Effective Exploration Based on the Structural Information Principles

Anonymous Authors[1]

Abstract

Traditional information theory presents a promising pathway for Reinforcement Learning (RL), facilitated by representation learning and entropy maximization in exploration. However, these methods primarily concentrate on modeling the uncertainty of RL's random variables, neglecting the structural information embedded within the state and action spaces. In this paper, we propose a novel **S**tructural **I**nformation principles-based **E**ffective **E**xploration framework, namely **SI2E**. The SI2E employs structural entropy to quantify uncertainty in state-action transitions. An innovative embedding principle is presented, defining entropy reduction in encoding trees as structural mutual information to capture dynamics-relevant representations. A unique intrinsic reward mechanism that maximizes value-conditional structural entropy, is designed to promote uniform coverage across the state-action space. Theoretical connections are established between SI2E and classical information-theoretic methodologies, underscoring the rationality of our framework. Comprehensive evaluations on various decision-making tasks in the MiniGrid and DeepMind Control Suite benchmarks demonstrate that SI2E significantly outperforms state-of-the-art exploration baselines in terms of final performance and sample efficiency, with maximum improvements of 37.63% and 44.0%, respectively.

1. Introduction

Reinforcement Learning (RL) has emerged as a pivotal technique for interacting with complex environments and addressing challenging sequential decision-making problems including computer games (Vinyals et al., 2019; Badia et al., 2020), robotic control tasks (Andrychowicz et al., 2017; Liu & 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 essential for optimizing agent policies and reducing the risk of suboptimal outcomes, particularly in high-dimensional and sparse-reward scenarios (Zhang et al., 2021b).

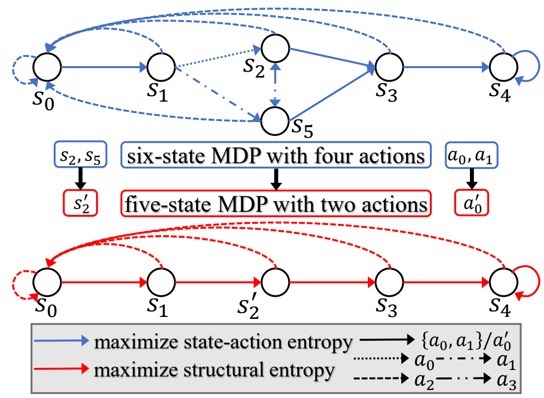


Figure 1: By incorporating structural information inherent in the state and action spaces, we simplify the original six-state Markov Decision Process (MDP) with four actions to a five-state MDP with two actions, effectively reducing the state-action space from $24(6\times4)$ to $10(5\times2)$. Here, 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 maximizing structural entropy would selectively focus on crucial transitions (red color), avoiding redundant transitions between s_2 and s_5.

Recently advancements in information-theoretic approaches have shown promise for exploration in self-supervised settings. The maximum entropy framework over action space (Haarnoja et al., 2017) leads 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, resulting in vulnerability to imbalanced state-value distributions in supervised settings. To

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

mitigate this, the concept of value-conditional state entropy is introduced, which computes intrinsic rewards based on 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 these successes, existing information-theoretic exploration methods have a critical limitation: they overlook the structural information inherent in state and action spaces. In Figure 1, we illustrate a clear example where policies that incorporate structural information reduce the size of the state-action space and avoid redundant transitions within the same vertex community, thus enhancing effectiveness and efficiency.

Therefore, we propose SI2E, a unified framework based on structural information principles, tailored for effective exploration in high-dimensional and sparse-reward environments. Initially, we embed state-action pairs into a low-dimensional space and quantify the uncertainty of random walks between the embedding and observation spaces as structural entropy. To capture dynamics-relevant information, we define the average entropy reduction achieved through encoding trees as structural mutual information and present an innovative embedding principle for state-action representation. Specifically, this principle maximizes this mutual information with subsequent observations while minimizing it with current observations. Following this, we analyze value differences in state-action pairs to construct a distribution graph and its optimal encoding tree, thereby revealing the hierarchical community structure inherent in the state-action space. We then design an intrinsic reward mechanism that maximizes this graph's high-dimensional structural entropy, thus promoting uniform explorations both within and across these communities. Furthermore, we establish theoretical connections between our framework and classical information-theoretic explorations, demonstrating the rationality and advantage of SI2E. Our extensive evaluations across diverse and challenging tasks in the MiniGrid and DeepMind Control Suite benchmarks have demonstrated SI2E's significant improvements in final performance and sample efficiency, surpassing state-of-the-art exploration baselines. For further research, the source codes are accessible via an anonymous link[1]. Our contributions are summarized as follows:

• A novel framework based on structural information principles, namely SI2E, is proposed for effective exploration in high-dimensional RL environments with sparse rewards.

• An innovative structural mutual information principle is presented to enhance the acquisition of dynamics-relevant representations for state-action pairs.

[1] `https://anonymous.4open.science/r/cced6655/`

• A unique intrinsic rewards mechanism that maximizes value-conditional structural entropy is designed to promote uniform coverage across the state-action space.

• Theoretical connections between our framework and classical information-theoretic exploration methods are established to demonstrate the generality and advantage of SI2E.

• Our experiments on various challenging tasks demonstrate that SI2E significantly improves final performance and sample efficiency by up to 37.63% and 44.0%, 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 reference.

2.1. Traditional Information Principles

Consider the random variable pair $Z = (X, Y)$ with a joint distribution probability denoted as $p(x, y) \in (0, 1)$. The marginal probabilities $p(x)$ and $p(y)$ are $p(x) = \sum_y p(x, y)$ and $p(y) = \sum_x p(x, y)$, respectively. The joint Shannon entropy (Shannon, 1953) of X and Y is given by $H(X, Y) = -\sum_{(x,y)} [p(x, y) \cdot \log p(x, y)]$, which quantifies the uncertainty inherent 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, and satisfies the following relation:

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \tag{1}$$

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 $a_t \in \mathcal{A}$ after observing $o_t \in \mathcal{O}$, yielding a transition to a new observation $o_{t+1} \sim \mathcal{P}(o_t, a_t)$ and a reward $r_t^e \in \mathbb{R}$. And the policy network π is optimized to maximize the cumulative long-term expected discounted reward.

Maximum State Entropy Exploration. In environments characterized by 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:

$$\widehat{H}_{KL}(X) = \frac{d_x}{n} \sum_{i=0}^{n-1} \log d(x_i) + C, \tag{2}$$

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

Information Bottleneck Principle. In the supervised learning paradigm, the objective of representation learning is to transform an input source X into a representation Z, aimed at 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, thereby capturing the relevant features of Y in Z. Concurrently, the IB principle imposes a complexity constraint by minimizing the mutual information $I(Z;X)$ between Z and X, which discards irrelevant features. To reconcile these dual objectives, the IB principle employs a Lagrangian multiplier, facilitating the trade-off between the richness of the representation and its complexity.

2.3. Structural Information Principles

Departing from the traditional information principle applied to random variables, structural entropy (Li & Pan, 2016) is devised to quantify dynamic uncertainty within complex networks under a hierarchical partitioning structure known as "encoding tree". This entropy is conceptualized as the minimum number of bits required to encode a vertex accessible via one random walk on a undirected and weighted graph $G = (V, E)$.

The encoding tree T of G is characterized as a rooted tree with the following properties: 1) Each tree node α in T corresponds to a subset of graph vertices $T_\alpha \subseteq V$. 2) The subset T_λ of tree root λ encompasses all vertices in V. 3) Each subset T_ν of a leaf node ν in T only contains a single vertex v, thus $T_\nu = \{v\}$. 4) For each non-leaf node α, the number of its children is assumed as l_α, with the i-th child specified as α_i. The collection of subsets $T_{\alpha_1}, \ldots, T_{\alpha_{l_\alpha}}$ constitutes a sub-partition of T_α.

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

$$H^T(G) = -\sum_{\alpha \in T, \alpha \neq \lambda} \left[\frac{g_\alpha}{\mathrm{vol}(G)} \cdot \log \frac{\mathrm{vol}(\alpha)}{\mathrm{vol}(\alpha^-)} \right], \tag{3}$$

$$H^K(G) = \min_T H^T(G), \tag{4}$$

where g_α is the weighted sum of edges connecting vertices within the subset T_α to vertices outside T_α. And $\mathrm{vol}(\alpha)$ is the degree sum of all vertices in the subset T_α.

3. The Proposed SI2E Framework

In this work, we propose a novel exploration framework that leverages dynamic-relevant representations derived from the mutual structural information principle and maximizes structural entropy to enhance the coverage of the state-action space conditioned by the agent policy. Initially, we define structural mutual information and apply it to generate representations for state-action pairs (see Section 3.1). Subsequently, we describe the process of maximizing structural entropy, which functions as an intrinsic reward to train the policy network (see Section 3.2). The overall architecture of our framework is illustrated in Figure 2.

