Generalizing Reinforcement Learning through Fusing Self-Supervised Learning into Intrinsic Motivation

Keyu Wu, Min Wu, Zhenghua Chen*, Yuecong Xu, Xiaoli Li

Institute for Infocomm Research , A*STAR, Singapore
{wu_keyu, wumin}@i2r.a-star.edu.sg, {chen0832, xuyu0014}@e.ntu.edu.sg, xlli@i2r.a-star.edu.sg

Abstract

Despite the great potential of reinforcement learning (RL) in solving complex decision-making problems, generalization remains one of its key challenges, leading to difficulty in deploying learned RL policies to new environments. In this paper, we propose to improve the generalization of RL algorithms through fusing Self-supervised learning into Intrinsic Motivation (SIM). Specifically, SIM boosts representation learning through driving the cross-correlation matrix between the embeddings of augmented and non-augmented samples close to the identity matrix. This aims to increase the similarity between the embedding vectors of a sample and its augmented version while minimizing the redundancy between the components of these vectors. Meanwhile, the redundancy reduction based self-supervised loss is converted to an intrinsic reward to further improve generalization in RL via an auxiliary objective. As a general paradigm, SIM can be implemented on top of any RL algorithm. Extensive evaluations have been performed on a diversity of tasks. Experimental results demonstrate that SIM consistently outperforms the state-of-the-art methods and exhibits superior generalization capability and sample efficiency.

Introduction

Reinforcement learning (RL) is capable of learning from previous experiences automatically without any labeled data, and has demonstrated great potential in solving various complex decision-making problems. By combining with convolutional neural networks (CNNs), vision-based RL has achieved prominent success in a broad range of areas, such as video games (Mnih et al. 2015), continuous control (Lillicrap et al. 2015), robot grasping (Kalashnikov et al. 2018), robot navigation (Wu et al. 2021) and so on. Despite its remarkable achievements, vision-based RL remains plagued by poor generalization capabilities, which significantly limits its deployment in real-world applications. Often, RL agents trained in one environment can hardly generalize well to unseen scenarios, even after they have been trained in a large number of diverse yet semantically similar environments (Gamrian and Goldberg 2019). The adaptation problems of vision-based RL agents can be further exacerbated due to the high-dimensional and partially-observable nature of pixel inputs.

As one of the well explored solutions, domain randomization typically creates a variety of environments with randomly sampled properties and finds the best policy which works across all of the environments (Tobin et al. 2017). Nevertheless, domain randomization can lead to large model complexity and is sensitive to sampling distribution and choice of randomized parameters. Another typical class of solutions is domain adaptation, which learns domain-invariant representations and mitigates the distribution discrepancy between the source and target domains (Bousmalis et al. 2018; Gamrian and Goldberg 2019). However, the target domain data are assumed to be accessible in domain adaptation settings while RL agents typically need to generalize to truly unknown environments.

Furthermore, the incorporation of data augmentations on input observations can also result in significantly improved generalization capability of RL agents (Laskin et al. 2020; Kostrikov, Yarats, and Fergus 2020; Raileanu et al. 2020). Recently, studies have also demonstrated that various self-supervised learning methods can be applied to improve both sample efficiency and generalization of RL policies substantially via better representation learning (Srinivas, Laskin, and Abbeel 2020; Hansen et al. 2020; Hansen and Wang 2021). However, prior works only combine self-supervised representation learning with reinforcement learning naively while it is promising to achieve greater improvement through linking self-supervised learning to reinforcement learning more ingeniously.

In this paper, we propose to improve the generalization of RL algorithms through fusing Self-supervised learning into Intrinsic Motivation (SIM). Similar to SODA (Hansen and Wang 2021), SIM decouples augmentation from policy learning to simplify RL optimization. It only uses augmented data to perform auxiliary representation learning while strictly using non-augmented samples for policy learning. In contrast to SODA, SIM does not require any prediction networks, momentum updates, or stop-gradients. It feeds the augmented and non-augmented data to two identical networks respectively, and makes the cross-correlation matrix between the two embeddings close to the identity matrix. This is to increase the similarity between the embeddings of a sample and its augmented version while mini-

*Corresponding Author

mizing the redundancy between the components of the vectors. In this way, the mutual information between the latent representations of augmented and non-augmented data is maximized so as to ignore features that are irrelevant to the RL tasks. Meanwhile, distinguished from all the existing methods, we also convert the redundancy reduction based self-supervised loss to an intrinsic reward to further improve RL generalization via an auxiliary objective. Since the self-supervised loss can be used to measure the quality of the embeddings, this intrinsic motivation is capable of guiding RL agents to learn the latent representations more efficiently. Since SIM is generic and does not require any change to the underlying algorithms, it can be implemented on top of any RL method. Empirical evaluations have been performed on a diversity of tasks in the DeepMind Control suite benchmark (Tassa et al. 2018), and experimental results demonstrate the superiority of SIM which outperforms the baselines and improves RL generalization significantly.

