Hierarchical State Abstraction Based on Structural Information Principles

Xianghua Zeng[1], **Hao Peng**[1], **Angsheng Li**[1,2], **Chunyang Liu**[3], **Lifang He**[4], **Philip S. Yu**[5]
[1] State Key Laboratory of Software Development Environment, Beihang University
[2] Zhongguancun Laboratory
[3] Didi Chuxing
[4] Department of Computer Science & Engineering, Lehigh University
[5] Department of Computer Science, University of Illinois at Chicago
{zengxianghua, penghao, angsheng}@buaa.edu.cn, liangsheng@gmail.zgclab.edu.cn,
liuchunyang@didiglobal.com, lih319@lehigh.edu, psyu@uic.edu.

Abstract

State abstraction optimizes decision-making by ignoring irrelevant environmental information in reinforcement learning with rich observations. Nevertheless, recent approaches focus on adequate representational capacities resulting in essential information loss, affecting their performances on challenging tasks. In this article, we propose a novel mathematical **S**tructural **I**nformation principles-based **S**tate **A**bstraction framework, namely **SISA**, from the information-theoretic perspective. Specifically, an unsupervised, adaptive hierarchical state clustering method without requiring manual assistance is presented, and meanwhile, an optimal encoding tree is generated. On each non-root tree node, a new aggregation function and condition structural entropy are designed to achieve hierarchical state abstraction and compensate for sampling-induced essential information loss in state abstraction. Empirical evaluations on a visual gridworld domain and six continuous control benchmarks demonstrate that, compared with five SOTA state abstraction approaches, SISA significantly improves mean episode reward and sample efficiency up to 18.98 and 44.44%, respectively. Besides, we experimentally show that SISA is a general framework that can be flexibly integrated with different representation-learning objectives to improve their performances further.

1 Introduction

Reinforcement Learning (RL) is a promising approach to intelligent decision-making for a variety of complex tasks, such as robot walking [Collins *et al.*, 2005], recommending systems [Ie *et al.*, 2019], automating clustering [Zhang *et al.*, 2022], abnormal detection [Peng *et al.*, 2021], and multi-agent collaboration [Baker *et al.*, 2020; Peng *et al.*, 2022]. In the RL setting, agents often learn to maximize their rewards in environments with high-dimensional and noisy observations, which requires suitable state representations [Jong and Stone, 2005; Kaiser *et al.*, 2019]. A valid solution is state abstraction, which can ignore irrelevant environmental information to compress the original state space, thereby considerably simplifying the decision process [Abel *et al.*, 2016; Laskin *et al.*, 2020b].

Prior work defines state-abstraction types via aggregation functions that group together "sufficiently similar" states for reductions in task complexity [Li *et al.*, 2006; Hutter, 2016; Abel *et al.*, 2016; Abel *et al.*, 2018]. However, their abstraction performances heavily depend on manual assistance due to high sensitivity to aggregation parameters, such as approximate abstraction's predicate constant and transitive abstraction's bucket size. On the other hand, recent work transfers state abstraction into a representation-learning problem and incorporates various learning objectives to enable state representations with desirable properties [Gelada *et al.*, 2019; Zhang *et al.*, 2020; Zang *et al.*, 2022; Zhu *et al.*, 2022]. Despite their adequate representational capacities, these approaches discard some essential information about state dynamics or rewards, making them hard to characterize the environment accurately. Therefore, balancing irrelevant and essential information is vital for decision-making with rich observations. Recently, Markov state abstraction [Allen *et al.*, 2021] is introduced to realize this balance, reflecting the original rewards and transition dynamics while guaranteeing its representational capacity. Nevertheless, representation learning based on sampling from finite replay buffers inevitably induces essential information loss in Markov abstraction, affecting its performance on challenging tasks. Although multi-agent collaborative role discovery based on structural information principles has been proposed [Zeng *et al.*, 2023], it is not available in the RL scenario of a single agent.

In this paper, we propose a novel mathematical **S**tructural **I**nformation principles-based hierarchical **S**tate **A**bstraction framework, namely SISA, from the information-theoretic perspective. The critical insight is that SISA combines hierarchical state clustering and aggregation of different hierarchies to achieve sample-efficient hierarchical abstraction. Inspired by the structural entropy minimization principle [Li and Pan, 2016; Li *et al.*, 2018], we first present an unsupervised, adaptive hierarchical state clustering method without requiring manual assistance. It consists of structuralization, sparsification, and optimization modules, to construct an optimal encoding tree. Secondly, an effective autoencoder structure and representation-learning objectives are adopted to learn state embeddings and refine the hierarchical clustering. Thirdly, for non-root tree nodes of different heights, we define a new aggregation function using the assigned structural entropy as each child node's weight, thereby achieving the hierarchical state abstraction. The hierarchical abstraction from leaf nodes to the root on the optimal encoding tree is an automatic process of ignoring irrelevant information and preserving essential information. Moreover, a new conditional structural entropy is designed to reconstruct the relation between original states to compensate for sampling-induced essential information loss. Furthermore, SISA is a general framework and can be flexibly integrated with various representation-learning abstraction approaches, e.g., Markov abstraction [Allen *et al.*, 2021] and SAC-AE [Yarats *et al.*, 2021], for improving their performances. Extensive experiments are conducted in both offline and online environments with rich observations, including one gridworld navigation task and six continuous control benchmarks. Comparative results and analysis demonstrate the performance advantages of the proposed state abstraction framework over the five latest SOTA baselines. All source codes and experimental results are available at Github[1].

The main contributions of this paper are as follows: 1) Based on the structural information principles, an innovative, unsupervised, adaptive hierarchical state abstraction framework (SISA) without requiring manual assistance is proposed to optimize RL in rich environments. 2) A novel aggregation function leveraging the assigned structural entropy is defined to achieve hierarchical abstraction for efficient decision-making. 3) A new conditional structural entropy reconstructing state relations is designed to compensate for essential information loss in abstraction. 4) The remarkable performance on challenging tasks shows that SISA achieves up to 18.98 and 44.44% improvements in the final performance and sample efficiency than the five latest SOTA baselines.

