Evaluating Vision Transformer Methods for Deep Reinforcement Learning from Pixels

Tianxin Tao* Daniele Reda* Michiel van de Panne

***Abstract*— Vision Transformers (ViT) have recently demonstrated the significant potential of transformer architectures for computer vision. To what extent can image-based deep reinforcement learning also benefit from ViT architectures, as compared to standard convolutional neural network (CNN) architectures? To answer this question, we evaluate ViT training methods for image-based reinforcement learning (RL) control tasks and compare these results to a leading convolutional-network architecture method, RAD. For training the ViT encoder, we consider several recently-proposed self-supervised losses that are treated as auxiliary tasks, as well as a baseline with no additional loss terms. We find that the CNN architectures trained using RAD still generally provide superior performance. For the ViT methods, all three types of auxiliary tasks that we consider provide a benefit over plain ViT training. Furthermore, ViT masking-based tasks are found to significantly outperform ViT contrastive-learning.**

I. Introduction

Image-based reinforcement learning is an important and growing area, as cameras provide a ubiquitous and inexpensive way to acquire observations in complex and unstructured environments. However, in RL, it remains common to give privileged access to compact state descriptors that may not be available in real-world settings. This is because extending such methods to work with images is non-trivial, usually requiring significant additional computation and algorithmic improvements to cope with the high-dimensional nature of images as inputs. In recent years, end-to-end deep reinforcement learning from visual inputs has yielded impressive results in domains such as robotics control tasks [1], [2], Atari games [3] and autonomous driving [4]. However, such methods remain data-intensive and are brittle with respect to confounding visual factors, including dynamic backgrounds, other agents, and changing camera perspectives.

In principle, end-to-end RL can learn representations directly while learning the policy. However, prior work has observed that RL is bounded by a "representation learning bottleneck" in the sense that a considerable portion of the learning period must be spent acquiring good representations of the observation space. To mitigate this, the learning of good state representations can be aided by auxiliary losses that guide the learning of a suitable representation [5], [6]. They can also be learned fully in advance, in an unsupervised fashion, in support of the subsequent control tasks to be learned [7], [8], [9]. Recently, data augmentation techniques, already popular in computer vision, have shown to significantly improve performances in RL from pixels [10], [11], [12], [13].

More recently, computer vision is seeing a potential shift in network architectures from convolutional neural networks (CNNs) to vision transformers (ViT), as the latter are being repeatedly shown to learn good representations for the downstream tasks. Recent advances in Transformers [14] and ViT [15] raise the obvious question as to whether or not this type of architecture will also benefit image-based deep reinforcement learning. Since ViT usually requires significantly more data to train on, a common methodology is to first train using a self-supervised objective and then fine-tune the representation as needed for downstream tasks. Motivated by this approach, we investigate whether ViT assisted with the existing self-supervised training objectives can assist in learning image-based RL policies.

Our primary contributions are as follows:

- We adapt and implement three existing self-supervised learning methods for computer vision tasks as auxiliary tasks for ViT-based RL policies;
- We evaluate and compare the above ViT-based appraoaches with RAD, a leading CNN-based RL method. Our results point to better overall performance for RAD. For the ViT methods, we find that masking-based methods outperform contrastive learning.

II. Related Work

Reinforcement learning from pixels has been approached using RL algorithms including DQN [3] and DDPG [16]. These algorithms were applied to Atari environments and not more complex dynamical control environments like the locomotion ones in Mujoco [17], PyBullet [18] or Deepmind Control Suite (DMControl Suite) [19]. These algorithms are not explicitly designed for image-based input, and they learn faster with lower dimensional state representations.

