1. Structural mutual information quantifies the average difference between the required encoding bits of accessible vertices of a single variable and the joint variable. The following theorem outlines the connection between  $I^{SI}(X; Y)$  and  $I(X; Y)$ , with a detailed proof in Appendix B.2.

**Theorem 3.4.** For a tuning parameter  $0 \leq \epsilon \leq 1$ , its holds for the structural mutual information  $I^{SI}(X; Y)$ , traditional mutual information  $I(X; Y)$ , and joint Shannon entropy  $H(X, Y)$  that:

$$I(X; Y) \leq I^{SI}(X; Y) \leq I(X; Y) + (1 - \epsilon) \cdot H(X, Y). \quad (6)$$

## 4 The Proposed SI2E Framework

In this section, we describe the detailed designs of the proposed SI2E framework, which captures dynamic-relevant state-action representations through structural mutual information (see Section

4.1) and enhances state-action coverage conditioned by the agent’s policy by maximizing structural entropy (see Section 4.2). The overall architecture of our framework is illustrated in Figure 2.

#### 4.1 State-action Representation Learning

To effectively learn dynamics-relevant state-action representations, we present an innovative embedding principle that maximizes the structural mutual information with subsequent states and minimizes it with current states.

**Structural Mutual Information Principle.** In this phase, the input variables at timestep  $t$  encompass the current observation  $O_t$  and the action  $A_t$ , with the target being the subsequent observation  $O_{t+1}$ . We denote the encoding of observations  $O_t$  and  $O_{t+1}$  as states  $S_t$  and  $S_{t+1}$ , respectively. We aim to generate a latent representation  $Z_t$  for the tuple  $(S_t, A_t)$ , which preserves information relevant to  $S_{t+1}$  while compressing information pertinent to  $S_t$ . This embedding process mentioned above is detailed as follows:

$$S_t = f_s(O_t), \quad S_{t+1} = f_s(O_{t+1}), \quad Z_t = f_z(S_t, A_t), \quad (7)$$

where  $f_s$  and  $f_z$  are the respective encoders for states and state-action pairs (step I. a in Figure 2). For the state-action embeddings  $Z_t$ , we construct two undirected bipartite graphs,  $G_{zs}$  and  $G_{zs'}$ , as shown in step I. b of Figure 2. These graphs represent the joint distributions of  $Z_t$  with the current states  $S_t$  and subsequent states  $S_{t+1}$ . In step I. c of Figure 2, we generate 2-layer approximate binary trees for  $G_{zs}$  and  $G_{zs'}$  and calculate the mutual information  $I^{SI}(Z_t; S_t)$  and  $I^{SI}(Z_t; S_{t+1})$  using Equation 5. Building upon the Information Bottleneck (IB) [Tishby et al., 2000], we present an embedding principle that aims to minimize  $I^{SI}(Z_t; S_t)$  while maximizing  $I^{SI}(Z_t; S_{t+1})$ , as demonstrated in step I. d of Figure 2. When the joint distribution between variables  $Z_t$  and  $S_{t+1}$  shows a one-to-one correspondence-meaning for each  $z_t \in Z_t$  value, there is a unique  $s_{t+1} \in S_{t+1}$  corresponding to it, and vice versa-their mutual information takes its maximum value. We introduce a theorem to elucidate the equivalence between  $I^{SI}(Z_t; S_{t+1})$  and  $I(Z_t; S_{t+1})$  under this condition.

**Theorem 4.1.** *For a joint distribution of variables  $X$  and  $Y$  that shows a one-to-one correspondence,  $I^{SI}(X; Y)$  equals  $I(X; Y)$ .*

A detailed proof is provided in Appendix B.3. When  $Z_t$  and  $S_t$  are mutually independent, the mutual information  $I(Z_t; S_t)$  attains its minimum value. Our  $I^{SI}(Z_t; S_t)$  goes beyond this, incorporating the joint entropy  $H(Z_t, S_t)$  according to Theorem 3.4. This integration effectively eliminates the irrelevant information embedded in the representation variable  $Z_t$ , a significant step in our research. Consequently, structural mutual information can be considered a reasonable and desirable learning objective for acquiring dynamics-relevant state-action representations.

**Representation Learning Objective.** Due to the computational challenges of directly minimizing  $I^{SI}(Z_t; S_t)$ , we formulate a variational upper bound  $I(Z_t; S_t) + H(Z_t|S_t) + H(S_t)$  (see Appendix C.2). Noting that the term  $H(S_t)$  is extraneous to our model, we equate the minimization of  $I^{SI}(Z_t; S_t)$  to the minimization of  $I(Z_t; S_t)$  and  $H(Z_t|S_t)$ .

By employing a feasible decoder to approximate the marginal distribution of  $Z_t$ , we derive an upper bound of  $I(Z_t; S_t)$  (See Appendix C.3) as follows:

$$I(Z_t; S_t) \leq \sum [p(z_t, s_t) \cdot D_{KL}(p(z_t|s_t) || q_m(z_t))] \triangleq L_{up}. \quad (8)$$

To concurrently decrease the conditional entropy  $H(Z_t|S_t)$ , we introduce a predictive objective (See Appendix C.4) through a tractable decoder  $q_{z|s}$  for the conditional probability  $p(z_t|s_t)$  as follows:

$$H(Z_t|S_t) \leq \sum \left[ p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)} \right] \triangleq L_{z|s}, \quad (9)$$

where  $L_{z|s}$  represents the log-likelihood of  $Z_t$  given  $S_t$ .

To efficiently optimize  $I^{SI}(Z_t; S_{t+1})$ , we maximize its lower bound,  $I(Z_t; S_{t+1})$ , as detailed in Theorem 3.4. By utilizing an alternative decoder  $q_{s|z}$  for the conditional probability  $p(s_{t+1}|z_t)$ , we obtain a lower bound of  $I(Z_t; S_{t+1})$  (See Appendix C.5) as follows:

$$I(Z_t; S_{t+1}) \geq \sum [p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t)] \triangleq L_{s|z}, \quad (10)$$

where  $L_{s|z}$  denotes the log-likelihood of  $S_{t+1}$  conditioned on  $Z_t$ .

Within our SI2E framework, the definitive loss for representation learning is a combination of the above bounds,  $L = L_{up} + L_{z|s} + \eta \cdot L_{s|z}$ , where  $\eta$  is a Lagrange multiplier used to maintain equilibrium among the specified terms.

## 4.2 Maximum Structural Entropy Exploration

We have designed a unique intrinsic reward mechanism to address the challenge of imbalance exploration towards low-value states in traditional entropy strategies, as discussed by [Kim et al., 2023]. Specifically, we generate a hierarchical state-action structure based on the agent’s policy and define value-conditional structural entropy as an intrinsic reward for effective exploration.

**Hierarchical State-action Structure.** Derived from the history of agent-environment interactions, we extract state-action pairs (step II. a in Figure 2) to form a complete graph  $G_{sa}$  (step II. b in Figure 2) that encapsulates the value relationships caused by the agent’s policy. Within this graph, any two vertices  $v_i$  and  $v_j$  is connected by an undirected edge whose weight  $w_{ij}$  is determined as:  $w_{ij} = \|\pi(s_t^i, a_t^i) - \pi(s_t^j, a_t^j)\|_2$ . The state-action pairs  $(s_t^i, a_t^i)$  and  $(s_t^j, a_t^j)$  are associated with vertices  $v_i$  and  $v_j$ , respectively. We minimize the 2-dimensional structural entropy of this graph  $G_{sa}$  to generate its 2-layer optimal encoding tree, denoted as  $T_{sa}^*$  (step II. c in Figure 2). This tree  $T_{sa}^*$  delineates a hierarchical community structure among the state-action vertices, with the root node corresponding to a community encompassing all vertices. Each intermediate node in  $T_{sa}^*$  corresponds to a sub-community, including vertices that share similar  $\pi$  values.

**Value-conditional Structural Entropy.** To measure the extent of the policy’s coverage across the state-action space, we construct an additional distribution graph  $G'_{sa}$  (step II. d in Figure 2). The graph  $G'_{sa}$  shares the same vertex set as  $G_{sa}$ . The following proposition confirms the existence of such a graph, with a detailed proof provided in Appendix B.4.

**Proposition 4.2.** *Given positive visitation probabilities  $p(s_t^0, a_t^0), \dots, p(s_t^{n-1}, a_t^{n-1})$  for all state-action pairs, there exists a weighted, undirected, and connected graph  $G'_{sa}$ , where each vertex’s degree  $d_i$  equals its visitation probability  $p(s_t^i, a_t^i)$ .*

In the graph  $G'_{sa}$ , the set of all state-action vertices is denoted as  $V_0$ , and the set of all state-action sub-communities is denoted as  $V_1$ . The Shannon entropies associated with the distribution of visitation probabilities for these sets are represented as  $H(V_0)$  and  $H(V_1)$ , respectively, where  $H(V_0) = H(S_t, A_t)$ . Within the 2-layer state-action community represented by  $T_{sa}^*$ , we define the structural entropy of  $G'_{sa}$  using Equation 2, denoted as  $H^{T_{sa}^*}(G'_{sa})$  (step II. e in Figure 2). The following theorem delineates the relationship between the value-conditional entropy  $H^{T_{sa}^*}(G'_{sa})$  with the state-action Shannon entropy  $H(S_t, A_t)$ . A detailed proof is provided in Appendix B.5.

**Theorem 4.3.** *For a tuning parameter  $0 \leq \zeta \leq 1$ , it holds for the structural entropy  $H^{T_{sa}^*}(G'_{sa})$  and the Shannon entropy  $H(S_t, A_t)$  that:*

$$\zeta \cdot H(S_t, A_t) \leq H(V_0) - H(V_1) \leq H^{T_{sa}^*}(G'_{sa}) \leq H(S_t, A_t), \quad (11)$$

where  $H(V_0) - H(V_1)$  is a variational lower bound of  $H^{T_{sa}^*}(G'_{sa})$ . On the one hand, the term  $H(V_0)$  ensures maximal coverage of the entire state-action space, analogous to the traditional Shannon entropy. On the other hand, the term  $H(V_1)$  mitigates uniform coverage among state-action sub-communities with diverse  $\pi$  values, thus addressing the challenge of imbalance exploration. By identifying the hierarchical state-action structure caused by the agent’s policy, the SI2E achieves enhanced maximum coverage exploration, thereby guaranteeing its exploration advantage.

