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

© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230359

Pretraining the Vision Transformer Using Self-Supervised Methods for Vision Based Deep Reinforcement Learning

Manuel Goulão[a,b,c;*] **and Arlindo L. Oliveira**[a,b]

[a]Instituto Superior Técnico
[b]INESC-ID
[c]NeuralShift
ORCiD ID: Manuel Goulão https://orcid.org/0000-0001-6478-2038, Arlindo L. Oliveira https://orcid.org/0000-0001-8638-5594

Abstract. The Vision Transformer architecture has shown to be competitive in the computer vision (CV) space where it has dethroned convolution-based networks in several benchmarks. Nevertheless, convolutional neural networks (CNN) remain the preferential architecture for the representation module in reinforcement learning. In this work, we study pretraining a Vision Transformer using several state-of-the-art self-supervised methods and assess the quality of the learned representations. To show the importance of the temporal dimension in this context we propose an extension of VICReg to better capture temporal relations between observations by adding a temporal order verification task. Our results show that all methods are effective in learning useful representations and avoiding representational collapse for observations from the Atari Learning Environment (ALE) which leads to improvements in data efficiency when we evaluated in reinforcement learning (RL). Moreover, the encoder pretrained with the temporal order verification task shows the best results across all experiments, with richer representations, more focused attention maps and sparser representation vectors throughout the layers of the encoder, which shows the importance of exploring such similarity dimension. With this work, we hope to provide some insights into the representations learned by ViT during a self-supervised pretraining with observations from RL environments and to understand which properties arise in the representations that lead to the best-performing agents.

1 Introduction

In recent years, a new architecture for vision-based tasks that does not use convolutions called the Vision Transformer (ViT) [18] has shown impressive results in several benchmarks. This architecture presents much weaker inductive biases when compared to a CNN, which can result in lower data efficiency. The Vision Transformer, unlike the CNNs, can capture relations between parts of an image (patches) that are far apart from each other, thus deriving global information that can help the model perform better in certain tasks. When the model is pretrained, using supervised or self-supervised learning, it manages to surpass in some cases the best convolution-based models in terms of task performance. Nonetheless, despite the successes obtained in computer vision these results are yet to be seen in reinforcement learning. Moreover, while some areas of machine learning have transitioned to large pretrained models, current Deep RL research is still largely based on small neural networks that are trained from *tabula rasa*.

Despite the successes of deep reinforcement learning agents in the last decade, these still require a large amount of data or interactions to learn good policies. This data inefficiency makes current methods difficult to apply to environments where interactions are more expensive or data is scarce, which is the case in many real-world applications. In environments where the agent does not have full access to the current state, i.e. partially observable environments, this problem becomes even more prominent, since the agent not only needs to learn the state-to-action mapping but also a state representation function that tries to be informative about the current state given an observation. In contrast, humans, when learning a new task, already have a well-developed visual system and a good model of the world which are components that allow us to easily learn new tasks. Previous works have tried to tackle the sample inefficiency problem by using auxiliary learning tasks [43, 45, 24], that try to help the network's encoder to learn good representations of the observations given by the environments. These tasks can be supervised or unsupervised and can happen during a pretraining phase or during a reinforcement learning phase in a joint-learning or decoupled-learning scheme.

Recent results have shown that self-supervised learning is very useful in computer vision. Increased interest in this area has resulted in the appearance of new and improved methods that train a network to learn important features from the data using only the data itself as supervision. A common approach to evaluating such methods is to train a network composed of the pretrained encoder, with the parameters frozen, and a linear layer using popular datasets, like ImageNet. These evaluations have shown that these methods can achieve high scores in different benchmarks, which shows how well the current state-of-the-art methods are able to encode useful information from the given images without being task-specific. Additionally, it has been shown that pretraining a network using self-supervised learning (or unsupervised) adds robustness to the network and gives better generalization capabilities [19].

