Contrastive Inductive Bias Controlling Networks for Reinforcement Learning

Dongxu Li LIDONGXU6C@MAIL.USTC.EDU.CN
Shaochen Wang SAMWANG@MAIL.USTC.EDU.CN
Kang Chen CK6@MAIL.USTC.EDU.CN
Bin Li BINLI@MAIL.USTC.EDU.CN
University of Science and Technology of China

Abstract

Effective learning in an visual-based environment is essential for reinforcement learning (RL) agent, while it has been empirically observed that learning from high dimensional observations such as raw pixels is sample-inefficient. For common practice, RL algorithms for image input often use encoders composed of CNNs to extract useful features from high dimensional observations. Recent studies have shown that CNNs have strong inductive bias towards image styles rather than content (i.e. agent shapes), while content is the information that RL algorithms should focus on. Inspired by this, we suggest reducing the intrinsic style bias of CNNs by proposing Contrastive Inductive Bias Controlling Networks for RL. It can help RL algorithms effectively focus on truly noteworthy information like agents' own characteristics. Our approach incorporates two transfer networks and feature encoder with contrastive learning methods, guiding RL algorithms to learn more efficiently with sampling. Extensive experiments show that the extended framework greatly enhances the performance of existing model-free methods (i.e. SAC), enabling it to reach state-of-the-art performance on the DeepMind control suite benchmark.

Keywords: Reinforcement learning, Contrastive learning, Inductive bias, Style transfer

1. Introduction

Deep reinforcement learning algorithms direct end-to-end training from pixels are promising and meaningful, which play an important role in the field of control and robotics (Andrychowicz et al., 2020). Notable success has been achieved in many areas including video games (Abadi et al., 2016), autonomous driving (Lillicrap et al., 2015) and robots (Kalashnikov et al., 2018).

In the past few years, academia has come a long way on tackling the inefficiency of reinforcement learning from high dimensional observations(i.e. pixels). Research methods can be mainly divided into two categories: (i) Auxiliary tasks to help the perception of agents; (ii) Predicting model of the future world. The former class of methods uses auxiliary tasks to learn better representations, including alternate tasks (Dwibedi et al., 2018), reconstruction tasks (Jaderberg et al., 2016) self-prediction (Schwarzer et al., 2020), or contrastive learning (Srinivas et al., 2020). However, these auxiliary tasks are not directly related to the goal of training and therefore remove of obtaining the truly needed representations for RL algorithms.

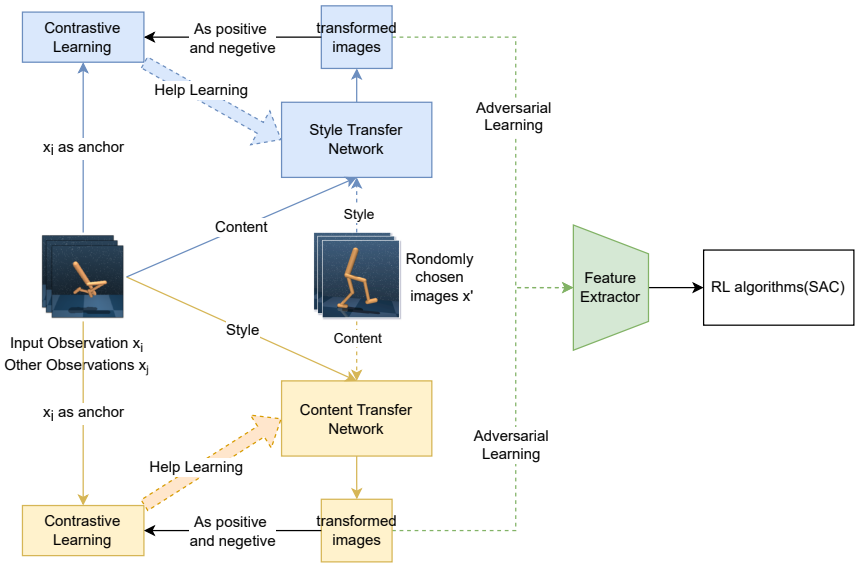

Figure 1: Our Contrastive Inductive Bias Controlling (CIBC) Networks. The framework consists of three main sub-components: a style-transfer network, a content-transfer network, and a subsequent feature extractor. Two transfer networks transform the input observations and jointly train adversarially to the subsequent feature extractor. The feature extractor is trained to help the reinforcement learning algorithm focus on the truly noteworthy parts of the input information.