**Estimation and Intrinsic Reward.** Considering the impracticality of directly acquiring visitation probabilities, we employ the  $k$ -NN entropy estimator in Equation 1 to estimate the lower bound:

$$H(V_0) - H(V_1) \approx \frac{d_z}{n_0} \cdot \sum_{i=0}^{n_0-1} \log d(v_i^0) - \frac{d_z}{n_1} \cdot \sum_{i=0}^{n_1-1} \log d(v_i^1) + C, \quad v_i^0 \in V_0, v_i^1 \in V_1, \quad (12)$$

where  $d_z$  is the dimension of state-action embedding,  $n_0$  and  $n_1$  are the vertex numbers in  $V_0$  and  $V_1$ , and  $d(v)$  is twice the distance from vertex  $v$  to its  $k$ -th nearest neighbor. By ignoring the constant term in Equation 12, we define the intrinsic reward  $r_t^i$  and train RL agents to address the target task using a combined reward  $r_t = r_t^e + \beta \cdot r_t^i$  (step II. f in Figure 2), where  $\beta$  is a positive hyperparameter that modulates the trade-off between exploration and exploitation. The pseudocode, complexity analysis, and limitations of our framework are provided in Appendix A.

Table 1: Summary of success rates and required steps to achieve target rewards in MiniGrid and MetaWorld tasks: “average value  $\pm$  standard deviation” and “average improvement”. **Bold**: the best performance, underline: the second performance.

MiniGrid Navigation	RedBlueDoors-6x6		SimpleCrossingS9N1		KeyCorridorS3R1	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
A2C	-	-	88.18 $\pm$ 3.46	570.08 $\pm$ 15.87	86.57 $\pm$ 2.26	658.74 $\pm$ 21.03
A2C+SE	-	-	88.59 $\pm$ 4.62	394.39 $\pm$ 66.14	87.20 $\pm$ 4.94	463.86 $\pm$ 38.27
A2C+VCSE	79.82 $\pm$ 7.26	1161.90 $\pm$ 241.59	91.30 $\pm$ 1.92	204.02 $\pm$ 25.60	86.01 $\pm$ 0.91	190.20 $\pm$ 6.11
A2C+SI2E	<b>85.80</b> $\pm$ 1.48	<b>461.90</b> $\pm$ 61.53	<b>93.64</b> $\pm$ 1.63	<b>139.17</b> $\pm$ 27.03	<b>94.20</b> $\pm$ 0.42	<b>129.06</b> $\pm$ 6.11
Abs.(%) Avg.	5.98(7.49) $\uparrow$	700.0(60.25) $\downarrow$	2.34(2.56) $\uparrow$	64.85(31.79) $\downarrow$	7.00(8.03) $\uparrow$	61.14(32.15) $\downarrow$
MiniGrid Navigation	DoorKey-6x6		DoorKey-8x8		Unlock	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
A2C	92.67 $\pm$ 8.47	567.20 $\pm$ 96.57	-	-	92.48 $\pm$ 11.96	669.78 $\pm$ 154.74
A2C+SE	93.18 $\pm$ 6.81	476.34 $\pm$ 94.63	72.60 $\pm$ 20.32	1515.81 $\pm$ 324.28	91.34 $\pm$ 18.37	634.37 $\pm$ 240.51
A2C+VCSE	94.08 $\pm$ 2.58	336.75 $\pm$ 19.84	94.32 $\pm$ 11.09	1900.96 $\pm$ 398.65	93.12 $\pm$ 3.43	405.22 $\pm$ 52.22
A2C+SI2E	<b>97.04</b> $\pm$ 1.52	<b>230.60</b> $\pm$ 19.85	<b>98.58</b> $\pm$ 3.11	<b>1090.96</b> $\pm$ 125.77	<b>97.13</b> $\pm$ 3.35	<b>309.14</b> $\pm$ 53.71
Abs.(%) Avg.	2.96(3.15) $\uparrow$	106.15(31.52) $\downarrow$	4.26(4.52) $\uparrow$	424.85(28.03) $\downarrow$	4.01(4.31) $\uparrow$	96.08(23.71) $\downarrow$
MetaWorld Manipulation	Button Press		Door Open		Drawer Open	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
DrQv2	94.55 $\pm$ 4.64	105.0 $\pm$ 5.0	-	-	-	-
DrQv2+SE	93.05 $\pm$ 7.67	95.0 $\pm$ 5.0	-	-	25.31 $\pm$ 7.40	-
DrQv2+VCSE	89.80 $\pm$ 3.29	77.5 $\pm$ 2.5	80.90 $\pm$ 10.19	-	82.74 $\pm$ 7.46	175.0 $\pm$ 5.0
DrQv2+SI2E	<b>99.60</b> $\pm$ 0.57	<b>62.5</b> $\pm$ 7.5	<b>95.77</b> $\pm$ 1.05	<b>87.5</b> $\pm$ 2.5	<b>95.96</b> $\pm$ 3.00	<b>82.5</b> $\pm$ 2.5
Abs.(%) Avg.	5.05(5.34) $\uparrow$	15.0(19.35) $\downarrow$	14.87(18.38) $\uparrow$	-	13.22(15.98) $\uparrow$	92.5(52.86) $\downarrow$
MetaWorld Manipulation	Faucet Close		Faucet Open		Window Open	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
DrQv2	53.33 $\pm$ 1.92	-	-	-	88.18 $\pm$ 1.50	192.5 $\pm$ 2.5
DrQv2+SE	92.36 $\pm$ 3.66	71.25 $\pm$ 6.25	-	-	93.14 $\pm$ 2.03	172.5 $\pm$ 2.5
DrQv2+VCSE	94.21 $\pm$ 1.74	60.0 $\pm$ 5.0	87.23 $\pm$ 5.29	67.5 $\pm$ 5.0	93.17 $\pm$ 1.45	127.5 $\pm$ 7.5
DrQv2+SI2E	<b>99.37</b> $\pm$ 1.18	<b>27.5</b> $\pm$ 2.5	<b>97.06</b> $\pm$ 1.39	<b>51.25</b> $\pm$ 3.75	<b>99.46</b> $\pm$ 0.35	<b>77.5</b> $\pm$ 2.5
Abs.(%) Avg.	5.16(5.48) $\uparrow$	32.5(54.17) $\downarrow$	9.83(11.27) $\uparrow$	16.25(24.07) $\downarrow$	6.29(6.75) $\uparrow$	50.0(39.22) $\downarrow$

## 5 Experiments

In this section, we present a comprehensive suite of comparative experiments on MiniGrid [Chevalier-Boisvert et al., 2018], MetaWorld [Yu et al., 2020], and the DeepMind Control Suite (DMControl) [Tunyasuvunakool et al., 2020] to evaluate the effectiveness of SI2E in terms of both final performance and sample efficiency. Consistent with previous work [Zeng et al., 2023c], we measure the required steps to attain specified rewards (0.9 times SI2E’s convergence reward) as a benchmark for assessing sample efficiency. For the SI2E implementation, we employ a randomly initialized encoder optimized to minimize the combined loss  $L$ . All experiments are conducted with 10 different random seeds, and the learning curves are delineated in Appendix E.

### 5.1 MiniGrid Evaluation

Initially, we assess our framework on navigation tasks using the MiniGrid benchmark, which includes goal-reaching tasks in sparse-reward environments. This setting is partially observable: the agent receives a  $7 \times 7 \times 3$  embedding of the immediate surrounding grid rather than the entire grid. For comparative purposes, we employ the A2C agent [Mnih et al., 2016] with Shannon entropy (SE) [Seo et al., 2021] and value-based state entropy (VCSE) [Kim et al., 2023] as our baselines. Table 1 (upper) displays the average values and standard deviations of success rates and required steps for various navigation tasks. The tasks encompass navigation with obstacles (SimpleCrossingS9N1), long-horizon navigation (RedBlueDoors, DoorKey, and Unlock), and long-horizon navigation with obstacles (KeyCorridorS3R1). The SI2E consistently exhibits enhanced final performance and sample efficiency across tasks, with an average success rate increase of 4.92%, from 89.97% to 94.40%, and an average decrease in required steps of 38.10%, from 635.65K to 393.47K. In the RedBlueDoors task, where baseline performances are inadequate, our SI2E significantly improves the success rate from 79.82% to 85.80% and reduces the required steps from 1161.90K to 461.90K.

### 5.2 MetaWorld Evaluation

We further evaluate the SI2E framework on visual manipulation tasks from the MetaWorld benchmark, which presents exploration challenges due to its large state space. We select the model-free DrQv2 algorithm as the underlying RL methodology. Adhering to the setup of [Seo et al., 2023], we employ the same camera configuration and normalize the reward with a scale of 1. We summarize the success rates and required steps for all exploration methods across six MetaWorld tasks in Table 1 (lower).

Table 2: Summary of average episode rewards for control tasks in DMControl, encompassing two cartpole tasks characterized by sparse rewards: “average value  $\pm$  standard deviation” and “average improvement” (absolute value(%)). **Bold**: the best performance, **underline**: the second performance.

Domain, Task	Hopper Stand	Cheetah Run	Quadruped Walk	Pendulum Swingup	Cartpole Balance	Cartpole Swingup
DrQv2	$87.59 \pm 11.70$	$229.28 \pm 123.93$	$289.79 \pm 24.17$	$424.21 \pm 246.96$	$998.97 \pm 22.95$	—
DrQv2+SE	$313.39 \pm 94.15$	$228.82 \pm 126.21$	$290.27 \pm 24.20$	$10.80 \pm 2.92$	$993.80 \pm 75.24$	$219.69 \pm 62.21$
DrQv2+VCSE	$711.32 \pm 30.84$	<u><math>456.26 \pm 22.20</math></u>	$243.74 \pm 29.91$	<u><math>824.17 \pm 99.59</math></u>	<u><math>998.65 \pm 9.58</math></u>	<u><math>707.76 \pm 50.38</math></u>
DrQv2+MADE	$717.09 \pm 112.94$	$366.59 \pm 53.74$	$262.63 \pm 23.92$	$672.11 \pm 34.63$	$996.16 \pm 40.60$	$704.18 \pm 41.75$
DrQv2+SI2E (Ours)	<b><math>797.17 \pm 53.21</math></b>	<b><math>464.08 \pm 29.32</math></b>	<b><math>399.51 \pm 29.05</math></b>	<b><math>885.50 \pm 38.28</math></b>	<b><math>999.58 \pm 2.97</math></b>	<b><math>795.09 \pm 90.49</math></b>
Abs.(%) Avg. $\uparrow$	80.08(11.17)	7.82(1.71)	109.24(37.63)	61.33(7.44)	0.93(0.09)	87.33(12.34)

Our SI2E framework enables the DrQv2 agent to solve all tasks with an average success rate of 97.87 after an average of 64.79K environmental steps, significantly outperforming other baselines. Specifically, in the Door Open task, all baselines struggle to achieve a meaningful success rate with a satisfactory number of environmental steps. This result demonstrates the SI2E’s effectiveness in improving and accelerating agent exploration in challenging tasks with expansive state-action spaces.