3.1. Structural Mutual Information Principle

To effectively learn dynamics-relevant representations for state-action pairs, we present a structural mutual information principle in SI2E. This principle defines the average entropy reduction obtained by encoding trees as mutual structural information. Drawing inspiration from the traditional Information Bottleneck (IB) (Tishby et al., 2000), our principle maximizes this mutual information with subsequent observations while minimizing it with current observations.

In the representation phase, the input variables at timestep t comprise 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. Our objective is 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:

$$S_t = f_s(O_t), S_{t+1} = f_s(O_{t+1}), Z_t = f_z(S_t, A_t), \tag{5}$$

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 $G_{zs'}$ (step I. b in Figure 2). These graphs represent the joint distributions P of Z_t with the current states S_t and subsequent states S_{t+1}. For clarity, we take the graph G_{zs} as an example, and operations applied to $G_{zs'}$ follow a similar methodology. In G_{zs}, each state-action pair z_t in Z_t is connected to states s_t in S_t via weighted edges. The weight $p(z_t, s_t)$ is the joint probability of the tuple (z_t, s_t). Notably, there are no edges connecting state-action pairs directly or states alone, and the total sum of the weights of all edges equals 1. In the graph G_{zs}, each random walk signifies a matching between two vertices z_t and s_t. The inherent uncertainty of these random walks, quantified by structural entropy, serves as a metric for refining the matching between variables Z_t and S_t.

Structural Mutual Information. In this subsection, we focus on reducing the structural entropy of graph G_{zs} by op-

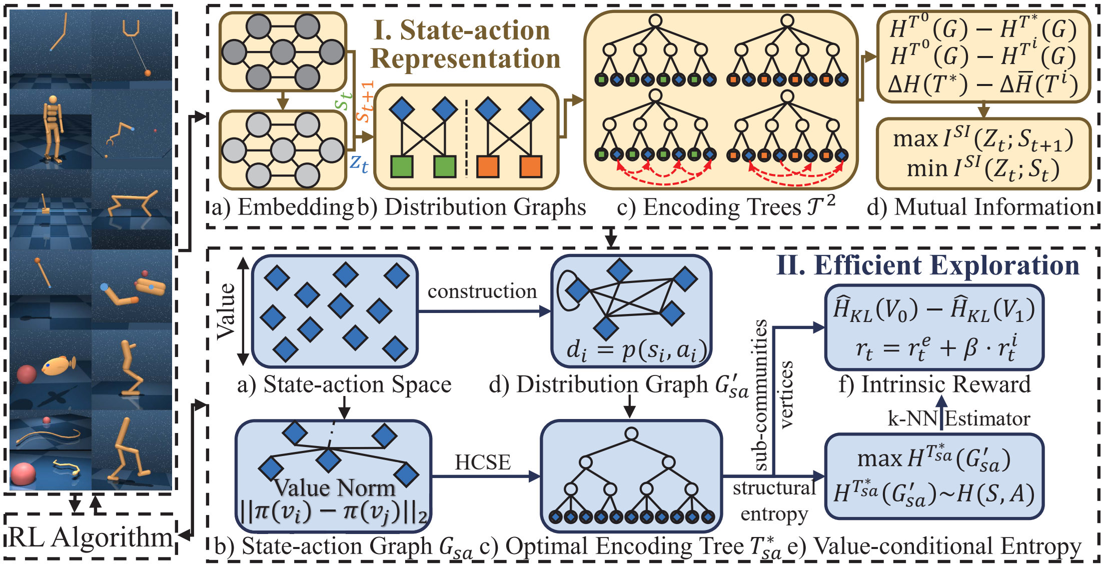


Figure 2: Overview of SI2E. The SI2E framework comprises two primary phases: state-action representation and efficient exploration. Initially, SI2E embeds the action-observation history and defines structural mutual information to generate dynamic-relevant representations of state-action pairs. Following this phase, SI2E constructs a state-action graph and maximizes its structural entropy to maximize the coverage of the state-action space, conditioned by agent policy.

timizing its hierarchical structure, the encoding tree, thereby refining the matching between Z_t and S_t. To attain a one-to-one matching between these variables, this optimization is further restricted to 2-layer approximate binary trees, denoted as $\mathcal{T}^2$ (step I. c in Figure 2). In this tree structure, each intermediate node (neither a root nor a leaf) is constrained to have exactly two child nodes.

For the graph G_{zs}, we initialize a one-layer encoding tree T^0_{zs}, where each node α's parent is assigned as the root λ, denoted as $\alpha^- = \lambda$. Utilizing the stretch operator from HCSE algorithm (Pan et al., 2021), we engage in an iterative and greedy optimization of T^0_{zs}, as detailed in Appendix A.2. The optimal encoding tree T^*_{zs} for G_{zs} is obtained through:

$$T^*_{zs} = \min_{T \in \mathcal{T}^2} H^T(G_{zs}). \tag{6}$$

Based on above definition of $\mathcal{T}^2$, we derive the following proposition, the proof of which is detailed 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^*, there does not exist any non-root node α such that both v_i and v_j are included in its subset T_α.*

Each intermediate node $\alpha \in T^*_{zs}$ corresponds to a subset T^*_α comprising exactly one state-action vertex and one state vertex, representing the optimal one-to-one matching under the joint distribution of variables Z_t and S_t. We refer to the i-th intermediate node in T^*_{zs}, ordered from left to right, as α_i. The state-action and state vertices within subset $T^*_{\alpha_i}$ are labeled as z^i_t and s^i_t.

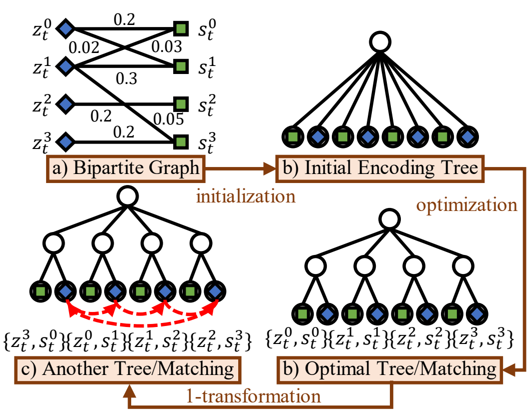


Figure 3: Illustration from joint distribution to one-to-one matching structure. a) Bipartite distribution graph, b) Initial encoding tree, c) Optimal encoding tree and one-to-one matching, d) Alternative encoding trees and one-to-one matching via transformation operations.

To illustrate the optimal matching's advantage, we introduce an l-transformation applied to T^*_{zs}. Given an integer parameter $l > 0$, this transformation forms a new 2-layer approximate binary tree T^l_{zs}, representing another one-to-one matching between Z_t and S_t.

Definition 3.2. For each intermediate node α_i in T^l_{zs}, the state-action and state vertices in $T^l_{\alpha_i}$ are specified as follows:

$$T^l_{\alpha_i} = \{z^{i'}_t, s^i_t\}, \quad i' = (i + l) \bmod n, \tag{7}$$

where n is the number of state-action pairs in Z_t, which is equal to the number of states in S_t.

To facilitate an intuitive understanding of the process described above, we present a clear example in Figure 3, which includes four states and four state-action pairs.

We now formally define the structural mutual information (step I. d in Figure 2), denoted as I^{SI}, which measures the average variation in entropy reductions caused by T^*_{zs} before and after applying above l-transformation.

Definition 3.3. The structural mutual information $I^{SI}(Z_t; S_t)$ between Z_t and S_t is defined as follows:

$$\begin{aligned}\Delta H(T_{zs}) &= H^{T^0_{zs}}(G_{zs}) - H^{T_{zs}}(G_{zs}),\\ I^{SI}(Z_t; S_t) &= \Delta H(T^*_{zs}) - \frac{1}{n} \cdot \sum_{l=0}^{n-1} \Delta H(T^l_{zs}).\end{aligned} \tag{8}$$

The detailed derivation is presented in Appendix C.1.

Intuitively, $I^{SI}(Z_t; S_t)$ reflects the advantage of the optimal one-to-one matching between variables Z_t and S_t under a specific joint distribution.

Connection with Traditional Mutual Information. Distinct from traditional mutual information, our structural mutual information extends to incorporate matching structures between variables, aligning with two representation objectives of the information bottleneck. Specifically, as the joint distribution of Z_t and S_t approaches a one-to-one distribution, the advantage of optimal matching structure, as represented by T^*_{zs}, becomes apparent, leading to a high value of $I^{SI}(Z_t; S_t)$. In contrast, when the joint distribution shifts towards uniform distribution, the advantage of T^*_{zs} diminish and the value of $I^{SI}(Z_t; S_t)$ becomes low. Furthermore, we introduce a theorem to elucidate the equivalence between traditional and structural mutual information in relation to these objectives.