2 Background

2.1 Markov Decision Process

In RL, the problem to resolve is described as a Markov Decision Process (MDP) [Bellman, 1957], a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the original state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward function, $\mathcal{P}(s' \mid s, a)$ is the transitioning probability from state $\mathrm{s} \in \mathcal{S}$ to state $\mathrm{s}' \in \mathcal{S}$ conditioning on an action $a \in \mathcal{A}$, and $\gamma \in [0, 1)$ is the discount factor. At each timestep, an agent chooses an action $a \in \mathcal{A}$ according to its policy function $a \sim \pi(s)$, which updates the environment state $s' \sim \mathcal{P}(s, a)$, yielding a reward $r \sim \mathcal{R}(s, a) \in \mathbb{R}$. The goal of the agent is to learn a policy that maximizes long-term expected discounted reward.

2.2 State Abstraction

Following Markov state abstraction [Allen *et al.*, 2021], we define state abstraction as a function f_ϕ that projects each original state $s \in \mathcal{S}$ to an abstract state $z \in \mathcal{Z}$. When applied to an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, the state abstraction induces a new abstract decision process $\mathcal{M}_\phi = (\mathcal{Z}, \mathcal{A}, \mathcal{R}_\phi, \mathcal{P}_\phi, \gamma)$, where typically $|\mathcal{Z}| \ll |\mathcal{S}|$.

[1] https://github.com/RingBDStack/SISA

2.3 Structural Information Principles

Structural information principles were first proposed to measure the dynamical uncertainty of a graph, called structural entropy [Li and Pan, 2016]. They have been widely applied to optimize graph classification and node classification [Wu *et al.*, 2022a; Wu *et al.*, 2022b; Zou *et al.*, 2023; Wang *et al.*, 2023; Yang *et al.*, 2023], obfuscate community structures [Liu *et al.*, 2019], and decode the chromosomes domains [Li *et al.*, 2018]. By minimizing the structural entropy, we can generate the optimal partitioning tree, which we name an "encoding tree".

We suppose a weighted undirected graph $G = (V, E, W)$, where V is the vertex set[2], E is the edge set, and $W : E \mapsto \mathbb{R}^+$ is the weight function of edges. Let $n = |V|$ be the number of vertices and $m = |E|$ be the number of edges. For each graph vertex $v \in V$, the weights sum of its connected edges is defined as its degree d_v.

Encoding tree. The encoding tree of graph G is a rooted tree defined as follows: 1) For each node $\alpha \in T$, a vertex subset T_α in G corresponds with α, $T_\alpha \subseteq V$. 2) For the root node λ, we set $T_\lambda = V$. 3) For each node $\alpha \in T$, we mark its children nodes as $\alpha^\wedge\langle i\rangle$ ordered from left to right as i increases, and $\alpha^\wedge\langle i\rangle^- = \alpha$. 4) For each node $\alpha \in T$, L is supposed as the number of its children; then all vertex subsets $T_{\alpha^\wedge\langle i\rangle}$ are disjointed, and $T_\alpha = \bigcup_{i=1}^{L} T_{\alpha^\wedge\langle i\rangle}$. 5) For each leaf ν, T_ν is a singleton subset containing a graph vertex.

One-dimensional structural entropy. The one-dimensional structural entropy[3] measures the dynamical complexity of the graph G without any partitioning structure and is defined as:

$$H^1(G) = -\sum_{v \in V} \frac{d_v}{vol(G)} \cdot \log_2 \frac{d_v}{vol(G)}, \tag{1}$$

where $vol(G) = \sum_{v \in V} d_v$ is the volume of G.

K-dimensional structural entropy. An encoding tree T, whose height is at most K, can effectively reduce the dynamical complexity of graph G, and the K-dimensional structural entropy measures the remaining complexity. For each node $\alpha \in T, \alpha \neq \lambda$, its assigned structural entropy is defined as:

$$H^T(G; \alpha) = -\frac{g_\alpha}{vol(G)} \log_2 \frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}, \tag{2}$$

where g_α is the sum of weights of all edges connecting vertices in T_α with vertices outside T_α. $\mathcal{V}_\alpha$ is the volume of T_α, the sum of degrees of vertices in T_α. Given the encoding tree T, the K-dimensional structural entropy of G is defined as:

$$H^K(G) = \min_T \left\{ \sum_{\alpha \in T, \alpha \neq \lambda} H^T(G; \alpha) \right\}, \tag{3}$$

where T ranges over all encoding trees whose heights are at most K, and the dimension K constraints the maximal height of the encoding tree T.

[2] Vertices are defined in the graph, and nodes are in the tree.

[3] It is another form of Shannon entropy using the stationary distribution of the vertex degrees.

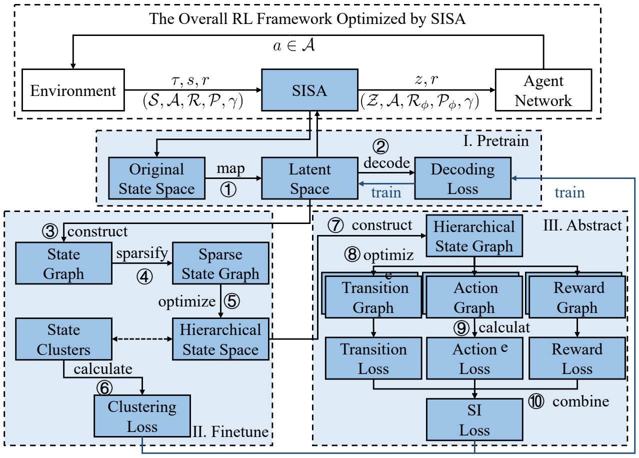


Figure 1: The proposed SISA framework.

3 The SISA Framework

This section describes the detailed design of the structural information principles-based state abstraction and how to apply SISA to optimize RL.

