ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230548

Improving Visual Reinforcement Learning with Discrete Information Bottleneck Approach

Haitao Wang[a] **and Hejun Wu**[a;*]

[a]Department of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

Abstract. Contrastive learning has been used to learn useful low-dimensional state representations in visual reinforcement learning (RL). Such state representations substantially improve the sample efficiency of visual RL. Nevertheless, existing contrastive learning-based RL methods have the problem of unstable training. Such instability comes from the fact that contrastive learning requires an extremely large batch size (e.g., 4096 or larger), while current contrastive learning-based RL methods typically set a small batch size (e.g., 512). In this paper, we propose an approach of discrete information bottleneck (DIB) to address this problem. DIB applies the technique of discretization and information bottleneck to contrastive learning in representing the state with concise discrete representation. Using this discrete representation for policy learning results in more stable algorithm training and higher sample efficiency with a small batch size. We demonstrate the advantage of discrete state representation of DIB on several continuous control tasks in the DeepMind Control suite. In the experiments, DIB outperforms prior visual RL methods, both model-based and model-free, in terms of performance and sample efficiency.

1 Introduction

Deep reinforcement learning (RL) is limited by a low sample efficiency when the state inputs are pixels. For example, RL algorithms for Atari games require hundreds of millions of time steps in training to learn a good policy. The key insight to solving this inefficiency problem is learning of better low-dimensional state representations. However, due to the reward sparsity, it is hard for the policy learner of RL to acquire useful low-dimensional state representations from high-dimensional images [27].

To solve this problem, researchers have introduced various auxiliary tasks that provide extra learning signals to learn a low-dimensional state representation for visual RL. Some typical auxiliary tasks include state reconstruction [34], future prediction [25], and world model learning [16]. A number of methods have been proposed to use contrastive learning to construct new auxiliary tasks to reduce the dimension of states in samples for the training of RL. Such contrastive auxiliary tasks improve the sample efficiency of an RL algorithm by extracting the most task-relevant features from high-dimensional state data. Consequently, the RL algorithm can make full use of each sample for training. During the training process, contrastive learning is formalized to maximize a contrastive loss to perform state representation learning. For instance, CPC [31] designs a probabilistic contrastive loss that induces the latent space to capture task-relevant information. CURL [20] improves the sample efficiency by using a contrastive loss, called InfoNCE [31], which enhances the task-relevance within the state data of each sample. RCRL [22] constructs a holistic contrastive loss according to the feedback of the long-term return from RL algorithms in addition to rewards of individual samples. The results of RCRL demonstrate the effectiveness of using such a contrastive loss.

Nevertheless, most prior contrastive learning-based RL methods still face the difficulty of training instability. Numerous experiments have shown that directly incorporating the contrastive loss into an RL algorithm leads to training instability, which in turn leads to poor performance and large variance [20, 28]. This may be because contrastive learning requires a large batch size, such as 4096 or 8192, however, the batch size is usually set to 512 in previous visual RL algorithms. This limitation makes contrastive learning hard to release its greatest potential to visual RL algorithms. Therefore, it is of particular interest to develop a robust version of the contrastive learning-based RL method to improve sample efficiency further.

In this paper, we propose an approach of **D**iscrete **I**nformation **B**ottleneck (DIB) to further improve the sample efficiency of model-free RL algorithms. DIB uses discrete contrastive learning to learn low-dimensional discrete state representations for RL tasks. Our motivation is that discrete representations are a natural fit for complex reasoning, control, and predictive learning. For example, language is inherently discrete and images can often be described concisely by language [32]. In order to learn a discrete state representation, we design a probabilistic discrete layer for the encoder to learn discrete state representations. Then we construct a new discrete contrastive loss and use an iterative optimization algorithm to alleviate the non-differentiability problem in discrete state representation architectures. We theoretically prove the strong connection between the proposed discrete contrastive loss and mutual information. The proposed learning approach conforms to the learning principle of the information bottleneck theory (IB). This theory defines what is a good representation, in terms of the fundamental trade-off between a concise representation and a representation with good predictive power. With this predictive power, DIB can learn more concise state representation.

We illustrate the effectiveness of the proposed method using Soft Actor-Critic [8] for continuous control benchmarks. In our experiments, DIB outperforms other state-of-the-art baselines, both model-based and model-free, in terms of performance and sample efficiency. Our study empirically demonstrates that discrete state representation can stabilize the training process, accelerate the convergence rate of

* Corresponding Author. Email: wuhejun@mail.sysu.edu.cn.

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

the algorithm, and reduce the variance, under a small batch size (i.e., 256). As far as we know, our results suggest for the first time that discrete state representation can be beneficial to model-free RL.

2 Related Work

Visual Reinforcement Learning. Successes of self-supervised learning in NLP and CV [5, 13, 3] have inspired successes in visual RL. Early representative works show how reconstruction loss could improve the sample efficiency of visual RL, such as CPC [31], SAC-AE [34], PlaNet [9], and SLAC [21]. Then, a series of self-supervised works made great progress in visual RL. In the model-free settings, many methods leverage contrastive learning to learn a good representation of the state to accelerate the policy learning of RL, such as CURL [20], RCRL [22], SPR [25], and ATC [28]. Some methods have proposed that the use of data augmentation techniques can significantly improve the sample efficiency of RL. Typical methods of this kind include DrQ [18], RAD [19], DrQ-v2 [33], etc. In the model-based settings, world models can be used as auxiliary tasks to explore the environment and lead to better performance, such as Dreamer [10], Dreamer-v2 [11], Dreamer-v3 [12], DreamerPro [4], and APV [26].