The main contributions of our work are summarized as follows:

- We propose a novel method, SIM, to improve the generalization of RL algorithms through fusing self-supervised learning into intrinsic motivation.
- We introduce a redundancy reduction based self-supervised learning method to enhance representation learning. Moreover, we convert this self-supervised loss to an intrinsic reward to further improve the generalization of RL agents via an auxiliary objective.
- We show that SIM outperforms the state-of-the-art baselines by a large margin.

Related Works

Self-Supervised Learning

Self-supervised learning empowers the learning of effective feature representations from unlabeled data which can benefit a wide variety of downstream tasks. So far, many auxiliary data prediction tasks have been proposed for self-supervised learning, such as relative position prediction (Doersch, Gupta, and Efros 2015), jigsaw puzzle (Noroozi and Favaro 2016), colorization (Zhang, Isola, and Efros 2016), context encoder (Pathak et al. 2016), rotation prediction (Gidaris, Singh, and Komodakis 2018), and so on.

As an important subclass of self-supervised learning, contrastive learning has gained growing popularity due to its encouraging progress. The rationale of contrastive learning is to minimize the distances between positive pairs while enlarging distances between negative pairs. For example, the CPC approach translates a generative modeling problem to a classification problem which distinguishes feature representation from a set of unrelated samples (Oord, Li, and Vinyals 2018). SimCLR learns visual representations via maximizing agreement between different augmented views of the same sample in the latent space (Chen et al. 2020). By introducing a queue-based dynamic dictionary, MoCo trains the representation encoder through matching the encoded query to a dictionary of encoded keys using a contrastive loss (He et al. 2020).

Different from the above contrastive learning approaches, BYOL (Grill et al. 2020) learns invariant feature representations without using negative samples, and instead adopts a bootstrapping mechanism. It relies on two sets of networks, where the target network has the same architecture as the online one but is updated through polyak averaging. It then trains the online network on one augmented view to predict the representations of a different augmented view generated by the target network. Through solving the problem from a different perspective, Barlow Twins (Zbontar et al. 2021) feeds two augmented versions of the training samples into the same network and tries to make the cross-correlation matrix between these two group of embeddings close to the identity. The objective is to keep the embeddings of different augmented versions of the same samples similar while minimizing the redundancy between these representation vectors. In this way, self-supervised representation learning can be achieved without negative samples, prediction networks, momentum updates, or stop-gradients.

Generalization in RL

Generalization remains one of the key challenges for RL. So far, different strategies have been proposed to address this issue. One of the leading solutions is domain randomization, which creates a variety of environments with randomly sampled properties and figures out the best policy that works across all the environments. For example, the simulation environment is rendered with random textures to train the adaptive policies (Tobin et al. 2017). Despite its effectiveness, domain randomization is sensitive and can lead to large model complexity. Another widely explored solution is domain adaptation, which reduces the distribution divergence between the source and target domains. By adopting domain adaptation in RL, adaptive robot grasping policy is learned through Sim2Real transfer (Bousmalis et al. 2018) and observation adaptation for video games is achieved through image-to-image translation (Gamrian and Goldberg 2019). Nevertheless, domain adaptation assumes the target domain data are accessible while RL agents often need to generalize to unknown scenarios. Moreover, a variety of works on data augmentation, such as RAD (Laskin et al. 2020), DrQ (Kostrikov, Yarats, and Fergus 2020) and DrAC (Raileanu et al. 2020), prove that data augmentation can be very effective as well as efficient to improve the generalization of RL agents as well.