For the first category of methods, training a convolutional encoder alongside the value and policy networks is a frequently used method for visual RL algorithms (Yarats et al., 2019; Srinivas et al., 2020). However most visual RL algorithms are simple exploitation of encoders migrated from the field of computer vision. Actually, in contrast to the strong adaptability of human visual recognition systems, CNNs are vulnerable to different styles of images among different domains. Recently a series of studies have shown that standard CNNs have an inductive bias that differs from human vision. It can be explained by the fact that while people prefer to use content to recognize objects (Landau et al., 1988), CNNs prefer to use style(i.e., texture) (Baker et al., 2018; Geirhos et al., 2018; Hermann et al., 2020). For this reason, the encoder composed of CNNs before visual RL tends to focus on the style of the high dimensional observations(i.e. pixels) rather than the agents' own information for pixel input in RL. According to the general view, visual RL algorithms need to acquire some characteristics of the agent itself in the pixels information for training, which can also be considered as the content of the picture. However, it is the information

that the encoders prefer less. Since RL algorithms require content information of the agent from input pixels rather than style information preferred by the encoder, it will definitely affect the efficiency of visual RL algorithms.

To solve the problem of inductive bias presented in the encoder and the mismatch between auxiliary task objectives and subsequent tasks, we propose Contrastive Inductive Bias Controlling (CIBC) Networks for reinforcement learning showed in figure 1. We attempt to eliminate the problem of induction bias by means of adversarial learning. It is a feasible way to obtain networks that focus on unnecessary information as adversarial targets by style transfer methods (Nam et al., 2021). The whole framework combines two transfer networks with contrastive method on top of a feature extractor. Contrastive learning is a self-supervised learning method, which is introduced into the framework to help the framework better learn the representations of different transfer network outputs. The style-transfer network and the content-transfer network work together adversarially to subsequent feature extractor. Both transfer networks and feature extractors are trained with contrastive learning approaches. The training goal of transfer networks is to make the subsequent feature extractor focus on interfering features such as image style, while the goal of the feature extractor is to get rid of the interference of the previous two parallel transfer networks. Such a network framework allows the elimination of the inductive bias that arises when simply utilizing encoders, which in turn improves the efficiency of subsequent RL algorithms.

This paper makes the following key contribution: We present a simple and highly portable framework CIBC network, controlling the inductive bias problem of visual RL algorithms due to the use of encoders composed of CNNs. It has the following advantages: Firstly, it can be seamlessly integrated with the training pipeline of most previous RL algorithms and does not require the introduction of multiple additional hyperparameters. Secondly, as a approach of the first type but distinct from previous RL algorithms, it is closely integrated with the original RL algorithms. Our approach focuses on information that is more relevant to subsequent RL algorithms: agents' own characteristics, which is a proven effective direction worthy of auxiliary missions' attention.

Our approach substantially improves the sample efficiency of subsequent RL algorithms with minimal changes to the RL architecture. It obtains state-of-the-art performance over prior state-of-the-art model-free method DrQ-v2 (Yarats et al., 2021a) across most tasks on DeepMind Control Suite (Tassa et al., 2018) benchmark in terms of sample efficiency. In addition, the performance is comparable to that of the model-based methods in some tasks.

2. Related work

In this section, we provide a brief introduction to the relevant work our model build on.

2.1. Inductive biases of CNNs

Baker et al. (2018) presented a trained Deep CNN with object silhouettes that preserved overall shape but were filled with surface texture taken from other objects, and they found that the shape cues seemed to play little or none role in the classification of animal pictures. Geirhos et al. (2018) observed that trained CNNs are more likely to make style-biased decisions for ambiguous stimuli (e.g. images that are stereotyped as different categories).

In addition, recent studies have also pointed out that convolutional neural networks perform poorly when only global shapes are given but local textures are not given (Ballester and Araujo, 2016; Geirhos et al., 2018). Our work takes inspiration from the tendency of CNNs to learn image style rather than image content.

2.2. Style Processing

We leverage convolutional feature statistics to realize style processing and build style transfer and content transfer networks. This is attributed to previous work on processing feature statistics in CNNs to change the image style. Gatys et al. (2015) showed that the feature statistics of CNN can effectively capture the stylistic information of images. Huang and Belongie (2017) proposed Adaptive instance normalization (AdaIN) and demonstrated that tuning the mean and variance of the convolutional feature map can easily alter the style of the image. Nam et al. (2021) proposed Style-Agnostic Networks to reduce the style bias and made the model more robust under domain shift. An auxiliary style-biased network G_s was built in this network utilizing a content randomization module which was also a variant of AdaIN. Karras et al. (2019) proposed Style-GAN and obtained impressive image generation results by repeatedly applying the AdaIN operation in the generative network.