### 5.3 DMControl Evaluation

Subsequently, we evaluate our framework across various continuous control tasks within the DMControl suite. As the foundational agent, we choose the same DrQv2 algorithm, which operates on pixel-based observations. We incorporate a state-action exploration baseline, MADE [Zhang et al., 2021b], for a more comprehensive comparison. We evaluate all exploration methods across six continuous control tasks, documenting the episode rewards in Table 2. Observations reveal that SI2E remarkably increases the mean episode reward in each DMControl task. Specifically, in the Cartpole Swingup task characterized by sparse rewards, our framework boosts the average reward from 707.76 to 795.09, resulting in a 12.34% improvement in the final performance. Moreover, we compare the sample efficiency of SI2E and the best-performing baseline in Appendix E.3.

These results not only demonstrate the effectiveness of SI2E in acquiring dynamics-relevant representations for state-action pairs but also highlight its potential to motivate agents to explore the state-action space. To better understand the rationality and advantage of the SI2E framework, we provide visualization experiments in Appendix E.4.

### 5.4 Ablation Studies

To further investigate the impact of two critical components within the SI2E framework, embedding principle (Section 4.1) and intrinsic reward mechanism (4.2), we perform ablation studies on MetaWorld and DMControl tasks, focusing on two distinct variants: (i) SI2E-DB, which utilizes the DB bottleneck [Bai et al., 2021] for learning state-action representations, and (ii) SI2E-VCSE, employing the state-of-the-art VCSE approach [Kim et al., 2023] for calculating intrinsic rewards. As depicted in Figure 3, SI2E surpasses all variants regarding final performance and sample efficiency. This outcome underscores the essential role of these critical components in conferring SI2E’s superior capabilities. Additional ablation studies for the parameters  $\beta$  and  $n$  are available in Appendix E.5.

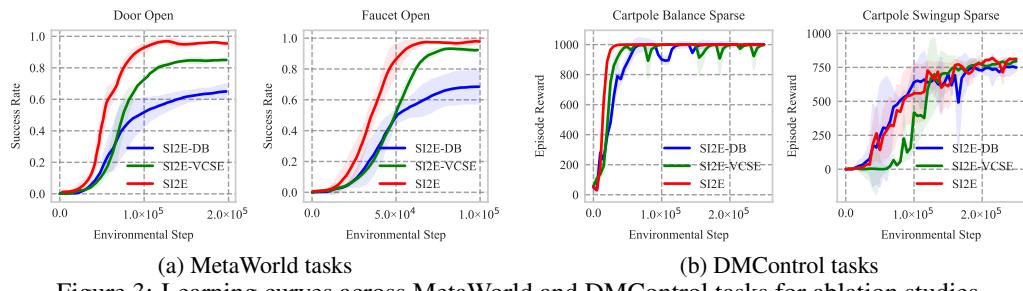


Figure 3: Learning curves across MetaWorld and DMControl tasks for ablation studies.

## 6 Related Work

### 6.1 Maximum Entropy Exploration

Maximum entropy exploration has evolved from focusing initially on unsupervised methods to incorporating task rewards in more advanced supervised models. In the unsupervised paradigm,

agents autonomously acquire behaviors by using state entropy as an intrinsic reward for exploration [Liu and Abbeel, 2021, Mutti et al., 2022, Yang and Spaan, 2023]. In contrast, in the supervised paradigm, agents aim to maximize state entropy in conjunction with task rewards [Seo et al., 2021, Yuan et al., 2022].

However, these methods face challenges due to imbalances in the distributions of states with differing policy values. To address this issue, a value-based approach [Kim et al., 2023] has been proposed, which integrates value estimates into the entropy calculation to ensure balanced exploration. Nevertheless, the effectiveness of this approach heavily depends on the partitioning structure of states according to policy values, requiring prior knowledge about downstream tasks.

In this work, we leverage structural information principles to derive the hierarchical state-action structure in an unsupervised manner. We further define the value-conditional structural entropy as an intrinsic reward to achieve more effective agent exploration. Compared to current maximum entropy explorations, SI2E introduces an additional sub-community entropy and minimizes this entropy to motivate the agent to explore specific sub-communities with high policy values. This approach helps avoid redundant explorations within low-value sub-communities and achieves enhanced maximum coverage exploration. Our method allows for balanced exploration without requiring prior knowledge of downstream tasks, effectively addressing the limitations of previous approaches.

## 6.2 Representation Learning

Novelty Search [Tao et al., 2020] and Curiosity Bottleneck [Kim et al., 2019b] leverage the Information Bottleneck principle for effective representation learning. Additionally, the EMI method [Kim et al., 2019a] maximizes mutual information in both forward and inverse dynamics to develop desirable representations. However, these methods are limited by the lack of an explicit mechanism to address the white noise issue in the state space. To overcome this challenge, the Dynamic Bottleneck model [Bai et al., 2021] is introduced for robust exploration in complex environments.

Our work defines structural mutual information to measure the structural similarity between two variables for the first time. Additionally, we present an innovative embedding principle that incorporates the entropy of the representation variable. This approach more effectively eliminates irrelevant information than the traditional information bottleneck principle.

## 6.3 Structural Information Principles

Since the introduction of structural information principles [Li and Pan, 2016], these principles have significantly transformed the analysis of network complexities, employing metrics such as structural entropy and partitioning trees. This innovative approach has not only deepened the understanding of network dynamics—even in the context of multi-relational graphs [Cao et al.]—but has also led to a wide array of applications across different domains. The application of structural information principles has extended to various fields, including graph learning [Wu et al., 2022], skin segmentation [Zeng et al., 2023a], and the analysis of social networks [Peng et al., Zeng et al., 2024, Cao et al., 2024]. In the domain of reinforcement learning, these principles have been instrumental in defining hierarchical action and state abstractions through encoding trees [Zeng et al., 2023b,c], marking a significant advancement in robust decision-making frameworks.

## 7 Conclusion

We propose SI2E, a novel exploration framework based on structural information principles. This framework defines structural mutual information to effectively capture state-action representations relevant to environmental dynamics. It maximizes the value-conditional structural entropy to enhance coverage across the state-action space. We have established theoretical connections between SI2E and traditional information-theoretic methodologies, underscoring the framework’s rationality and advantages. Through extensive and comparative evaluations, SI2E significantly improves final performance and sample efficiency over state-of-the-art exploration methods. Our future work includes expanding the height of encoding trees and the range of experimental environments. Our goal is for SI2E to remain a robust and adaptable tool in reinforcement learning, particularly suited to high-dimensional and sparse-reward contexts.

## Acknowledgments

The corresponding authors are Hao Peng and Angsheng Li. This work is supported by the National Key R&D Program of China through grant 2021YFB1714800, NSFC through grants 61932002, 62322202, and 62432006, Beijing Natural Science Foundation through grant 4222030, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 246Z0102G, Hebei Natural Science Foundation through grant F2024210008, and Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020.

## References

- Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.
- 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.
- Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International conference on machine learning*, pages 507–517. PMLR, 2020.
- Chenjia Bai, Lingxiao Wang, Lei Han, Animesh Garg, Jianye Hao, Peng Liu, and Zhaoran Wang. Dynamic bottleneck for robust self-supervised exploration. *Advances in Neural Information Processing Systems*, 34:17007–17020, 2021.
- Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.
- Yuwei Cao, Hao Peng, Angsheng Li, Chenyu You, Zhifeng Hao, and S Yu Philip. Multi-relational structural entropy. In *The 40th Conference on Uncertainty in Artificial Intelligence*.
- Yuwei Cao, Hao Peng, Zhengtao Yu, and S Yu Philip. Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8255–8264, 2024.
- Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. 2018.
- Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.
- Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.
- Riashat Islam, Raihan Seraj, Pierre-Luc Bacon, and Doina Precup. Entropy regularization with discounted future state distribution in policy gradient methods. *ArXiv*, 2019.
- Daejin Jo, Sungwoong Kim, Daniel Nam, Taehwan Kwon, Seungeun Rho, Jongmin Kim, and Donghoon Lee. Leco: Learnable episodic count for task-specific intrinsic reward. *Advances in Neural Information Processing Systems*, 35:30432–30445, 2022.
- Dongyoung Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Accelerating reinforcement learning with value-conditional state entropy exploration. *arXiv preprint arXiv:2305.19476*, 2023.
- Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. In *International Conference on Machine Learning*, pages 3360–3369. PMLR, 2019a.

- Youngjin Kim, Wontae Nam, Hyunwoo Kim, Ji-Hoon Kim, and Gunhee Kim. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International conference on machine learning*, pages 3379–3388. PMLR, 2019b.
- Angsheng Li and Yicheng Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62:3290–3339, 2016.
- Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.
- Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.
- Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, pages 16223–16239. PMLR, 2022.
- Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.
- Yicheng Pan, Feng Zheng, and Bingchen Fan. An information-theoretic perspective of hierarchical clustering. *arXiv preprint arXiv:2108.06036*, 2021.
- 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.
- Hao Peng, Jingyun Zhang, Xiang Huang, Zhifeng Hao, Angsheng Li, Zhengtao Yu, and Philip S Yu. Unsupervised social bot detection via structural information theory. *ACM Transactions on Information Systems*, pages 1–42.
- Óscar Pérez-Gil, Rafael Bareja, Elena López-Guillén, Luis M Bergasa, Carlos Gomez-Huelamo, Rodrigo Gutiérrez, and Alejandro Diaz-Díaz. Deep reinforcement learning based control for autonomous vehicles in carla. *Multimedia Tools and Applications*, 81(3):3553–3576, 2022.
- Sahaya Beni Prathiba, Gunasekaran Raja, Kapal Dev, Neeraj Kumar, and Mohsen Guizani. A hybrid deep reinforcement learning for autonomous vehicles smart-platooning. *IEEE Transactions on Vehicular Technology*, 70(12):13340–13350, 2021.
- Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pages 9443–9454. PMLR, 2021.
- Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.
- Claude Shannon. The lattice theory of information. *Transactions of the IRE professional Group on Information Theory*, 1(1):105–107, 1953.
- Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.
- Ruo Yu Tao, Vincent François-Lavet, and Joelle Pineau. Novelty search in representational space for sample efficient exploration. *Advances in Neural Information Processing Systems*, 33:8114–8126, 2020.
- Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

- Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm\_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.
- Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, and etc. Alphastar: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- Shanchuan Wan, Yujin Tang, Yingtao Tian, and Tomoyuki Kaneko. Deir: efficient and robust exploration through discriminative-model-based episodic intrinsic rewards. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4289–4298, 2023.
- Junran Wu, Xueyuan Chen, Ke Xu, and Shangzhe Li. Structural entropy guided graph hierarchical pooling. In *ICML*, pages 24017–24030. PMLR, 2022.
- Qisong Yang and Matthijs TJ Spaan. Cem: Constrained entropy maximization for task-agnostic safe exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10798–10806, 2023.
- Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021.
- Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.
- Mingqi Yuan, Man-On Pun, and Dong Wang. Rényi state entropy maximization for exploration acceleration in reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 2022.
- Guangjie Zeng, Hao Peng, Angsheng Li, Zhiwei Liu, Chunyang Liu, Philip S Yu, and Lifang He. Unsupervised skin lesion segmentation via structural entropy minimization on multi-scale superpixel graphs. In *IEEE ICDM*, pages 768–777, 2023a.
- Xianghua Zeng, Hao Peng, and Angsheng Li. Effective and stable role-based multi-agent collaboration by structural information principles. *Proceedings of the AAAI Conference on Artificial Intelligence*, (10):11772–11780, Jun. 2023b.
- Xianghua Zeng, Hao Peng, Angsheng Li, Chunyang Liu, Lifang He, and Philip S Yu. Hierarchical state abstraction based on structural information principles. In *IJCAI*, pages 4549–4557, 2023c.
- Xianghua Zeng, Hao Peng, and Angsheng Li. Adversarial socialbots modeling based on structural information principles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 392–400, 2024.
- Chuheng Zhang, Yuanying Cai, Longbo Huang, and Jian Li. Exploration by maximizing rényi entropy for reward-free rl framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10859–10867, 2021a.
- Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E Gonzalez, and Stuart Russell. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34:9663–9680, 2021b.
- Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

## A Framework Details

### A.1 Notations

Table 3: Glossary of Notations.