Theorem 3.4. *For a joint distribution of variables X and Y that is either one-to-one or uniform, $I^{SI}(X; Y)$ equals $I(X; Y)$ when $n \to \infty$, where n is the number of possible values of the variable X.*

A detailed proof is provided in Appendix B.2.

Consequently, structural mutual information can be considered a reasonable and desirable learning objective for acquiring dynamics-relevant representations.

State-action Representation. Building upon the information bottleneck, we present an innovative embedding principle that aims to minimize the mutual information $I^{SI}(Z_t; S_t)$ while maximizing $I^{SI}(Z_t; S_{t+1})$, as shown in step I. d in Figure 2.

Due to the computational challenges of directly minimizing $I^{SI}(Z_t; S_t)$, we formulate a variational upper bound (see Appendix C.2). Noting that the term $\frac{1}{2}H(S_t) + \log 2$ 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:

$$I(Z_t; S_t) \leq \sum \left[p(z_t, s_t) \cdot D_{KL}(p(z_t|s_t)||q_m(z_t))\right] \triangleq L_{up}. \tag{9}$$

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:

$$H(Z_t|S_t) \leq \sum \left[p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)}\right] \triangleq L_{z|s}, \tag{10}$$

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 the lower bound of $\Delta H(T^*_{zs'})$ and minimize the upper bound of $\sum_{i=0}^{n-1} \Delta H(T^i_{zs'})$, detailed in Appendix C.2, as follows:

$$\Delta H(T^*_{zs'}) \geq \frac{1}{n} \cdot I(Z_t; S_{t+1}), \tag{11}$$

$$\sum_{i=0}^{n-1} \Delta H(T^i_{zs'}) \leq H(Z_t|S_{t+1}) + 2H(S_{t+1}). \tag{12}$$

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:

$$I(Z_t; S_{t+1}) \geq \sum \left[p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t)\right] \triangleq L_{s|z}, \tag{13}$$

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

Employing the decoder $q_{z|s}$, we calculate predictive objectives to maximize the lower bound of the conditional entropy terms in Equation 12 as follows:

$$\begin{aligned}H(Z_t|S_{t+1}) &= \sum \left[p(z_t, s_{t+1}) \cdot \log \frac{1}{q_{z|s}(z_t|s_{t+1})}\right] - D_{KL}(p||q_{z|s})\\ &\leq \sum \left[p(z_t, s_{t+1}) \cdot \log \frac{1}{q_{z|s}(z_t|s_{t+1})}\right] \triangleq L'_{z|s}.\end{aligned} \tag{14}$$

Within our SI2E framework, the definitive loss for representation learning is a combination of the above bounds:

$$L = (\frac{n-2}{2n} \cdot L_{up} + \frac{1}{2} L_{z|s}) + \eta(\frac{1}{n} \cdot L_{s|z} + \frac{1}{n} \cdot L'_{z|s}), \tag{15}$$

where η is a Lagrange multiplier modulating to maintain equilibrium among the specified terms.

3.2. Maximum Structural Entropy Exploration

To address the challenge of imbalance exploration towards low-value states in traditional entropy strategies, as discussed by (Kim et al., 2023), we design a unique intrinsic reward mechanism. This mechanism aims to facilitate uniform exploration among state-action pairs, regardless of their differing values. Specifically, we generate the hierarchical state-action structure based on agent policy and define the value-conditional structural entropy as intrinsic reward for policy training.

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 inherent in agent policy. Within this graph, each pair of state-action vertices v_i and v_j is connected by an undirected edge. The associated weight w_{ij} is determined:

$$w_{ij} = ||\pi(s_t^i, a_t^i) - \pi(s_t^j, a_t^j)||_2, \tag{16}$$

where 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 and vertices that share similar π values are grouped within the same sub-community.

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. A detailed proof is provided in Appendix B.3.

Proposition 3.5. *Given positive visitation probabilities $p(s_t^0, a_t^0), \dots, p(s_t^{n-1}, a_t^{n-1})$ of 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 3, denoted as $H^{T_{sa}^*}(G'_{sa})$ (step II. e in Figure 2).

Connection with Traditional Shannon Entropy. 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 of this connection is provided in Appendix B.4.

Theorem 3.6. *For a tuning parameter $0 \leq \epsilon \leq 1$, it holds for the structural entropy $H^{T_{sa}^*}(G'_{sa})$ and the Shannon entropy $H(S_t, A_t)$ that:*

$$\epsilon \cdot H(S_t, A_t) \leq H(V_0) - H(V_1) \leq H^{T_{sa}^*}(G'_{sa}) \leq H(S_t, A_t), \tag{17}$$

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 the 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 values, thus addressing the challenge of imbalance exploration. Through the identification of the hierarchical state-action structure influenced by policy values, the SI2E achieves an enhanced maximum coverage exploration, where ϵ represents the degree of enhancement. This approach effectively guarantees our framework's exploration advantage.

Estimation and Intrinsic Reward. Considering the impracticality of directly acquiring visitation probabilities, we employ the k-NN entropy estimator in Equation 2 to estimate the lower bound $H(V_0) - H(V_1)$ as follows:

$$H(V_0) - H(V_1) \approx \frac{d_z}{n_0} \cdot \sum_{i=0}^{n_0-1} \log d(v_i^0) - \frac{d_z}{n_1} \cdot \sum_{i=0}^{n_1-1} \log d(v_i^1) + C, \quad v_i^0 \in V_0, v_i^1 \in V_1, \tag{18}$$

where d_z denotes the dimension of state-action embedding, n_0 and n_1 denote the numbers of vertices in V_0 and V_1, and $d(v)$ denotes twice the distance from vertex v to its k-th nearest neighbor. By ignoring the constant term in Equation 18, 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 β is a positive hyperparameter that modulates the trade-off between exploration and exploitation. The pseudocode and complexity analysis of our framework are provided in Appendix A.

4. Experiments

In this section, we present a comprehensive suite of comparative experiments on navigation tasks from MiniGrid (Chevalier-Boisvert et al., 2018) and control tasks from the DeepMind Control Suite (DMControl) (Tunyasuvunakool et al., 2020) to evaluate the effectiveness of our exploration framework, SI2E, in terms of both final performance and sample efficiency. To showcase the generality of SI2E, we integrate it into various reinforcement learning algorithms, particularly Advantage Actor-Critic (A2C) (Mnih et al., 2016) and DrQv2 (Yarats et al., 2021).

4.1. MiniGrid Evaluation

Setup. Initially, we assess our framework on navigation tasks using the MiniGrid benchmark, which includes goal-

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

Navigation Task	LavaGapS7		SimpleCrossingS9N1		KeyCorridorS3R1	
	Success Rate (%)	Required Step (K)	Success Rate (%)	Required Step (K)	Success Rate (%)	Required Step (K)
A2C	84.02 ± 7.03	609.43 ± 58.81	88.18 ± 3.46	570.08 ± 15.87	87.31 ± 2.52	649.07 ± 31.87
A2C+SE	87.65 ± 8.51	661.21 ± 92.91	88.59 ± 4.62	391.83 ± 63.78	86.35 ± 3.60	440.99 ± 84.75
A2C+VCSE	93.13 ± 1.06	86.91 ± 1.41	91.25 ± 1.93	200.12 ± 21.69	89.88 ± 2.72	183.68 ± 7.68
A2C+SI2E	**93.85** ± 0.90	**63.36** ± 13.89	**93.25** ± 1.58	**139.13** ± 27.04	**93.75** ± 0.32	**128.98** ± 6.22
Abs.(%) Avg.	0.72(0.77) $\uparrow$	23.55(27.10) $\downarrow$	2.00(2.19) $\uparrow$	60.99(30.48) $\downarrow$	3.87(4.31) $\uparrow$	54.70(29.78) $\downarrow$
Navigation Task	DoorKey-6x6		DoorKey-8x8		Unlock	
	Success Rate (%)	Required Step (K)	Success Rate (%)	Required Step (K)	Success Rate (%)	Required Step (K)
A2C	92.67 ± 8.47	566.88 ± 96.25	1.57 ± 1.05	–	92.48 ± 11.96	667.73 ± 155.0
A2C+SE	93.18 ± 6.81	469.18 ± 87.81	72.60 ± 20.32	–	91.34 ± 18.37	630.74 ± 242.25
A2C+VCSE	95.55 ± 2.42	335.21 ± 21.25	95.81 ± 11.20	1854.74 ± 395.06	94.59 ± 3.40	399.46 ± 48.89
A2C+SI2E	**96.19** ± 1.37	**230.60** ± 19.85	**96.61** ± 2.82	**1094.16** ± 128.40	**95.14** ± 3.16	**306.41** ± 52.13
Abs.(%) Avg.	0.64(0.67) $\uparrow$	104.61(31.21) $\downarrow$	0.80(0.83) $\uparrow$	760.58(41.01) $\downarrow$	0.55(0.58) $\uparrow$	93.05(23.29) $\downarrow$

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 with Shannon entropy (SE) and value-based state entropy (VCSE) as our baselines. Consistent with the original implementations, we set $k = 5$ for all exploration methods. For the A2C+SI2E implementation, we employ a randomly initialized encoder, which is then optimized by minimizing the loss in Equation 15.