In recent years, studies have demonstrated that it is promising to improve RL generalization more significantly through leveraging the advances in self-supervised learning. For instance, PAD explores the use of self-supervision tasks, such as inverse dynamics prediction, to enable continued training during deployment and thereby improve the generalization of vision-based RL (Hansen et al. 2020). CURL adopts the MoCo mechanism in RL and enforces consistencies in observations via matching embeddings of two augmented versions of the raw observation (Srinivas, Laskin, and Abbeel 2020). More recently, by using the concepts of BYOL in RL, SODA learns invariant representations through bootstrapping (Hansen and Wang 2021). Rather than aligning two different augmented views, SODA aligns

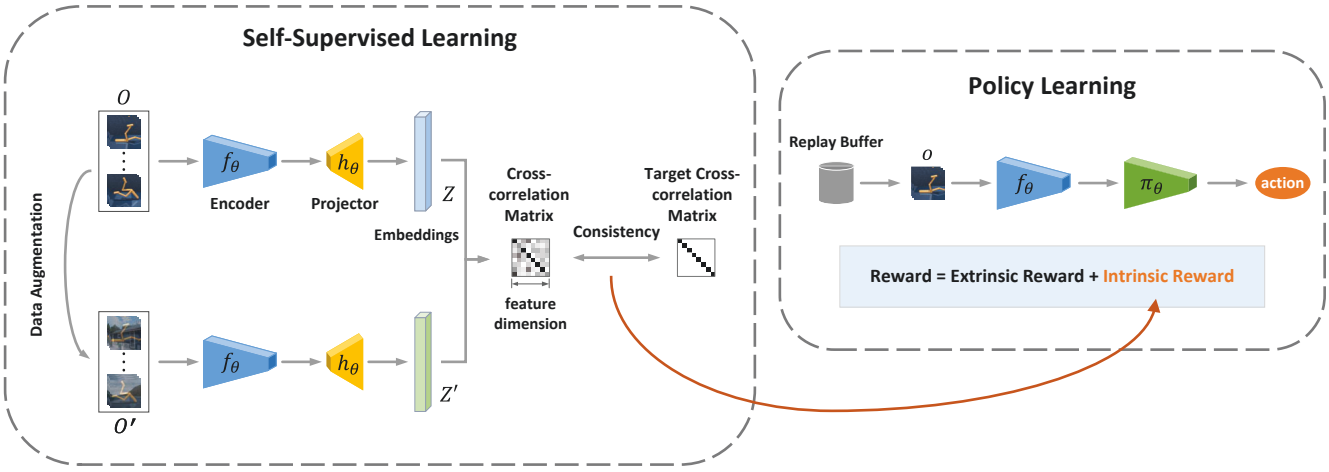


Figure 1: SIM architecture. SIM trains the RL network jointly with an auxiliary self-supervised network. The two networks share a common encoder and the self-supervised loss is converted as an intrinsic reward for RL agent's progress toward better representation learning. Left: representation learning via redundancy reduction based self-supervised learning with augmented observations. Right: policy learning with non-augmented observations.

augmented images to their non-augmented counterparts in latent space to ensure that data augmentation is only used for representation learning in order to reduce the difficulty of RL optimization. In this paper, we employ the mechanism of Barlow Twins to improve the representation learning in RL without requiring any negative samples, prediction networks, momentum updates, or stop-gradients. Moreover, we propose to fuse self-supervised learning into intrinsic motivation through converting the self-supervised loss to an intrinsic reward to further improve the generalization of RL via an auxiliary objective.

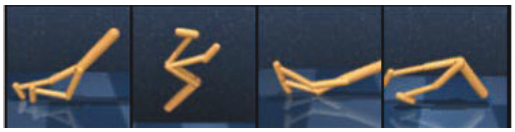
Method

We propose to improve the generalization of RL through fusing Self-supervised learning into Intrinsic Motivation (SIM). Since SIM does not require any modification to the underlying base RL method, it is a general paradigm that can be implemented on top of any RL algorithm. In this work, we train SIM alongside the Soft Actor-Critic (SAC) algorithm (Haarnoja et al. 2018) to demonstrate its effectiveness. Generally, the objective of SIM is to achieve effective representation learning so that the important information shared between an observation and its augmented counterpart is well encoded.