Motivated by the potential of the Vision Transformer, in particular when paired with a pretraining phase, and the increasing interest in self-supervised tasks for DRL, we study pretraining ViT using

* Email: manuel.silva.goulao@tecnico.ulisboa.pt

state-of-the-art (SOTA) self-supervised learning methods for images. However, unlike images from datasets like ImageNet or MSCOCO, observations from reinforcement learning environments share similarities in more dimensions, for example, time [45, 5], semantics [20, 60], and behavior [1]. To show the importance of these dimensions in comparison to current SOTA methods we propose extending VICReg (Variance Invariance Covariance Regularization) [7] with a temporal order verification task [36] to help the model better capture the temporal relations between consecutive observations. We named this approach Temporal Order Verification-VICReg or in short TOV-VICReg. While we could have adapted any of the other methods, we opted for VICReg due to its computational performance, simplicity, and robustness against collapse.

We evaluate the different pretrained encoders in a data-efficiency regime and a linear probing task to determine which methods produce a better initialization for the model, assess if any pretrained model shows signs of representational collapse, and conduct a series of experiments to better understand the properties present in the representations. In our discussion, we also highlight some of the challenges that we faced during the experiments and propose some changes that can alleviate them.

Our main contributions are:

- A proposal to combine two pretext tasks, VICReg and temporal order verification, to capture temporal relations between consecutive observations in reinforcement learning environments, in Section 4.
- The evaluation and comparison of the different self-supervised learning methods in Reinforcement Learning (Section 6.1) and linear probing task (Section 6.2) based on imitation learning. The ViT pretrained with TOV-VICReg appears as the best performing model.
- A comparison of the different pretrained models using cosine similarity between the representations, attention maps and ratio of zeros in each layer of ViT (Section 8). The results show that TOV-VICReg produces richer representations, more focused attention maps and sparser representation vectors.

2 Related Work

Vision Transformer for vision-based Deep RL Recent work, has compared the Vision Transformer to convolution-based architectures with a similar number of parameters and shows that ViT is very data inefficient even when paired with an auxiliary task [48].

Pretraining representations Previous work has explored, similarly to our approach, pretraining representations using self-supervised methods which led to great data-efficiency improvements in the fine-tuning phase [43, 59] or superior results in evaluation tasks, like AtariARI [5]. Others have pretrained representations using RL algorithms, like DQN, and transferred those learned representations to a new learning task [52].

Joint learning and augmentations In recent years, adding an auxiliary loss to the RL loss, usually called joint learning, has become a common approach by many proposed methods. Curl [44] adds a contrastive loss using a siamese network with a momentum encoder. Another work studies different joint-learning frameworks using different self-supervised methods [34]. SPR [42] uses an auxiliary task that consists in training the encoder followed by an RNN to predict the encoder representation k steps into the future. PSEs [1] combines a policy similarity metric (PSM), that measures the similarity of states in terms of the behaviour of the policy in those states, and a contrastive task for the embeddings (CME) that helps to learn more robust representations. PBL [24] learns representations through an interdependence between an encoder, which is trained to be informative about the history that led to that observation, and an RNN that is trained to predict the representations of future observations. Proto-RL [58] uses an auxiliary self-supervised objective to learn representations and prototypes [11], and uses the learned prototypes to compute intrinsic rewards that will push the agent to explore the environment.

A big contributor to the success of some joint learning methods has been the use of augmentations. Methods like DrQ [31] and RAD [32] pair an RL algorithm, like SAC, with image augmentations to improve data efficiency and generalization of the algorithms without using any auxiliary function.

Self-Supervised learning for image sequences Multiple works propose simple pretext tasks to train encoders to capture information from image sequences. These pretexts tasks can be playback speed classification [57], a temporal order classification [36, 33, 56], a jigsaw game [4] or a masked modelling task [46]. A different approach uses contrastive learning. In this category, we can find methods that maximise the similarity between image sequences [21], use autoregressive models to predict frames multiple steps in the future [35], and maximize the similarity between temporally adjacent frames [30].

In the context of RL, works have also explored learning representations that have temporal information encoded. ATC (Augmented Temporal Contrast) [45] trains an encoder to compute temporally consistent representations using contrastive learning, and the ST-DIM (SpatioTemporal DeepInfoMax) [5] captures spatial-temporal information by maximizing the mutual information between features of two consecutive observations.