2.3. Contrastive Learning

Contrastive Learning refers to learning representations that are subject to similarity restrictions. The framework is usually structured according to similar and dissimilar pairs. It can be understood as a dictionary lookup task, where positives and negatives represent a set of key corresponding to the query(or anchor). A classic application scenario of contrastive learning is instance discrimination (Wu et al., 2018). If the query and the key value are data augmentations of the same instance(i.e. image), then they are positive or negative pairs. There are some kinds of loss functions to choose from in contrastive learning. Van den Oord et al. (2018) proposed InfoNCE and achieved strong performance on four distinct domains: speech, images, text, reinforcement learning in 3D environments. Triplet was proposed by Wang and Gupta (2015) and Chopra et al. (2005) have proposed Siamese before.

2.4. Visual Reinforcement Learning

Inspired by the success of representation learning in computer vision, auto-encoders are applied to some visual reinforcement learning and have proven to be successful in practice. Works like SAC-AE (Yarats et al., 2019), SLAC (Lee et al., 2020) prove the validity of this idea in visual RL. In addition, other methods of self-supervised learning have been introduced to this area, such as contrastive learning in CURL (Srinivas et al., 2020) and ATC (Stooke et al., 2021), contrastive cluster assignment in Proto-RL (Yarats et al., 2021b) and data augmentation in RAD (Laskin et al., 2020), DrQ (Kostrikov et al., 2020) and Drq-v2 (Yarats et al., 2021a). Ye et al. (2021) proposed a sample efficient model-based visual RL algorithm EffientZero achieves 190.4% mean human performance and 116.0% median performance on the Atari 100k benchmark. All these methods mentioned above have greatly reduced the gap between state-based RL and image-based RL. As a representative of the excellent and clever model-free approaches to Visual RL, Drq-v2 (Yarats et al., 2021a) has obtained state-of-the-art performance on DeepMind Control Suite (Tassa et al., 2018).

3. Background

3.1. Reinforcement Learning from Pixels

Image-based control can be expressed in terms of an infinite-horizon partially observable Markov decision process(POMDP) (Bellman, 1957; Kaelbling et al., 1998). The tuple (O, A, p, r, γ) can be used to describe POMDPs. O represents a high-dimensional observation space(i.e. images). A represents the action space, and p is the transition dynamics $p = Pr(o'_t|o_{\leq t}, a_t)$, which means the probability distribution of next observation o'_t considering the previous observations $o_{\leq t}$ and current action a_t. r is a reward function which maps the current observation and action to a reward $r : \mathcal{O} \times \mathcal{A} = \mathcal{R}$. $\gamma \in [0, 1)$ is the discount factor. The goal of our training is to find the most appropriate policy $\pi(a_t|s_t)$ to maximize the cumulative discounted return $\mathbb{E}_\pi = r_0 + \sum_{t=1}^{\infty} \gamma^t r_t$.

3.2. Soft Actor Critic Algorithm

SAC (Haarnoja et al., 2018) is an off-policy model-free RL algorithm that aims to find the optimal policy to maximize the expected maximum-entropy trajectory returns for MDP (S, A, p, r, γ). SAC is a variant of actor-critic method which learns a policy π_ψ and two critics Q_{ϕ_1},Q_{ϕ_2} during the training process. SAC works well when the task input is from state observations, but fails to learn effective policies from pixels.

3.3. Contrastive Learning

One of the main innovations of our method is to introduce contrastive learning into the training process of the networks. Through contrastive learning, the networks can improve the ability to discriminate the style and content of high-dimensional information, so as to shift attention to where RL algorithms need to pay attention. Contrastive learning (Hadsell et al., 2006; LeCun et al., 2006; Van den Oord et al., 2018; Wu et al., 2018) often is interpreted as learning a dictionary lookup task. A query q(also referred to as anchor in contrastive learning), keys $\mathbb{K} = \{k_1, k_2, \dots\}$(also referred to as target in contrastive learning) and the partition of $\mathbb{K} : P(\mathbb{K}) = (\{k_+\}, \mathbb{K}\backslash\{k_+\})$ (positive and negative) are given for contrastive learning. The objective of contrastive learning is to guarantee that q matches k_+ with a relatively higher degree than any key in $\mathbb{K}\backslash\{k_+\}$. Dots products$(q^T k)$ (Wu et al., 2018; He et al., 2020) and bilinear products (Van den Oord et al., 2018; Henaff, 2020) are the most widely used models of the similarity between the anchor and the target. We use the InfoNCE loss (Van den Oord et al., 2018) as a criterion for contrastive learning:

$$L_q = \log \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{K-1} \exp(q^T W k_i)} \tag{1}$$

The equation 1 can be interpreted as the log-loss of the K-way classifier whose correct label is k_+.