In recent years, a focus on RL from pixel-based input has narrowed the gap between learning from compact state observations and high-dimensional observations, generally using CNNs. Learning a representation with an auxiliary decoder is known to improve efficiency in learning good representations [4], [6]. Data augmentation methods have recently been adopted in RL, in contrast to computer vision where this has a longer history. RAD [13] proposes to augment images with random cropping, and other augmentations to improve data-efficiency. DrQ [10] and DrQ-v2 [11] propose to augment both the input image and the Q-function. This acts as a regularization technique and allows efficient

*The authors contribute equally

Tianxin Tao, Daniele Reda and Michiel van de Panne are with Department of Computer Science, University of British Columbia, Vancouver, BC, Canada. Email: {taotianx, dreda, van}@cs.ubc.ca.

learning of action-value approximations for images. Later, stochastic data augmentation is proposed to further stabilize the training of agent [20]. SUNRISE proposes the use of an ensemble of Bellman back-ups and a novel exploration strategy to enhance the performance [21]. Predicting future internal representations has also been demonstrated to be an effective objective for RL from pixels [22]. CURL [12] uses a contrastive loss [23], [24] to match representations of the same image processed with different augmentations. The contrastive objective for RL is then extended to incorporate temporal prediction [25], [26] and curiosity-driven rewards [27]. Additionally, model-based approaches learn the latent dynamics corresponding to image-based observations to improve sampling efficiency [7], [28], [29], [30].

Multiple methods have been proposed to learn self-supervised representations with ViT, and later fine tune them for downstream computer vision tasks. Popular ideas include contrasting the visual representations of different augmented views of the images [31], [32], discriminating the latent representations [33], [34], and predicting targets with masked inputs [35], [36], [37], [38].

In this work, we aim to analyze the usage of ViTs for reinforcement learning from pixels. We seek to understand the extent to which additional self-supervised losses can guide the learning of efficient representations in an RL context, and to compare this to CNN-based methods such as RAD.

III. Experimental Design

Convolutional architectures paired with data augmentation techniques have shown great progress in RL from pixels. However, whether ViT can bring any benefit to the problem remains unanswered yet. Therefore, we study the impact of a plain ViT encoder and various self-supervised losses proposed with ViT on the RL from pixels problem. We embed three recently proposed self-supervised objectives with ViT into the standard RL pipeline as auxiliary tasks. More specifically, we employ the idea of *Data2Vec* [38], *MAE* [35], and momentum contrastive learning [32], [12].

A high-level summary of our empirical study is illustrated in Figure 1. We keep the standard RL training fixed at all times, and test the performance of adding different ViT auxiliary tasks, as well as comparing to a CNN-encoder, as implemented for RAD. In this pipeline, the stacked images are first augmented via random cropping, and then embedded to a shared representation using an encoder. This shared representation is then used both for RL training, after being flattened and passed through fully-connected layers, and the auxiliary training tasks, if any. For ViT, we adopt the ViT architecture as both the encoder and decoder. For a fair comparison, we choose a small ViT encoder with a similar order of magnitude of learnable parameters as its convolutional counterpart.

We now introduce each self-supervised objective. Please refer to the original paper for more details [38], [35], [32], [12].

A. Data2Vec

Data2Vec [38] is a unified self-supervised model proposed across multiple modalities with different specialized structures, including images, speeches and natural language processing. In this work, we focus on its variant operating on images. Data2Vec adopts the idea of predicting the internal representation of a masked input to match the representation of the original input. A visual depiction is shown in Figure 2a.

The self-supervision task receives the original stack of images o_t and its masked version o'_t as input. The observation o_t is encoded with an encoder E_θ. The output of the encoder summed with the activation value of the K-last layers of the encoder forms the target vector t. Instead, the masked input o'_t is encoded with a momentum encoder $E'_{\theta'}$ and then passed through a two-layer decoder to form a prediction vector p. The goal is to minimize the difference between the prediction p and the target t.