Results and Analysis. Table 1 displays the average values and standard deviations of success rates and environmental steps required to attain specified rewards in various navigation tasks. Consistent with previous work (Zeng et al., 2023b), these target rewards are set as 0.9 times the convergence reward of SI2E for each task, serving as a benchmark for assessing sample efficiency. The tasks encompass navigation with obstacles (LavaGapS7 and SimpleCrossingS9N1), long-horizon navigation (DoorKey and Unlock), and long-horizon navigation with obstacles (KeyCorridorS3R1). SI2E consistently exhibits improved sample efficiency across tasks, with a minimum enhancement of 23.29%. Notably, in navigation scenarios with obstacles where baseline performances are inadequate, SI2E secures a 2.42% average increment in success rate. This advantage highlight SI2E's exceptional final performance. The learning curves are delineated in the Appendix E.1.

4.2. DeepMind Control Suite Evaluation

Setup. Subsequently, we evaluate our framework across a range of continuous control tasks within the DMControl suite. To ensure a challenging evaluation, we choose the model-free RL algorithm DrQv2, which operates on pixel-based observations, as the foundational agent. For a more comprehensive comparison, we incorporate a state-action exploration baseline, MADE (Zhang et al., 2021b). The value of k is set to 12 for both SI2E and the comparative baselines. For the implementation of DrQv2+SI2E, the original ICM module (Pathak et al., 2017) is replaced by representation learning guided by the structural mutual information principle.

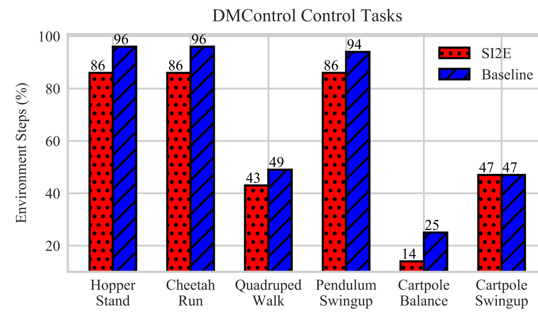


Figure 4: Comparison of sample efficiency between SI2E and the best-performing baseline in DMControl, focusing on the required environmental steps to reach the reward target, expressed as a proportion of the total $250K$ steps.

Results and Analysis. We conduct evaluations of all exploration methods across six control tasks within the DMControl suite, documenting the episode rewards in Table 2. Observations reveal that SI2E significantly 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. Detailed learning curves for all control tasks are available in Appendix E.2. 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 4, SISA demonstrates an average improvement of 11.06% in sample efficiency. This improvement is reflected in a reduction in the required environmental steps, decreasing from 67.83% to 60.33% of the total steps required to achieve the reward target. These results imply that SI2E effectively generates dynamics-relevant state-action representations, thereby motivating the agent to thoroughly explore the state-action space by leveraging the inherent structural information.

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, <u>underline</u>: the second performance.

Domain, Task	Hopper Stand	Cheetah Run	Quadruped Walk	Pendulum Swingup	Cartpole Balance	Cartpole Swingup
DrQv2	87.59 ± 11.70	229.28 ± 123.93	289.79 ± 24.17	424.21 ± 246.96	998.97 ± 22.95	–
DrQv2+SE	313.39 ± 94.15	228.82 ± 126.21	<u>290.27</u> ± 24.20	10.80 ± 2.92	993.80 ± 75.24	219.69 ± 62.21
DrQv2+VCSE	711.32 ± 30.84	<u>456.26</u> ± 22.20	243.74 ± 29.91	<u>824.17</u> ± 99.59	<u>998.65</u> ± 9.58	<u>707.76</u> ± 50.38
DrQv2+MADE	<u>717.09</u> ± 112.94	366.59 ± 53.74	262.63 ± 23.92	672.11 ± 34.63	996.16 ± 40.60	704.18 ± 41.75
DrQv2+SI2E (Ours)	**797.17** ± 53.21	**464.08** ± 29.32	**399.51** ± 29.05	**885.50** ± 38.28	**999.58** ± 2.97	**795.09** ± 90.49
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)

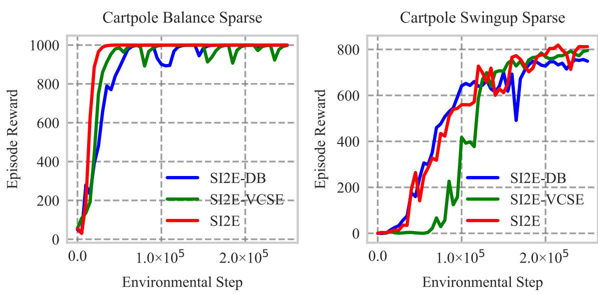


Figure 5: Learning curves across two DMControl for ablation studies.

4.3. Ablation Studies

To further investigate the impact of two critical components within the SI2E framework, structural mutual information and value-conditional structural entropy, we perform ablation studies on sparse-reward tasks in DMControl, focusing on two distinct variants: (i) SI2E-DB, which utilizes the DB bottleneck for learning state-action representations, and (ii) SI2E-VCSE, employing the state-of-the-art VCSE approach for calculating intrinsic rewards. As depicted in Figure 5, SI2E significantly surpasses its variants in terms of final performance and sample efficiency. This outcome underscores the essential role of these critical components in conferring SI2E's superior capabilities. Additional ablation studies are available in Appendix E.3.

5. Related Work

5.1. Maximum Entropy Exploration

Maximum entropy exploration in RL has evolved from initially focusing on unsupervised methods to more advanced supervised models incorporating task rewards. In the unsupervised paradigm, agents autonomously acquire behaviors by using state entropy as the intrinsic reward for exploration, independent of task rewards (Liu & Abbeel, 2021; Mutti et al., 2022; Yang & Spaan, 2023). 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 mitigate this issue, a value-based approach (Kim et al., 2023) is proposed, which integrates value estimates into the entropy calculation to ensure a balanced exploration.

5.2. Information Bottleneck Principle

In RL, 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' limitation is 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.

5.3. Structural Information Principles

Since the introduction of structural information principles (Li & Pan, 2016), these concepts have brought significant changes in the analysis of network complexities, utilizing metrics like structural entropy and partitioning trees. This innovative approach has not only deepened the understanding of network dynamics but has also sparked a wide range of applications. In the RL domain, these principles have played a crucial role in defining hierarchical action and state abstractions within encoding trees (Zeng et al., 2023a;b), representing a substantial advancement in the development of robust decision-making frameworks.

6. 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 and maximizes the structural entropy to enhance coverage across the state-action space. We have established theoretical connections between SI2E and traditional information-theoretic methodologies, highlighting the framework's rationality. Through extensive and comparative evaluations, SI2E exhibits significant improvements in terms of final performance and sample efficiency over state-of-the-art exploration methods. Looking ahead, our plan 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.

Impact Statement

This paper contributes to advancing the field of Machine Learning. While there are numerous potential societal implications of our research, we believe none require specific emphasis in this context.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. Agent57: Outperforming the atari human benchmark. In *International conference on machine learning*, pp. 507–517. PMLR, 2020.

Bai, C., Wang, L., Han, L., Garg, A., Hao, J., Liu, P., and Wang, Z. Dynamic bottleneck for robust self-supervised exploration. *Advances in Neural Information Processing Systems*, 34:17007–17020, 2021.

Bellman, R. A markovian decision process. *Journal of Mathematics and Mechanics*, pp. 679–684, 1957.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic gridworld environment for openai gym. 2018.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Islam, R., Seraj, R., Bacon, P.-L., and Precup, D. Entropy regularization with discounted future state distribution in policy gradient methods. *ArXiv*, 2019.

Kim, D., Shin, J., Abbeel, P., and Seo, Y. Accelerating reinforcement learning with value-conditional state entropy exploration. *arXiv preprint arXiv:2305.19476*, 2023.

Kim, H., Kim, J., Jeong, Y., Levine, S., and Song, H. O. Emi: Exploration with mutual information. In *International Conference on Machine Learning*, pp. 3360–3369. PMLR, 2019a.

Kim, Y., Nam, W., Kim, H., Kim, J.-H., and Kim, G. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International conference on machine learning*, pp. 3379–3388. PMLR, 2019b.