3.1 Overall RL Framework Optimized by SISA

For better descriptions, we first introduce how to apply SISA to optimize RL framework. The optimized RL framework consists of three modules: Environment, Agent Network $\mathcal{Q}$, and the proposed state abstraction SISA, as shown in Fig. 1. The decision process in the environment is labeled as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where the original state space $\mathcal{S}$ is high-dimensional and noisy. SISA described in the following subsection takes action-observation history τ as input and maps each original environment state $s \in \mathcal{S}$ to an abstract state $z \in \mathcal{Z}$, where $|\mathcal{Z}| \ll |\mathcal{S}|$. Moreover, the agent makes decisions based on its individual network $\mathcal{Q}$ taking the abstract state z and reward r as inputs, which induces a new abstract decision process $\mathcal{M}_\phi = (\mathcal{Z}, \mathcal{A}, \mathcal{R}_\phi, \mathcal{P}_\phi, \gamma)$.

3.2 Hierarchical State Abstraction

As shown in Fig. 1, SISA includes pretrain, finetune, and abstract stages. In the pretrain stage, we map the original state space to a dense low-dimensional latent space and adopt representation-learning objectives to decode. In the finetune stage, we sparsify a state graph, optimize its encoding tree to obtain a hierarchical state structure, and calculate a clustering loss. In the abstract stage, we construct a hierarchical state graph and extract transition, action, and reward relations to calculate a structural information (SI) loss.

Prtrain. For tractable decision-making in high-dimensional and noisy environments, we utilize representation-learning objectives to compress the state space via an abstraction function, as the level-0 abstraction.

To this end, we adopt the encoder-decoder structure [Cho *et al.*, 2014] to learn abstract state representations, mapping the state space S to a low-dimensional and dense abstract state space Z. In the encoder, we encode each state $s \in S$ as a d-dimensional embedded representation $z \in Z$ via the abstraction function $f_\phi : S \to Z$, as the step 1 in Fig. 2[4]. In the decoder, we decode each abstract representation z and select the training objectives in Markov state abstraction, including constructive and adversarial objectives, for calculating the decoding loss L_{de} to guarantee Markov property in the pretrain stage, as the step 2 in Fig. 2. Given the action-observation history τ, the encoder-decoder structure is trained end-to-end by minimizing L_{de}. Furthermore, the abstraction function f_ϕ will be further optimized in the finetune and abstract stages by minimizing the clustering loss L_{cl} and SI loss L_{si}.

Finetune. Instead of aggregation condition definitions [Abel *et al.*, 2018] or representation learning in the original state space [Gelada *et al.*, 2019; Laskin *et al.*, 2020a; Zhang *et al.*, 2020], we present an unsupervised, adaptive hierarchical clustering method without requiring manual assistance to obtain the hierarchical structure of environment states. Specifically, we construct a weighted, undirected, complete state graph according to state correlations, minimize its structural entropy to get the optimal encoding tree, and calculate the clustering loss based on Kullback-Leibler (KL) divergence.

Firstly, for states s_i and s_j with $i \neq j$, we calculate the cosine similarity between abstract representations z_i and z_j to measure their correlation $\mathcal{C}_{ij} \in [-1, 1]$. Intuitively, the larger the value of $\mathcal{C}_{ij}$ represents the more similarity between states s_i and s_j, which should belong to the same cluster with a more significant probability. We take states as vertices and for any two vertices s_i and s_j, assign $\mathcal{C}_{i,j}$ to the undirected weighted edge (s_i, s_j), $w_{ij} = \mathcal{C}_{ij}$, thereby constructing the complete graph G, as the step 3 in Fig. 2. In G, vertices represent states in S, namely $V = S$, edges represent state correlations, and edge weight quantifies the cosine similarity between states. We define edge weight whose absolute value approaches 0 as trivial weight.

Secondly, we realize sparsification of the state graph to eliminate negative interference of trivial weights. Following the construction of cancer cell neighbor networks [Li *et al.*, 2016], we minimize the one-dimensional structural entropy to sparsify graph G into a k-nearest neighbor (k-NN) graph G_k, as the step 4 in Fig. 2. We retain the most significant k edge weights for each vertex to construct G_k, calculate its one-dimensional structural entropy $H^1(G_k)$, select parameter k of the minimum structural entropy as k^*, and output G_{k^*} as the sparse state graph G^*. Moreover, we initialize an encoding tree T of G^*: 1) We generate a root node λ and set its vertex subset $T_\lambda = S$ as the whole state space; 2) We generate a leaf node ν with $T_\nu = \{s\}$ for each state $s \in S$, and set it as a child node of λ, $\nu^- = \lambda$.

Thirdly, we realize the hierarchical state clustering by optimizing the encoding tree T from 1 layer to K layers. In our work, two operators *merge* and *combine* are introduced from the deDoc [Li *et al.*, 2018] to optimize the sparse graph G^* by minimizing its K-dimensional structural entropy, as the step 5 in Fig. 2. We define two nodes possessing a common father node in the encoding tree T are brothers. The *merge*

[4] For better understanding, we set $S = \{s_0, s_1 \ldots, s_{11}\}$ in the original state space as an example.