Discrete Representation Learning. Discrete representation learning is a central machine learning task because of the compactness of the representations and ease of convergence [15]. In clustering and unsupervised hashing [14], discrete representation learning plays an important role. Recently, SOMVAE [7] uses discrete representation learning to learn discrete representations of time series and achieve superior performance. In reinforcement learning, some model-based methods [23, 11] have solved visual discrete control problems with a discrete world model. However, the main purpose of the world model is to obtain long-horizon imagination. They are not shown to support direct visual continuous control.

Information Bottleneck. The information bottleneck (IB) was first proposed in [30]. The principle of IB is to learn a good representation that can retain task-relevance information while reducing task-irrelevance information [17, 2]. Since IB is plagued by computational mutual information, many methods are devoted to finding its lower bound. For example, DVIB [1] presents a variational approximation to approach the objective of IB. MIB [6] extends IB under the setting of multi-view learning.

3 Background

The problem for Soft actor-critic (SAC) is a Markov Decision Process (MDP). An MDP can be described as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state collection, $\mathcal{A}$ the action collection, $P(s_{t+1}|s_t, a_t)$ the transition function of environment, $R(s_t, a_t)$ the reward function, and γ the discount factor. Soft actor-critic (SAC) is an off-policy actor-critic RL method. The policy evaluation and policy improvement of SAC are executed alternately at each iteration. SAC solves continuous action space by using deep neural networks, in which the Q-function Q_θ (critic) and policy π_ϕ (actor) are approximated. In the policy evaluation step, the soft Q-function with parameters θ are optimized to minimize the soft Bellman residual:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(Q(s_t,a_t) - r_t - \gamma\overline{V}(s_{t+1})\right)^2\right] \quad (1)$$

where $\mathcal{D}$ is the replay buffer, and $\overline{V}$ the target value function, which is approximated via a Monte Carlo estimate of the following expectation:

$$\overline{V}(s_t) = \mathbb{E}_{a_t\sim\mathcal{D}}\left[Q_{\overline{\theta}}(s_t,a_t) - \alpha log\pi(a_t|s_t)\right] \quad (2)$$

where $\overline{\theta}$ is the delayed parameter. In the step of policy improvement, the actor parameters ϕ are optimized to update the policy. The loss is as follows:

$$J_\pi(\phi) = \mathbb{E}_{a_t\sim\pi_\phi}\left[\alpha log(\pi_\phi(a_t|s_t)) - Q_\theta(s_t,a_t)\right] \quad (3)$$

4 Methodology

In this section, we combine the strong model-free RL algorithm (e.g., SAC) with discrete contrastive loss as a novel auxiliary loss to improve the sample efficiency. Then, we further establish the connection between discrete contrastive loss and mutual information, reformulating the unsupervised auxiliary task under the broader framework of information bottleneck (IB). Without loss of generality, we assume that all of the following vectors are row vectors. For the sake of clarity, frequently used notations and corresponding descriptions are summarized in Table 1. The full notations table is in the Appendix.

4.1 Discrete Contrastive Learning

Contrastive Learning as Auxiliary Task. Many deep reinforcement learning methods combine an RL algorithm with contrastive learning as the auxiliary task to improve the sample efficiency. Formally, given a minibatch of training instances $\{o^k\}_{k=1}^N$ (N is the batch size) from the replay buffer, the contrastive learning-based RL methods first transform each instance into query v_1^k and key v_2^k by data augmentation (we use subscript 1 for query and 2 for key). After that, v_1^k and v_2^k are fed into a query encoder f_{θ_1} and a key encoder f_{θ_2} to produce *continuous representations*, respectively.

$$z_1^k = f_{\theta_1}(v_1^k) \quad (4)$$

$$z_2^k = f_{\theta_2}(v_2^k) \quad (5)$$

The representation vector z_1^k and z_2^k should be similar since they come from the same observation o^k. The aim of contrastive learning is to achieve this goal by minimizing the following loss:

$$L = \frac{1}{N}\sum_{k=1}^{N}(\ell(z_1^k, z_2^k)) \quad (6)$$

where ℓ is a certain contrastive loss such as InfoNCE [31] and Margin [24]. During training, the observations are passed to the RL algorithm while the query-key pairs are passed to the contrastive learning loss.

Discrete Contrastive Learning. As mentioned in the section of the introduction, the unstable training problem hinders contrastive learning from releasing its greatest potential to further sample efficiency of RL. Our DIB is proposed using discrete state representation and the information bottleneck approach to address this problem.

We first introduce a probabilistic discrete representation layer into our model. Specifically, given a key or a query v_t^k ($t = \{1, 2\}$), we first compute the probability:

$$p_t^k = \sigma(f_{\theta_t}(v_t^k)) \quad (7)$$

where σ is the sigmoid function and is applied to its argument in an element-wise way. Then, the discrete representations of v_t^k are