Li, A. and Pan, Y. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62:3290–3339, 2016.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Mutti, M., De Santi, R., and Restelli, M. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, pp. 16223–16239. PMLR, 2022.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Pan, Y., Zheng, F., and Fan, B. An information-theoretic perspective of hierarchical clustering. *arXiv preprint arXiv:2108.06036*, 2021.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Pérez-Gil, Ó., Barea, R., López-Guillén, E., Bergasa, L. M., Gomez-Huelamo, C., Gutiérrez, R., and Diaz-Diaz, A. Deep reinforcement learning based control for autonomous vehicles in carla. *Multimedia Tools and Applications*, 81(3):3553–3576, 2022.

Prathiba, S. B., Raja, G., Dev, K., Kumar, N., and Guizani, M. A hybrid deep reinforcement learning for autonomous vehicles smart-platooning. *IEEE Transactions on Vehicular Technology*, 70(12):13340–13350, 2021.

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., and Lee, K. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pp. 9443–9454. PMLR, 2021.

Shannon, C. The lattice theory of information. *Transactions of the IRE professional Group on Information Theory*, 1 (1):105–107, 1953.

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23 (3-4):301–321, 2003.

Tao, R. Y., François-Lavet, V., and Pineau, J. Novelty search in representational space for sample efficient exploration. *Advances in Neural Information Processing Systems*, 33: 8114–8126, 2020.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., Lillicrap, T., Heess, N., and Tassa, Y. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., and etc. Alphastar: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Yang, Q. and Spaan, M. T. Cem: Constrained entropy maximization for task-agnostic safe exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10798–10806, 2023.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021.

Yuan, M., Pun, M.-O., and Wang, D. Rényi state entropy maximization for exploration acceleration in reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 2022.

Zeng, X., Peng, H., and Li, A. Effective and stable role-based multi-agent collaboration by structural information principles. *Proceedings of the AAAI Conference on Artificial Intelligence*, (10):11772–11780, Jun. 2023a.

Zeng, X., Peng, H., Li, A., Liu, C., He, L., and Yu, P. S. Hierarchical state abstraction based on structural information principles. In *IJCAI*, pp. 4549–4557, 2023b.

Zhang, C., Cai, Y., Huang, L., and Li, J. Exploration by maximizing rényi entropy for reward-free rl framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10859–10867, 2021a.

Zhang, T., Rashidinejad, P., Jiao, J., Tian, Y., Gonzalez, J. E., and Russell, S. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34:9663–9680, 2021b.

A. Framework Details

A.1. Notations

Table 3: Glossary of Notations.

Notation	Description
$X; Y$	Random variables
$x; y$	Variable values
p	Probability
$H; I$	Shannon entropy; Mutual information
$\mathcal{O}; \mathcal{A}$	Observation space; Action space
$O; S; A$	Observation, state, action variables
$o; s; a$	Single observation, state, action
$z; Z$	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
$v; V$	Vertex; Vertex set
$e; E; w$	Edge; Edge set; Edge weight
$\alpha; T$	Tree node; Encoding tree
$\lambda; \nu$	Root node; Leaf node
$H^T; H^K; \Delta H$	Structural entropy; Entropy reduction
$\mathcal{T}$	Approximate binary trees
I^{SI}	Structural mutual information

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

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:     ***# ΔH is the entropy reduction caused by one stretch operation***
5:     $(\alpha_i^*, \alpha_j^*) \leftarrow \arg\max\{\Delta H\}$
6:     **if** $\Delta H = 0$ **then**
7:         Break
8:     **end if**
9:     Create a new tree node α'
10:     $\alpha' \leftarrow (\alpha_i^*)^-, (\alpha_i^*)^- \leftarrow \alpha', (\alpha_j^*)^- \leftarrow \alpha'$
11: **end while**

A.3. The Pseudocode of SI2E

Algorithm 2 Effective Exploration based on Structural Information Principles

```
1: Input: batch size n, update interval t_up
2: Initialize: policy π, encoder functions f_s and f_z, decoder functions q_m, q_{z|s}, q_{s|z}, replay buffer B
3: for each episode do
4:     for each environmental step t do
5:         Collect transition τ_t = (s_t, a_t, s_{t+1}, r_t^e) using the encoder f_s and policy π
6:         Sample a batch {τ_{t_i}}_{i=1}^n from 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 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 + β · r_t^i
12:        Update τ'_t = (s_t, a_t, s_{t+1}, r_t) and augment B with τ'_t
13:        if t mod t_up = 0 then
14:            # Structural Mutual Information Principle
15:            Compute representation losses L_up, L_{z|s}, L_{s|z}, and L'_{z|s}
16:            Update encoder and decoder functions to minimize the combined loss L as defined in Equation 15
17:            Update agent policy π using B
18:        end if
19:    end for
20: end for
```

A.4. Time Complexity of SI2E

Within the SI2E framework, we analyze the time complexities of critical components independent of the underling RL algorithm. During the state-action representation phase (lines 6 to 11 in Algorithm 2), 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 (lines 12 to 20 in Algorithm 2), the generation of hierarchical community structure incurs a $O(n \cdot \log^2 n)$ complexity, the construction of distribution graph leads to a complexity of $O(n^2)$, and value-conditional structural entropy is calculated with $O(n)$ time complexity.

B. Theorem Proofs

B.1. Proof of Proposition 3.1

Proof. For any two vertices $v_i \in V$ and $v_j \in V$, their corresponding tree nodes are denoted as α_i and α_j. These nodes' parents are initially assigned as the root node λ. Before executing one stretch operation on α_i and α_j, their structural entropies are defined as follows:

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

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

$$H(G;\alpha_i) = -\frac{d_i}{\mathrm{vol}(G)} \cdot \log \frac{d_i}{\mathrm{vol}(\alpha')}, \quad H(G;\alpha_j) = -\frac{d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_j}{\mathrm{vol}(\alpha')}, \tag{20}$$

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

$$g_{\alpha'} = d_i + d_j, \quad \mathrm{vol}(\alpha') = d_i + d_j. \tag{21}$$

The structural entropy of α' can be determined as:

$$H(G;\alpha') = -\frac{g_{\alpha'}}{\mathrm{vol}(G)} \cdot \log \frac{\mathrm{vol}(\alpha')}{\mathrm{vol}(G)} = -\frac{d_i + d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_i + d_j}{\mathrm{vol}(G)}. \tag{22}$$

The entropy reduction ΔH, consequent to the stretch operation on vertices v_i and v_j, is calculated to be zero:

$$\begin{aligned}
\Delta H &= \left[-\frac{d_i}{\mathrm{vol}(G)} \cdot \log \frac{d_i}{\mathrm{vol}(G)} - \frac{d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_j}{\mathrm{vol}(G)}\right] - \left[-\frac{d_i}{\mathrm{vol}(G)} \cdot \log \frac{d_i}{d_i + d_j} - \frac{d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_j}{d_i + d_j} - \frac{d_i + d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_i + d_j}{\mathrm{vol}(G)}\right] \\
&= \left[-\frac{d_i}{\mathrm{vol}(G)} \cdot \log \frac{d_i}{\mathrm{vol}(G)} - \frac{d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_j}{\mathrm{vol}(G)}\right] - \left[-\frac{d_i}{\mathrm{vol}(G)} \cdot \log \frac{d_i}{\mathrm{vol}(G)} - \frac{d_j}{\mathrm{vol}(G)} \cdot \log \frac{d_j}{\mathrm{vol}(G)}\right] \\
&= 0.
\end{aligned} \tag{23}$$

Given the zero change in entropy, as per line 5 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. □

B.2. Proof of Theorem 3.4

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

$$p(x_i, y_j) = \begin{cases} p(x_i) = p(y_j) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

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

$$\begin{aligned}
I^{SI}(X;Y) &= \Delta H(T^*_{xy}) - \frac{1}{n} \cdot \sum_{l=0}^{n-1} \Delta H(T^l_{xy}) \\
&= \sum_i \left[p(x_i, y_i) \cdot \log \frac{2}{p(x_i) + p(y_i)}\right] - \frac{1}{n} \cdot \sum_{i,j} \left[p(x_i, y_j) \cdot \log \frac{2}{p(x_i) + p(y_j)}\right] \\
&= \sum_i \left[p(x_i) \cdot \log \frac{1}{p(x_i)}\right] - \frac{1}{n} \cdot \sum_i \left[p(x_i) \cdot \log \frac{1}{p(x_i)}\right] \\
&= H(X) - \frac{1}{n} \cdot H(X) \\
&= (1 - \frac{1}{n}) \cdot H(X).
\end{aligned} \tag{25}$$

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

$$\begin{aligned} I(X;Y) &= \sum_{i,j}\left[p(x_i,y_j)\cdot\log\frac{p(x_i,y_j)}{p(x_i)\cdot p(y_j)}\right] \\ &= \sum_{i}\left[p(x_i)\cdot\log\frac{1}{p(x_i)}\right] \\ &= H(X). \end{aligned} \tag{26}$$

Consequently,

$$\lim_{n\to\infty}\frac{I^{SI}(X;Y)}{I(X;Y)} = \lim_{n\to\infty}(1-\frac{1}{n}) = 1. \tag{27}$$

Hence, $I^{SI}(Z_t;S_t)$ equals $I(Z_t;S_t)$ when $n\to\infty$.