Architecture Overview

The network architecture of SIM is depicted in Figure 1. Generally, SIM trains the reinforcement learning network jointly with an auxiliary self-supervised learning network. The RL policy network with parameters θ can be split into two parts, i.e., an encoder f_θ and a policy π_θ, so that given an input observation o, the action can be derived as $a = \pi_\theta(f_\theta(o))$. The encoder f_θ contains 11 convolutional layers with 32 filters and the policy π_θ contains 4 fully-connected layers. Meanwhile, the proposed auxiliary self-supervised learning task shares a common encoder f_θ with the policy network and is composed of the shared encoder f_θ and a projector h_θ. The projector h_θ has 3 fully-connected layers where the first two layers are followed by a batch normalization layer and rectified linear units. Particularly, the outputs of the encoder and the projector are referred as representations and embeddings respectively. The representations are used for the downstream RL tasks and mapped to actions through the policy network, while the embeddings are normalized along the batch dimension and fed to the loss function of the self-supervised learning task.

During training, observations are sampled from the replay buffer. Similar to SODA, SIM decouples the training data flow so that augmented data are only employed for representation learning to reduce the difficulty of RL optimization. In addition, instead of contrasting two augmented instances of the same image to a batch of negative samples, SIM learns invariance through maximizing similarity between the embeddings of the augmented observations and their non-augmented counterparts without the demand for negative samples. Therefore, both the non-augmented and augmented observations are fed to the self-supervised learning network and mapped to two sets of embeddings via a pair of identical networks. As shown in Figure 2, we augment the observations using the random overlay method (Hansen and Wang 2021) which produces an augmented view through linearly interpolating between an observation and an image. With this strong data augmentation, f_θ is encouraged to learn features that are shared across different views while discarding factors of variation that are irrelevant to the RL tasks. Since it is common in the RL setting to stack consecutive frames as observations to infer temporal information, we apply augmentations randomly across the batch while the same image is applied to all frames of a given observation to retain the temporal information.



(a) non-augmented samples (walker_walk)

(b) random overlay samples (walker_walk)

Figure 2: Data augmentation based on the random overlay method.

Self-Supervised Representation Learning

Inspired by the self-supervised learning method proposed in (Zbontar et al. 2021), we also apply the redundancy reduction based method to learn invariant feature representations. However, our goal is to align the embeddings of augmented and non-augmented observations rather than those of two different augmented views. Without negative samples, we achieve self-supervised representation learning through maximizing similarity between the embedding vectors while reducing redundancy between their components.

First, given an original observation o, a data augmentation can be sampled so that $t \sim \mathcal{T}$. Then, by applying data augmentation t, an augmented observation $o' = t(o)$ can be generated so that o and o' can be regarded as two different views of the same underlying state. Considering a batch of observations O, a batch of augmented observation O' can be produced by applying augmentations randomly across O. As shown in Figure 1, these two batches of different views are fed to the encoder f_θ followed by the projector h_θ to generate two batches of embedding vectors Z and Z' respectively. It is worth mentioning that the two embeddings are normalized along the batch direction so that each unit has zero mean over the batch.

Let $\mathcal{C}$ be a cross-correlation matrix computed between the two sets of embedding vectors, the loss function of the self-supervised task can be expressed as:

$$\mathcal{L}_{\text{SSL}} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \tag{1}$$

where

$$\mathcal{C}_{ij} = \frac{\sum_b \mathbf{z}_{b,i} \mathbf{z}'_{b,j}}{\sqrt{\sum_b (\mathbf{z}_{b,i})^2} \sqrt{\sum_b (\mathbf{z}'_{b,j})^2}} \tag{2}$$

In these equations, λ is a positive constant used to trade off the two terms of the loss, b is the batch index, and i, j are indexes of the network output vector dimension. Therefore, $\mathcal{C}$ is a square matrix with the size same as the dimensionality of the network's output and each of its entry calculates the cosine similarity between $\mathbf{z}_{b,i}$ and $\mathbf{z}'_{b,j}$. In this way, the first term can drive the embedding invariant to the applied data augmentation while the second term can decorrelate the different vector components to avoid the inclusion of redundant information.

Without modifying either the architecture or inputs, the RL objective is optimized directly on non-augmented original observations o. Therefore, the policy network and self-supervised network are continuously optimized in an alternative manner during training while only the encoder f_θ and the policy π_θ are employed during evaluation.