3 Background

3.1 Vision Transformer

ViT [18] is a model, for image classification tasks, that doesn't rely on CNNs and uses self-attention mechanisms. The model wraps the encoder of a Transformer by using linear projections of the patches extracted from the input image as tokens and adding a classification token which after the computation will serve as the image representation. When compared to CNNs, ViT presents weaker image-specific inductive biases which can impact the sample-efficiency of the model during learning [16]. However, it has been shown that with enough data the image-specific inductive biases become less important [18]. Moreover, ViT can capture relations between patches that are far apart from each other, thus deriving global information that can help the model perform better in certain tasks.

3.2 Reinforcement Learning

The problem of an **agent** learning to solve a task in a certain **environment** can be defined as a Markov Decision Process (MDP). A MDP $\mathcal{M}$ is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $\mathcal{R}$ the reward function, and $\mathcal{T}$ the transition function. At each timestep the agent is in a state $s \in \mathcal{S}$ and takes an action $a \in \mathcal{A}$. Upon performing the action the agent receives from the environment a reward $r \in \mathcal{R}$ and a new state $s' \in \mathcal{S}$

which is determined by the transition function $\mathcal{T}(s', s, a)$. The MDP assumes that the Markov property holds in the environment, i.e. that the state transitions are independent and the agent only needs to know the current state to perform an action $P(a_t|x_0, x_1...x_t) = P(a_t|x_t)$. For the agent to decide what action to take it uses a policy function π, which gives a distribution over actions given a state, $\pi(a_t|s_t)$. This policy is evaluated using the function $V^{\pi}(s)$, which estimates the expected total discounted reward of an agent in a state s and that follows a policy π.

3.2.1 *DQN and Rainbow*

DQN [37] is a value-based method and uses a network with parameters ϕ that given a state s outputs a prediction of the distribution of Q values over actions, $Q_{\phi}(s, a)$. The network learns the Q function by minimizing the mean squared error: $(y - Q_{\phi}(s, a))^2$, where $y = r + \gamma \, max_{a'} Q_{\phi}(s', a')$.

Several works followed the DQN algorithm which introduced changes to improve performance. Rainbow [28] combines six improvements, Double Q-Learning [49], Prioritized Replay [41], Dueling Networks [53], Multi-step Learning [47], Distributional RL [8], and Noisy Nets [22] resulting in a more stable and sample efficient algorithm.

3.3 *Self-Supervised methods*

For this study we selected DINO [12], MoCo [14], MAE [26], and VICReg [7] since they are currently considered state-of-the-art, their official implementations are available in PyTorch, and each represents a different type of approach. MoCo [27] is a contrastive learning method meaning that it learns using a loss function that pulls the positive samples together and pushes the negative samples apart. MoCo, in particular, has three versions 1 [27], 2 [13], and 3[14]. In this work, we consider the most recent version (v3). On the other hand, non-contrastive methods (also called regularized) don't rely on the notion of positive and negative samples and only attempt to push different views from the same source together. To avoid collapse these methods use a set of tools that act as regularization, e.g. stop gradient, strong augmentations, and asymmetric siamese networks. From this class of methods, we consider DINO [12] and VICReg. Lastly, we also consider MAE [26], a masked reconstruction method, which consists in training an auto-encoder based on ViT to reconstruct an image with a set of patches masked.

4 TOV-VICReg

VICReg is a non-contrastive method that trains a network to be invariant to augmentations applied to the inputs while avoiding a trivial solution with the help of two additional losses, called variance and covariance, that act as regularizers over the embeddings. While VICReg is agnostic concerning the architectures used and even the weight sharing, in this work we consider the version where paths are symmetric, the weights are shared, and each path is composed of an encoder (also called backbone) and an expander. The expander is a network that increases the dimension of the representation vector in a non-linear way allowing the covariance loss to reduce dependencies and not only correlations of the representation vector. In addition, the expander also removes information that is not common to both representations.