4. Contrastive Inductive Bias Controlling Networks

We propose a Contrastive Inductive Bias Controlling Networks to help RL algorithms to be able to focus more on the agents' own characteristics with high-dimensional input informa-

tion rather than on other distracting information of pictures (e.g. style). It contains three main sub-components: a contrastive style transfer network, a contrastive content transfer network, and a feature extractor. By confusing the original image randomized style and other images randomized the same style, the style transfer network is encouraged to apply higher attention to the image style during training. For the content transfer network, it is also encouraged to apply higher attention to the image style during training, by discriminating between the original image randomized intermediate content and other images randomized with the same content but different styles. These two networks are trained together adversarially to make the feature extractor less style-biased. In this work, the statistics of CNN features are used as stylistic representations, and spatial configurations of CNNs are utilized as content representations. The main method of contrastive learning used in the article is instance discrimination.

4.1. Contrastive Style Transfer Network

In the contrastive style-transfer network, we implement the style transfer through the style randomization module. The style randomization(SR) module interpolates feature statistics between different inputs, thus randomizing the style. In the training process, the original image is used as the anchor, the original image with the style transfer is used as the positive, and other chosen images with the same style transfer are used as the negative. The InfoNCE loss (Van den Oord et al., 2018) of contrastive learning is calculated and its opposite is taken as the training loss function of the style transfer network, which reduces the similarity between the original image and the transformed image that has the same content with it but undergoned style transfer. In this way, the style transfer network is encouraged to focus more on the style of the images, so that to make the subsequent feature extractor less style-biased.

Given an initial input image x and another image x' chosen at random. We input them to the feature extractor of style transfer network G_{f_s} to obtain the intermediate feature layer y and $y' \in \mathbb{R}^{D\times H\times W}$, where H and W are the dimensions of spatial space. D is the number of channels. And then we calculate the mean and standard deviation for each channel $\mu(y)$ and $\sigma(y) \in \mathbb{R}^D$:

$$\mu(y) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} y_{hw} \tag{2}$$

$$\sigma(y) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(y_{hw} - \mu(y))^2 + \epsilon} \tag{3}$$

ϵ is a small value preventing a standard deviation of 0. The style randomization module uses the calculated mean and standard deviation to perform a style transfer through adaptive instance normalization (AdaIN) (Huang and Belongie, 2017), and then obtain the intermediate feature maps $SR(y, y')$ after transfer:

$$\hat{\mu} = \alpha \cdot \mu(y) + (1-\alpha)\cdot\mu(y') \tag{4}$$

$$\hat{\sigma} = \alpha \cdot \sigma(y) + (1-\alpha)\cdot\sigma(y') \tag{5}$$

$$SR(y, y') = \hat{\sigma} \cdot (\frac{y - \mu(y)}{\sigma(y)}) + \hat{\mu} \tag{6}$$

$\alpha \in (0, 1)$ is an interpolation weight. Afterwards, the obtained intermediate feature maps are fed into the subsequent feature extractor. And then processing the obtained style transformed images with the instance discrimination method of contrastive learning.

Due to the brittleness of RL algorithms (Henderson et al., 2018), complex discrimination may destabilize RL algorithms. In addition, it may also reduce the efficiency of RL algorithms. We therefore use a simple instance discrimination, which can be considered as the maximization of the mutual information between the original image and the transformed image. Input the intermediate feature maps of the original image y and the obtained intermediate feature maps of transfered images $SR(y, y')$ to the subsequent feature extractor G_f and get their intermediate feature maps z and z'

$$z = G_f(y) = G_f(G_{f_s}(x)) \tag{7}$$

$$SR(z, z') = G_f(SR(y, y')) = G_f(SR(G_{f_s}(x), G_{f_s}(x'))) \tag{8}$$

Taking the intermediate feature maps of original image z_i as the anchor, the intermediate feature maps of transformed original image $SR(z_i, z')$ as the positive, and other images with different contents but suffered the same style transfer $SR(z_{j(j\neq i)}, z')$ as the negative. Referring to the form of the formula for InfoNCE loss 1, we can obtain the optimization objective of the contrastive style transfer network as:

$$\max_{G_{f_s}} L_{style} = \log \frac{\exp(z_i \cdot W \cdot SR(z_i, z'))}{\exp(z_i \cdot W \cdot SR(z_i, z')) + \sum_{j(j\neq i)} \exp(z_j \cdot W \cdot SR(z_j, z'))} \tag{9}$$

The above optimization objective can be viewed as minimizing the mutual information between the original image and the transfer image, and maximizing the mutual information between different images with different content but undergoned the same style transfer. This optimization objective will allow the contrastive style transfer network to exert a stronger attention on the style and thus rely on the style to make decisions.