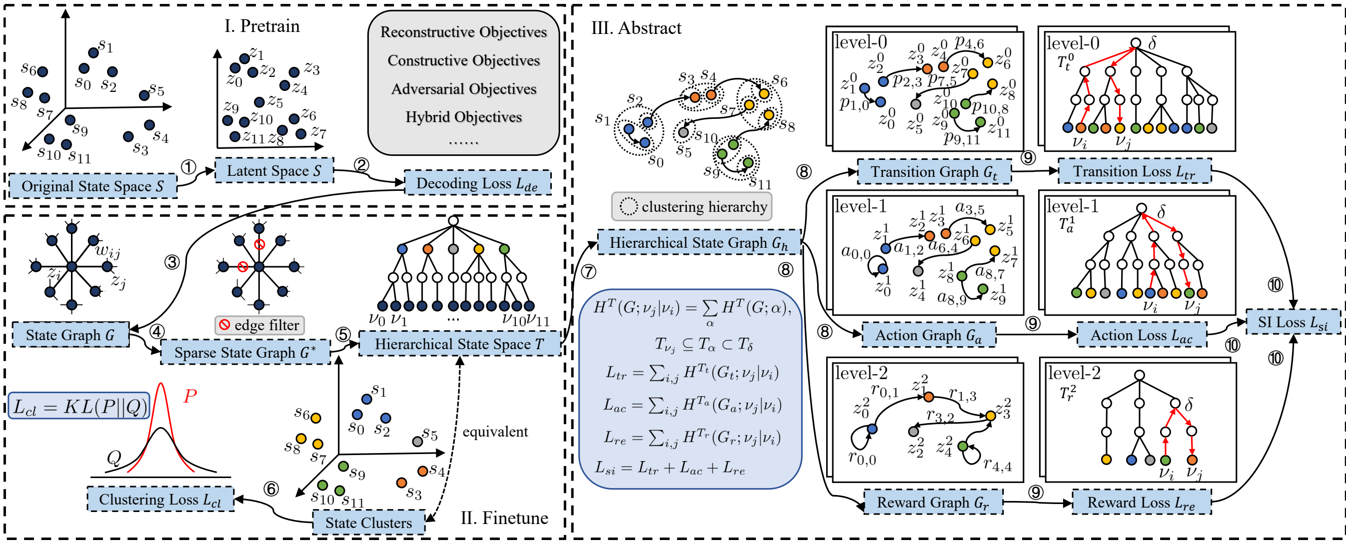


Figure 2: The proposed state abstraction SISA.

and *combine* are operated on brother nodes and marked as T_{mg} and T_{cb}. We summarize the encoding tree optimization as an iterative algorithm, as shown in Algorithm 1. At each iteration, we traverse all brother nodes β_1 and β_2 in T (lines 4 and 9) and greedily execute operator T_{mg} or T_{cb} to realize the maximum structural entropy reduction ΔSE if the tree height does not exceed K (lines 5 and 10). When no brother nodes satisfy $\Delta SE > 0$ or the tree height exceeds K, we terminate the iterative algorithm and output the optimal encoding tree T^*. The tree T^* is a hierarchical clustering structure of the state space S, where the root node λ corresponds to S, $T_\lambda = S$, each leaf node ν corresponds to a singleton containing a single state $s \in S$, $T_\nu = \{s\}$, and other tree nodes correspond to state clusters with different hierarchies.

Finally, we choose each child $\lambda^\wedge\langle i\rangle$ of the root node λ as a cluster center and define a structural probability distribution among its corresponding vertex set $T_{\lambda^\wedge\langle i\rangle}$ to calculate its embedding C_i. For each vertex $s_j \in T_{\lambda^\wedge\langle i\rangle}$, we define its distribution probability using the sum of the assigned structural entropies of nodes on the path connecting its corresponding leaf node ν_j and node $\lambda^\wedge\langle i\rangle$ as follows:

$$p_{\lambda^\wedge\langle i\rangle}(s_j) = \exp(-\sum_{T_{\nu_j} \subseteq T_\alpha \subset T_{\lambda^\wedge\langle i\rangle}} H^{T^*}(G;\alpha)), \quad (4)$$

where α is any node on the path connecting ν_j and $\lambda^\wedge\langle i\rangle$. For the cluster center $\lambda^\wedge\langle i\rangle$, we calculate its embedding C_i by:

$$C_i = \sum_{s_j \in T_{\lambda^\wedge\langle i\rangle}} p_{\lambda^\wedge\langle i\rangle}(s_j) \cdot z_j, \quad (5)$$

where z_j is the abstract representation of state s_j. Based on the abstract representations and cluster center embeddings, we generate a soft assignment matrix Q, where Q_{ij} represents the probability of assigning i-th state s_i to j-th cluster center $\lambda^\wedge\langle j\rangle$. We derive a high-confidence assignment matrix P from Q and calculate the clustering loss L_{cl} as follows:

$$L_{cl} = KL(P\|Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}. \quad (6)$$

Algorithm 1: The Iterative Optimization Algorithm

Input: T
Output: T^*
1 Initialize β_1^*, β_2^*
2 **while** *True* **do**
3 $\Delta SE \leftarrow 0$
4 **for** *each brother nodes β_1 and β_2 in T* **do**
5 $\beta_1^*, \beta_2^* \leftarrow$ maximize ΔSE caused by the operator T_{mg} via Eq. (3)
6 **if** $\Delta SE > 0$ **then**
7 Execute the operator T_{mg} on β_1^*, β_2^*
8 Continue
9 **for** *each brother nodes β_1 and β_2 in T* **do**
10 $\beta_1^*, \beta_2^* \leftarrow$ maximize ΔSE caused by the operator T_{cb} via Eq. (3)
11 **if** $\Delta SE > 0$ **then**
12 Execute the operator T_{cb} on β_1^*, β_2^*
13 **else**
14 Break
15 $T^* \leftarrow T$
16 **return** T^*

3.3 Abstraction on Optimal Encoding Tree

To compensate for essential information loss induced by sampling, we leverage structural information principles to design an aggregation function on the optimal encoding tree for achieving hierarchical abstraction while accurately characterizing the original decision process.