Notation	Description
$X; Y$	Random variables/Vertex Sets
$x; y$	Variable values/Vertices
$p$	Probability
$H; I$	Shannon entropy; Mutual information
$\mathcal{O}; \mathcal{A}$	Observation space; Action space
$O; S; A$	Observation, state, action variables/Vertex sets
$o; s; a$	Single observation, state, action/vertex
$z; Z$	Single embedding; Embedding variables
$\mathcal{P}$	Transition function
$r; \mathcal{R}$	Reward; Reward function
$\pi; \gamma; t$	Policy network; Discount factor; Timestep
$f; q$	Encoder; Decoder
$G$	Graph
$v; V$	Single vertex; Vertex set
$e; E; w$	Edge; Edge set; Edge weight
$\alpha; T$	Tree node; Encoding tree
$\lambda; \nu$	Root node; Leaf node
$H^T, H^K, H^{SI}; \Delta H$	Structural entropy; Entropy reduction
$\mathcal{T}$	Approximate binary trees
$I^{SI}$	Structural mutual information

### A.2 Tree Optimization on $\mathcal{T}^2$

In this subsection, we have provided additional explanations and illustrative examples for the encoding tree optimization on  $\mathcal{T}^2$ . As shown in Figure 4, the stretch operator is executed over sibling nodes  $\alpha_i$  and  $\alpha_j$  that share the same parent node,  $\lambda$ . The detailed steps of this operation are as follows:

$$\alpha'^- = \lambda, \quad \alpha_i^- = \alpha', \quad \alpha_j^- = \alpha', \quad (13)$$

where  $\alpha'$  is the added tree node via the stretch operation.

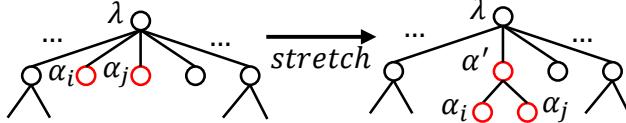


Figure 4: Stretch operation on sibling nodes within the encoding tree.

The corresponding variation in structural entropy,  $\Delta H$ , due to the stretch operation is calculated as:

$$H^T(G; \alpha_i) = -\frac{g_{\alpha_i}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha_i)}{\text{vol}(G)}, \quad H^T(G; \alpha_j) = -\frac{g_{\alpha_j}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha_j)}{\text{vol}(G)}, \quad (14)$$

$$H^{T'}(G; \alpha_i) = -\frac{g_{\alpha_i}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha_i)}{\text{vol}(\alpha')}, \quad H^{T'}(G; \alpha_j) = -\frac{g_{\alpha_j}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha_j)}{\text{vol}(\alpha')}, \quad (15)$$

$$H^{T'}(G; \alpha') = -\frac{g_{\alpha'}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)}, \quad (16)$$

$$\begin{aligned} \Delta H &= H^T(G; \alpha_i) + H^T(G; \alpha_j) - H^{T'}(G; \alpha_i) - H^{T'}(G; \alpha_j) - H^{T'}(G; \alpha') \\ &= (H^T(G; \alpha_i) - H^{T'}(G; \alpha_i)) + (H^T(G; \alpha_j) - H^{T'}(G; \alpha_j)) - H^{T'}(G; \alpha') \\ &= -\frac{g_{\alpha_i}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)} - \frac{g_{\alpha_j}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)} + \frac{g_{\alpha'}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)} \\ &= -\frac{g_{\alpha_i} + g_{\alpha_j} - g_{\alpha'}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)}. \end{aligned} \quad (17)$$

As shown in Algorithm 1, the HCSE algorithm iteratively and greedily selects the pair of sibling nodes that cause the maximum entropy variation,  $\Delta H$  to execute one stretch optimization.

---

**Algorithm 1** The Encoding Tree Optimization on  $\mathcal{T}^2$ 


---

```

1: Input: one-layer initial encoding tree  $T$ 
2: Output: the optimal encoding tree  $T^* \in \mathcal{T}^2$ 
3: while True do
4:    $(\alpha_i, \alpha_j) \leftarrow$  maximize the entropy reduction  $\Delta H$  caused by one stretch operation
5:   if  $\Delta H = 0$  then
6:     Break
7:   end if
8:   Create a new tree node  $\alpha'$ 
9:    $(\alpha')^- \leftarrow \lambda, (\alpha_i)^- \leftarrow \alpha', (\alpha_j)^- \leftarrow \alpha'$ 
10: end while

```

---

### A.3 Intuitive Example of Optimal Encoding Tree

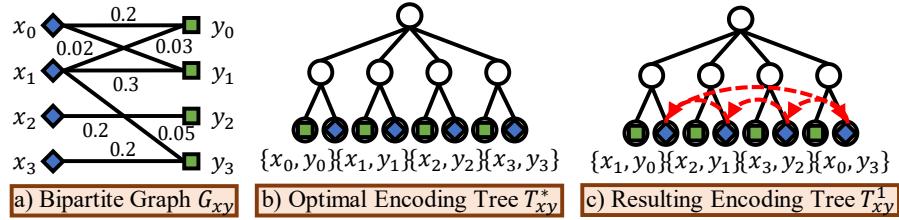


Figure 5: Illustration from a joint distribution to 2-layer approximate binary trees: a) Bipartite distribution graph, b) Optimal encoding tree, c) Resulting encoding tree via a 1-transformation.

#### A.4 The Pseudocode of SI2E

---

**Algorithm 2** Effective Exploration based on Structural Information Principles

---

```

1: Input: batch size  $n$ , update interval  $t_{\text{up}}$ 
2: Initialize: agent's policy  $\pi$ , encoder functions  $f_s$  and  $f_z$ , decoder functions  $q_m$ ,  $q_{z|s}$ ,  $q_{s|z}$ , replay buffer  $\mathcal{B}$ 
3: for each episode do
4:   for each environmental step  $t$  do
5:     Collect transition  $\tau_t = (s_t, a_t, s_{t+1}, r_t^e)$  using the encoder  $f_s$  and policy  $\pi$ 
6:     Sample a batch  $\{\tau_i\}_{i=1}^n$  from  $\mathcal{B}$  including the variables  $S_t$ ,  $A_t$ , and  $S_{t+1}$ 
7:     Adopt encoder functions  $f_z$  to obtain state-action embeddings  $Z_t$ 
8:     # Maximum Structural Entropy Exploration
9:     Construct the state-action graph  $G_{sa}$  according to the policy  $\pi$  and generate its hierarchical community structure  $T_{sa}^*$ 
10:    Employ the k-NN estimator to estimate the lower bound  $H(V_0) - H(V_1)$  and compute intrinsic reward  $r_t^i$ 
11:    Compute total reward  $r_t = r_t^e + \beta \cdot r_t^i$ 
12:    Update  $\tau'_t = (s_t, a_t, s_{t+1}, r_t)$  and augment  $\mathcal{B}$  with  $\tau'_t$ 
13:    if  $t \bmod t_{\text{up}} = 0$  then
14:      # Structural Mutual Information Principle
15:      Compute representation losses  $L_{\text{up}}$ ,  $L_{z|s}$ , and  $L_{s|z}$ 
16:      Update encoder and decoder functions to minimize the combined loss  $L$ 
17:      Update agent policy  $\pi$  using  $\mathcal{B}$ 
18:    end if
19:  end for
20: end for

```

---

#### A.5 Complexity Analysis of SI2E

Within the SI2E framework, we analyze the time complexities of critical components independent of the underlying RL algorithm. During the state-action representation phase, the construction of bipartite graphs takes  $O(n^2)$  time complexity, the generation of 2-layer approximate binary trees requires  $O(n \cdot \log^2 n)$  time complexity, and the calculation of mutual information involves a time complexity of  $O(n^2)$ . During the effective exploration phase, the generation of hierarchical community structure incurs a  $O(n \cdot \log^2 n)$  complexity, the construction of the distribution graph leads to a complexity of  $O(n^2)$ , and value-conditional structural entropy is calculated with  $O(n)$  time complexity.

#### A.6 Limitations

Our work, which is a result of thorough research, aims to address the limitations of information theory methods and structural information theory research in reinforcement learning. Therefore, we have selected the state-of-the-art information theory exploration method as the baseline in our evaluation. Despite the current limitations in height due to complexity and cost issues, the encoding tree structure in the SI2E framework holds immense potential. In our future research, we are optimistic about expanding its height further and conducting more research on the advantages and restrictions brought by this expansion.