VICReg uses three loss functions: **invariance** is the mean of the square distance between each pair of embeddings from the same original image, as shown in Equation 1, where Z, and Z' are two sets of embeddings, of size N, that result from computing two different augmentations of N sources, and z_j denotes the *j-th* embedding in the set; **variance** is a hinge loss that computes, over the batch, the standard deviation of the variables in the embedding vector and pushes that value to be above a certain threshold, as shown in Equation 2, where d denotes the number of dimensions of the embedding vector, and Z^j is the set of the *j-th* variables in the set of embedding Z; **covariance** is a function that computes the sum of the squared off-diagonal coefficients of a covariance matrix computed over a batch of embeddings, as shown in Equation 3, to decorrelate the variables from the embedding. While the invariance loss function tries to make the model invariant to augmentations, i.e. output the same representation vector, the other two functions act as regularizers by pushing the variables of the embedding vector to vary above a certain threshold and decorrelating the variables in each embedding vector.

$$i(Z, Z') = \frac{1}{N} \sum_{j}^{N} \left\| z_j - z'_j \right\|_2^2 \tag{1}$$

$$v(Z) = \frac{1}{d} \sum_{j}^{d} \max(0, \gamma - \sqrt{Var(Z^j)}) \tag{2}$$

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [\mathrm{Cov(Z)}]_{i,j}^2 \tag{3}$$

TOV-VICReg or Temporal-Order-Verification-VICReg extends VICReg to better capture the temporal relations between consecutive observations and consequently encode extra information that can be useful in the deep reinforcement learning phase. To achieve that we add a new temporal order verification task, as proposed at Shuffle-and-Learn [36], that consists of a binary classification task where a linear layer learns to predict if three given representation vectors are in the correct order or not. Like the other losses, we also employ a coefficient for the temporal loss and in most of our experiments, the value is 0.1. Figure 1 visually illustrates TOV-VICReg.

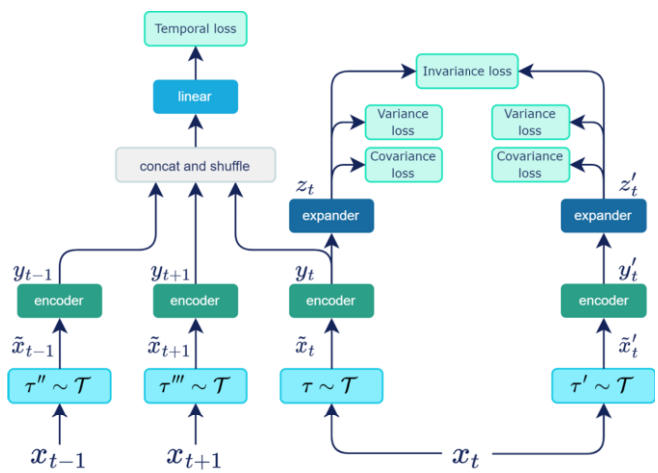

Figure 1. TOV-VICReg architecture

At each step we sample three consecutive observations, $\{x_{t-1},\ x_t,\ x_{t+1}\}$. x_t is processed by two different augmentations,

and like VICReg these are the augmentations used in BYOL [23]. x_{t-1} and x_{t+1} are processed by two simple augmentations composed of a color jitter and a random grayscale. The x_t augmentations are computed by the VICReg computation path and the resultant embeddings are used with VICReg loss functions, i.e. variance, invariance, and covariance. For the **temporal order verification task** we encode the augmentation of x_{t-1} and x_{t+1}, and concatenate those two representations with one of the representations of x_t. In our case we used the one that was augmented without solarize, obtaining the vector $\{y_{t-1}, y_t, y_{t+1}\}$. Finally, we randomly permute the order of the representations in the vector and feed the resultant concatenated vector to a linear layer with a single output node that predicts if the given concatenated vector has the representations in the right order or not. The **temporal loss** used to optimize the model for this task is a binary cross entropy loss.

5 Pre-Training Methodology

We pretrained five encoders, one using our proposed method TOV-VICReg and four using state-of-the-art self-supervised methods: MoCo v3 [14], DINO [12], VICReg [7], and MAE [26]. For this study, the encoder used is a Vision Transformer, more precisely the ViT tiny. We use a patch size of 8 for all SSL methods except MAE where the value is 7 since it requires the observation size (84) to be divisible by the patch size. Our experiments show that the patch size of 7 used for MAE does not affect the results. Moreover, the implementation we use is an adaptation of the timm library [55] implementation, which can be found in the source code of the DINO method. The dataset used is a set of observations from 10 of the 26 games in the Atari 100k benchmark, all available in the DQN Replay Dataset [2]. For each game, we use three checkpoints (1,25,50) with a size of one hundred thousand data points (observations), which makes up a total of three million data points (~55 hours). The pretraining phase is 10 epochs with two warmup epochs. We used the official code bases of all the self-supervised methods and tried to change the least amount of hyperparameters.