Assuming a *uniform distribution* for variables X and Y, the joint probability of a tuple (x_i,y_j) is given by:

$$p(x_i,y_j) = \frac{1}{n^2}, \quad \forall i,j. \tag{28}$$

Subsequently, $I^{SI}(X;Y)$ and $I(X;Y)$ are determined as follows:

$$\begin{aligned} I^{SI}(X;Y) &= n\cdot\frac{1}{n^2}\cdot\log\frac{2}{\frac{1}{n}+\frac{1}{n}} - \frac{1}{n}\cdot n^2\cdot\frac{1}{n^2}\log\frac{2}{\frac{1}{n}+\frac{1}{n}} = 0, \\ I(Z_t;S_t) &= n^2\cdot\frac{1}{n^2}\cdot\log\frac{\frac{1}{n^2}}{\frac{1}{n}\cdot\frac{1}{n}} = 0. \end{aligned} \tag{29}$$

Hence, $I^{SI}(Z_t;S_t)$ equals $I(Z_t;S_t)$. □

B.3. Proof of Proposition 3.5

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 $p_0\le p_1$. 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 3.5.
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 $(p'_0,\ldots,p'_{k-1})$ with $\sum_{i=0}^{k-1}p'_i=1$.
- Modify the weight of all edges connected to each vertex v_i ($0\le i\le k$) by a factor $\frac{(1-p_n)p_i}{p'_i}$.
- For each vertex v_i, create an edge with weight p_ip_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. □

B.4. Proof of Theorem 3.6

In the tree T^*_{sa}, we denote the i-th intermediate node as α_i and its j-th child node as α_{ij}. The single state-action vertex in the corresponding subset of α_{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:

$$\mathrm{vol}(G) = \sum_{i,j}p(s_{ij},a_{ij}) = 1, \quad \mathrm{vol}(\alpha_{ij}) = g_{\alpha_{ij}} = p(s_{ij},a_{ij}). \tag{30}$$

The 2-dimensional value-conditional structural entropy $H^{T^*_{sa}}(G'_{sa})$ is calculated through the following expressions:

$$
\begin{aligned}
H^{T^*_{sa}}(G'_{sa}) &= -\sum_i \left[g_{\alpha_i} \cdot \log \mathrm{vol}(\alpha_i) + \sum_j \left[g_{\alpha_{ij}} \cdot \log \frac{\mathrm{vol}(\alpha_{ij})}{\mathrm{vol}(\alpha_i)} \right] \right] \\
&= -\sum_i \left[g_{\alpha_i} \cdot \log \mathrm{vol}(\alpha_i) + \sum_j \left[p(s_{ij}, a_{ij}) \cdot \log \frac{p(s_{ij}, a_{ij})}{\mathrm{vol}(\alpha_i)} \right] \right] \\
&= \sum_{i,j} \left[p(s_{ij}, a_{ij}) \cdot \log \frac{1}{p(s_{ij}, a_{ij})} \right] - \sum_i \left[\mathrm{vol}(\alpha_i) \cdot \log \frac{1}{\mathrm{vol}(\alpha_i)} \right] + \sum_i \left[g_{\alpha_i} \cdot \log \frac{1}{\mathrm{vol}(\alpha_i)} \right] \\
&= H(V_0) - H(V_1) + \sum_i \left[g_{\alpha_i} \cdot \log \frac{1}{\mathrm{vol}(\alpha_i)} \right] \\
&\geq H(V_0) - H(V_1).
\end{aligned} \tag{31}
$$

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

$$
0 \leq \log_{p(s_{ij}, a_{ij})} \frac{p(s_{ij}, a_{ij})}{\mathrm{vol}(\alpha_i)} \leq 1. \tag{32}
$$

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

$$
\mathrm{vol}(\alpha_i) \cdot \log \frac{1}{\mathrm{vol}(\alpha_i)} = \sum_j \left[(1 - \epsilon_{ij}) \cdot p(s_{ij}, a_{ij}) \cdot \log \frac{1}{p(s_{ij}, a_{ij})} \right]. \tag{33}
$$

Selecting the minimal ϵ-value as ϵ^* allow us to reformulate Equation 31 as follows:

$$
H(V_1) = \sum_i \left[(\mathrm{vol}(\alpha_i)) \cdot \log \frac{1}{\mathrm{vol}(\alpha_i)} \right] \leq (1 - \epsilon^*) \cdot H(S_t, A_t), \tag{34}
$$

$$
H(V_0) = H(S_t, A_t), \tag{35}
$$

$$
\epsilon^* \cdot H(S_t, A_t) \leq H(V_0) - H(V_1) \leq H^{T^*_{sa}}(G'_{sa}) \leq H(S_t, A_t). \tag{36}
$$

C. Detailed Derivations

C.1. Derivation of $I^{SI}(Z_t; S_t)$

The structural entropy of G_{zs} under T_{zs}^0 is calculated as follows:

$$H^{T_{zs}^0}(G_{zs}) = -\sum_{\alpha \in T_{zs}^0, \alpha \neq \lambda} \left[\frac{g_\alpha}{\mathrm{vol}(G_{zs})} \cdot \log \frac{\mathrm{vol}(\alpha)}{\mathrm{vol}(G_{zs})} \right] = -\sum_{z_t \in Z_t} \frac{p(z_t)}{2} \cdot \log \frac{p(z_t)}{2} - \sum_{s_t \in S_t} \frac{p(s_t)}{2} \cdot \log \frac{p(s_t)}{2}, \tag{37}$$

where the degree sum of all vertices in G_{zs} equals 2, $\mathrm{vol}(G_{zs}) = 2$. For each node $\alpha \in T_{zs}^0$ whose corresponding subset is $\{s_t\}$ or $\{z_t\}$, it holds that $g_\alpha = \mathrm{vol}(\alpha) = p(s_t)$ or $g_\alpha = \mathrm{vol}(\alpha) = p(z_t)$.

For each node $\alpha_i \in T_{zs}^*$ with subset $\{z_t^i, s_t^i\}$, the entropy sum of this node and its children is calculated as follows:

$$-\frac{g_{\alpha_i}}{2} \cdot \log \frac{\mathrm{vol}(\alpha_i)}{2} - \frac{p(z_t^i)}{2} \cdot \log \frac{p(z_t^i)}{\mathrm{vol}(\alpha_i)} - \frac{p(s_t^i)}{2} \cdot \log \frac{p(s_t^i)}{\mathrm{vol}(\alpha_i)}. \tag{38}$$

The reduction in structural entropy $\Delta H(T_{zs}^*)$ achieved through the optimal encoding tree T_{zs}^* is determined by:

$$\begin{aligned}
\Delta H(T_{zs}^*) &= H^{T_{zs}^0}(G_{zs}) - H^{T_{zs}^*}(G_{zs}) \\
&= \sum_i \left[-\frac{p(z_t^i)}{2} \cdot \log \frac{p(z_t^i)}{2} - \frac{p(s_t^i)}{2} \cdot \log \frac{p(s_t^i)}{2} \right] - \sum_i \left[-\frac{g_{\alpha_i}}{2} \cdot \log \frac{\mathrm{vol}(\alpha_i)}{2} - \frac{p(z_t^i)}{2} \cdot \log \frac{p(z_t^i)}{\mathrm{vol}(\alpha_i)} - \frac{p(s_t^i)}{2} \cdot \log \frac{p(s_t^i)}{\mathrm{vol}(\alpha_i)} \right].
\end{aligned} \tag{39}$$

Proposition 3.1 ensures that:

$$\mathrm{vol}(\alpha_i) = p(z_t^i) + p(s_t^i), \quad g_{\alpha_i} = p(z_t^i) + p(s_t^i) - 2p(z_t^i, s_t^i). \tag{40}$$

Consequently, we can reformulate the entropy reduction $\Delta H(T_{xy}^*)$ as follows:

$$\begin{aligned}
\Delta H(T_{zs}^*) &= \sum_i \left[\frac{p(z_t^i)}{2} \cdot \log \frac{2}{p(z_t^i) + p(s_t^i)} + \frac{p(s_t^i)}{2} \cdot \log \frac{2}{p(z_t^i) + p(s_t^i)} - \frac{p(z_t^i) + p(s_t^i) - 2 \cdot p(z_t^i, s_t^i)}{2} \cdot \log \frac{2}{p(z_t^i) + p(s_t^i)} \right] \\
&= \sum_i \left[p(z_t^i, s_t^i) \cdot \log \frac{2}{p(z_t^i) + p(s_t^i)} \right].
\end{aligned} \tag{41}$$