Algorithm 1: SIM

θ: random initialized network parameters
N_{RL}: RL batch size
N_{SSL}: SSL batch size
$\mathcal{B}$: RL replay buffer
n: SSL update frequency
μ: intrinsic reward weight

for each iteration **do**
  Sample a batch of N_{RL} transitions from $\mathcal{B}$
  Augment sampled observations
  Calculate $\mathcal{L}_{\text{SSL}}(o, o'; \theta_{t-1})$ and $\mathcal{L}_{\text{SSL}}(o, o'; \theta_t)$
  Calculate intrinsic reward and add it to each reward in the batch
  Optimize RL objective
  if step $\bmod n = 0$ **then**
    Sample a batch of N_{SSL} observations from $\mathcal{B}$
    Augment sampled observations
    Compute embeddings of augmented and non-augmented observations
    Optimize SSL objective
  end if
end for

Fusion of Self-Supervised Learning into Intrinsic Motivation

In RL, intrinsic motivation is originally proposed to facilitate the exploration and its rationale is to predict the consequences caused by taking an action (Pathak et al. 2017; Burda et al. 2018). However, in this work, our goal is to utilize intrinsic motivation to further facilitate the representation learning via better leveraging the useful information provided by the self-supervised network. Since the self-supervised auxiliary task contains adequate information to measure the quality of the embeddings, an intrinsic reward converted from the self-supervised loss can be promising to assist the RL agents to learn the feature representations more efficiently as well as effectively. As a result, the supervision signals during RL training are contributed from both the extrinsic and intrinsic rewards so that the total reward function R is defined as the sum of these two types of rewards expressed as:

$$R_t = R_t^e + \mu R_t^i \tag{3}$$

where R_t^e, R_t^i and μ represent the extrinsic reward, intrinsic reward, and weighting coefficient, respectively. To better stimulate the representation learning, we define the intrinsic

Hyperparameter	Value
Observation rendering	100×100
Observation downsampling	84×84
Stacked frames	3
Action repeat	2 (finger) 8 (cartpole) 4 (otherwise)
Discount factor γ	0.99
Episode length	1,000
Base RL algorithm	SAC
Replay buffer size	500,000
Initial steps	1000
Learning rate (actor, critic, SSL)	1e-3
Learning rate (α)	1e-4
Initial temperature	0.1
Trade-off constant λ	3.9e-3
Update frequency (actor, critic target, SSL)	2

Table 1: Hyperparameters used for the DMControl experiments.

reward as:

$$R_t^i = \frac{\mathcal{L}_{\text{SSL}}(o, o'; \theta_{t-1}) - \mathcal{L}_{\text{SSL}}(o, o'; \theta_t)}{\mathcal{L}_{\text{SSL}}(o, o'; \theta_{t-1})} \tag{4}$$

where θ_t and θ_{t-1} denote the current network parameters and those at the previous step, respectively. Since $\mathcal{L}_{\text{SSL}}$ is computed based on the whole batch, the same intrinsic reward is assigned to all the samples in the batch. By introducing this auxiliary objective, the RL agent is motivated to optimize the encoder more efficiently to achieve improved representation learning.

We summarize the training procedure of SIM in Algorithm 1.

Experiments

Setup

DeepMind Control Suite (DMControl) (Tassa et al. 2018) is a widely used benchmark dataset for vision-based RL algorithm comparison. It contains a variety of challenging and diverse continuous control tasks. We evaluate the RL agents on 5 tasks from DMControl in which the agents observe only raw pixels without access to state information.

For benchmarking, we compare the SIM with five different baselines, i.e., CURL (Srinivas, Laskin, and Abbeel 2020), RAD (Laskin et al. 2020), DrQ (Kostrikov, Yarats, and Fergus 2020), PAD (Hansen et al. 2020) and SODA (Hansen and Wang 2021). All the methods are implemented on top of the SAC algorithm. In particular, RAD and DrQ achieve policy adaptation through data augmentations while CURL, PAD and SODA generalize the policy via adding self-supervised objectives.

All the RL agents are trained in a fixed training environment and evaluated in unseen environments as shown in Figure 3. Particularly, the `color_hard` environment randomizes the colors of background, foreground and the agent itself, while the `video_easy` and `video_hard` replace the background with natural videos. We evaluate the performance of different methods at 500k environment steps because all methods have achieved their optimal performance on most environments at this point. For each test environment, the methods are evaluated across 100 randomly initialized episodes per seed.