To avoid learning trivial solutions (i.e., degenerating the representation into a uniform vector), the authors apply parameter-free layer normalization on the output value a^i of the last i^{th} layer. The target t can be mathematically expressed as: $t = \sum_{i=0}^{K} \text{LayerNorm}(a^i)$

Given the target t and prediction p, the parameters of the encoder and the decoder are updated through the Smooth L1 loss as follows:

$$\mathcal{L}_{Data2Vec}(t,p) = \begin{cases} \frac{1}{2}(t-p)^2/\beta & \text{if} \|t-p\| \leq \beta, \\ \|t-p\| - \frac{1}{2}\beta & \text{otherwise.} \end{cases} \quad (1)$$

The momentum encoder $E'_{\theta'}$ is updated through Polyak averaging of the encoder E_θ's parameters:

$$\theta' = (1-\tau)\theta + \tau\theta'. \quad (2)$$

B. Masked Autoencoders

MAE [35] adopts a simple-yet-effective idea: given an image with masked patches, the self-supervised objective is to reconstruct the original unmasked image. Both the encoder and decoder use a ViT architecture. A visual representation of MAE is depicted in Figure 2b.

The encoder only operates on the unmasked patches and encodes the unmasked patches to a latent representation z_t. The decoder then receives the concatenation of both the unmasked embeddings and the masked patches as input, and transforms the input into images p. All the masked patches are uniformly represented by a learnable vector z'. The positional information of each patch is indicated by the position embedding of the decoder. We implement a lightweight decoder to save computational resources. The reconstruction target t is the pixel value of the masked patches only, and the pixels are normalized per patch to improve the performance as the authors of [35] suggest. The reconstruction loss $\mathcal{L}_{MAE}$ is computed as the mean squared error between the target t and reconstructed images p only on the masked portion.

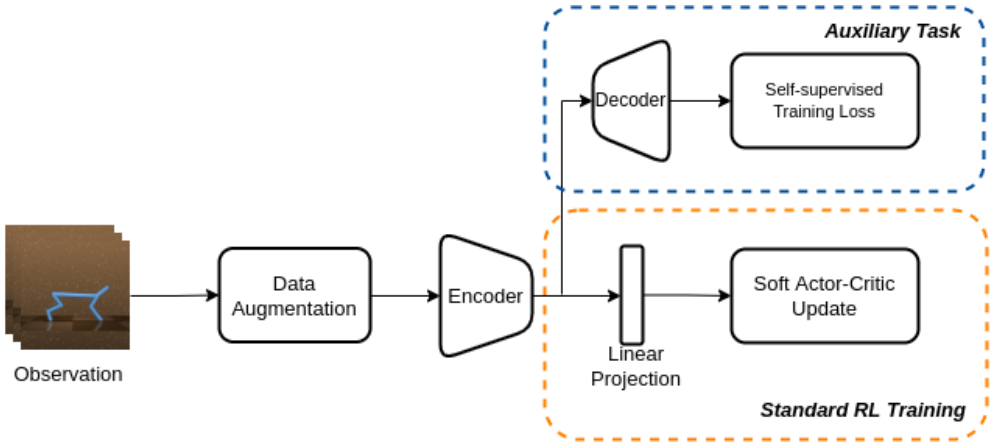
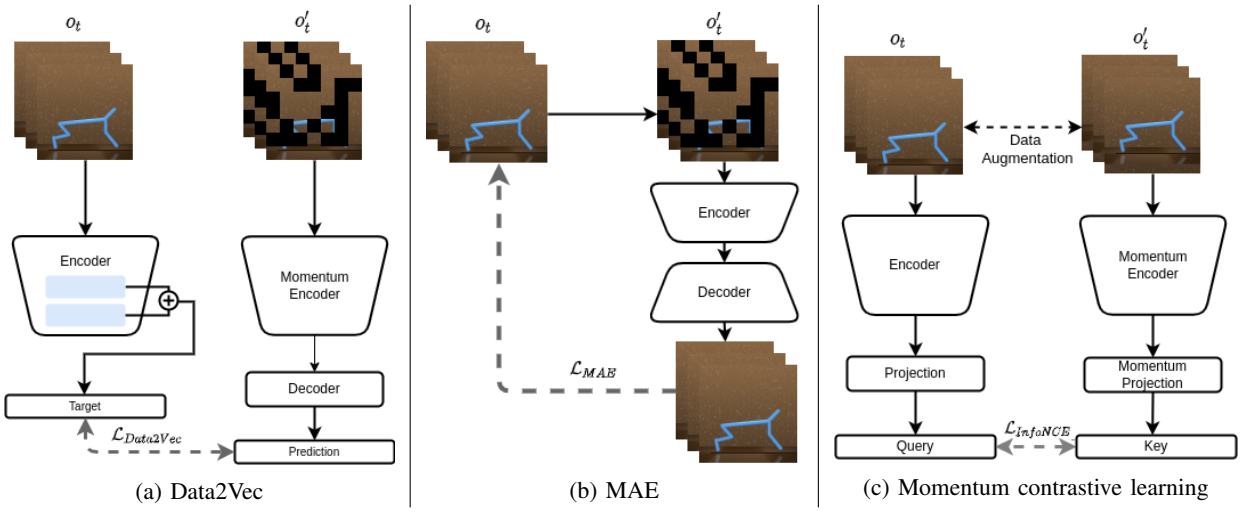