## B Theorem Proofs

### B.1 Proof of Proposition 3.1

*Proof.* For any two vertices  $v_i \in V$  and  $v_j \in V$  without any edge connecting them, their corresponding tree nodes are denoted as  $\alpha_i$  and  $\alpha_j$ . These nodes' parents are initially assigned as the root node  $\lambda$ . Before executing one stretch operation on vertices  $\alpha_i$  and  $\alpha_j$ , their structural entropies are calculated as follows:

$$H(G; \alpha_i) = -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{\text{vol}(G)}, \quad H(G; \alpha_j) = -\frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{\text{vol}(G)}, \quad (18)$$

where  $d_i$  and  $d_j$  are the degrees of vertices  $v_i$  and  $v_j$ . Post-stretch operation, their structural entropies are given by:

$$H(G; \alpha_i) = -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{\text{vol}(\alpha')}, \quad H(G; \alpha_j) = -\frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{\text{vol}(\alpha')}, \quad (19)$$

where  $\alpha'$  are their new common parent node. The absence of an edge between  $v_i$  and  $v_j$  ensures that:

$$g_{\alpha'} = d_i + d_j, \quad \text{vol}(\alpha') = d_i + d_j. \quad (20)$$

The structural entropy of  $\alpha'$  can be determined as:

$$H(G; \alpha') = -\frac{g_{\alpha'}}{\text{vol}(G)} \cdot \log \frac{\text{vol}(\alpha')}{\text{vol}(G)} = -\frac{d_i + d_j}{\text{vol}(G)} \cdot \log \frac{d_i + d_j}{\text{vol}(G)}. \quad (21)$$

The entropy reduction  $\Delta H$ , consequent to the stretch operation on vertices  $v_i$  and  $v_j$ , is calculated as:

$$\begin{aligned} \Delta H &= \left[ -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{\text{vol}(G)} - \frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{\text{vol}(G)} \right] \\ &\quad - \left[ -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{d_i + d_j} - \frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{d_i + d_j} - \frac{d_i + d_j}{\text{vol}(G)} \cdot \log \frac{d_i + d_j}{\text{vol}(G)} \right] \\ &= \left[ -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{\text{vol}(G)} - \frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{\text{vol}(G)} \right] - \left[ -\frac{d_i}{\text{vol}(G)} \cdot \log \frac{d_i}{\text{vol}(G)} - \frac{d_j}{\text{vol}(G)} \cdot \log \frac{d_j}{\text{vol}(G)} \right] \\ &= 0. \end{aligned} \quad (22)$$

Given the zero reduction in entropy, as per lines 5 and 6 of the optimization algorithm for  $\mathcal{T}^2$  (See Appendix A.2), the stretch operation involving  $v_i$  and  $v_j$  is omitted from the optimization process.  $\square$

### B.2 Proof of Theorem 3.4

*Proof.* The difference between the mutual information  $I^{SI}(X; Y)$  and  $I(X; Y)$  is expressed as:

$$\begin{aligned} I^{SI}(X; Y) - I(X; Y) &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{2}{p(x_i) + p(y_j)} \right] - \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)} \right] \\ &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \left[ \frac{2}{p(x_i) + p(y_j)} \cdot \frac{p(x_i) \cdot p(y_j)}{p(x_i, y_j)} \right] \right] \\ &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \left[ \frac{1}{p(x_i, y_j)} \cdot \frac{2 \cdot p(x_i) \cdot p(y_j)}{p(x_i) + p(y_j)} \right] \right] \\ &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \left[ \frac{1}{p(x_i, y_j)} \cdot \frac{2}{\frac{1}{p(x_i)} + \frac{1}{p(y_j)}} \right] \right]. \end{aligned} \quad (23)$$

Given the conditions  $p(x_i, y_j) \leq p(x_i) \leq 1$  and  $p(x_i, y_j) \leq p(y_j) \leq 1$ , the following inequalities are satisfied:

$$\frac{1}{p(x_i)} + \frac{1}{p(y_j)} \leq \frac{2}{p(x_i, y_j)}, \quad \frac{1}{p(x_i, y_j)} \cdot \frac{2}{\frac{1}{p(x_i)} + \frac{1}{p(y_j)}} \geq 1, \quad (24)$$

$$0 \leq \log_{p(x_i, y_j)} p(x_i) \leq 1, \quad 0 \leq \log_{p(x_i, y_j)} p(y_j) \leq 1. \quad (25)$$

By defining  $\epsilon_i = \log_{p(x_i, y_j)} p(x_i)$  and  $\epsilon_j = \log_{p(x_i, y_j)} p(y_j)$ , we obtain the following results:

$$\begin{aligned} I^{SI}(X; Y) - I(X; Y) &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \left[ \frac{1}{p(x_i, y_j)} \cdot \frac{2}{\left[ \frac{1}{p(x_i, y_j)} \right]^{\epsilon_i} + \left[ \frac{1}{p(x_i, y_j)} \right]^{\epsilon_j}} \right] \right] \\ &\leq (1 - \min(\epsilon_i, \epsilon_j)) \cdot \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{1}{p(x_i, y_j)} \right] \\ &= (1 - \min(\epsilon_i, \epsilon_j)) \cdot H(X, Y). \end{aligned} \quad (26)$$

Therefore,

$$0 \leq I^{SI}(X; Y) - I(X; Y) \leq (1 - \min(\epsilon_i, \epsilon_j)) \cdot H(X, Y), \quad (27)$$

$$I(X; Y) \leq I^{SI}(X; Y) \leq I(X; Y) + (1 - \min(\epsilon_i, \epsilon_j)) \cdot H(X, Y). \quad (28)$$

□

### B.3 Proof of Theorem 4.1

*Proof.* For a one-to-one correspondence of variables  $X$  and  $Y$ , the joint probability of a tuple  $(x_i, y_j)$  is as follows:

$$p(x_i, y_j) = \begin{cases} p(x_i) = p(y_j) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (29)$$

The calculation for  $I^{SI}(X; Y)$  is carried out in the following manner:

$$\begin{aligned} I^{SI}(X; Y) &= \sum_{l=0}^n \left[ H^{SI}(X) + H^{SI}(Y) - H^{T_{xy}^l}(X, Y) \right] \\ &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{2}{p(x_i) + p(y_j)} \right] \\ &= \sum_i \left[ p(x_i) \cdot \log \frac{1}{p(x_i)} \right] \\ &= H(X). \end{aligned} \quad (30)$$

Similarly, the calculation for  $I(X; Y)$  proceeds as follows:

$$\begin{aligned} I(X; Y) &= \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)} \right] \\ &= \sum_i \left[ p(x_i) \cdot \log \frac{1}{p(x_i)} \right] \\ &= H(X). \end{aligned} \quad (31)$$

Hence,  $I^{SI}(X; Y)$  equals  $I(X; Y)$  when the joint distribution between  $X$  and  $Y$  shows a one-to-one correspondence. □

### B.4 Proof of Proposition 4.2

*Proof.* Now, we employ mathematical induction to demonstrate the existence of the graph  $G'_{sa}$ .

**Base Case ( $n = 2$ ):** Suppose the degree distribution of two vertices is given by  $(p_0, p_1)$  with  $p_0 \leq p_1$ .

We construct the graph  $G'_{sa}$  as follows:

- Create an edge with weight  $p_0$  between  $v_0$  and  $v_1$ .
- Add a self-connected edge at vertex  $v_1$  with weight  $p_1 - p_0$ .

**Inductive Step ( $n = k$ ):** Assume that, the graph  $G'_{sa}$  with  $k$  vertices exists and satisfies Proposition 4.2.

**Inductive Case ( $n = k + 1$ ):** Consider the addition of a new vertex  $v_k$  to construct  $G'_{sa}$  with  $k + 1$

vertices using the following steps:

- Start with a subgraph that includes the first  $k$  vertices, yielding a distribution of  $(p'_0, \dots, p'_{k-1})$  with  $\sum_{i=0}^{k-1} p'_i = 1$ .
- Modify the weight of all edges connected to each vertex  $v_i$  ( $0 \leq i \leq k$ ) by a factor  $\frac{(1-p_n)p_i}{p'_i}$ .
- For each vertex  $v_i$ , create an edge with weight  $p_i p_n$  connecting it to the new vertex  $v_k$ .
- Add a self-connected edge at vertex  $v_k$  with weight  $p_k^2$ .

These modifications ensure that the degree distribution of the graph remains consistent with the addition of the new vertex, thus completing the inductive step and proving the existence of  $G'_{sa}$  for any  $n$ .  $\square$

## B.5 Proof of Theorem 4.3

In the tree  $T_{sa}^*$ , we denote the  $i$ -th intermediate node as  $\alpha_i$  and its  $j$ -th child node as  $\alpha_{ij}$ . The single state-action vertex in the corresponding subset of  $\alpha_{ij}$  is assumed as  $(s_{ij}, a_{ij})$ . In the graph  $G'_{sa}$ , the degree of any state-action vertex  $(s_{ij}, a_{ij})$  is equated to its visitation probability, thereby:

$$\text{vol}(G'_{sa}) = \sum_{i,j} p(s_{ij}, a_{ij}) = 1, \quad \text{vol}(\alpha_{ij}) = g_{\alpha_{ij}} = p(s_{ij}, a_{ij}). \quad (32)$$

The 2-dimensional value-conditional structural entropy  $H^{T_{sa}^*}(G'_{sa})$  is calculated through the following expressions:

$$\begin{aligned} H^{T_{sa}^*}(G'_{sa}) &= - \sum_i \left[ g_{\alpha_i} \cdot \log \text{vol}(\alpha_i) + \sum_j \left[ g_{\alpha_{ij}} \cdot \log \frac{\text{vol}(\alpha_{ij})}{\text{vol}(\alpha_i)} \right] \right] \\ &= - \sum_i \left[ g_{\alpha_i} \cdot \log \text{vol}(\alpha_i) + \sum_j \left[ p(s_{ij}, a_{ij}) \cdot \log \frac{p(s_{ij}, a_{ij})}{\text{vol}(\alpha_i)} \right] \right] \\ &= \sum_{i,j} \left[ p(s_{ij}, a_{ij}) \cdot \log \frac{1}{p(s_{ij}, a_{ij})} \right] - \sum_i \left[ \text{vol}(\alpha_i) \cdot \log \frac{1}{\text{vol}(\alpha_i)} \right] + \sum_i \left[ g_{\alpha_i} \cdot \log \frac{1}{\text{vol}(\alpha_i)} \right] \\ &= H(V_0) - H(V_1) + \sum_i \left[ g_{\alpha_i} \cdot \log \frac{1}{\text{vol}(\alpha_i)} \right] \\ &\geq H(V_0) - H(V_1). \end{aligned} \quad (33)$$

Given that  $p(s_{ij}, a_{ij}) \cdot \text{vol}(\alpha_i) \leq p(s_{ij}, a_{ij}) \leq \text{vol}(\alpha_i)$ , this inequalities hold that:

$$0 \leq \log_{p(s_{ij}, a_{ij})} \frac{p(s_{ij}, a_{ij})}{\text{vol}(\alpha_i)} \leq 1. \quad (34)$$

By defining  $\zeta_{ij} = \log_{p(s_{ij}, a_{ij})} \frac{p(s_{ij}, a_{ij})}{\text{vol}(\alpha_i)}$ , we obtain that:

$$\begin{aligned} \text{vol}(\alpha_i) \cdot \log \frac{1}{\text{vol}(\alpha_i)} &= \sum_j p(s_{ij}, a_{ij}) \cdot \frac{1}{\text{vol}(\alpha_i)} \\ &= \sum_j p(s_{ij}, a_{ij}) \cdot \log \frac{1}{[p(s_{ij}, a_{ij})]^{1-\zeta_{ij}}} \\ &= \sum_j \left[ (1 - \zeta_{ij}) \cdot p(s_{ij}, a_{ij}) \cdot \log \frac{1}{p(s_{ij}, a_{ij})} \right]. \end{aligned} \quad (35)$$