4.2. Contrastive Content Transfer Network

In order to make the feature extractor unable to recognize the information of the agents' own characteristics in pictures, we also use a contrastive content transfer network to make the subsequent feature extractor unable to distinguish the content information in pictures. The contrastive content transfer network uses a content randomization module. It will keep the style of the picture but change its content.

For an input x and another image x' chosen at random, we input them to the feature extractor of content transfer network G_{f_c} and get their corresponding intermediate feature maps y and y'. AdaIN method is used to get the content of y' and preserve the style of y:

$$CR(y, y') = \sigma(y) \cdot (\frac{y' - \mu(y')}{\sigma(y')}) + \mu(y) \tag{10}$$

This can be considered as a transformation of the content of original image x. Afterwards, similar to the style transfer network, the obtained intermediate feature maps are input

to the subsequent feature extractor G_f and get the subsequent intermediate feature maps noted as z and z'.

$$CR(z, z') = G_f(CR(y, y')) = G_f(CR(G_{f_c}(x), G_{f_c}(x'))) \tag{11}$$

Taking the intermediate feature maps of original image z_i as the anchor, the intermediate feature maps of transformed original image $CR(z_i, z')$ as the positive, and other images with different styles but suffered the same content transfer as the negative, denoted as $CR(z_{j(j\neq i)}, z')$. The optimization goal of the contrastive content transfer network is:

$$\min_{G_{f_c}} L_{content} = \log \frac{\exp(z_i \cdot W \cdot CR(z_i, z'))}{\exp(z_i \cdot W \cdot CR(z_i, z')) + \sum_{j(j\neq i)} \exp(z_j \cdot W \cdot CR(z_j, z'))} \tag{12}$$

The optimization goal above can be considered as maximizing the similarity of the original images and their transformed editions with content transfer, so that the contrastive content transfer network pays less attention to the content of pictures and more attention to other distracting information in pictures (e.g., style). In this way the feature extractor of the content transformation network G_{f_c} is trained adversarially to the subsequent feature extractor G_f.

Algorithm 1 Optimization Process of CIBC Networks

Input : Total number of environment steps T, transfer update interval T_s, encoder update intervals T_f, replay buffer $\mathcal{D}$

Output: Actor network, Q network

for *each timestep* $t \leftarrow 1$ **to** T **do**
  $a_t \sim \pi(\cdot|s_t)$, $s'_t \sim p(\cdot|s_t, a_t)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r, s'_t)$
  if $t\%T_s == 0$ **then**
    UpdateStyleNetwork($\mathcal{D}$), UpdateContentNetwork($\mathcal{D}$)
  end
  if $t\%T_f == 0$ **then**
    UpdateEncoderNetwork($\mathcal{D}$)
  end
  UpdateCritic($\mathcal{D}$), UpdateActor($\mathcal{D}$)
  UpdateEncoderNetwork($\mathcal{D}$)
end

4.3. Feature Extractor Implementation

The feature extractor G_f is trained to fool the prior contrastive style transfer network and contrastive content transfer network. So its training goal is to distinguish between the original image and the transformed image, and then extract the really needed information from RL algorithms. So one part of its optimization goal is exactly contrary to style transfer networks, to maximize the similarity between original images and their style transformed

Algorithm 2 Optimization Process of Transfer Networks

Input : mini-batch size N, replay buffer $\mathcal{D}$, feature extractor of style transfer network G_{f_s}, feature extractor of content transfer network G_{f_c}, subsequent feature extractor G_f, replay buffer $\mathcal{D}$

procedure UpdateStyleNetwork($\mathcal{D}$)

$(s_i, a_i, r, s'_{i+1})_{i=1}^{N} \in \mathcal{D}$

for *each i=1,...,N* **do**

$z_i = G_f(y_i) = G_f(G_{f_s}(s_i))$

$SR(z_i, z') = G_f(SR(y, y')) = G_f(SR(G_{f_s}(x), G_{f_s}(x')))$

$L_{style} = \log \frac{\exp(z_i \cdot W \cdot SR(z_i, z'))}{\exp(z_i \cdot W \cdot SR(z_i, z')) + \sum_{j(j \neq i)} \exp(z_j \cdot W \cdot SR(z_j, z'))}$

$\theta_s = \theta_s + \lambda_s \cdot \nabla_{\theta_s} L_{style}$

end

procedure UpdateContentNetwork($\mathcal{D}$)

$(s_i, a_i, r, s'_{i+1})_{i=1}^{N} \in \mathcal{D}$

for *each i=1,...,N* **do**

$z_i = G_f(y_i) = G_f(G_{f_s}(s_i))$

$CR(z_i, z') = G_f(CR(y, y')) = G_f(CR(G_{f_s}(x), G_{f_s}(x')))$

$L_{content} = \log \frac{\exp(z_i \cdot W \cdot CR(z_i, z'))}{\exp(z_i \cdot W \cdot CR(z_i, z')) + \sum_{j(j \neq i)} \exp(z_j \cdot W \cdot CR(z_j, z'))}$