Fig. 1: High-level overview of our training pipeline for both ViT and CNN encoders. We keep the standard RL updates unchanged when we switch to ViT auxiliary tasks. We use random cropping as augmentation for RL training. For auxiliary ViT tasks proposed in [38], [35], we apply random masking as the augmentation stategy, and we adopt another randomly cropped view as augmented sample for contrastive learning [35], [12].



(a) Data2Vec (b) MAE (c) Momentum contrastive learning

Fig. 2: A detailed representation of the different auxiliary tasks used to guide the learning of the encoder. (a) Data2vec receives two variations of the same image and tries to match the reconstruction. (b) MAE uses an encoder-decoder architecture receiving as input an image with missing patches and tries to reconstruct the original image. (c) The contrastive loss (InfoNCE) draws together similar image pairs (different augmentations of the same image) while pushing apart dissimilar image pairs.

C. Momentum Contrastive Learning

Contrastive learning [23], [31], [39], [40] is designed to learn representations that obey similarity constraints in a self-supervised manner. The process of contrastive learning can be understood as a dictionary look-up task. Given an encoded query q and a dictionary of encoded keys $K = \{k_0, k_1, k_2, \ldots, k_n\}$, among which one key k^+, defined as the positive key, matches the query q, the objective of contrastive learning is to ensure the distance between the query q and the positive key k^+ is closer than the distance of the query q and any other key in the dictionary $K \setminus k^+$. The keys are commonly encoded via a momentum encoder [31], [41], [35] to enhance training stability. We adopt the loss design of CURL[12] where the distance is computed as bilinear products ($q^T W k$) with a learnable matrix W. InfoNCE is applied as the loss to ensure the similarity constraints:

$$\mathcal{L}_{\text{InfoNCE}} = \log \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{K-1} \exp(q^T W k_i)}, \quad (3)$$

As shown in Figure 2c, the encoder and momentum encoder compress the two sets of image stacks $\{o_t, o'_t\}$ into latent vectors as queries and keys. We treat the image stack yielded from the same image but augmented differently as the positive key k^+, and other image stacks in the batch as the negative keys $K \setminus k^+$. The momentum encoder is a moving average of the encoder updated according to Equation 2.

IV. Experiments and Results

We evaluate different choices of self-supervised learning losses, including *Data2Vec* [38], *MAE* [35], and contrastive