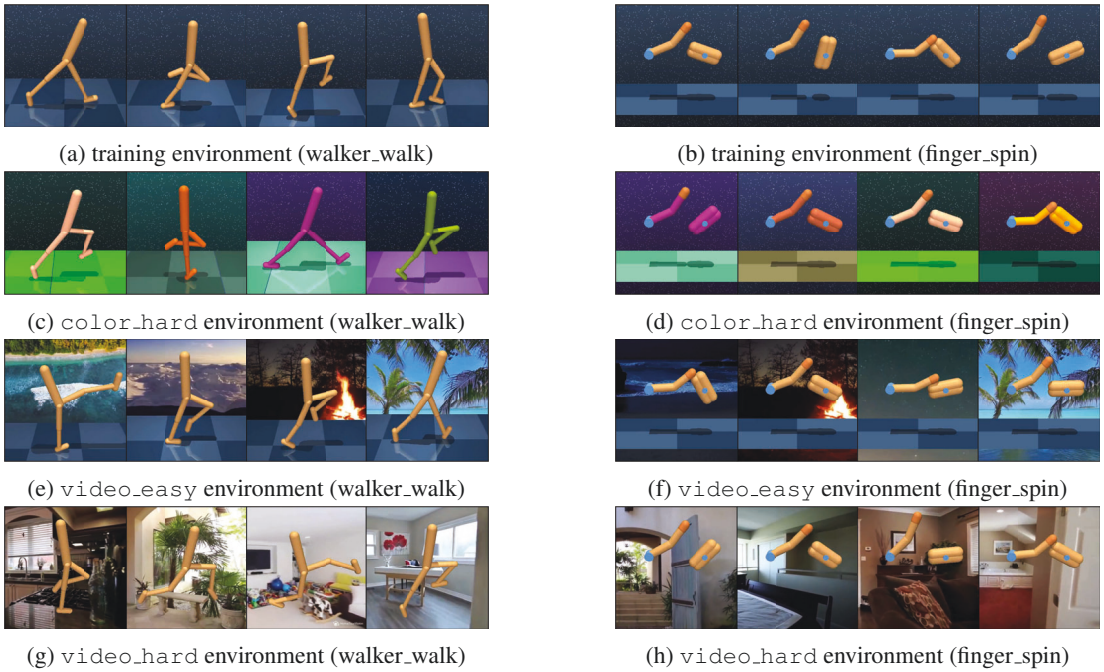
During training, we use a batch size of 128 for both RL and SSL tasks. The three linear layers of the projector h_θ contain 2048, 4096, 4096 output units, respectively. Both $\mathcal{L}_{\text{RL}}$ and $\mathcal{L}_{\text{SSL}}$ are optimized using Adam (Kingma and Ba 2014). The SSL update frequency is set to 2 so that the SSL objective is only optimized after every second RL update. The other hyperparameters are listed in Table 1.

Generalization to Unseen Environments

After training, we evaluate the generalization capability of SIM in the unseen `color_hard`, `video_easy` and `video_hard` environments and compare it to a number of recently proposed state-of-the-art approaches. Generally, SIM is much more superior compared to the baseline methods. The experimental results in the `color_hard` environment are shown in Table 2, where the best result for each task is highlighted in bold. It is worth noting that SIM outperforms the baseline methods in all tasks. Particularly, it improves the performance by as much as 16% on the `walker_walk` task and 21% on the `finger_spin` task. Similarly, SIM also significantly outperforms the baseline methods in the `video_easy` and `video_hard` as demonstrated in Table 3 and Table 4, respectively. In the `video_easy` environment, SIM outperforms the state-of-the-art methods in all tasks except `ball_in_cup_catch`. And its maximum performance improvement is 17% on the `finger_spin` task. In the extremely challenging `video_hard` environment, an improvement of 20% is achieved on both the `walker_walk` and `finger_spin` tasks.

However, although SIM outperforms all previous methods except SODA dramatically in the `video_hard` environment, there is still large room for improvement on the majority of tasks and it performs worse on the `cartpole_swingup` and `ball_in_cup_catch` tasks compared to SODA. We conjecture that this performance gap can be addressed by further improving the removal of task-irrelevant features so as to reduce observational overfitting. This can be achieved through improving the encoder architecture, introducing better self-supervised tasks, modifying the RL update rules and enhancing the interaction between RL and SSL objectives.