6 Representations Evaluation

To evaluate the pretrained Vision Transformers we perform two experiments. In the first experiment, we evaluate the pretrained representations in a reinforcement learning setting and compare the data-efficiency gains. In the second experiment, we evaluate the pretrained representations using a linear probing task based on imitation learning.

6.1 Data-Efficiency in RL

To evaluate the pretrained Vision Transformers in reinforcement learning and compare data-efficiency gains, we trained in the 10 games used for pre-training for 100k steps using the Rainbow algorithm [28], with the DER [50] hyperparameters. The only difference between the agents at the start is the representation module. We chose two networks to compare against, the Nature CNN [38], and a ResNet that has a similar number of parameters similar to the ViT tiny. Moreover, we use a learning rate two orders of magnitude smaller for the encoder (1×10^{-6}), which previous works [43] and experiments performed by us have shown to be beneficial. To report our results we follow the rliable [3] evaluation framework, where the scores of all games are normalized and treated as one single task.

Figure 2 shows the aggregate metrics of seven different encoders on 10 Atari games with training runs of 100k steps. The first five (ViT+<method>) are ViT tiny models pretrained with five different self-supervised methods, while the last three (ViT, ResNet, and Nature CNN) are randomly initialized models.

Starting with the randomly initialized models we can assess that the Nature CNN and the ResNet are the most sample efficient models, with ViT far behind.

Regarding the pretrained encoders, ViT, when pretrained with TOV-VICReg, performs better than the other pretrained encoders and the non-pretrained ViT in all metrics except in the median.

The observed difference between the behavior of the mean and the median is explained by the fact that the distribution of scores obtained by TOV-VICReg has a long tail to the right. In fact, TOV-VICReg is the method that more commonly exhibits behavior that surpasses human performance, as shown by the optimality gap, pointing to the possibility that in some fraction of the cases, it finds good representations that allow the agent to learn a good policy faster. It is worth noting that we found a higher variance in the results of our proposed method, when compared to the remaining methods and non-pretrained models.

All self-supervised methods prove to be effective in improving the data-efficiency of ViT with MoCo showing the best results in IQM among the SOTA methods, followed by DINO, VICReg and MAE, respectively.

ViT+TOV-VICReg when compared to Nature CNN, which has far fewer parameters, and ResNet, with a similar number of parameters, seems to closely match their sample-efficiency performance. Furthermore, the difference between the ViT+TOV-VICReg and ViT+VICReg shows that exploring temporal relations results in better representations. Lastly, comparing the ViT+TOV-VICReg with the non-pretrained ViT shows that a good self-supervised method with 3 million data points can help close the sample-efficiency gap while remaining a more complex and capable model.

6.2 Linear Probing

Evaluating representations computed by a pretrained encoder is a difficult task. One possible option is assessing improvements in data efficiency in a reinforcement learning task, as we did in the previous section. However, the results usually suffer from a high level of uncertainty which requires us to run dozens of training runs, thus making it computationally expensive. Another possible path would be using previously proposed benchmarks like the AtariARI benchmark [5], which tries to evaluate representations using the RAM states as ground truth labels. However, this only works for 22 Atari games (out of 62) and requires the encoder to use the full observation provided by the environments (160x210). For those reasons, we use a different evaluation task that is more efficient, allowing us to test more pretrained models during the research process ($\sim$ 50 min per game), and flexible, meaning that we can use it in different environments.

Our second experiment consists in linear probing pretrained encoders in an imitation learning task. We present the results in Table 1, where we compare the pretrained encoders against a random classifier, i.e. uniform sampling, a randomly initialized ViT and a non-frozen encoder which we use as a goal. All methods were trained for 100 epochs except the latter which we trained for 300. The results are aligned with the results from the previous section with all methods showing improvements in comparison to the randomly initialized ViT. Once again ViT+TOV-VICReg shows better performance than the remaining pretrained encoders.