100K STEP SCORES	ViT	ViT w/ Data2Vec	ViT w/ Mae	ViT w/ Contrastive	RAD
FINGER, SPIN	293.06 ± 260.74	646.52 ± 247.59	***899.08±106.28***	850.36 ± 103.52	823.16 ± 169.13
CARTPOLE, SWING	852.73 ± 15.64	721.91 ± 283.46	*864.45±4.90*	844.48 ± 27.11	**871.50±9.87**
REACHER, EASY	125.72 ± 60.36	185.98 ± 73.35	383.52 ± 234.86	*440.52±88.83*	**897.56±73.12**
CHEETAH, RUN	263.54 ± 22.79	287.46 ± 36.44	*366.78±49.08*	270.40 ± 147.41	**584.30±15.83**
WALKER, WALK	562.92 ± 105.89	*587.84±101.82*	263.57 ± 228.29	377.67 ± 258.46	**875.04±138.11**
CUP, CATCH	172.06 ± 182.79	410.50 ± 192.20	***946.34±21.87***	804.28 ± 235.83	661.63 ± 140.37
500K STEP SCORES					
FINGER, SPIN	758.82 ± 381.72	***975.54±8.80***	951.62 ± 53.20	917.80 ± 127.19	843.86 ± 166.31
CARTPOLE, SWING	861.76 ± 14.02	866.74 ± 12.31	*868.34±10.96*	853.49 ± 32.11	**872.03±7.23**
REACHER, EASY	230.08 ± 77.42	234.28 ± 91.76	362.04 ± 83.65	*539.70±119.56*	**910.26±50.58**
CHEETAH, RUN	480.15 ± 50.48	539.86 ± 104.35	*551.58±50.48*	442.82 ± 96.81	**837.36±26.83**
WALKER, WALK	841.77 ± 48.30	*895.56±37.94*	844.29 ± 43.90	892.41 ± 78.93	**970.20±7.08**
CUP, CATCH	486.34 ± 394.58	963.04 ± 4.82	***973.64±5.78***	888.18 ± 132.07	925.16 ± 20.84

TABLE I: Average test reward for the environments in DMControl Suite at 100K and 500K training steps. Method achieving the best performance is highlighted in bold text while the best performance among the ViT variants is labelled in italic text.

learning approaches [12], [31] with ViT on continuous control tasks in DMControl Suite [19]. We choose a vanilla ViT pipeline without any auxiliary task as the baseline for comparison, and also report the results for RAD [13], the state-of-the-art (SOTA) convolutional architecture for RL from pixels for a complete view. We keep the number of parameters as similar as possible across all the experiments. To make our comparisons as fair as possible we kept the following details fixed in all ViT-based experiments: (i) Soft Actor-Critic (SAC) as RL algorithm, (ii) structure and dimensionality of ViT encoders, (iii) learning rate for auxiliary tasks and RL updates. We use hyperparameter values from the respective original papers and keep hyperparameters unchanged across the experiments, except for the masking ratio used in [35], [31]. Random cropping is selected as the image augmentation strategy for all the experiments in the standard RL training. Further implementation and hyperparameter details are described in subsection VI-B.

For evaluation, we use the average total reward at 100K and 500K gradient update scores, as listed in Table I. Each entry is evaluated by averaging across 5 runs with different random seeds. Corresponding learning curves are illustrated in Figure 3. Compared with a plain ViT model, adding any auxiliary task improves the learning performance. Among the three auxiliary training tasks, the masking-based tasks (*Data2Vec* and *MAE*) are clearly better than the contrastive learning one. *MAE* outperforms *Data2Vec* in most environments except the `WALKER,WALK` environment. When convolutional models (*RAD*) are also considered for comparison, *RAD* still achieves the SOTA performance in 4 out of the 6 experiments, while *Data2Vec* and *MAE* with ViT architecture reach the best accumulated reward in the `FINGER, SPIN` and `CUP,CATCH` environments.

V. Discussion and Future Work

Adding reconstruction-based auxiliary tasks [38], [35] with ViT encoders can significantly enhance performance while encoders built with CNNs still obtain the best performance in the majority of our experiments. *Data2Vec* and *MAE* both predict the features of a masked version of the input. *Data2Vec* reconstructs the deep features encoded by the momentum encoder while *MAE* chooses the original pixel values as the reconstruction target. The contrastive learning framework improves the performance of a plain ViT to a limited extent, which is analogous to the study on CNN [13], [12]. We suspect that a lack of diversity in the training images could cause the inferior performance of contrastive learning. Unlike the diverse *ImageNet* dataset often used as the baseline for computer vision research, images collected by the RL agent are more similar to each other. As a result, positive and negative keys are sometimes too similar to be distinguished for the contrastive objective.