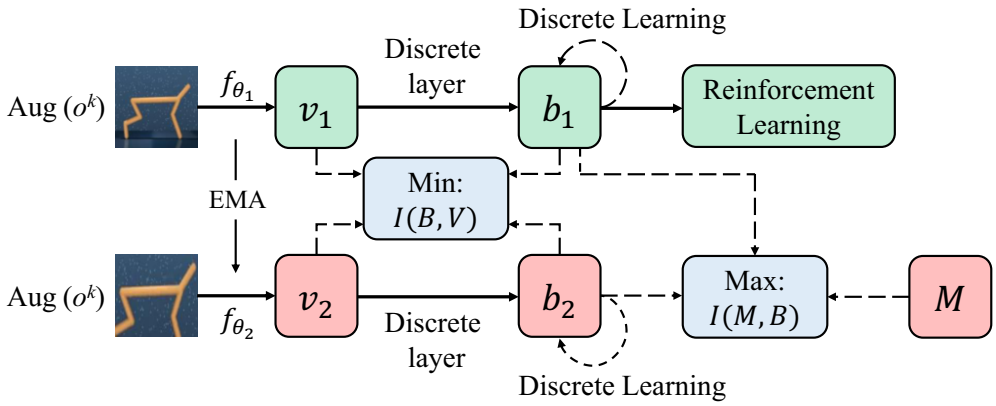


Figure 1: Illustration of the proposed DIB method. The pixel observations are data-augmented twice and then fed into the encoder f_θ to form query v_1 and key v_2, respectively. A discrete layer is introduced to generate the discrete representation b_1 and b_2 for query and key. The generated b_1 and b_2 are optimal by a non-gradient discrete learning algorithm. Only the queries are passed to the RL algorithm. During the gradient update step, the query encoder f_{θ_1} is trained via maximizing $I(M,B)$ and simultaneously minimizing $I(B,V)$. The key encoder f_{θ_2} weights are the moving average (EMA) of the query weights similar to MoCo [13].

Table 1: Main notations used in this paper

Notation	Description
o	training instances (observation)
v_1	query
v_2	key
z	continuous state representation
b	discrete state representation
m	similarity indicator
f_θ	query/key encoder
N	batch size

generated by sampling from the multivariate Bernoulli distribution as:

$$b_t^k \sim Bernoulli(p_t^k) \tag{8}$$

where the d-th element $[b_t^k]_d \in \{-1, 1\}$ is generated according to the corresponding probability $[p_t^k]_d$. The range of d is the length of the vector b_t^k. Due to the non-differentiability of discrete representation, we cannot directly apply existing contrastive loss, e.g., InfoNCE, to learn compact representation. Hence, we choose to design a discrete learning module for learning compact discrete state representation.

Let $\Theta_{ij} = \lambda b_1^i b_2^{j(T)}$ be the inner product of b_1^i and b_2^j, where λ is a hyper-parameter denoting a scale factor for tuning. To achieve the goal of contrastive learning, we want Θ_{ij} to be as large/small as possible if v_1^i and v_2^j come from the same/different observation o. For example, if $i = j$, Θ_{ij} should be large and vice versa. We define the likelihood of the similarity indicators m as follows:

$$p(m|\mathcal{V}) = \prod_{i,j=1}^{N} p\left(m_{ij}|v_1^i, v_2^j\right) \tag{9}$$

where $\mathcal{V} \triangleq \{v_1^1, v_2^1, v_1^2, v_2^2, ..., v_1^N, v_2^N\}$ is the set of query and key in a minibatch. $m_{ij} = 1$ indicates that v_1^i and v_2^j come from the same observation o while $m_{ij} = 0$ means that v_1^i and v_2^j are dissimilar. The probability $p(m_{ij}|v_1^i, v_2^j)$ is defined as follows:

$$p\left(m_{ij}|v_1^i, v_2^j\right) = \begin{cases} A_{ij}, & \text{if } m_{ij} = 1, \\ 1 - A_{ij}, & \text{if } m_{ij} = 0. \end{cases} \tag{10}$$

where we define $A_{ij} = \frac{1}{1+e^{-\Theta_{ij}}}$. Then the log-likelihood can be derived as the following equation:

$$\begin{aligned} L_{cl} &= -\frac{1}{N} \log p\left(m|\mathcal{V}\right) \\ &= -\frac{1}{N} \sum_{i,j=1}^{N} \log A_{ij}^{m_{ij}} (1 - A_{ij})^{1-m_{ij}} \\ &= -\frac{1}{N} \sum_{i,j=1}^{N} [m_{ij}\Theta_{ij} - \log(1 + e^{\Theta_{ij}})] \end{aligned} \tag{11}$$

We call Eq. (11) the discrete contrastive loss. Eq. (11) reveals that minimizing the discrete contrastive loss will make the inner product Θ_{ij} of b_1^i and b_2^j as large as possible, if they come from the same observation o^k. Subsequently, the learned b_1^i and b_2^j will be more and more similar during training. Hence, this property is reasonable, and it matches the goal of contrastive learning, as well. Moreover, compared with continuous representation, our discrete representation can make the query and its positive key closer. We will demonstrate this property in the section of experiments.

Discrete Learning Algorithm. As mentioned above, since the discrete representation is not differentiable, it is much difficult to directly optimize b_t^k with the whole model. Therefore, we use a surrogate strategy to update b_1 and b_2. Specifically, we first derive a lower bound of -Eq. (11) as follows:

$$\widetilde{L}_{cl}([b_t]_d) = [b_t]_d \left(\frac{\partial L}{\partial [b_t]_d} - H[b_t]_d\right) \tag{12}$$

where $[b_t]_d$ is the d-th element of b_t, $H = -N\lambda^2 I$, and I is the identity matrix. Then we adopt a widely used optimization principle, i.e., maximizing the lower bound, to solve the discrete contrastive loss. Due to the limit of space, we provide the proof of (12) in Appendix.

We here directly give the solution of maximizing (12) as follows:

$$[\bar{b}_t]_d = sign\left(\frac{\partial L}{\partial [b_t]_d} - H[b_t]_d\right) \tag{13}$$

where $sign([x]_d) = 1$, ($[x]_d > 0$) and $sign([x]_d) = -1$, ($[x]_d < 0$).