The optimal encoding tree T^* represents a hierarchical clustering structure of the state space S, where each tree node corresponds to a state cluster and the height is its clustering hierarchy. Given the action-observation and reward histories, we firstly sample randomly to construct a hierarchical state graph G_h, where vertices represent states and edges represent state transitions with action and reward information, as the

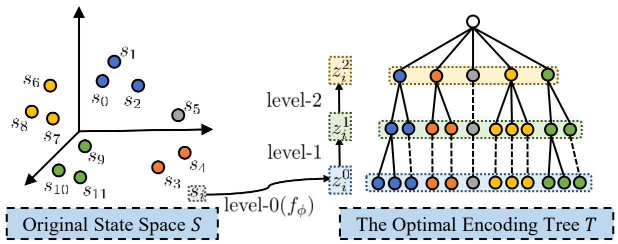


Figure 3: The hierarchical abstraction in the SISA.

step 7 in Fig. 2. Because of construction by sampling, there is an inevitable essential loss of reward or action information between states in the hierarchical graph G_h. Secondly, we define an aggregation function on the optimal encoding tree to achieve hierarchical abstraction from leaf nodes to the root, as shown in Fig. 3. For each leaf node ν_i with $T_{\nu_i} = \{s_i\}$, we define the level-0 abstraction via function f_ϕ described in the pretrain stage and get its level-0 abstract representation z_i^0:

$$z_i^0 = f_\phi(s_i). \tag{7}$$

For each non-leaf node α_i whose height is h, we design an aggregation function using the assigned structural entropy as each child node's weight to achieve the level-h abstraction:

$$z_i^h = \sum_{j=1}^{L} \frac{H^{T^*}(G; \alpha_i^\wedge\langle j\rangle)}{\sum_{l=1}^{L} H^{T^*}(G; \alpha_i^\wedge\langle l\rangle)} \cdot z_{li+j-1}^{h-1}, \tag{8}$$

where L is the number of children nodes of α_i and li is its most left child's index in tree nodes whose height is $h-1$. Thirdly, we extract three kinds of state relations (transition, action, and reward) from the hierarchical graph G_h to construct multi-level transition, action, and reward graphs, respectively, as the step 8 in Fig. 2. For convenience, we take the level-0 transition graph G_t^0 as an example, and operations on graphs of different relations or levels are similar. In G_t^0, vertices represent the level-0 abstract representations and edge weights quantify the transition probabilities between states via sampling. Fourthly, we minimize the K-dimensional structural entropy of G_t^0 to generate its optimal encoding tree T_t^0 and calculate the level-0 transition loss L_{tr}^0, as the step 9 in Fig. 2. Furthermore, we design a conditional structural entropy to reconstruct the state relation to compensate for sampling-induced essential information loss. For any two leaf nodes ν_i and ν_j in T_t^0, we find their common father node δ and calculate conditional structural entropy to quantify the transition probability from z_i^0 to z_j^0 as follows:

$$p(z_j^0|z_i^0) = H^{T_t^0}(G_t^0; z_j^0|z_i^0) = \sum_{T_{\nu_j} \subseteq T_\alpha \subset T_\delta} H^{T_t^0}(G_t^0; \alpha), \tag{9}$$

where α is any node on the path connecting the father node δ and leaf ν_j. And we decode the abstract representations to reconstruct transition probabilities for calculating L_{tr}^0. Finally, as the step 10 in Fig. 2, the SI loss L_{si} is calculated as:

$$L_{si} = L_{tr} + L_{ac} + L_{re} = \sum_{i=1}^{K} (L_{tr}^i + L_{ac}^i + L_{re}^i), \tag{10}$$

where K is the maximal encoding tree height.

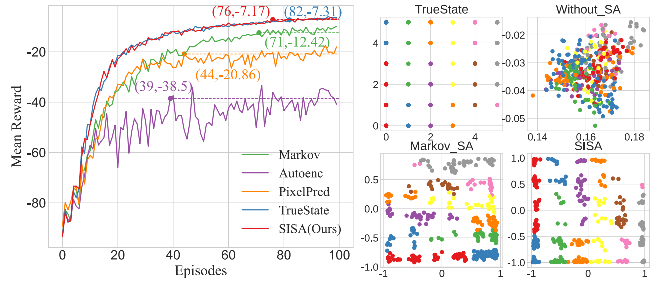


Figure 4: (left) Mean episode rewards for the visual gridworld navigation task. (right) Visualization of 2-D state abstractions for the 6×6 visual gridworld domain.

4 Experiments

In this section, we conduct extensive empirical and comparative experiments, including offline abstraction for visual gridworlds and online abstraction for continuous control. And we evaluate final performance by measuring the mean reward of each episode and evaluate sample efficiency by measuring how many steps it takes to achieve the best performance. Similar to other works [Laskin *et al.*, 2020a; Zhang *et al.*, 2020], all experimental results are illustrated with the average and deviation of performances with different random seeds for fair evaluations. By default, we set the maximal encoding tree height in SISA as 3, $K = 3$. All experiments are conducted on a 3.00GHz Intel Core i9 CPU and an NVIDIA RTX A6000 GPU.

4.1 Offline Abstraction for Visual Gridworlds

Experimental setup. First, we evaluate SISA for offline state abstraction in a visual gridworld domain, where each discrete position is mapped to a noisy image, like experiments in Markov abstraction [Allen *et al.*, 2021]. The agent only has access to these noisy images and uses a uniform random exploration policy over four directional actions to train the SISA framework offline. Then, we froze the framework that maps images to abstract states while training DQN [Mnih *et al.*, 2015]. We compare SISA against three baselines, including pixel prediction [Kaiser *et al.*, 2019], reconstruction [Lee *et al.*, 2020], and Markov abstraction [Allen *et al.*, 2021].

Evaluations. Fig. 4 shows the learning curves of SISA and three baselines for the visual gridworld navigation task. For reference, we also include a learning curve for DQN trained on ground-truth positions without abstraction, labeled as TrueState. Each curve's starting point of convergence is marked in brackets. As shown in Fig. 4 (left), SISA converges at 76.0 epochs and achieves a -7.17 mean episode reward. It can be observed that SISA significantly outperforms other baselines and matches the performance of the TrueState. Moreover, we visualize the 2-D abstract representations for the 6×6 gridworld domain and denote ground-truth positions with different colors in Fig. 4 (right). In SISA, the hierarchical clustering based on the structural information principles effectively reconstructs relative positions of the gridworld better than baselines, resulting in its advantage.