Selecting the minimal  $\zeta$ -value as  $\zeta^*$  allow us to reformulate Equation 33 as follows:

$$H(V_1) = \sum_i \left[ (\text{vol}(\alpha_i)) \cdot \log \frac{1}{\text{vol}(\alpha_i)} \right] \leq (1 - \zeta^*) \cdot H(S_t, A_t), \quad (36)$$

$$H(V_0) = H(S_t, A_t), \quad (37)$$

$$\zeta^* \cdot H(S_t, A_t) \leq H(V_0) - H(V_1) \leq H^{T_{sa}^*}(G'_{sa}) \leq H(S_t, A_t). \quad (38)$$

## C Detailed Derivations

### C.1 Derivation of $I^{SI}(X; Y)$

For each intermediate node  $\alpha_i \in T_{xy}^*$  with vertex subset  $\{x_i, y_i\}$ , the entropy sum of this node and its children is calculated as follows:

$$\begin{aligned} & -\frac{g_{\alpha_i}}{\text{vol}(G_{xy})} \cdot \log \frac{\text{vol}(\alpha_i)}{\text{vol}(G_{xy})} - \frac{p(x_i)}{\text{vol}(G_{xy})} \cdot \log \frac{p(x_i)}{\text{vol}(\alpha_i)} - \frac{p(y_i)}{\text{vol}(G_{xy})} \cdot \log \frac{p(y_i)}{\text{vol}(\alpha_i)} \\ & = -\frac{g_{\alpha_i}}{2} \cdot \log \frac{\text{vol}(\alpha_i)}{2} - \frac{p(x_i)}{2} \cdot \log \frac{p(x_i)}{\text{vol}(\alpha_i)} - \frac{p(y_i)}{2} \cdot \log \frac{p(y_i)}{\text{vol}(\alpha_i)}. \end{aligned} \quad (39)$$

The  $H^{SI}(X) + H^{SI}(Y) - H^{T_{xy}^*}(X, Y)$  term in Equation 5 is calculated as follows:

$$\sum_i \left[ -\frac{p(x_i)}{2} \cdot \log \frac{p(x_i)}{2} - \frac{p(y_i)}{2} \cdot \log \frac{p(y_i)}{2} \right] - \sum_i \left[ -\frac{g_{\alpha_i}}{2} \cdot \log \frac{\text{vol}(\alpha_i)}{2} - \frac{p(x_i)}{2} \cdot \log \frac{p(x_i)}{\text{vol}(\alpha_i)} - \frac{p(y_i)}{2} \cdot \log \frac{p(y_i)}{\text{vol}(\alpha_i)} \right]. \quad (40)$$

Proposition 3.1 ensures that:

$$\text{vol}(\alpha_i) = p(x_i) + p(y_i), \quad g_{\alpha_i} = p(x_i) + p(y_i) - 2p(x_i, y_i). \quad (41)$$

Consequently, we can reformulate Equation 40 as follows:

$$\begin{aligned} & \sum_i \left[ \frac{p(x_i)}{2} \cdot \log \frac{2}{p(x_i) + p(y_i)} + \frac{p(y_i)}{2} \cdot \log \frac{2}{p(x_i) + p(y_i)} - \frac{p(x_i) + p(y_i) - 2 \cdot p(x_i, y_i)}{2} \cdot \log \frac{2}{p(x_i) + p(y_i)} \right] \\ & = \sum_i \left[ p(x_i, y_i) \cdot \log \frac{2}{p(x_i) + p(y_i)} \right]. \end{aligned} \quad (42)$$

For any integer  $l > 0$ , the  $H^{SI}(X) + H^{SI}(Y) - H^{T_{xy}^l}(X, Y)$  term in Equation 5 is given by:

$$\sum_i \left[ p(x_{i'}, y_i) \cdot \log \frac{2}{p(x_{i'}) + p(y_i)} \right], \quad i' = (i + l) \bmod n. \quad (43)$$

Consequently,

$$I^{SI}(X; Y) = \sum_{l=0}^n \left[ H^{SI}(X) + H^{SI}(Y) - H^{T_{xy}^l}(X, Y) \right] = \sum_{i,j} \left[ p(x_i, y_j) \cdot \log \frac{2}{p(x_i) + p(y_j)} \right]. \quad (44)$$

### C.2 Upper Bound of $I^{SI}(Z_t; S_t)$

Theorem 3.4 assures the following inequality:

$$I^{SI}(X; Y) \leq I(X; Y) + (1 - \epsilon) \cdot H(X, Y), \quad 0 \leq \epsilon \leq 1. \quad (45)$$

Given the relationship between joint entropy and conditional entropy in traditional information theory, this inequality can be reformulated as follows:

$$\begin{aligned} I^{SI}(X; Y) & \leq I(X; Y) + (1 - \epsilon) \cdot H(X, Y) \\ & = I(X; Y) + (1 - \epsilon) \cdot H(X|Y) + (1 - \epsilon) \cdot H(Y) \\ & \leq I(X; Y) + H(X|Y) + H(Y). \end{aligned} \quad (46)$$

### C.3 Upper Bound of $I(Z_t; S_t)$

Through the non-negativity of KL-divergence, the following upper bound of  $I(Z_t; S_t)$  holds that:

$$\begin{aligned} I(Z_t; S_t) & = \sum \left[ p(z_t, s_t) \cdot \log \frac{p(z_t|s_t)}{p(z_t)} \right] \\ & = \sum \left[ p(z_t, s_t) \cdot \log \frac{p(z_t|s_t)}{q_m(z_t)} \right] - D_{KL}(p||q_m) \\ & \leq \sum [p(z_t, s_t) \cdot D_{KL}(p(z_t|s_t)||q_m(z_t))]. \end{aligned} \quad (47)$$

#### C.4 Upper Bound of $H(Z_t|S_t)$

Through the non-negativity of KL-divergence, the following upper bound of  $H(Z_t|S_t)$  holds that:

$$\begin{aligned} H(Z_t|S_t) &= \sum \left[ p(z_t, s_t) \cdot \log \frac{1}{p(z_t|s_t)} \right] \\ &= \sum \left[ p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)} \right] - D_{KL}(p||q_{z|s}) \\ &\leq \sum \left[ p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)} \right]. \end{aligned} \quad (48)$$

#### C.5 Lower Bound of $I(Z_t; S_{t+1})$

Leveraging the non-negative Shannon entropy and KL-divergence, we obtain the lower bound of  $I(Z_t; S_{t+1})$ :

$$\begin{aligned} I(Z_t; S_{t+1}) &= \sum \left[ p(z_t, s_{t+1}) \cdot \log \frac{p(s_{t+1}|z_t)}{p(s_{t+1})} \right] \\ &= \sum [p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t)] + H(S_{t+1}) + D_{KL}(p||q_{s|z}) \\ &\geq \sum [p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t)]. \end{aligned} \quad (49)$$

## D Experimental Details

In the experiments conducted for this work, we utilize a single NVIDIA RTX A1000 GPU and eight Intel Core i9 CPU cores clocked at 3.00GHz for each training run. The total number of environmental steps was set to  $3000K/1000K$  for the MiniGrid benchmark,  $200K/100K$  for the MetaWorld benchmark, and  $250K$  for the DeepMind Control Suite (DMControl).

### D.1 Implementation Details

**A2C implementation details.** In our implementation of the A2C algorithm, we utilize the official RE3 implementation<sup>2</sup>, adhering to the pre-established hyperparameters set, except where explicitly noted. State representations in baselines are derived using a fixed encoder that is randomly initialized, and intrinsic rewards are normalized based on the standard deviation computed from sample data, consistent with the original methodology. However, this normalization process is omitted in the intrinsic reward calculations for VCSE and SI2E implementations. Across all exploration methods, we maintain fixed scale parameters  $\beta = 0.005$  and  $k = 5$ , in line with the original framework. The comprehensive hyperparameters for the A2C algorithm are detailed in Table 4.

Table 4: Hyperparameters for the A2C algorithm on the MiniGrid benchmark.

Hyperparameter	Value
number of updates between two savings	100
number of processes	16
number of frames in training	$3e6/1e6$
scale parameter $\beta$	0.005
batch size	256
number of frames per process before update	5
discount factor	0.99
learning rate	0.001
GAE coefficient	0.95
maximum norm of gradient	0.5

**DrQv2 implementation details.** For the DrQv2 algorithm, we employ its official implementation<sup>3</sup> [Yarats et al., 2021], maintaining the original hyperparameter settings unless specified otherwise. A fixed noise level of 0.2 and  $k = 12$  are used for all exploration methods, including SE, VCSE, MADE, and SI2E. In the calculation of intrinsic rewards in baselines, we train the Intrinsic Curiosity Module [Pathak et al., 2017] using representations from the visual encoder to measure vertex distance in the estimation of value-conditional structural entropy. Specific hyperparameters in the DrQv2 are summarized in Table 5.

Table 5: Hyperparameters for the DrQv2 algorithm on the DeepMind Control Suite.

Hyperparameter	Value
number of frames stacked	3
number of times each action is repeated	2
number of frames for an evaluation	10000
number of episodes for each evaluation	10
number of worker threads for the replay buffer	4
replay buffer size	$1e6$
batch size	64
discount factor	0.99
learning rate	0.0001
feature dimensionality	50
hidden dimensionality	1024
scale parameter $\beta$	0.1

<sup>2</sup><https://github.com/younggyoseo/RE3>

<sup>3</sup><https://github.com/facebookresearch/drqv2>

## D.2 Environment Details

**MiniGrid Experiments.** In our MiniGrid benchmark experiments, we encompass six navigation tasks, including RedBlueDoors, SimpleCorssingS9N1, KeyCorridor, DoorKey-6x6, DoorKey-8x8, and Unlock, with visual representations provided in Figure 6. Notably, all tasks are employed in their original forms without any modifications.

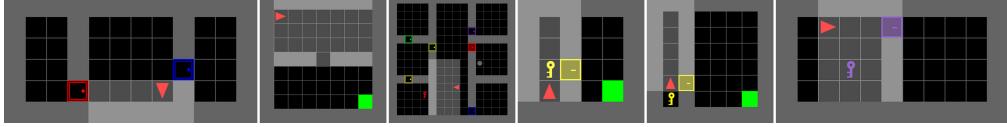


Figure 6: Examples of navigation tasks used in our MiniGrid experiments include: (a) RedBlueDoors, (b) SimpleCorssingS9N1, (c) KeyCorridor, (d) DoorKey-6x6, (e) DoorKey-8x8, (f) Unlock.