Since transformer-based architectures are notorious for being data-hungry, we are also interested in further improving the sampling efficiency via pre-training. *First pre-train, then finetune* is a well-established paradigm for learning computer vision models to mitigate the issue of insufficient data. Due to a lack of datasets for control tasks, and more specifically RL, what serves as the best dataset for pre-training remains an open question. We plan to investigate the following options: (i) a large, general and diverse dataset such as ImageNet, (ii) a dataset generated from an expert policy, (iii) a dataset generated from a random policy, and (iv) a dataset containing a mixture of expert and non-expert demonstrations.

VI. Conclusion

This paper empirically explored the potential of several ViT methods for RL and compared the results to CNNs trained with RAD, a leading method for image-based RL for CNNs. Specifically, we evaluate vanilla ViTs, as well as the addition of various self-supervised losses (reconstructive and contrastive) to ViT as auxiliary tasks to the RL training. Our results show that ViTs trained with auxiliary tasks (self-supervised losses) are helpful for the RL-from-pixels problem, but they currently still fall short of what can be achieved with CNNs by RAD.

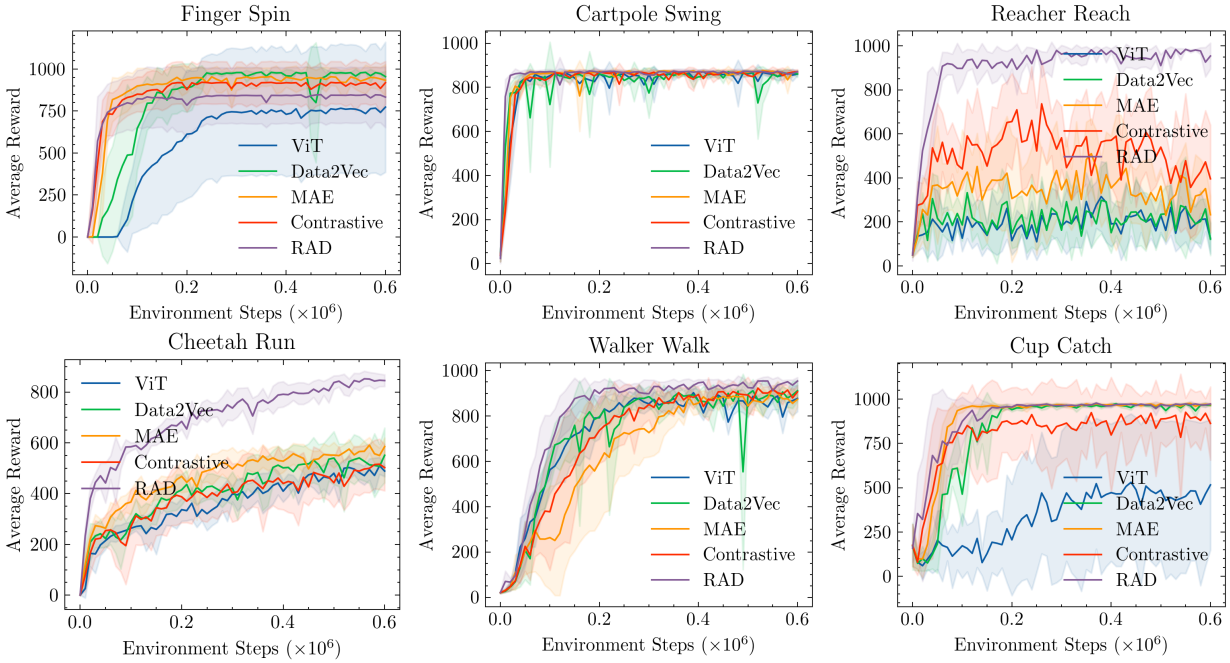


Fig. 3: Learning curves of Data2Vec, MAE and contrastive learning on DMControl Suite.