After optimization, we use $\bar{b}_1$ and $\bar{b}_2$ to represent the final solution of b_1 and b_2, and they serve as a regression target for the encoder $f_{\theta_1}(\cdot)$ and $f_{\theta_2}(\cdot)$. We use the mean square error as the regression loss, as shown in Eq. (14)

$$L_1 = \|f_{\theta_1}(v_1) - \bar{b}_1\|_F^2 + \|f_{\theta_2}(v_2) - \bar{b}_2\|_F^2 \tag{14}$$

For the query encoder f_{θ_1} parametrized by θ_1 and key encoder f_{θ_2} parametrized by θ_2, we perform the update $\theta_2 = \tau\theta_2 + (1-\tau)\theta_1$ and encode any key using Stop Gradient.

4.2 Connections to Information Bottleneck

In this section, we demonstrate that our discrete contrastive loss can be connected to mutual information. Furthermore, we show that our learning approach conforms to the learning principle of the technique of broader information bottleneck.

Connections to Mutual Information. Considering that b_t^k is the discrete stochastic features of v_t^k, we take expectation over the loss in (11) and obtain the loss as:

$$L_{cl} = -\mathbb{E}_{p_{(\mathcal{B}|\mathcal{V})}}[\log p(m|\mathcal{V})] \tag{15}$$

where $\mathcal{B} \triangleq \{b_1^1, b_2^1, b_1^2, b_2^2, ..., b_1^N, b_2^N\}$ is the set of discrete representation of $\mathcal{V}$ and $p_{(\mathcal{B}|\mathcal{V})} = \prod_{i=1}^{N}\prod_{t=1}^{2} p(b_t^i|v_t^i)$. The loss function in (15) only considers the situation under a minibatch. Usually, the loss should be optimized over all samples from the replay buffer. Without loss of generality, using $p(v,m)$ to denote the distribution of all query-key sets, the discrete contractive loss averaged over all query-key is:

$$\begin{aligned} L_{cl} &= -\mathbb{E}_{p_{(v,m)}}\mathbb{E}_{p_{(\mathcal{B}|\mathcal{V})}}[\log q(m|b)] \\ &= -\int p(b,m)\log q(m|b)dmdb \end{aligned} \tag{16}$$

where $p(b,m) = \int p(v,m)p(b|v)dv$. The distribution $q(m|b) = \prod_{k\in\mathcal{B}} q(m|b_1^k, b_2^k)$, where $\mathcal{B}$ is the replay buffer, defined as:

$$q(m|b_1,b_2) = \begin{cases} \dfrac{1}{1+\mathrm{e}^{-\lambda b_1 b_2^T}}, & \text{if } m=1, \\ \dfrac{\mathrm{e}^{-\lambda b_1 b_2^T}}{1+\mathrm{e}^{-\lambda b_1 b_2^T}}, & \text{if } m=0. \end{cases} \tag{17}$$

Using the fact that the KL-divergence is always positive [1], we can derive the inequality $\int p(b,m)\log p(m|b)dmdb > \int p(b,m)\log q(m|b)dmdb$. According to this inequality, we have:

$$L_{cl} \geq -\int p(b,m)\log p(m|b)dmdb = H(M|B) \tag{18}$$

where M and B denote the random variables of similarity and discrete state representations, respectively. Then, subtracting entropy $H(M)$ on both sides, we obtain:

$$L_{cl} - H(M) \geq -I(M,B) \tag{19}$$

Algorithm 1: DIB for Reinforcement Learning

```
Initialize the query and key encoders f_{θ1} and f_{θ2}.
Initialize the parameters φ for the base RL algorithm.
A replay buffer R = ∅.
Given a batch of training samples N, the loss function for the
  base RL algorithm is L_RL(f_{θ1}, φ; N).
for t = 1, 2, ..., T do
    Rollout the current policy and store the samples to replay
      buffer R;
    Draw a batch of samples N from R;
    Update the parameters with the loss fuction:
      L_RL + L_DIB.
end
```

where the equality $I(M,B) = H(M) - H(M|B)$ is used. From Eq. (19), $L_{cl} - H(M)$ is actually an upper bound of the negative mutual information, $-I(M,B)$. Since $H(M)$ is a constant, minimizing the loss L_{cl} is equivalent to minimizing the upper bound of negative mutual information. Therefore, we can conclude that the proposed discrete contrastive loss, L_{cl}, essentially maximizes the mutual information, i.e., $\max_\theta I(M,B)$.

Learning under IB. Information bottleneck (IB) [30] is to maximize the mutual information between the representations and their label. The objective of IB is to maximize L_{IB} as follows:

$$L_{IB} = I(M,B) - \beta I(B,V) \tag{20}$$

where β is a Lagrange multiplier, V the random variable of the input. Apparently, our DIB can be understood under the IB framework by setting β to 0. According to recent work [1], better semantic representations can be obtained if an appropriate value of β is selected. Subsequently, we can train our model under the broader IB framework by taking the $I(B,V)$ into account. We seek to maximize the lower bound of $I(B,V)$ due to its computational intractability. According to [1], we can obtain:

$$I(B,V) \leq \mathbb{E}_{p(v)}[KL(p(b|v)|q(b))] \tag{21}$$

where $q(b)$ can be any distribution of b. Combing (18) and (21), we can obtain the lower bound for L_{IB}:

$$L_{IB} \geq -L_{cl} - \beta\mathbb{E}_{p(v)}[KL(p(b|v)|q(b))] + H(M) \tag{22}$$