$\theta_c = \theta_c - \lambda_c \cdot \nabla_{\theta_s} L_{content}$

end

editions, because the content of the images is of most interest to the feature extractor does not change:

$$\min_{G_f, W} L_{adv_s} = \log \frac{\exp(z_i \cdot W \cdot SR(z_i, z'))}{\exp(z_i \cdot W \cdot SR(z_i, z')) + \sum_{j(j \neq i)} \exp(z_j \cdot W \cdot SR(z_j, z'))} \tag{13}$$

At the same time, it also has to get rid of the interference of the content transfer network and make effective judgments on the output of the contrastive content transfer network. Since InfoNCE loss1 can be considered as a log loss function for k classification with label k_+. This part of the optimization goal is to out a uniform distribution prediction for images that have undergone content transfer networks. We can obtain the optimization objective as follows:

$$\min_{G_f, W} L_{adv_c} = \log \left| \frac{\exp(z_i W \cdot CR(z_i, z'))}{\exp(z_i W \cdot CR(z_i, z')) + \sum_{j(j \neq i)}^{K-1} \exp(z_j W \cdot CR(z_j, z'))} - \frac{1}{K} + 1 \right| \tag{14}$$

Finally, the total optimization objective of the feature extractor can be expressed as:

$$\min_{G_f, W} L_{total} = \lambda_1 \cdot L_{adv_s} + \lambda_2 \cdot L_{adv_c} \tag{15}$$

λ_1 and λ_2 are adjustable coefficients to control the trade-off between the content and style biases.

4.4. Implementation Details

Our Contrastive Inductive Bias Controlling(CIBC) Networks can be easily integrated with existing RL algorithms. In this paper we integrate it with SAC. The input images are fed into the contrastive style transfer network and contrastive content transfer network respectively. These two networks are updated alternately with the subsequent feature extractor. The features obtained after the feature extractor are input to subsequent policy and value networks for the update of RL algorithms. The gradients updated by RL algorithms are also back-propagated to the feature extractor to guide the update of the feature extractor together. Algorithm 1 summarizes the whole process of CIBC Networks training. Algorithm 2 summarizes the optimization process of two transfer networks. Considering the training time of the algorithm, we found that T_s of 100 can also be used to guide RL algorithms well and does not have a significant impact on the clock time.

5. Experimental Study

In this section, we evaluated our algorithm on visual continuous control tasks from DMC (Tassa et al., 2018), which is a widely used benchmark for sample efficiency. We compare our method with the previous model-free methods and model-based methods in terms of sample efficiency and perform the ablation experiment of the algorithm afterwards.

5.1. Setup

5.1.1. Environment

The experimental results were evaluated mainly on the Mujoco(Todorov et al., 2012) task provided by DMControl suite. DMC offers a wide variety of tasks with different levels of difficulty. Lots of model-free approaches proposed to improve sampling efficiency have been experimented on this benchmark, giving us sufficient baseline for comparison.

5.1.2. Training Details

In the experiments, the input is a stack of 3 consecutive RGB images of size 84×84, stacked along the channel dimensions. Each evaluation query was averaged over last 10 episodes returns. In all graphs, we draw the average of the 10 seeds' performance, as well as the shaded area representing the 95% confidence interval. The complete training details and hyperparameter settings are recorded in Appendix A.

5.2. DMC Experiments Compared to Model-Free Methods

According to common practice, we use real environment steps to compare the performance of different algorithms on the task and thus are not affected by the action-repeat hyperparameter. Some tasks that included sparse rewards are excluded (e.g. acrobot and tetrapods) because they require modifying the SAC algorithm to incorporate multi-step returns, which