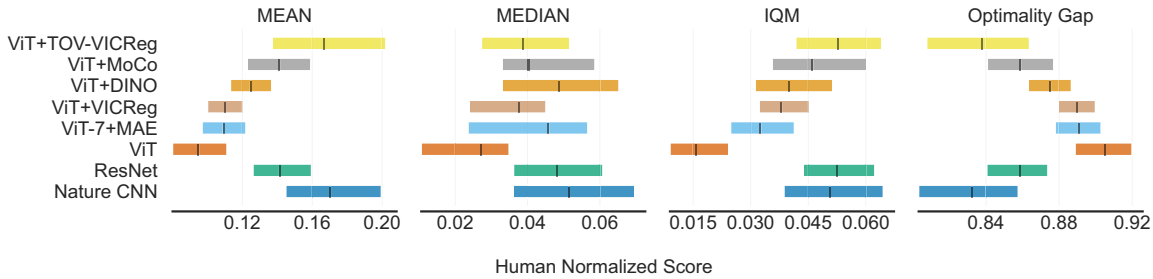

Figure 2. The eval runs across the different games are normalized and treated as a single task. The IQM corresponds to the Inter-Quartile Mean among all the runs, where the top and bottom 25% are discarded and the mean is calculated over the remaining 50%. The Optimality Gap refers to the number of runs that fail to surpass the human average score, i.e. 1.0.

		Random init	Pre-trained encoders					W/o freeze
Game	Random Classifier	ViT	ViT+ DINO	ViT+ MoCo	ViT+ VICReg	ViT-7+ MAE	ViT+ TOV-VICReg	Nature CNN
Alien	0.0556	0.0147	0.0470	0.0646	0.0695	0.0988	0.1003	**0.1021**
Assault	0.1519	0.1770	0.2536	0.2557	0.3704	0.3065	0.3044	**0.6673**
BankHeist	0.0608	0.0756	0.1059	0.1083	0.1467	0.1523	0.1622	**0.2080**
Breakout	0.2509	0.2183	0.3591	0.2765	0.4077	0.3099	0.3285	**0.5907**
Chopper Command	0.0563	0.0176	0.0383	0.2019	0.1298	0.3088	**0.3225**	0.2660
Freeway	0.3999	0.6843	0.6850	0.6972	0.6971	0.6942	0.7041	**0.8885**
Frostbite	0.0565	0.0367	0.0517	0.0744	0.0664	0.1001	**0.1021**	0.1019
Kangaroo	0.0603	0.0562	0.0877	0.1374	0.1259	0.0737	0.2184	**0.3311**
MsPacman	0.1121	0.0780	0.1215	0.1168	0.1400	0.1500	0.1527	**0.2063**
Pong	0.1644	0.0718	0.1447	0.2730	0.2337	0.1223	0.2853	**0.4340**
Mean	0.1369	0.1430	0.1894	0.2206	0.2387	0.1706	0.2680	**0.3796**

Table 1. F1-scores for each game evaluated and mean of the F1-scores. We trained all the encoders in all games separately for 100 epochs over a dataset of 100k observations and evaluate 10k unseen observations. The rightmost column shows the results of a Nature CNN encoder that was not frozen during the training phase and which we use as a goal for the remaining.

7 Collapse Evaluation

A significant phenomenon when doing self-supervised training is the collapse of the representations, which can be seen in three forms: representational collapse, dimensional collapse, and informational collapse. Representational collapse refers to the features of the representation vector collapsing to a single value for every input, leading to a variance of the features of zero, or close to zero. In dimensional collapse, the representations don't use the full representation space, which can be measured by calculating the singular values of the covariance matrix calculated over the representations. Informational collapse corresponds to the case where the features of the representation vector are correlated and therefore are representing the same information.