Domain, Task	ball_in_cup-catch	cartpole-swingup	cheetah-run	finger-spin	reacher-easy	walker-walk
DBC	168.95 ± 84.76	317.74 ± 77.49	432.24 ± 181.43	805.90 ± 78.85	191.44 ± 69.07	331.97 ± 108.40
SAC-AE	929.24 ± 39.14	$\underline{839.23} \pm 15.83$	663.71 ± 9.16	898.08 ± 30.23	917.24 ± 38.33	895.33 ± 56.25
RAD	$\underline{937.97} \pm \mathbf{6.77}$	$825.62 \pm \underline{9.80}$	$\underline{802.53} \pm 8.73$	835.20 ± 93.26	$908.24 \pm \underline{25.62}$	907.08 ± 13.02
CURL	899.03 ± 30.61	824.46 ± 18.53	$309.49 \pm \mathbf{8.15}$	949.57 ± 15.71	$\underline{919.71} \pm 28.03$	$885.03 \pm \mathbf{9.88}$
Markov	919.10 ± 38.14	814.94 ± 17.61	642.79 ± 65.92	$\underline{969.91} \pm \mathbf{8.41}$	806.34 ± 131.40	$\underline{918.44} \pm 12.58$
SISA(Ours)	$\mathbf{946.29} \pm \underline{8.63}$	$\mathbf{858.21} \pm \mathbf{6.31}$	$\mathbf{806.67} \pm \underline{8.61}$	$\mathbf{970.45} \pm \underline{8.75}$	$\mathbf{924.52} \pm \mathbf{19.04}$	$\mathbf{921.64} \pm \underline{12.43}$
Abs.(%) Avg. ↑	8.32(0.89)	18.98(2.26)	4.14(0.52)	0.54(0.06)	4.81(0.52)	3.20(0.35)

Table 1: Summary of the mean episode rewards for different tasks from DMControl: "average value $\pm$ standard deviation" and "average improvement" (absolute value(%)). **Bold**: the best performance under each category, <u>underline</u>: the second performance.

4.2 Online Abstraction for Continuous Control

Experimental setup. Next, we benchmark our framework in an online setting with a challenging and diverse set of image-based, continuous control tasks from the DeepMind Control suite (DMControl) [Tunyasuvunakool *et al.*, 2020]. The online experiments are conducted on six DMControl environments: $ball\_in\_cup$-$catch$, $cartpole$-$swingup$, $cheetah$-run, $finger$-$spin$, $reacher$-$easy$, and $walker$-$walk$, to examine the sample efficiency and final performance. The Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018] is chosen as a traditional RL algorithm, combined with SISA and different state abstraction baselines. The compared state-of-the-art baselines consist of random data augmentation RAD [Laskin *et al.*, 2020b], contrastive method CURL [Laskin *et al.*, 2020a], bisimulation method DBC [Zhang *et al.*, 2020], pixel-reconstruction method SAC-AE [Yarats *et al.*, 2021], and Markov abstraction [Allen *et al.*, 2021].

Evaluations. We evaluate all compared methods in six environments from the DMControl suite and summarize averages and deviations of mean episode rewards in Table 1. It can be seen that SISA improves the average mean episode reward in each DMControl environment. Specifically, SISA achieves up to 18.98 (2.26%) improvement from 839.23 to 858.21 in average value, which corresponds to its advantage on final performance. In terms of stability, SISA reduces the standard deviation in two environments. And in the other four environments, SISA achieves the second lowest deviations (8.63, 8.61, 8.75, and 12.43), where it remains very close to the best baseline. The reason is that, SISA minimizes the structural entropy to realize the optimal hierarchical state clustering without any manual assistance and therefore guarantees its stability.

On the other hand, the sample-efficiency results of DMControl experiments are shown in Fig. 5. In each experiment, we set the mean reward target as 0.9 times the final performance of SISA and choose the best baseline as the compared method. In contrast to classical baselines, SISA takes fewer steps to finish the mean episode reward target and thereby achieves higher sample efficiency. In particular, SISA achieves up to 44.44% improvement in sample efficiency, reducing the environment steps from $45k$ to $25k$ to obtain an 851.661 mean reward in $ball\_in\_cup$-$catch$ task.

In summary, in the online setting where reward information is available, SISA establishes a new state of art on DMControl regarding final performance and sample efficiency. The hierarchical abstraction on the optimal encoding tree effectively compensates for essential information loss in state compression to maintain the original task characteristics, guaranteeing SISA's advantages. Fig. 6 shows the learning curves of SISA and the three best-performing baselines in each task; similarly, their starting points of convergence are marked. SISA converges at 64000.0 timesteps and achieves an 858.21 mean episode reward, as shown in the $cartpole$-$swingup$ task.

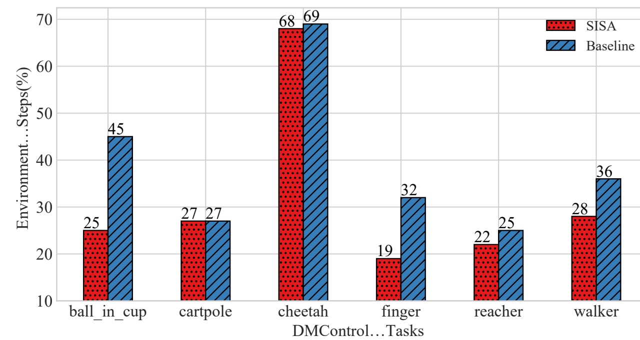


Figure 5: The sample-efficiency results for DMControl experiments.

4.3 Integrative Abilities

SISA is a general framework and can be flexibly integrated with various existing representation-learning abstraction approaches in the pretrain stage. Therefore, we integrate our framework with the Markov abstraction and SAC-AE, namely Markov-SISA and SAC-SISA, and choose two tasks ($ball\_in\_cup$-$catch$ and $cartpole$-$swingup$) to evaluate their performances. Each integrated framework achieves higher final performance and sample efficiency than the original approach, as shown in Fig. 7. The experimental results indicate that our abstraction framework can significantly optimize existing abstraction approaches in complex decision-making.