For any integer $l > 0$, the entropy reduction $\Delta H(T_{zs}^l)$ induced by T_{zs}^l is given by:

$$\Delta H(T_{zs}^l) = \sum_i \left[p(z_t^{i'}, s_t^i) \cdot \log \frac{2}{p(z_t^{i'}) + p(s_t^i)} \right], \quad i' = (i + l) \bmod n. \tag{42}$$

Consequently,

$$\sum_{l=0}^{n} \Delta H(T_{zs}^l) = \sum_{i,j} \left[p(z_t^i, s_t^j) \cdot \log \frac{2}{p(z_t^i) + p(s_t^j)} \right]. \tag{43}$$

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

Non-negativity of structural entropy $H^{T_{zs}^*}(G_{zs})$ assures the following inequality:

$$\begin{aligned}
\Delta H(T_{zs}^*) &= H^{T_{zs}^0}(G_{zs}) - H^{T_{zs}^*}(G_{zs}) \\
&\leq H^{T_{zs}^0}(G_{zs}) \\
&= \sum_i \left[-\frac{p(z_t^i)}{2} \cdot \log \frac{p(z_t^i)}{2} - \frac{p(s_t^i)}{2} \cdot \log \frac{p(s_t^i)}{2} \right] \\
&= \frac{1}{2} H(Z_t) + \frac{1}{2} H(S_t) + \frac{1}{2} \log 2 \cdot \sum_i p(s_t^i + z_t^i) \\
&= \frac{1}{2} I(Z_t; S_t) + \frac{1}{2} H(Z_t, S_t) + \log 2 \\
&= \frac{1}{2} I(Z_t; S_t) + \frac{1}{2} H(Z_t | S_t) + \frac{1}{2} H(S_t) + \log 2.
\end{aligned} \tag{44}$$

The established optimality of T^*_{zs} (see Equation 6) confirms that:

$$\begin{aligned} H(T^*_{zs}) &\leq H(T^l_{zs}), \quad \forall l > 0, \\ \Delta H(T^*_{zs}) &\geq \Delta H(T^l_{zs}), \quad \forall l > 0. \end{aligned} \tag{45}$$

This leads us to the conclusion that:

$$\frac{1}{n} \cdot \sum_{i=0}^{n} \Delta H(T^i_{zs}) \leq \Delta H(T^*_{zs}) \leq \frac{1}{2} I(Z_t; S_t) + \frac{1}{2} H(Z_t|S_t) + \frac{1}{2} H(S_t) + \log 2. \tag{46}$$

Subsequently,

$$\begin{aligned} \sum_{i=0}^{n} \Delta H(T^i_{zs}) - I(Z_t; S_t) &= \sum_{i,j} \left[p(z^i_t, s^j_t) \cdot \log \frac{2}{p(z^i_t) + p(s^j_t)} \right] - \sum_{i,j} \left[p(z^i_t, s^j_t) \cdot \log \frac{p(z^i_t, s^j_t)}{p(z^i_t) \cdot p(s^j_t)} \right] \\ &= \sum_{i,j} \left[p(z^i_t, s^j_t) \cdot \log \frac{2}{p(z^i_t, s^j_t)} \right] + \sum_{i,j} \left[p(z^i_t, s^j_t) \cdot \log \frac{1}{\frac{1}{p(z^i_t)} + \frac{1}{p(s^j_t)}} \right]. \end{aligned} \tag{47}$$

Given that $p(z^i_t, s^j_t) \leq p(z^i_t) \leq 1$ and $p(z^i_t, s^j_t) \leq p(s^j_t) \leq 1$, this inequalities hold that:

$$\begin{aligned} 2 \leq \frac{1}{p(z^i_t)} + \frac{1}{p(s^j_t)} &\leq \frac{2}{p(z^i_t, s^j_t)}, \\ 0 \leq \sum_{i=0}^{n} \Delta H(T^i_{zs}) - I(Z_t; S_t) &\leq H(Z_t, S_t), \end{aligned} \tag{48}$$

$$I(Z_t; S_t) \leq \sum_{i=0}^{n} \Delta H(T^i_{zs}) \leq I(Z_t; S_t) + H(Z_t, S_t) = H(Z_t) + H(S_t) \leq H(Z_t|S_t) + 2H(S_t). \tag{49}$$

By integrating the results of Equation 46 and Equation 49, we obtain that:

$$0 \leq I^{SI}(Z_t; S_t) \leq \frac{n-2}{2n} \cdot I(Z_t; S_t) + \frac{1}{2} H(Z_t|S_t) + \frac{1}{2} H(S_t) + \log 2. \tag{50}$$

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:

$$\begin{aligned} I(Z_t; S_t) &= \sum \left[p(z_t, s_t) \cdot \log \frac{p(z_t|s_t)}{p(z_t)} \right] \\ &= \sum \left[p(z_t, s_t) \cdot \log \frac{p(z_t|s_t)}{q_m(z_t)} \right] - D_{KL}(p||q_m) \\ &\leq \sum \left[p(z_t, s_t) \cdot D_{KL}(p(z_t|s_t)||q_m(z_t)) \right]. \end{aligned} \tag{51}$$

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:

$$\begin{aligned} H(Z_t|S_t) &= \sum \left[p(z_t, s_t) \cdot \log \frac{1}{p(z_t|s_t)} \right] \\ &= \sum \left[p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)} \right] - D_{KL}(p||q_{z|s}) \\ &\leq \sum \left[p(z_t, s_t) \cdot \log \frac{1}{q_{z|s}(z_t|s_t)} \right]. \end{aligned} \tag{52}$$

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 $I(Z_t; S_{t+1})$:

$$
\begin{aligned}
&I(Z_t; S_{t+1}) = \sum \left[p(z_t, s_{t+1}) \cdot \log \frac{p(s_{t+1}|z_t)}{p(s_{t+1})} \right] \\
&= \sum \left[p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t) \right] + H(S_{t+1}) + D_{KL}(p||q_{s|z}) \\
&\geq \sum \left[p(z_t, s_{t+1}) \cdot \log q_{s|z}(s_{t+1}|z_t) \right].
\end{aligned} \tag{53}
$$

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 number of environmental steps was set to $300K/100K$ for the MiniGrid 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[2], adhering to the pre-established hyperparameters set, except where explicitly noted. Consistent with the original methodology, state representations are derived using a fixed encoder that is randomly initialized, and intrinsic rewards are normalized based on the standard deviation computed from sample data. 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 β	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[3] (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, 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 β	0.1

[2] https://github.com/younggyoseo/RE3

[3] https://github.com/facebookresearch/drqv2

D.2. Environment Details

MiniGrid Experiments. In our MiniGrid benchmark experiments, we encompass six navigation tasks, including LavaGapS7, 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.

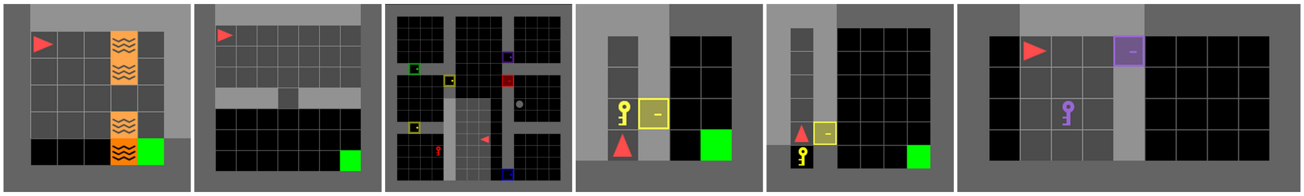


Figure 6: Examples of navigation tasks used in our MiniGrid experiments include: (a) LavaGapS7, (b) SimpleCorssingS9N1, (c) KeyCorridor, (d) DoorKey-6x6, (e) DoorKey-8x8, (f) Unlock.

DMControl Experiments. Our research in DMControl suite focues on six continuous control tasks, specifically Hopper Stand, Cheetah Run, Quadruped Walk, Pendulum Swingup, Cartpole Balance Sparse, and Cartpole Swingup Sparse. And visualizations of thes tasks are provided in Figure 7.

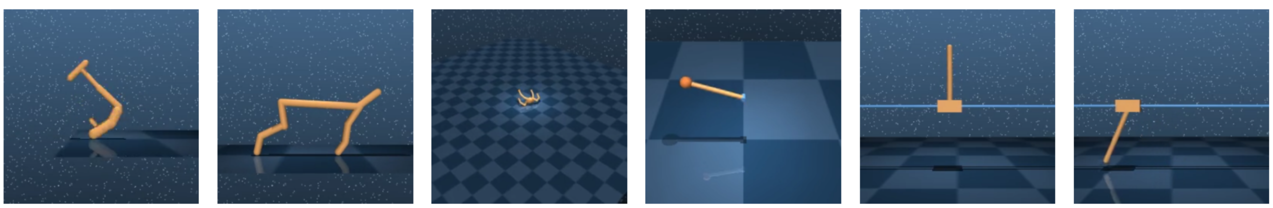


Figure 7: Examples of control tasks used in our DeepMind Control Suite experiments include: (a) Hopper Stand, (b) Cheetah Run, (c) Quadruped Walk, (d) Pendulum Swingup, (e) Cartpole Balance Sparse, (f) Cartpole Swingup Sparse.