APPENDIX

A. Background

1) Reinforcement Learning: We formulate the RL problem as a Markov Decision Process (MDP). An MDP consists of the following elements: states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, a dynamics function $p(s_{t+1}|s_t, a_t)$ denoting the transition probability of a state-action combination (s_t, a_t) and a reward function $R(s_t, a_t)$. Model-free reinforcement learning algorithm often targets at maximizing the expected return as $J = \sum_{t=0}^{T} \gamma^t R(s_t, a_t)$ given a discount factor $\gamma \in [0, 1]$. The state variable for RL from pixels problems are commonly represented as a stack of consecutive image frames to infer the status.

Soft Actor-Critic (SAC) [42] is applied widely to image-based RL problems because of its excellent sampling efficiency and exploration strategy. SAC learns a policy network $\pi_\theta(s_t)$ and a critic network $Q^\pi_\phi(s_t, a_t)$ by optimizing the expected return and an entropy regularization term concurrently. The parameters in the critic ϕ are updated by minimizing the squared Bellman error given the transition tuples $\tau_t = (s_t, a_t, s_{t+1}, r_t)$ stored in the replay buffer D:

$$\begin{aligned} \mathcal{L}(\phi) &= \mathbb{E}_{\tau\sim D}[Q_\phi(s_t, a_t) - (r_t + y(r_t, s_{t+1}))]^2, \\ y(r_t, s_{t+1}) &= \gamma(\min_{i=1,2} Q'_{\phi_i}(s_{t+1}, a') - \alpha \log_{\pi_\theta}(a'|s_{t+1})) \end{aligned} \tag{4}$$

The parameter α is the temperature value to balance the two terms, which is treated as learned parameter as in [43]. Q' is a slowly updated copy of the critic to improve training stability. The actor is learned by maximizing the weighted objective of the expected return and the policy entropy as:

$$\mathcal{L}(\theta) = \mathbb{E}_{a\sim\pi}[Q_\phi(s_t, a) - \alpha \log_{\pi_\theta}(a|s_t)]. \tag{5}$$

2) Data Augmentation in Reinforcement Learning: Data augmentation has been found to be essential for the performance of RL from pixels. Common data augmentation techniques include random cropping, color jittering, flipping, rotating and gray scaling. We refer the reader to a detailed study for the comparison between different data augmentation strategies [13]. We adopt the most effective random cropping strategy to the images for all the experiments in this work. Intuitively, random cropping enhances translational invariance to the perception module. In our experiments, the rendered images have 100 × 100 pixels, which are later randomly cropped to 84 × 84 pixels.

3) Vision Transformer: The transformer architecture [14] raised the bar in most domains of machine learning, and have become the state of the art method in NLP tasks. The Vision Transformer (ViT) [15] instead takes the transformer architecture and adapts it to make it suitable for images, and has shown incredible results at classification challenges. ViT operates by subdividing an image into fixed-size patches. Each patch is then flattened and linearly transformed into an embedding which is concatenated to a learnable 1D positional embeddings to allow for spatial awareness between the input patches. This embedding is then finally processed through the standard transformer architecture.

More recently, ViTs have been used with self-supervised learning to learn good representations, examples of which are Data2Vec [38], Masked Autoencoders [35]. These works apply transformations to the unlabelled images and use losses that try to draw closer together representation of inputs coming from the same original image. A more complete explanation on these methods is described in section III.

Hyperparameter	Value
Observation rendering	(100, 100)
Observation downsampling	(84, 84) (random cropping)
Replay buffer size	100000
Initial steps	1000
Stacked frames	3
Action repeat	2 FINGER, SPIN; WALKER, WALK 8 CARTPOLE, SWING 4 otherwise
SAC hidden units(MLP)	1024
Evaluation episodes	10
Evaluation frequency	10000
Optimizer	Adam
Encoder learning rate	$1e-3$
Actor learning rate	$1e-3$
Critic learning rate	$1e-3$
Temperature learning rate	$1e-4$
Batch Size	512
Encoder EMA τ	0.05
Critic function EMA τ	0.01
Discount factor γ	0.99
Initial temperature	0.1
Latent dimension	128
Critic update frequency	2
Patch size	(12, 12)
ViT depth	4
ViT MLP dimension	128
Attention head	8
K in *Data2Vec*	2
β in *Data2Vec*	2.0

TABLE II: Hyperparameters used in the experiments.