In our experiments, we set $q(b) = p(b|v_2^k)$ for query v_1^k; and $q(b)$ for key v_2^k can be defined similarity. Through making close the encoding distributions from different views of the same observation, our model is able to eliminate redundant information from different augmentations. Therefore, the overall objective function of DIB can be formulated in the following:

$$L_{DIB} = L_1 + \beta\mathbb{E}_{p(v)}[KL(p(b|v)|q(b))] \tag{23}$$

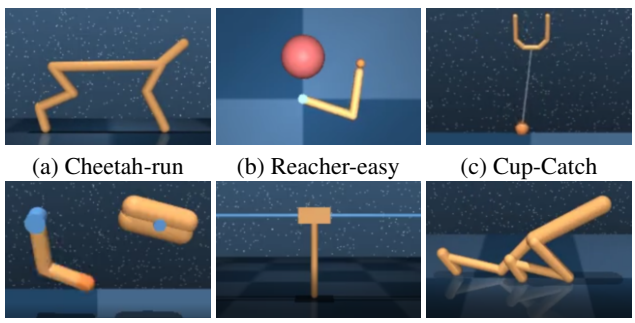
Intuitively, by encouraging the encoding distributions from different views of the same observation close to each other, the model can eliminate superfluous information from each view.

4.3 Incorporating DIB in RL

The training architecture is shown in Figure 1. We adopt DIB as an auxiliary task (as described in Algorithm 1) that helps the agent to

Table 2: Performance comparison of DIB and baselines on DeepMind Control suite at 100K and 500K environment steps. The results of all methods show the mean and standard deviation averaged three runs with different seeds.

500K Environment Steps	DIB (Ours)	Dreamer-v2	DrQ-v2	CURL	SAC-state
Finger-Spin	**983 ± 1**	415 ± 111	788 ± 123	815 ± 90	820 ± 20
Cartpole-Swingup-Sparse	**757 ± 24**	742 ± 24	383 ± 343	14 ± 9	755 ± 12
Reacher-Easy	**978 ± 3**	664 ± 75	751 ± 217	789 ± 107	788 ± 357
Cheetah-Run	592 ± 38	538 ± 98	**602 ± 33**	272 ± 35	454 ± 41
Walker-Run	**593 ± 61**	434 ± 61	436 ± 136	198 ± 21	632 ± 40
Cup-Catch	**968 ± 5**	749 ± 246	843 ± 172	739 ± 137	969 ± 5
100K Environment Steps					
Finger-Spin	**806 ± 72**	324 ± 52	325 ± 292	582 ± 48	621 ± 55
Cartpole-Swingup-Sparse	**31 ± 23**	20 ± 13	22 ± 7	0 ± 0	533 ± 229
Reacher-Easy	**545 ± 56**	157 ± 60	249 ± 138	335 ± 130	729 ± 76
Cheetah-Run	**447 ± 28**	253 ± 65	273 ± 123	145 ± 51	203 ± 13
Walker-Run	**140 ± 18**	120 ± 21	103 ± 63	76 ± 19	246 ± 52
Cup-Catch	**616 ± 199**	318 ± 212	358 ± 228	572 ± 271	949 ± 7



(a) Cheetah-run (b) Reacher-easy (c) Cup-Catch

(d) Finger-spin (e) Cartpole-swingup (f) Walker-walk

Figure 2: The examples of continuous control tasks from the DMC.

learn a useful low-dimensional state representation. We simultaneously train our DIB with the RL agent by adding L_{DIB} as an auxiliary objective during training. The policy and value function in RL can directly take the representation of state to allow the models in RL to backprop to our encoder. This further improves the performance by accommodating the learned representations with the policy/value function. We demonstrate the effectiveness of DIB by building the agents on top of SAC in continuous control tasks in the next section.

5 Experiments

In this section, we implement DIB on the commonly used continuous control tasks from DeepMind Control Suite (DMC) [29]. We first present a comparison to prior methods, both model-free and model-based, in terms of sample efficiency and performance. We then present some ablation studies that guided the final version of DIB. Finally, we analyze the representations learned by DIB and CURL to explain why our method works.

5.1 Setups

Environments. We evaluate our DIB and the baselines on continuous control tasks using simulated robots from DMC, a widely used benchmark for visual RL. We choose the set of tasks based on those considered in CURL [20]. Specifically, we replace the Cartpole Swingup and Walker Walk with their more challenging counterparts, including Cartpole Swingup Sparse and Walker Run, and keep the remaining tasks.

Baselines. For DeepMind Control Suite with image inputs, we compare the following methods:

- Dreamer-v2 [11], a leading model-based RL agent that learns behaviors from a compact latent space of a discrete world model.
- CURL [20], performing off-policy control on top of the features extracted from the raw pixels using contrastive learning.
- DrQ-v2 [33], an off-policy actor-critic approach that uses data augmentation to learn policies directly from pixels.
- SAC-state [8], a basic actor-critic method using low-level state-based features as input.

Implementation Details. We use the SAC algorithm as the objective algorithm to work with DIB. This new algorithm is built on top of the publicly released implementation from CURL [20]. We add a discrete layer with a dimension of 64 in the state encoder for discrete state representation learning. The scale factor λ is set to 0.1, and the Lagrange multiplier is set to 0.05. The batch size for DIB is set to 256, while for the other baselines it is set to 512. Per common practice [20, 33], we employ the same action repeat of CURL and measure performance/sample efficiency in the environment steps, rather than the actor steps. The complete list of hyper-parameters is in the Appendix. All the experiments are performed on a computer with Intel I7- 4790K@4.0GHz CPU, GTX1080Ti GPU, and 64GB RAM. Each seed for DMC benchmarks takes about 3 days to finish.

Evaluation Protocol. In this work, we measure the *Performance* and *Sample efficiency* of our method and baselines.