E. Additional Experiments

E.1. Experiments on MiniGrid Benchmark

Figure 8 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. Specifically, on the LavaGapS7, SimpleCrossingS9N1, and KeyCorridorS3R1 tasks, A2C+SI2E achieves an average success rate of 93.62%, marking an improvement of $2.20(2.42\%)$ in final performance. In terms of sample efficiency, on the DoorKey-6x6, DoorKey-8x8, and Unlock tasks, SI2E converges before 50% of the total environmental steps are completed. This indicates SI2E's effectiveness in enhancing the agent's exploration of the state-action space, surpassing the baseline methods.

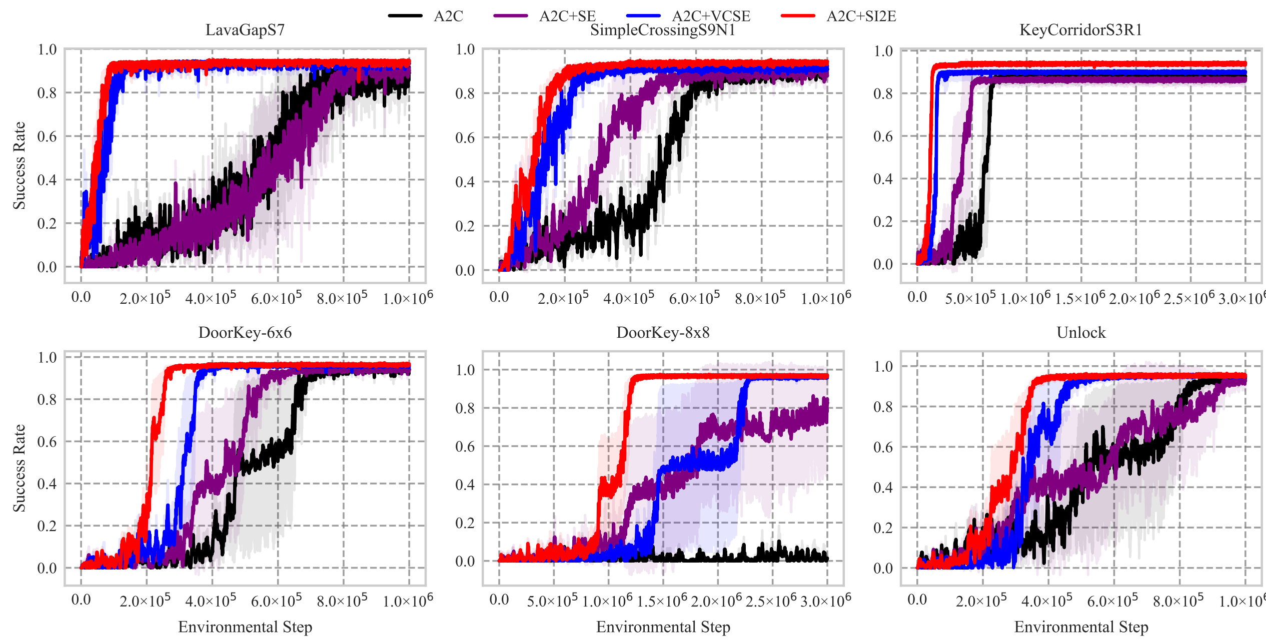


Figure 8: Learning curves for six navigation tasks in MiniGrid, measured in terms of success rate. The solid lines represent the interquartile mean, while the shaded regions indicate the standard deviation, both calculated across 10 runs.

E.2. Experiments on DMControl Suite

Figure 9 shows the learning curves for the DrQv2 algorithm when integrated with our SI2E framework and other exploration baselines, SE and VCSE. The variants are identified as DrQv2, DrQv2+SE, DrQv2+VCSE, 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.

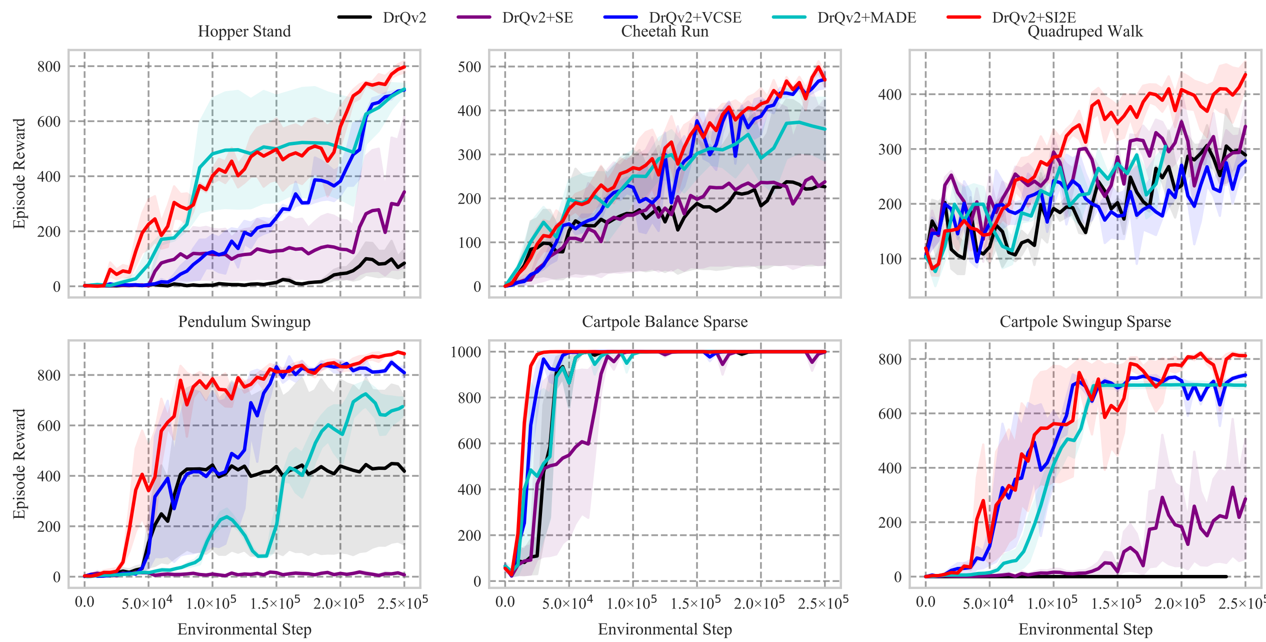
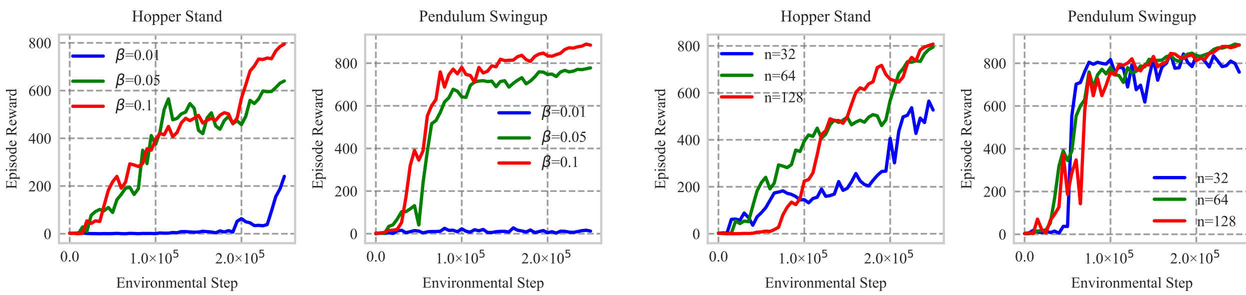


Figure 9: Learning curves for six continuous control tasks from DMControl Suite, measured in terms of episode reward. The solid lines represent the interquartile mean, while the shaded regions indicate the standard deviation, both calculated across 10 runs.

E.3. Ablation Studies

To investigate the influence of parameters β 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 10(a) illustrates that an increase in parameter β consistently enhances performance across both tasks, substantiating the effectiveness of our exploration method. Conversely, Figure 10(b) 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.



(a) Effect of scale parameter β (b) Effect of batch size n

Figure 10: Learning curves of SI2E with varied β and n values on Hopper Stand and Pendulum Swingup tasks. (a) shows the effect of the scale parameter β 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.