**MetaWorld Experiments.** In our evaluation using the MetaWorld benchmark, we conduct experiments on six manipulation tasks: Door Open, Drawer Open, Faucet Open, Window Open, Button Press, and Faucet Close. These tasks are visualized in Figure 7.



Figure 7: Examples of manipulation tasks used in our MetaWorld experiments include: (a) Door Open, (b) Drawer Open, (c) Faucet Open, (d) Window Open, (e) Button Press, (f) Faucet Close.

**DMControl Experiments.** Our research in DMControl suite focuses on six continuous control tasks, specifically Hopper Stand, Cheetah Run, Quadruped Walk, Pendulum Swingup, Cartpole Balance Sparse, and Cartpole Swingup Sparse. And visualizations of these tasks are provided in Figure 8.

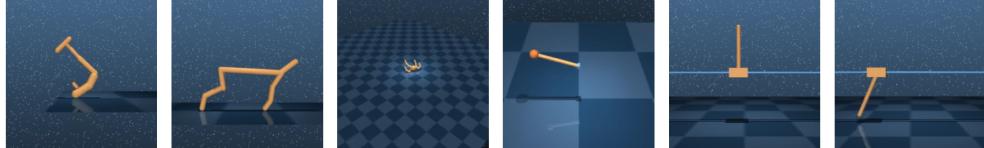


Figure 8: Examples of control tasks used in our DeepMind Control Suite experiments include: (a) Hopper Stand, (b) Cheetah Run, (c) Quadruped Walk, (d) Pendulum Swingup, (e) Cartpole Balance Sparse, (f) Cartpole Swingup Sparse.

## E Additional Experiments

### E.1 Experiments on MiniGrid Benchmark

Figure 9 illustrates the learning curves for the A2C algorithm integrated with our SI2E framework, as well as with other exploration baselines, SE and VCSE. The corresponding variants are labeled as A2C, A2C+SE, A2C+VCSE, and A2C+SI2E. These results demonstrate that SI2E consistently outperforms other baselines across various navigation tasks. In terms of final performance, A2C+SI2E achieves an average success rate of 94.40% across all navigation tasks, significantly outperforming the second-best A2C+VCSE, which has an average success rate of 89.78%. Regarding sample efficiency, SI2E converges in fewer than 50% of the total environmental steps in almost all tasks, except for DoorKey-8x8. This indicates SI2E’s effectiveness in enhancing the agent’s exploration of the state-action space, surpassing the baseline methods.

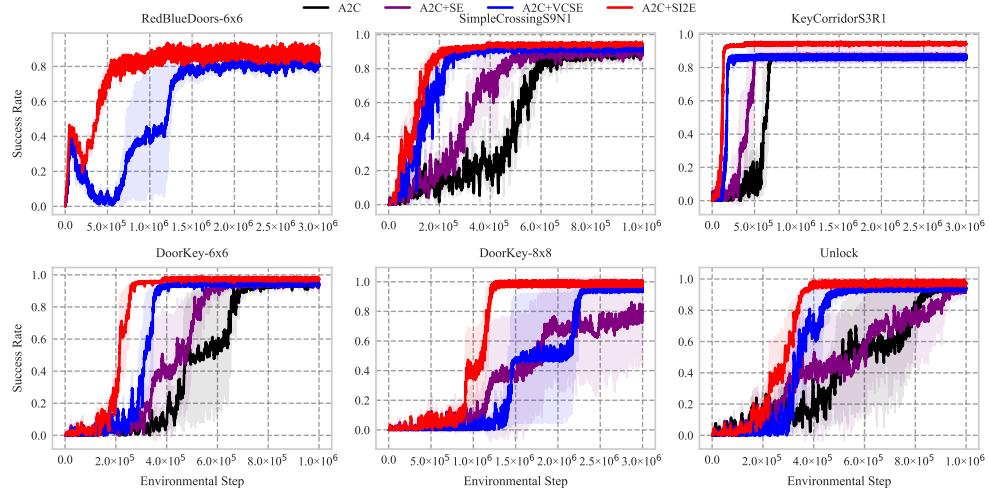


Figure 9: Learning curves for six navigation tasks in MiniGrid, measured in terms of success rate. The solid lines represent the interquartile mean, while the shaded regions indicate the standard deviation, both calculated across 10 runs.

To further substantiate our results on the Minigrid environment, we have introduced Leco[Jo et al., 2022] and DEIR[Wan et al., 2023], two additional state-of-the-art baselines representing advanced mechanisms for episodic intrinsic rewards. These results in Table 6 demonstrate that our method consistently maintains a performance advantage in terms of effectiveness and efficiency, even when compared to advanced episodic intrinsic reward mechanisms.

Table 6: Comparative results between SI2E and advanced episodic intrinsic reward mechanisms in MiniGrid benchmark: : “average value  $\pm$  standard deviation” and “average improvement”

MiniGrid Navigation	RedBlueDoors-6x6		SimpleCrossingS9N1		KeyCorridorS3R1	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
Leco	81.97 $\pm$ 10.81	817.47 $\pm$ 137.21	90.02 $\pm$ 4.13	417.59 $\pm$ 17.63	90.36 $\pm$ 0.57	520.43 $\pm$ 10.31
DEIR	78.32 $\pm$ 7.21	722.37 $\pm$ 81.93	91.47 $\pm$ 8.29	523.79 $\pm$ 31.27	91.81 $\pm$ 2.13	735.87 $\pm$ 9.24
SI2E	85.80 $\pm$ 1.48	461.90 $\pm$ 61.53	93.64 $\pm$ 1.63	139.17 $\pm$ 27.03	94.20 $\pm$ 0.42	129.06 $\pm$ 6.11
MiniGrid Navigation	DoorKey-6x6		DoorKey-8x8		Unlock	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
Leco	94.37 $\pm$ 3.41	571.31 $\pm$ 31.27	92.07 $\pm$ 19.11	2168.35 $\pm$ 293.52	94.48 $\pm$ 6.39	791.40 $\pm$ 82.39
DEIR	94.81 $\pm$ 5.13	410.25 $\pm$ 29.16	95.41 $\pm$ 13.27	1247.58 $\pm$ 231.42	95.13 $\pm$ 12.74	531.06 $\pm$ 131.84
SI2E	97.04 $\pm$ 1.52	230.60 $\pm$ 19.85	98.58 $\pm$ 3.11	1090.96 $\pm$ 125.77	97.13 $\pm$ 3.35	309.14 $\pm$ 53.71

## E.2 Experiments on MetaWorld Benchmark

Figure 10 summarizes the learning curves for the DrQv2 algorithm with different exploration methods, including SI2E (ours), SE, and VCSE. These variants are denoted as DrQv2, DrQv2+SE, DrQv2+VCSE, and DrQv2+SI2E, respectively. As shown in Figure 10, SI2E consistently and significantly achieves higher success rates with fewer environmental steps across all manipulation tasks. On average, SI2E improves the final performance by 10.21% and sample efficiency by 45.06%.

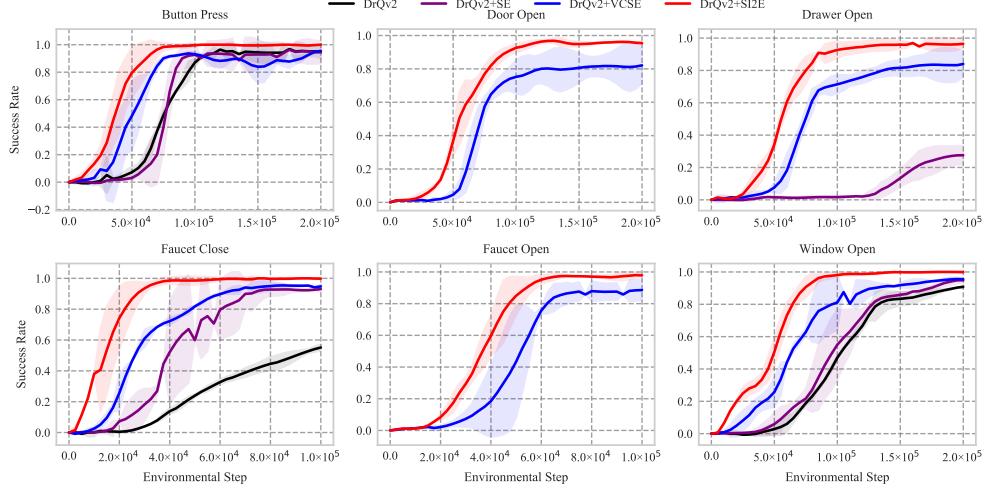


Figure 10: Learning curves for six manipulation tasks in MetaWorld, measured in terms of success rate. The solid lines represent the interquartile mean, while the shaded regions indicate the standard deviation, both calculated across 10 runs.

To provide additional validation, we conducted further experiments in the MetaWorld environment using the advanced TACO [Zheng et al., 2024] as the underlying agent. The new experimental results, as documented in Table 7, demonstrate the robustness and superior performance of our method with different underlying agents.

Table 7: Summary of success rates and required steps for the TACO agent in MetaWorld tasks: “average value  $\pm$  standard deviation” and “average improvement”. **Bold**: the best performance, **underline**: the second performance.

MetaWorld Manipulation	Button Press		Door Open		Drawer Open	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
TACO	73.56 $\pm$ 21.43	-	-	-	33.47 $\pm$ 2.15	-
TACO+VCSE	91.43 $\pm$ 5.51	95.0 $\pm$ 5.0	78.59 $\pm$ 4.22	-	70.93 $\pm$ 6.65	-
TACO+SI2E	<b>95.63</b> $\pm$ 0.62	<b>75.0</b> $\pm$ 5.0	<b>96.72</b> $\pm$ 4.98	<b>115.0</b> $\pm$ 10.0	<b>94.92</b> $\pm$ 2.06	<b>75.0</b> $\pm$ 5.0
Abs.(%) Avg.	4.2(4.59) $\uparrow$	20.0(21.05) $\downarrow$	18.13(23.07) $\uparrow$	-	23.99(33.82) $\uparrow$	-
MetaWorld Manipulation	Faucet Close		Faucet Open		Window Open	
	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )	Success Rate (%)	Required Step ( $K$ )
TACO	58.16 $\pm$ 4.23	-	49.48 $\pm$ 11.60	-	76.86 $\pm$ 5.79	-
TACO+VCSE	76.00 $\pm$ 0.75	-	80.40 $\pm$ 3.65	-	92.31 $\pm$ 3.17	<u>120.0</u> $\pm$ 5.0
TACO+SI2E	<b>93.78</b> $\pm$ 1.21	<b>40.0</b> $\pm$ 5.0	<b>92.09</b> $\pm$ 2.26	<b>51.25</b> $\pm$ 1.25	<b>97.03</b> $\pm$ 1.58	<u>70.0</u> $\pm$ 2.5
Abs.(%) Avg.	17.78(23.39) $\uparrow$	-	11.69(14.54) $\uparrow$	-	4.72(5.11) $\uparrow$	50.0(41.67) $\downarrow$

### E.3 Experiments on DMControl Suite

For each task, we benchmark the convergence reward of the best-performing baseline as the target and track the environmental steps required by both SI2E and this baseline to reach the target. As illustrated in Figure 11, SISA demonstrates an average improvement of 35.60% in sample efficiency. This improvement is reflected in a reduction in the required environmental steps, decreasing from 31.83% to 20.5% of the total steps required to achieve the reward target.