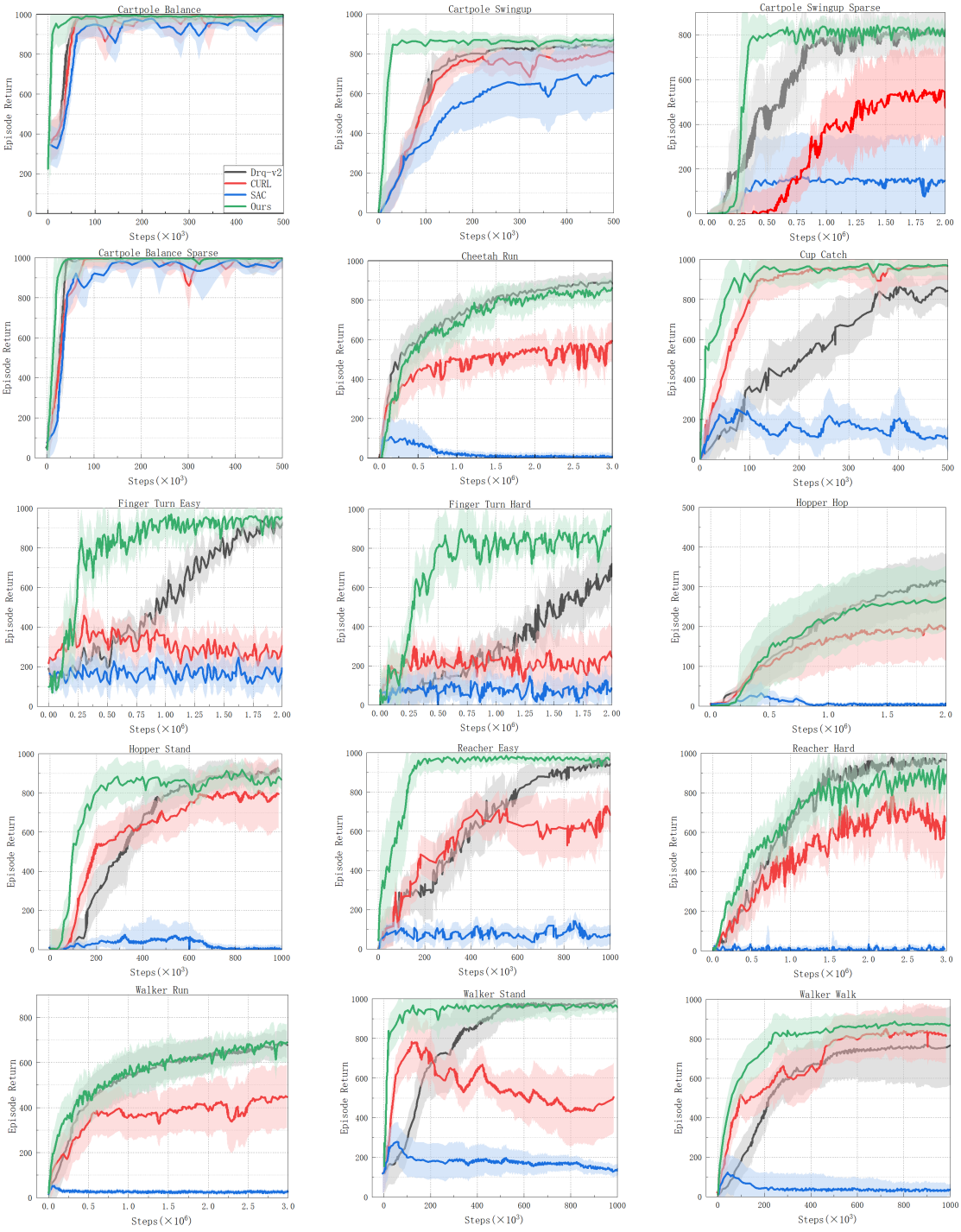

Figure 2: The DMC benchmark consists of 15 selected control tasks that offer various challenges, compared to other model-free methods. Our framework combined with SAC algorithm outperforms the previous state-of-the-art algorithm DrQ-v2 in most experimental environments.