4.4 Ablation Studies

We conduct ablation studies in the $finger$-$spin$ task to understand the functionalities of finetune and abstract stages in SISA. The finetune and abstract stages are removed from SISA, respectively, and we name the corresponding variants SISA-FI and SISA-AT. As shown in Fig. 8, SISA remarkably outperforms SISA-FI and SISA-AT in the final performance, sample efficiency, and stability, which shows that the finetune and abstract stages are both important for the SISA's advantages. Furthermore, the advantages over the SISA-AT variant are more significant.

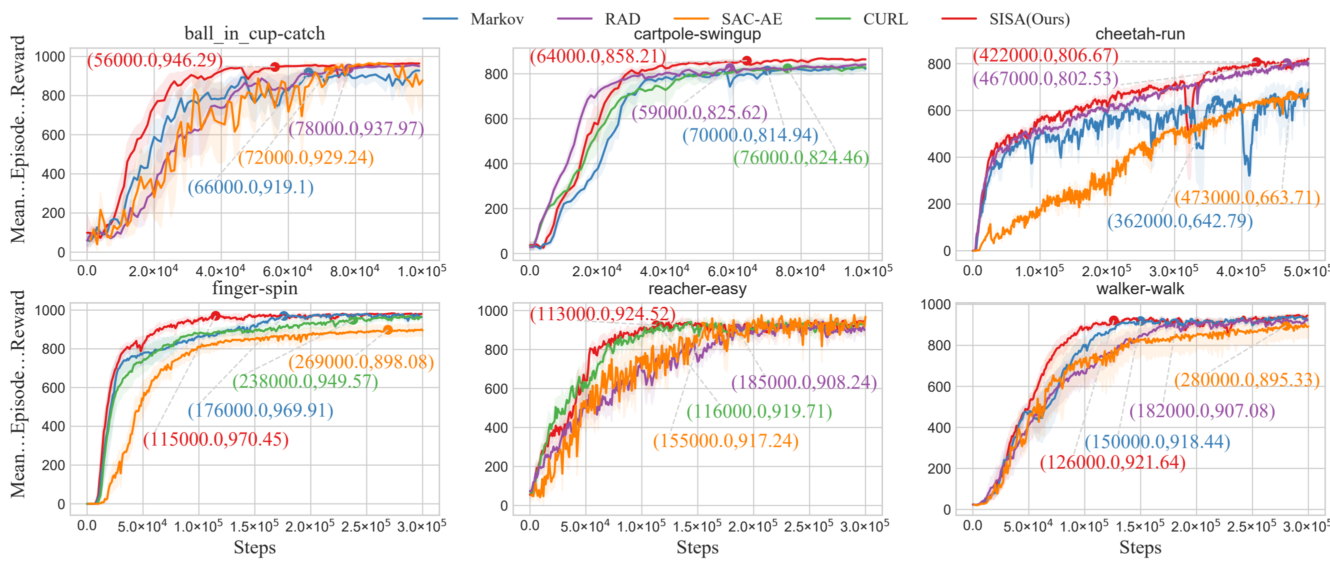
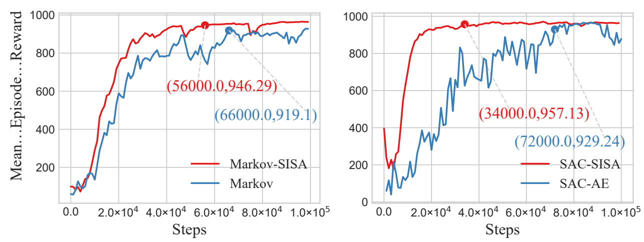
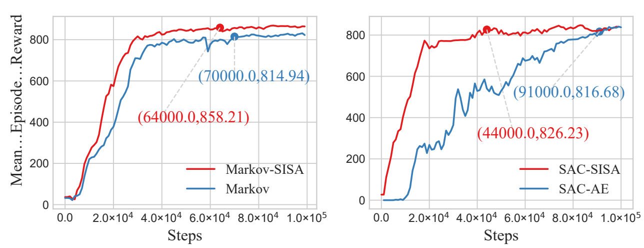


Figure 6: Mean episode rewards on six DMControl environments.



(a) ball_in_cup-catch



(b) cartpole-swingup

Figure 7: Mean episode rewards of the SISA integrated with abstraction methods Markov and SAC-AE.

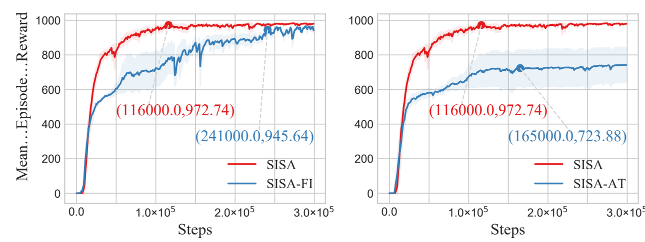


Figure 8: Mean episode rewards for ablation studies.

5 Related Work

State abstractions for sample-efficient RL. The SAC-AE [Yarats *et al.*, 2021] trains models to reproduce original states by pixel prediction and related tasks perfectly. Instead of prediction, the CURL [Laskin *et al.*, 2020a] learns abstraction by differentiating whether two augmented views come from the same observation. The DBC [Zhang *et al.*, 2020] trains a transition model and reward function end-to-end to learn approximate bisimulation abstractions, where original states are equivalent if their expected reward and transition dynamics are the same. To ensure the abstract decision process is Markov, Allen et al. [2021] introduce sufficient conditions to learn Markov abstract state representations. Recently, SimSR [Zang *et al.*, 2022] designs a stochastic approximation method to learn abstraction from observations to robust latent representations. IAEM [Zhu *et al.*, 2022] efficiently obtains abstract representations, by capturing action invariance. State abstraction is applied to three-valued semantics to find "failure" states under assumptions of imperfect information and perfect recall [Belardinelli *et al.*, 2023].

6 Conclusion

This paper proposes a general structural information principles-based hierarchical state abstraction (SISA) framework, from the information-theoretic perspective. To the best of our knowledge, it is the first work to incorporate the mathematical structural information principles into state abstraction to optimize decision-making with high-dimension and noisy observations. Evaluations of challenging tasks in the visual gridworld and DMControl suite demonstrate that SISA significantly improves final performance and sample efficiency over state-of-the-art baselines. In the future, we will evaluate SISA in other environments and further explore the hierarchical encoding tree structure in decision-making.