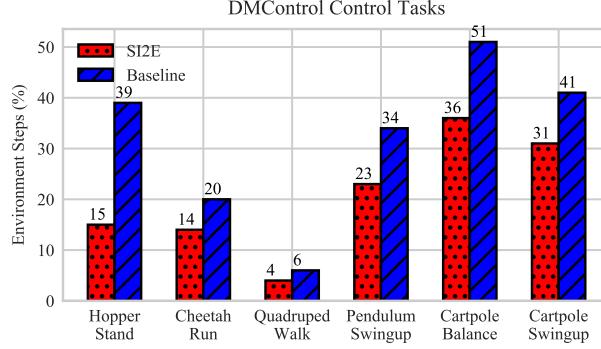


Figure 11: Comparison of sample efficiency between SI2E and the best-performing baseline in DMControl, focusing on the required environmental steps to reach the reward target, expressed as a proportion of the total 250K steps.

Figure 12 shows the learning curves for the DrQv2 algorithm when integrated with our SI2E framework and other exploration baselines, SE, VCSE, and MADE. The variants are identified as DrQv2, DrQv2+SE, DrQv2+VCSE, DrQv2+MADE, and DrQv2+SI2E. These results reveal that SI2E exploration significantly improves the sample efficiency of DrQv2 in both sparse reward and dense reward tasks, outperforming all other baselines. Particular in sparse reward tasks (Cartpole Balance and Cartpole Swingup), our framework successfully accelerates training and achieves higher episode rewards. This suggests that SI2E avoids exploring states that may not contribute to task resolution, thereby enhancing performance with a 5.17% improvement in final performance and a 15.28% increase in sample efficiency.

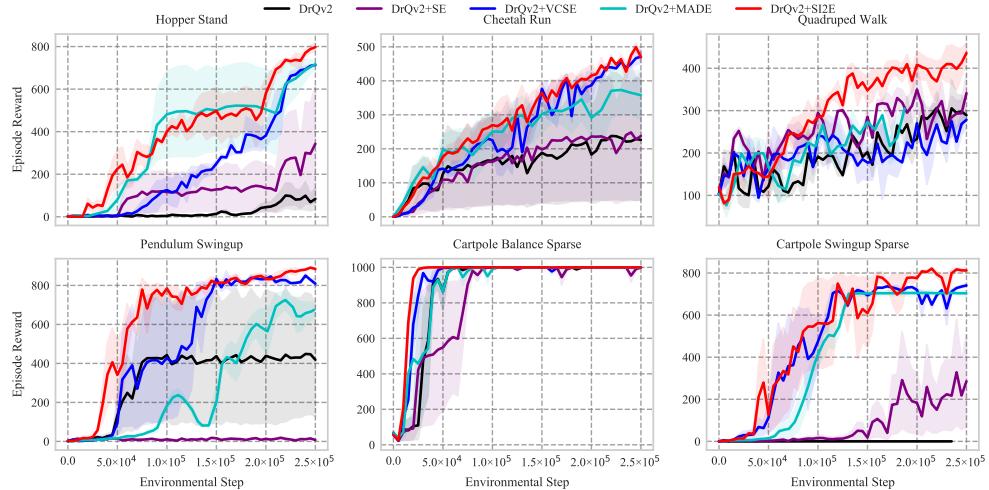


Figure 12: Learning curves for six continuous control tasks from DMControl Suite, measured in terms of episode reward. The solid lines represent the interquartile mean, while the shaded regions indicate the standard deviation, both calculated across 10 runs.

#### E.4 Visualization Experiment.

To understand the learned representation through our structural mutual information principle (Section 4.1), we use t-SNE to project the original observation  $O$  and state-action embedding  $Z$  into 2-dimensional vectors. Figure 13a visualizes these 2-dimensional vectors and measures the Euclidean distance between temporally consecutive vectors. Our presented principle aligns all movements on the same curve and reduces their distance by an average of 36.41, demonstrating its effectiveness for dynamics-relevant state-action embedding.

To analyze the benefits of our intrinsic reward mechanism (Section 4.2), we keep the map configuration constant in the SimpleCrossing task and display the exploration paths of various methods in Figure 13b. Our SI2E, in comparison to the SE and VCSE baselines, prevents biased exploration towards low-value areas and effectively explores high-value crucial areas, like the crossing point, with fewer environmental steps, ensuring its performance superiority.

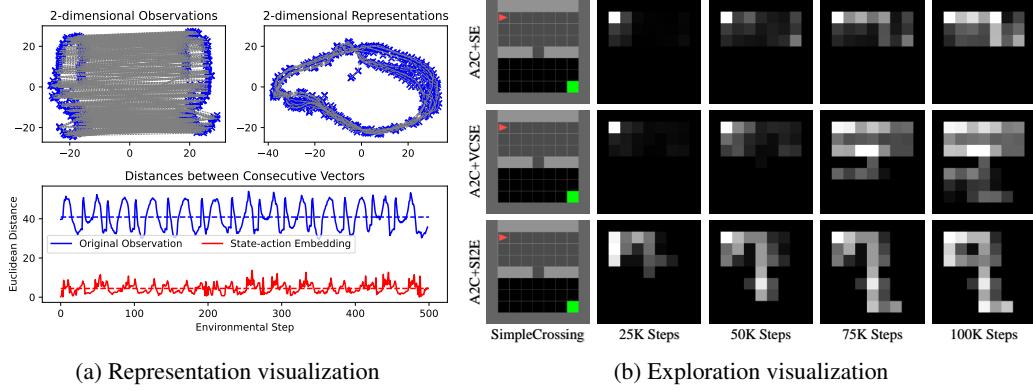


Figure 13: Visualization results of SI2E under DMControl and Minigrid tasks. (a) Visualization of representation learning based on structural mutual information principle. (b) Visualization of agent exploration through maximizing the value-conditional structural entropy.

Moreover, we have provided the visualizations of state coverage of our SI2E framework and other baselines (SE and VCSE) in the continuous Cartpole Balance task in Figure 14. Compared with the SE and VCSE baselines, our SI2E framework achieves a more efficient exploration strategy by minimizing occurrences in extreme positions and angles, resulting in a higher density in key areas.

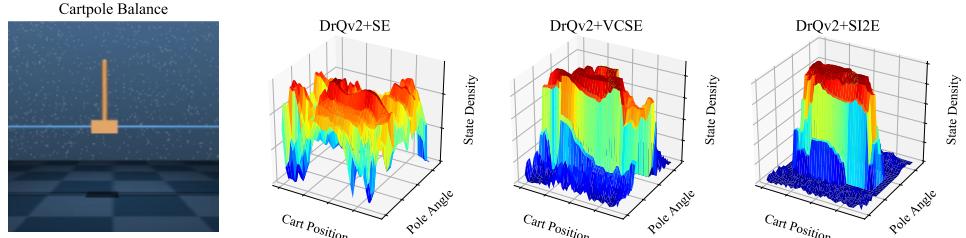


Figure 14: Visualization of agent exploration in the CartPole Balance task. Heat maps illustrate the final state densities for cart position and pole angle of the learned policies: (b)DrQv2+SE, (c) DrQv2+VCSE, and (d) DrQv2+SI2E.

## E.5 Ablation Studies

To investigate the influence of parameters  $\beta$  and  $n$  on our framework's performance, we incrementally adjust these parameters across two distinct tasks: DMControl tasks Hopper Stand and Pendulum Swingup. We meticulously document the resulting learning curves to assess the outcomes. Figure 15a illustrates that an increase in parameter  $\beta$  consistently enhances performance across both tasks, substantiating the effectiveness of our exploration method. Conversely, Figure 15b demonstrates that variations in batch size  $n$  yield comparable performance outcomes, particularly notable in Pendulum Swingup task, thereby confirming the SI2E's stability amidst variations in batch size.

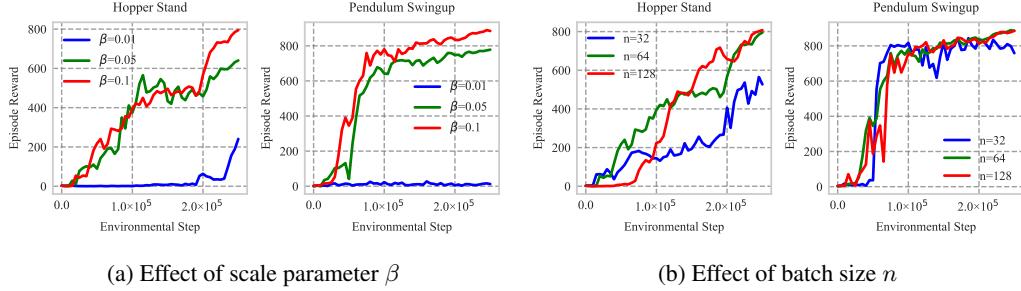


Figure 15: Learning curves of SI2E with varied  $\beta$  and  $n$  values on Hopper Stand and Pendulum Swingup tasks. (a) shows the effect of the scale parameter  $\beta$  on episode reward. (b) shows the effect of the batch size  $n$  on episode reward. The solid line represents the interquartile mean across 10 runs.

## 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: We have clearly outlined the contributions and scope of our paper 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: We have outlined the limitations of our paper in the conclusion and the details are provided in our appendix.

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: [Yes]

Justification: For each theoretical result, we have included the complete set of assumptions and a correct proof in our appendix.

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 source codes are accessible via an anonymous link in the introduction, and specific experimental setups are provided in the appendix.

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 main experimental results can be replicated using the source codes accessible via an anonymous link provided in the introduction.

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: We have provided detailed experimental settings (hyperparameter and optimizer) in our evaluation and appendix.

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: In our evaluation section, we have conducted multiple trials with different random seeds, and the charts reflect the average performance and standard deviation.

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: We have provided sufficient information on the computer resources in our appendix.

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: Our paper strictly maintains anonymity.

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: [NA]

Justification: Our work is focused on fundamental research in reinforcement learning and is not specifically tied to any particular applications. Furthermore, there is no direct path to negative applications.

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: Our 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 appropriately cited the original papers for our experimental datasets, including MiniGrid, MetaWorld, and the DeepMind Control Suite.

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](http://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: Our paper does not release new assets.

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: Our 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: Our 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.