Although sample efficiency is not the key consideration of our method, the training and evaluation curves are depicted in Figure 4 for completeness. It is observed that while SAC is able to converge in training, its generalization ability is poor during evaluation. Compared to SAC, both SIM and SODA substantially improve RL generalization. With the best performance, SIM exhibits a similar sample efficiency compared to SODA. Therefore, although strong augmentations can improve generalization at the cost of sample



(a) training environment (walker_walk)
(b) training environment (finger_spin)
(c) color_hard environment (walker_walk)
(d) color_hard environment (finger_spin)
(e) video_easy environment (walker_walk)
(f) video_easy environment (finger_spin)
(g) video_hard environment (walker_walk)
(h) video_hard environment (finger_spin)

Figure 3: Sample training and test environments. The agents are trained in a fixed training environment and evaluated in different unseen test environments.

DMControl (color_hard)	SAC	CURL	RAD	DrQ	PAD	SODA	SIM (w/o IM)	SIM
walker, walk	414±74	445±99	400±61	520±91	468±47	692±68	673±85	**803±33**
walker, stand	719±74	662±54	644±88	770±71	797±46	893±12	926±6	**940±2**
cartpole, swingup	592±50	454±110	590±53	586±52	630±63	805±28	835±8	**841±13**
ball_in_cup, catch	411±183	231±92	541±29	365±210	563±50	949±19	952±10	**953±7**
finger, spin	626±163	691±12	667±154	776±134	803±72	793±128	954±20	**960±6**

Table 2: Average episode reward obtained in the color_hard environment.

DMControl (video_easy)	SAC	CURL	RAD	DrQ	PAD	SODA	SIM (w/o IM)	SIM
walker, walk	616±80	556±133	606±63	682±89	717±79	768±38	779±65	**861±33**
walker, stand	899±53	852±75	745±146	873±83	935±20	955±13	958±15	**963±5**
cartpole, swingup	375±90	404±67	373±72	485±105	521±76	758±62	749±55	**770±18**
ball_in_cup, catch	393±175	316±119	481±26	318±157	436±55	**875±56**	704±184	820±135
finger, spin	447±102	502±19	400±64	533±119	691±80	695±97	801±34	**815±38**

Table 3: Average episode reward obtained in the video_easy environment.

DMControl (video_hard)	SAC	CURL	RAD	DrQ	PAD	SODA	SIM (w/o IM)	SIM
walker, walk	155±85	58±18	56±9	104±22	93±29	381±72	438±104	**459±67**
walker, stand	247±43	45±5	231±39	289±49	278±72	771±83	810±36	**827±24**
cartpole, swingup	152±14	114±15	110±16	138±9	123±24	**429±64**	390±50	367±47
ball_in_cup, catch	78±18	115±33	97±29	92±23	66±61	**327±100**	237±165	287±39
finger, spin	29±25	27±21	34±11	71±45	56±18	302±41	353±18	**362±9**

Table 4: Average episode reward obtained in the video_hard environment.