B. Implementation Details

We adopt the soft actor-critic implementation offered by [44], which tunes the temperature automatically with a constrained optimization [43]. We list the hyperparameters in Table. II. For experiments using masked images as input, we experiment with masking ratio from $\{30\%, 40\%, 50\%, 60\%, 75\%\}$, and select the 40% and 75% for *Data2Vec* and *MAE* respectively. The mask is represented as a learnable vector optimized by the auxiliary task. We also find that selecting a small batch size for the contrastive learning can stabilize the training. Therefore, the batch size is 512 for the SAC update but 128 for the contrastive objective.

In our experiments, the policy is trained jointly with the auxiliary tasks. We adopt the training setup from *SAC+AE* [6]. *SAC+AE* [6] blocks the gradient signals from the actor to update the shared encoder while the critic has the privilege to update the encoder, which has been proven to greatly improve the performance.

The decoder in *Data2Vec* consists of two-layer MLPs with ReLU activation function. We choose a light-weight ViT decoder for *MAE*, which comprises 2 layers with 64 hidden units and 4 heads. Similar to the conclusion in *MAE*, the capacity of the decoder has limited influence on the overall performance. In our implementation of momentum contrastive learning, the queries and keys are generated by a separate head other than the one used for RL, which can be viewed as the decoder in Figure 1.

C. Extra Discussion on Image Augmentation

In our work, we apply random cropping to all the experiments as the augmentation technique. The impact of data augmentation with ViT can be further investigated. A corresponding study is conducted with CNN [13]. Since ViT operates on image patches, diverse augmentation strategies [45], [46], [47], [48], [49] have been developed to improve training ViT for computer vision tasks. Finding the most effective augmentation technique for RL with ViT is still of great interest to the RL researchers.

REFERENCES

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[2] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne, "Learning locomotion skills for cassie: Iterative design and sim-to-real," in *Proc. Conference on Robot Learning (CORL 2019)*, 2019.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[5] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, *et al.*, "Urban driving with conditional imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 251–257.

[6] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving sample efficiency in model-free reinforcement learning from images," 2019.

[7] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *Advances in neural information processing systems*, vol. 31, 2018.

[8] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner, "Darla: Improving zero-shot transfer in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1480–1490.

[9] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," *Advances in neural information processing systems*, vol. 31, 2018.

[10] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International Conference on Learning Representations*, 2020.

[11] ——, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=GY6-6sTvGaf

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

[13] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 884–19 895, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

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

[17] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[18] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[19] Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess, "dm_control: Software and tasks for continuous control," 2020.

[20] N. Hansen, H. Su, and X. Wang, "Stabilizing deep q-learning with convnets and vision transformers under data augmentation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, "Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6131–6141.

[22] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," *arXiv preprint arXiv:2007.05929*, 2020.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[24] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv e-prints*, pp. arXiv–1807, 2018.

[25] K.-H. Lee, I. Fischer, A. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama, "Predictive information accelerates learning in rl," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 890–11 901, 2020.

[26] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9870–9879.

[27] T. Nguyen, T. M. Luu, T. Vu, and C. D. Yoo, "Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3471–3477.

[28] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[29] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[30] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.

[31] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.

[32] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9640–9649.

[33] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9650–9660.

[34] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," *arXiv preprint arXiv:2111.07832*, 2021.

[35] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.

[36] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," *arXiv preprint arXiv:2112.09133*, 2021.

[37] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[38] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," *arXiv preprint arXiv:2202.03555*, 2022.

[39] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4182–4192.

[40] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[41] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

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

[43] ——, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[44] D. Yarats and I. Kostrikov, "Soft actor-critic (sac) implementation in pytorch," https://github.com/denisyarats/pytorch_sac, 2020.

[45] E. Cubuk, B. Zoph, J. Shlens, Q. R. Le, and Randaugment, "Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017.

[46] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[47] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[48] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[49] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry, "Augment your batch: Improving generalization through instance repetition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8129–8138.