- In terms of performance, we follow the evaluation protocol of CURL [20]. We evaluate performance by measuring the ratio of the episode returns achieved by DIB versus the baselines at 100k and 500k environment steps.
- In terms of sample efficiency, we follow the evaluation protocol of DrQ-v2 [33]. We facilitate this comparison by computing an algorithm's performance measured by episode return with respect to environment steps. For each task, we train each model for 1M environment steps. Specifically, for CURL and SAC-state, we train

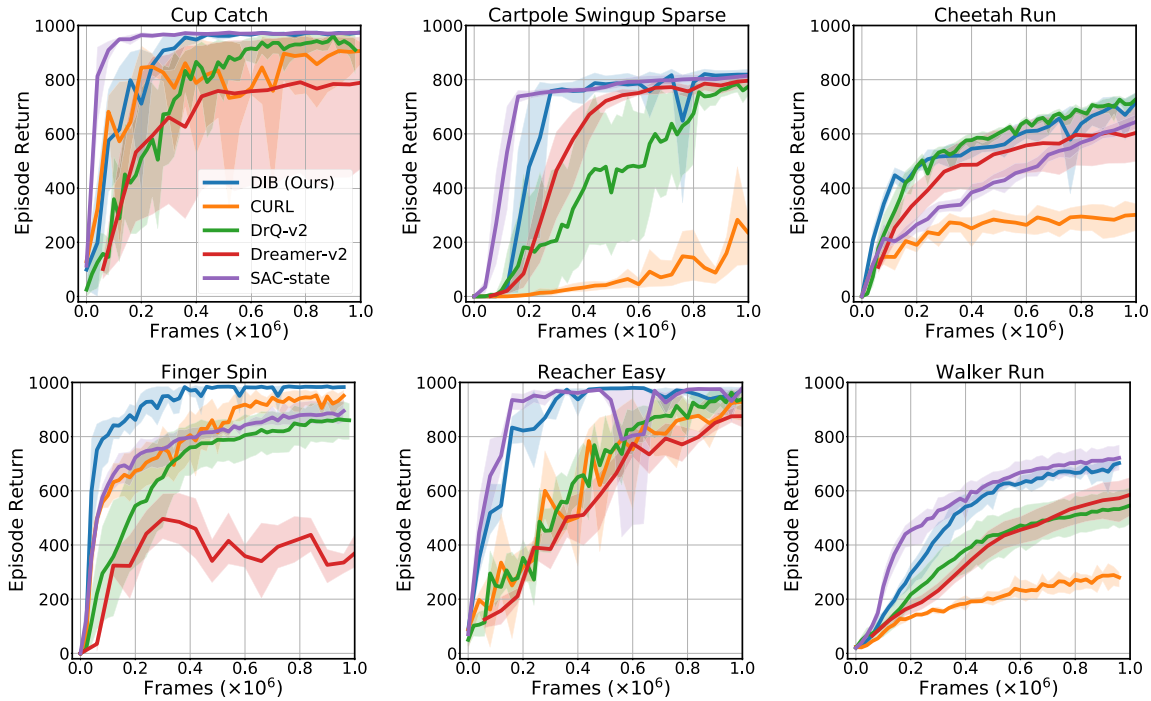


Figure 3: Learning Curves in DMC environment. DIB significantly outperforms all visual baselines on Cup Catch, Finger Spin, Walker Run, and Reacher easy.

them for 1M environment steps (about 125-500k actor steps with different action repeat). For Dreamer-v2 and DrQ-v2, the open-source implementations provide learning curves for DMC tasks.

In all figures and tables, the mean and standard deviation are computed from three random seeds.

5.2 *Comparison to Baselines*

We show the performance scores and learning curves in Table 2 and Figure 3 achieved by DIB and baselines.

Performance Comparison. From Table 2, we can see that DIB is the only visual RL method that achieves comparable or even better performance than state-based SAC on all DMC tasks. Specifically, in the task on Finger spin, Cartpole swingup sparse, and Cup Catch, DIB operating purely from pixels outperforms SAC operating from the low-dimensional physical state. We also notice that the standard deviation of DIB is significantly lower than the other visual baselines. This low standard deviation of our DIB indicates that 1) Our DIB can easily achieve close to optimal behavior; 2) The proposed discrete state representation can improve the training stability in the continuous control task of DMC under a small batch size. Therefore, our results suggest for the first time that discrete state representation can be beneficial to modal-free RL agents for continuous control tasks.

Sample Efficiency Comparison. The results in Figure 3 reveal that DIB outperforms model-free and the leading model-based methods in terms of sample efficiency across the six benchmarks. In comparison with CURL, a basic contrastive learning-based RL method, one can notice that DIB converges faster, is more stable, and has a smaller variance. Notably, the advantage of DIB is more pronounced on more challenging tasks (Cartpole Swingup Sparse, Cheetah Run, and Walker Run), where the exploration is especially challenging. As DIB can solve visual control problems with a small batch size,

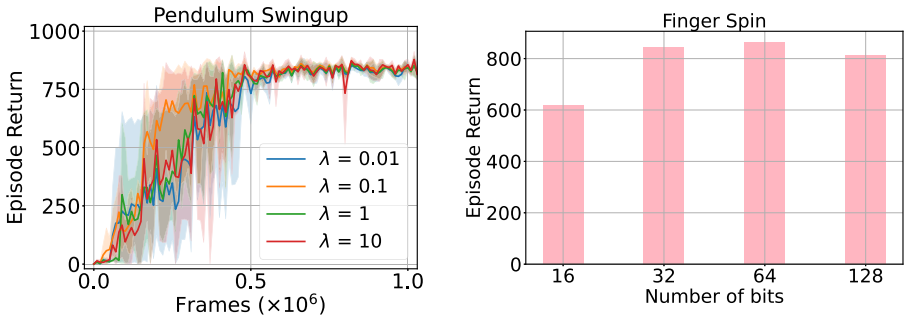
Table 3: Performance comparison between DIB and its variants on DeepMind Control suite at 100K and 500K environment steps. The results of all methods show the mean and standard deviation averaged three runs.