Acknowledgments

The corresponding authors are Hao Peng and Angsheng Li. This paper was supported by the National Key R&D Program of China through grant 2021YFB1714800, NSFC through grant 61932002, S&T Program of Hebei through grant 20310101D, Natural Science Foundation of Beijing Municipality through grant 4222030, CCF-DiDi GAIA Collaborative Research Funds for Young Scholars, the Fundamental Research Funds for the Central Universities, Xiaomi Young Scholar Funds for Beihang University, and in part by NSF under grants III-1763325, III-1909323, III-2106758, SaTC-1930941, and MRI-2215789, as well as Lehigh University's Accelerator and CORE grants S00010293 and 001250.

References

[Abel *et al.*, 2016] David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of the International Conference on Machine Learning*, pages 2915–2923. PMLR, 2016.

[Abel *et al.*, 2018] David Abel, Dilip Arumugam, Lucas Lehnert, and Michael Littman. State abstractions for lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 10–19. PMLR, 2018.

[Allen *et al.*, 2021] Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning markov state abstractions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8229–8241, 2021.

[Baker *et al.*, 2020] Bowen Baker, Ingmar Kanitscheider, Todor M. Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *Proceedings of the International Conference on Learning Representations*, pages 1–28, 2020.

[Belardinelli *et al.*, 2023] Francesco Belardinelli, Angelo Ferrando, and Vadim Malvone. An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information. *Artificial Intelligence*, page 103847, 2023.

[Bellman, 1957] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[Collins *et al.*, 2005] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[Gelada *et al.*, 2019] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *Proceedings of the International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[Hutter, 2016] Marcus Hutter. Extreme state aggregation beyond markov decision processes. *Theoretical Computer Science*, 650:73–91, 2016.

[Ie *et al.*, 2019] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2592–2599. International Joint Conferences on Artificial Intelligence Organization, 2019.

[Jong and Stone, 2005] Nicholas K Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 8, pages 752–757. Citeseer, 2005.

[Kaiser *et al.*, 2019] Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model based reinforcement learning for atari. In *Proceedings of the International Conference on Learning Representations*, 2019.

[Laskin *et al.*, 2020a] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[Laskin *et al.*, 2020b] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33:19884–19895, 2020.

[Lee *et al.*, 2020] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

[Li and Pan, 2016] Angsheng Li and Yicheng Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016.

[Li *et al.*, 2006] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *AI&M*, 2006.

[Li *et al.*, 2016] Angsheng Li, Xianchen Yin, and Yicheng Pan. Three-dimensional gene map of cancer cell types: Structural entropy minimisation principle for defining tumour subtypes. *Scientific Reports*, 6:1–26, 2016.

[Li *et al.*, 2018] Angsheng Li, Xianchen Yin, Bingxiang Xu, Danyang Wang, Jimin Han, Yi Wei, Yun Deng, Ying Xiong, and Zhihua Zhang. Decoding topologically associating domains with ultra-low resolution hi-c data by graph structural entropy. *Nature Communications*, 9:1–12, 2018.

[Liu *et al.*, 2019] Yiwei Liu, Jiamou Liu, Zijian Zhang, Liehuang Zhu, and Angsheng Li. Rem: From structural entropy to community structure deception. *Advances in Neural Information Processing Systems*, 32, 2019.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Peng *et al.*, 2021] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S. Yu. Reinforced neighborhood selection guided multi-relational graph neural networks. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–46, 2021.

[Peng *et al.*, 2022] Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and Philip S. Yu. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):980–998, 2022.

[Tunyasuvunakool *et al.*, 2020] 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.

[Wang *et al.*, 2023] Yifei Wang, Yupan Wang, Zeyu Zhang, Song Yang, Kaiqi Zhao, and Jiamou Liu. User: Unsupervised structural entropy-based robust graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[Wu *et al.*, 2022a] Junran Wu, Xueyuan Chen, Ke Xu, and Shangzhe Li. Structural entropy guided graph hierarchical pooling. In *Proceedings of the International Conference on Machine Learning*, pages 24017–24030. PMLR, 2022.

[Wu *et al.*, 2022b] Junran Wu, Shangzhe Li, Jianhao Li, Yicheng Pan, and Ke Xu. A simple yet effective method for graph classification. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2022.

[Yang *et al.*, 2023] Zhenyu Yang, Ge Zhang, Jia Wu, Jian Yang, Quan Z Sheng, Hao Peng, Angsheng Li, Shan Xue, and Jianlin Su. Minimum entropy principle guided graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 114–122, 2023.

[Yarats *et al.*, 2021] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021.

[Zang *et al.*, 2022] Hongyu Zang, Xin Li, and Mingzhong Wang. Simsr: Simple distance-based state representations for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8997–9005, 2022.

[Zeng *et al.*, 2023] Xianghua Zeng, Hao Peng, and Angsheng Li. Effective and stable role-based multi-agent collaboration by structural information principles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[Zhang *et al.*, 2020] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *Proceedings of the International Conference on Learning Representations*, 2020.

[Zhang *et al.*, 2022] Ruitong Zhang, Hao Peng, Yingtong Dou, Jia Wu, Qingyun Sun, Yangyang Li, Jingyi Zhang, and Philip S. Yu. Automating dbscan via deep reinforcement learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2620–2630, 2022.

[Zhu *et al.*, 2022] Zheng-Mao Zhu, Shengyi Jiang, Yu-Ren Liu, Yang Yu, and Kun Zhang. Invariant action effect model for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9260–9268, 2022.

[Zou *et al.*, 2023] Dongcheng Zou, Hao Peng, Xiang Huang, Renyu Yang, Jianxin Li, Jia Wu, Chunyang Liu, and Philip S. Yu. Se-gsl: A general and effective graph structure learning framework through structural entropy optimization. In *Proceedings of the Web Conference*, pages 1–12, 2023.