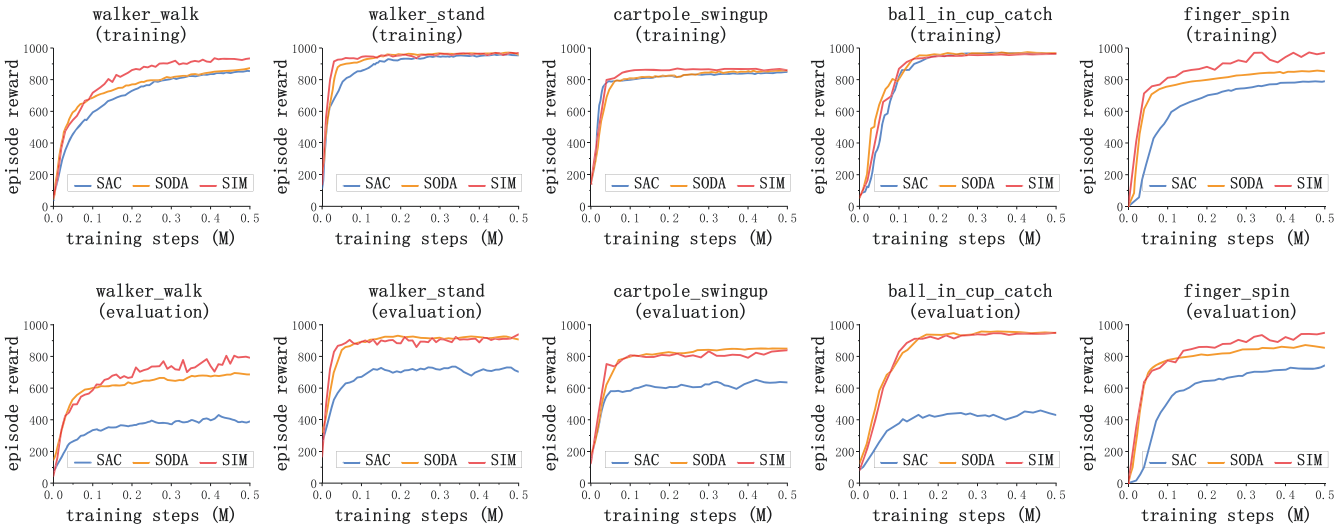


Figure 4: Training and evaluation performance. Top: average episode reward in the training environment. Bottom: generalization capability measured by average episode reward in the `color_hard` environment.

efficiency (Tian et al. 2020), SIM achieves better generalization capability without compromising the sample efficiency.

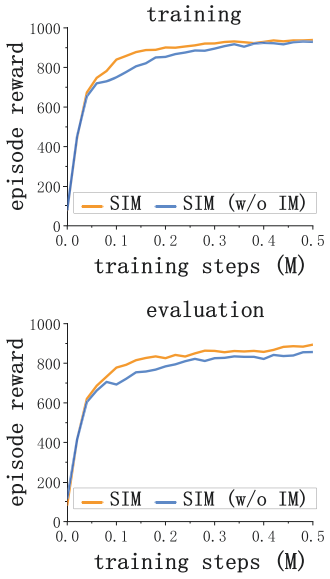


Figure 5: Mean episode reward averaged across all the 5 tasks from DMControl.

Ablation Study

Furthermore, we conduct ablation study to verify the effectiveness of fusing self-supervised learning into intrinsic motivation, and denote the SIM method without intrinsic reward as SIM (w/o IM). As shown in Table 2, Table 3 and Table 4, SIM outperforms SIM (w/o IM) consistently in all tasks except on the `cartpole_swingup` task in the `video_hard` environment. To illustrate the competency of intrinsic motivation more intuitively, we calculate the mean episode rewards averaged across all 5 tasks from DMControl during training and evaluation. As shown in Figure 5, although SIM and SIM (w/o IM) converge to similar performance during training, the superiority of SIM becomes notable in unseen test environments. Moreover, SIM also exhibits slightly better sample efficiency compared to SIM (w/o IM). Therefore, it demonstrates the importance of fusing self-supervised learning into intrinsic motivation in improving the generalization capability of RL agents.

Our code and more experimental results are available at https://github.com/KerryWu16/SIM.

Conclusion

In this paper, we propose to address the generalization challenge for RL agents via fusing Self-supervised learning into Intrinsic Motivation (SIM). We employ a redundancy reduction based self-supervised learning method to learn invariant representations through increasing the similarity between the embeddings of a sample and its augmented counterpart while minimizing the redundancy between the components of the vectors. Meanwhile, we also convert the self-supervised loss to an intrinsic reward to further improve RL generalization via an auxiliary objective. SIM can be implemented on top of any RL method without incurring any change to the underlying algorithm and experimental results have demonstrated its remarkable superiority.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its Career Development Award (Grant No. C210112046).

References

Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, 4243–4250. IEEE.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, 1422–1430.

Gamrian, S.; and Goldberg, Y. 2019. Transfer learning for related reinforcement learning tasks via image-to-image translation. In *International Conference on Machine Learning*, 2063–2072. PMLR.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hansen, N.; Sun, Y.; Abbeel, P.; Efros, A. A.; Pinto, L.; and Wang, X. 2020. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*.

Hansen, N.; and Wang, X. 2021. Generalization in Reinforcement Learning by Soft Data Augmentation. arXiv:2011.13389.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. 2018. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, 651–673.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; and Srinivas, A. 2020. Reinforcement Learning with Augmented Data. *arXiv preprint arXiv:2004.14990*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 69–84. Springer.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.

Raileanu, R.; Goldstein, M.; Yarats, D.; Kostrikov, I.; and Fergus, R. 2020. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*.

Srinivas, A.; Laskin, M.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–30. IEEE.

Wu, K.; Han, W.; Esfahani, M. A.; and Yuan, S. 2021. Learn to Navigate Autonomously through Deep Reinforcement Learning. *IEEE Transactions on Industrial Electronics*.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *European conference on computer vision*, 649–666. Springer.