500K Environment Steps	DIB	DIB$_\alpha$	DIB$_\beta$
Finger-spin	**983** $\pm$ **1**	944 $\pm$ 32	803 $\pm$ 104
Cartpole-swingup-sparse	**757** $\pm$ **24**	699 $\pm$ 19	29 $\pm$ 15
Reacher-easy	**978** $\pm$ **3**	896 $\pm$ 95	825 $\pm$ 121
Cheetah-run	**592** $\pm$ **38**	551 $\pm$ 41	331 $\pm$ 23
Walker-run	**593** $\pm$ **61**	546 $\pm$ 103	177 $\pm$ 30
Cup-catch	**968** $\pm$ **5**	963 $\pm$ 13	784 $\pm$ 159
100K Environment Steps			
Finger-spin	**806** $\pm$ **72**	669 $\pm$ 103	610 $\pm$ 31
Cartpole-swingup-sparse	**31** $\pm$ **23**	14 $\pm$ 11	5 $\pm$ 8
Reacher-easy	**545** $\pm$ **56**	370 $\pm$ 73	352 $\pm$ 121
Cheetah-run	**447** $\pm$ **28**	342 $\pm$ 13	301 $\pm$ 57
Walker-run	**140** $\pm$ **18**	135 $\pm$ 8	88 $\pm$ 20
Cup-catch	**616** $\pm$ **199**	587 $\pm$ 147	550 $\pm$ 243

our algorithm does not require high hardware demands and can run on a single GTX1080Ti GPU. This demonstrates that DIB is superior to existing contrastive learning-based RL algorithms in terms of computational efficiency.

5.3 *Parameter Analysis*

There are several key parameters, i.e., λ and the bits number of discrete representation, in the proposed DIB. To study the influence of parameters, we conduct experiments by varying the value of one parameter while fixing the others. Figure 4 (a) shows the learning curves on Pendulum Swingup task and (b) shows the episode return for 100k interactions in the finger spin task. We observe that promising results could be expected as long as λ and bit number are chosen properly, i.e., $\lambda = 0.1$ and $bits = 64$. Our experience demonstrate



(a) (b)

Figure 4: Performance influence of parameter λ and different bits of discrete representations.

that compact representations can achieve better control in complex visual environments. A high feature dimension may introduce additional noise, leading to decision-making errors by the agent.

5.4 *Ablation Study*

To further verify the effectiveness of the proposed method, we here compare with several baselines which are special cases of DIB: (1) DIB_α represents DIB without the IB loss; (2) DIB_β indicates DIB without the discrete contrastive loss, which performs off-policy control on top of the continuous feature. We train the ablated versions in DMC tasks. The performance scores shown in Table 3 demonstrate that both components are essential and effective for visual RL. We also find that DIB_β outperforms CURL on most DMC tasks, as the IB loss we designed in Eq. (22), can learn a concise representation by eliminating superfluous information from query and key. In addition, these results demonstrate that a good representation does improve the performance of visual RL.

5.5 *Representation Analysis*

We analyze the learned representation of DIB and CURL to demonstrate that our method attains a better representation, which may explain why our discrete representation succeeds. We show the cosine similarity of the representations between positive query-key pairs during the training on the task of Cheetah Run in Figure 5. First, the results in Figure 5 and the learning curves in Figure 3 show that when a good policy is learned, the representations of positive pairs learned by contrastive learning are similar. This may indicate that the contrastive learning-based RL algorithm learns policy better when the contrastive learning converges faster. In comparison to CURL, we find that the representations of the positive query-key pairs in DIB are more similar and converge faster. Therefore, we conclude that our contrastive loss can significantly accelerate the convergence of contrastive learning in RL, which makes DIB learn faster.

5.6 *Training Time*

Since our DIB requires an additional discrete learning algorithm, there may be concerns about the increase in training cost. To address this, we compare the training time of CURL and DIB. In Table 4, we report the training times of CURL and DIB for 1K interaction steps (which includes both interactions with the environments and updates of model parameters) across 6 DMControl tasks. Under the same batch size setting (512), DIB is on average 8% more computationally efficient than CURL.

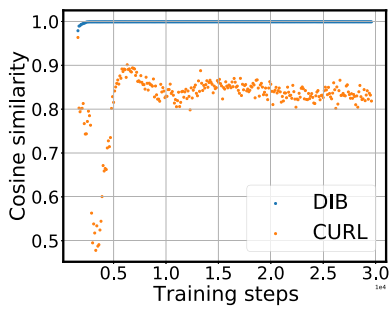


Figure 5: The cosine similarity between the representations of the query and its positive key during the training of CURL and DIB.

Table 4: Comparison of training time (1K interaction steps) between CURL and DIB.

Task	CURL (512)	DIB (512)
Finger-spin	163.1	149.1(−8.6%)
Cartpole-swingup-sparse	160.2	146.1(−8.8%)
Reacher-easy	162.10	147.7(−9.2%)
Cheetah-run	171.6	157.2(−8.1%)
Walker-run	177.9	164.9(−7.3%)
Cup-catch	165.4	151.3(−8.4%)

6 Conclusion

In this paper, we propose DIB, a discrete information bottleneck approach for RL. We design a new discrete contrastive loss to enable our model to learn more discrete state representations. Through this loss, our method can carry out stable algorithm training without a large batch size. From a theoretical perspective, we demonstrate the strong connection between the proposed discrete contrastive loss and mutual information. We have proven that the learning approach conforms to the learning principle of the information bottleneck theory. Extensive experiments on the DeepMind control suite have shown that our DIB can achieve state-of-the-art sample efficiency and performance on continuous control tasks. Our DIB provides a starting point to extend such discrete representation improvement to the model-free setting, as well.