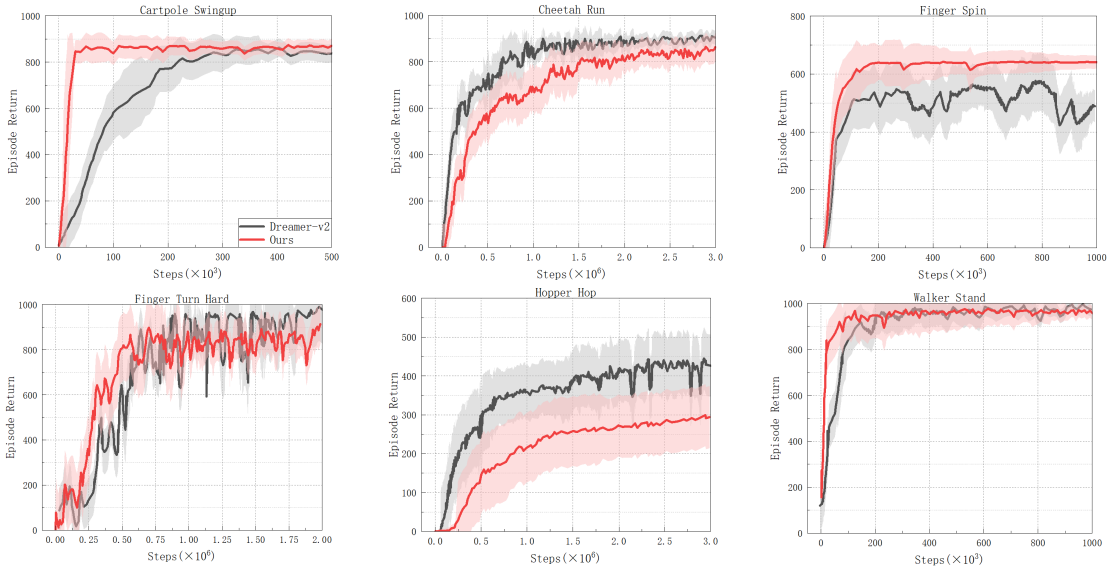

Figure 3: The DMC benchmark consists of 6 selected control tasks that offer various challenges, compared to Dreamer-v2, a model-based method

is beyond the scope of this work. In figure 2, We compare our approach to several state-of-the-art model-free RL algorithms for visual RL, including CURL (Srinivas et al., 2020), DrQ-v2 (Yarats et al., 2021a), and vanilla SAC (Haarnoja et al., 2018) augmented with the convolutional encoder of SAC-AE (Yarats et al., 2019). The experimental result shows that our algorithm achieves the best performance on most experiments environments. In Appendix B, we also evaluate our algorithm on the commonly used benchmarks based on the DeepMind control suite, namely the PlaNet (Hafner et al., 2019).

5.3. DMC Experiments Compared to Model-based Methods

We also compare our algorithm with a model-based approach Dreamer-v2 (Hafner et al., 2020), which is a very advanced model-based approach. It tends to obtain better sample complexity through a larger computational footprint. Due to hardware constraints, we perform a performance comparison between Dreamer-v2 and our algorithm on a limited number of tasks. In figure 3, although our algorithm is a model-free approach, it still exhibits a performance on some tasks that can compete with the advanced model-based algorithm Dreamer-v2 in terms of sample efficiency.

5.4. Ablation Studies

In figure 4, we show the results of the ablation experiments. The two transfer networks in the structure are masked separately. The result shows that the final effect of our algorithm is boosted by the synergy of the two transform networks, and any structure that keeps only a single transform network significantly reduces the experiments result in terms of sample

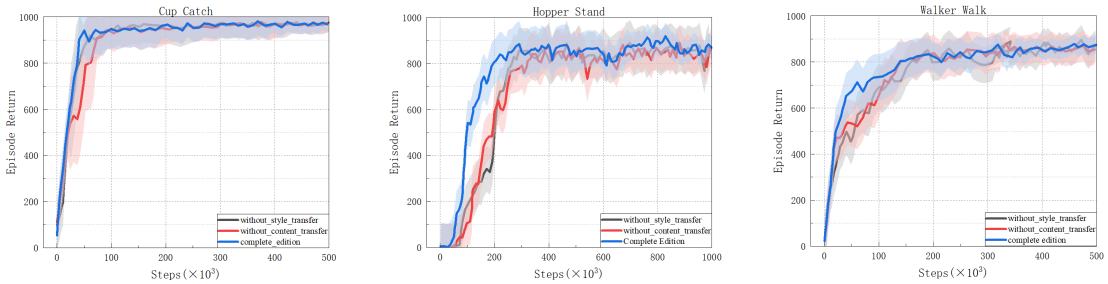

Figure 4: Ablation experiments: The results of the experiments without the style transfer network, without the content transfer network and the full version of the algorithm on the same task were tested separately. The experimental results show that the elimination of any of the structures significantly reduces the sampling efficiency.

efficiency. (The result of removing all of the two transform networks can be considered as the result of SAC+AE in figure 2).

6. Conclusion

In this work, we propose a migration-friendly framework: Contrastive Inductive Bias Controlling Networks for RL algorithms. It combines with SAC algorithm achieving state-of-the-art performance in terms of sample efficiency on most tasks on DMC. It demonstrates excellent performance on a continuous control environment by controlling the inductive bias with minimal changes to the RL algorithms. Besides, outstanding experimental results demonstrate that inductive bias in visual RL algorithms severely affects sample efficiency. It is a noteworthy direction for future study in visual RL. Due to its good portability, we believe that this antecedent framework can help RL algorithms achieve real deployments of RL in areas where sample efficiency is important, such as the real world.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grants No.U19B2044 and No.61836011.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al.

Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018.

Pedro Ballester and Ricardo Matsumura Araujo. On the performance of googlenet and alexnet applied to sketches. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. Learning actionable representations from visual observations. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1577–1584. IEEE, 2018.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015, 2020.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

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.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

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.

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

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with momentum predictive representations. *arXiv preprint arXiv:2007.05929*, 2(3), 2020.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021b.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34, 2021.