7 Acknowledgements

This paper was mainly supported by the National Natural Science Foundation of China (NSFC) (Grant No. 62272497 to H. Wu). This work was also supported by Science and Technology Program of Guangzhou, China (No. 202002020045).

References

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy, 'Deep variational information bottleneck', in *ICLR*, (2017).

[2] Chenjia Bai, Lingxiao Wang, Lei Han, Animesh Garg, Jianye Hao, Peng Liu, and Zhaoran Wang, 'Dynamic bottleneck for robust self-supervised exploration', in *Neural Information Processing Systems*, (2021).

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton, 'A simple framework for contrastive learning of visual representations', *arXiv*, **abs/2002.05709**, (2020).

[4] Fei Deng, Ingook Jang, and Sungjin Ahn, 'Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations', in *ICML*, volume 162, pp. 4956–4975, (2022).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: pre-training of deep bidirectional transformers for language understanding', in *NAACL*, pp. 4171–4186, (2018).

[6] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata, 'Learning robust representations via multi-view information bottleneck', in *ICLR*, (2020).

[7] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch, 'Som-vae: Interpretable discrete representation learning on time series', in *ICLR*, (2019).

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine, 'Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor', in *ICLR*, (2018).

[9] Danijar Hafner, T. Lillicrap, Ian S. Fischer, Ruben Villegas, David R Ha, Honglak Lee, and James Davidson, 'Learning latent dynamics for planning from pixels', *arXiv*, **abs/1811.04551**, (2019).

[10] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi, 'Dream to control: Learning behaviors by latent imagination', in *ICLR*, (2020).

[11] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba, 'Mastering atari with discrete world models', in *ICLR*, (2021).

[12] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap, 'Mastering diverse domains through world models', *arXiv preprint arXiv:2301.04104*, (2023).

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick, 'Momentum contrast for unsupervised visual representation learning', in *CVPR*, pp. 9726–9735, (2020).

[14] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama, 'Learning discrete representations via information maximizing self-augmented training', in *ICML*, volume 70, pp. 1558–1567, (2017).

[15] Qing-Yuan Jiang and Wu-Jun Li, 'Deep cross-modal hashing', *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3270–3278, (2016).

[16] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, K. Czechowski, D. Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, G. Tucker, and Henryk Michalewski, 'Model-based reinforcement learning for atari', *ArXiv*, **abs/1903.00374**, (2020).

[17] Jaekyeom Kim, Minjung Kim, Dongyeon Woo, and Gunhee Kim, 'Drop-bottleneck: Learning discrete compressed representation for noise-robust exploration', *ArXiv*, **abs/2103.12300**, (2021).

[18] Ilya Kostrikov, Denis Yarats, and R. Fergus, 'Image augmentation is all you need: Regularizing deep reinforcement learning from pixels', *arXiv*, **abs/2004.13649**, (2021).

[19] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas, 'Reinforcement learning with augmented data', in *NeurIPS*, (2020).

[20] Michael Laskin, Aravind Srinivas, and Pieter Abbeel, 'CURL: contrastive unsupervised representations for reinforcement learning', in *ICML*, volume 119, pp. 5639–5650, (2020).

[21] Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine, 'Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model', in *NeurIPS*, (2020).

[22] Guoqing Liu, Chuheng Zhang, Li Zhao, Tao Qin, Jinhua Zhu, Jian Li, Nenghai Yu, and Tie-Yan Liu, 'Return-based contrastive representation learning for reinforcement learning', in *ICLR*, (2021).

[23] Sebastian Risi and Kenneth O. Stanley, 'Deep neuroevolution of recurrent and discrete world models', in *GECCO*, pp. 456–462, (2019).

[24] Florian Schroff, Dmitry Kalenichenko, and James Philbin, 'Facenet: A unified embedding for face recognition and clustering', in *CVPR*, pp. 815–823, (2015).

[25] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman, 'Data-efficient reinforcement learning with self-predictive representations', in *ICLR*, (2021).

[26] Younggyo Seo, Kimin Lee, Stephen L. James, and Pieter Abbeel, 'Reinforcement learning with action-free pre-training from videos', in *ICML*, pp. 19561–19579, (2022).

[27] Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell, 'Loss is its own reward: Self-supervision for reinforcement learning', in *ICLR*, (2017).

[28] Adam Stooke, Kimin Lee, P. Abbeel, and M. Laskin, 'Decoupling representation learning from reinforcement learning', *arXiv*, **abs/2009.08319**, (2021).

[29] Yuval Tassa, S. Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess, 'dm_control: Software and tasks for continuous control', *Softw. Impacts*, **6**, 100022, (2020).

[30] Naftali Tishby, Fernando C Pereira, and William Bialek, 'The information bottleneck method', *ArXiv*, **physics/0004057**, (2000).

[31] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, 'Representation learning with contrastive predictive coding', *ArXiv*, **abs/1807.03748**, (2018).

[32] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, 'Neural discrete representation learning', *ArXiv*, **abs/1711.00937**, (2017).

[33] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto, 'Mastering visual continuous control: Improved data-augmented reinforcement learning', in *ICLR*, (2022).

[34] Denis Yarats, A. Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and R. Fergus, 'Improving sample efficiency in model-free reinforcement learning from images', in *AAAI*, (2021).