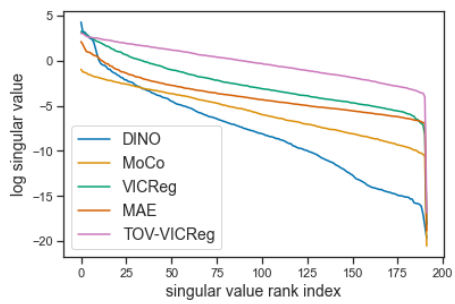

Figure 3. Logarithm of the singular values of the representation vector's covariance matrix sorted by value.

Dimensional Collapse All methods seem to avoid dimensional collapse, since most dimensions have a singular value larger than zero, as observed in Figure 3. However, we notice that some methods make better use of the space available since they present higher singular values. TOV-VICReg, in particular, seems to excel in this metric, even improving the results obtained by VICReg. It is worth noting that both VICReg and TOV-VICReg employ a covariance loss that helps decorrelate the embedding variables which may be contributing positively to these results. Furthermore, we used a covariance coefficient of 10 for TOV-VICReg and 1 for VICReg a change that according to our experiments culminates in the increase here observed.

Representational Collapse Results in the first row of Table 2 show the computed standard deviation of the representation vector over a batch of thousands of data points. DINO, VICReg and TOV-VICReg show a value well above zero, meaning that none of the methods suffered from representation collapse during training. On the other hand, MoCo shows a much smaller value of 0.178, which is still, is far from a complete collapse. Both VICReg and TOV-VICReg use a hinge loss that pushes the representation vector to have a standard deviation of 1 or above. While VICReg slowly converges to this value our method converges to roughly 1.65, which might be the result of adding a temporal order verification task.

Metric	DINO	MoCo	VICReg	MAE	TOV-VICReg
Std	0.979	0.178	1.003	0.475	1.648
Corr. Coef.	0.1764	0.1538	0.1531	0.1602	0.0780

Table 2. Average standard deviation and correlation coefficient of the representation vector

Informational Collapse We report in the second row of Table 2, the comparison of the average correlation coefficients of the representation vectors. TOV-VICReg performs better than the other methods, including VICReg, all of them with very similar coefficients. Like in the dimensional collapse, this result is in part due to the higher covariance coefficient used in TOV-VICReg which by design helps the model to decorrelate the representation's features. Increasing the coefficient in VICReg results in a lower correlation coefficient as well, but is still higher than TOV-VICReg.

8 Analysis of the Representations

In this section, we present different visualizations to better understand the representations learned by each of the pretrained encoders. Our goal with the following visualizations is to help us better understand the learned representations, give some intuitions about their properties, and understand which properties are present in the encoders that performed better in Section 6.1 and 6.2.

Cosine similarity Figure 6 presents a similarity matrix of the representations where we can observe that TOV-VICReg can better distinguish between observations of different games but also observations from the same game, as shown in Figure 7. MoCo, on the other hand, seems to make a good distinction between observations from the different games. However, as we can observe in the colour bar, all the representations are very similar to each other, which corroborates the results obtained in Section 7. Oppositely, VICReg and DINO manage to spread representations more, as we can see in the colour bars, but, the yellow squares in the diagonal show that the representations from the same game are more similar to each other which is corroborated by Figure 7. Given the empirical results, we believe that this capacity to distinguish observations from the same game might be a good indicator.

Attention visualisation Inspired by the results presented in the DINO work [12], we perform an analysis of the attention maps of the different pretrained encoders. In Figure 4, we can see the results of all methods for an observation from the game of Pong, where each method produces three attention maps, one for each self-attention head of the last block of the Vision Transformer. All pretrained ViT seem to attend at some level to important game features like the ball and the paddles. However, TOV-VICReg is the only method that doesn't spread the attention to other parts of the frame that we don't consider important to describe the current state of the game. When comparing to VICReg's attention maps we believe that the temporal order verification task greatly helped the attention of the model. In more visually complex games, e.g. Freeway or MsPacman, these attention maps start to be more difficult to analyze but it is still possible to discern some important features.

Sparsity Figure 5 shows the ratio of zeros of the representation vector after the activation of the MLP across the different layers of Vision Transformer. We can see that TOV-VICReg has a higher sparsity than the other methods and that the sparsity increases after each layer of the network. Sparsity has been exploited to scale transformers to larger sizes while maintaining a reasonable number of floating point operations.

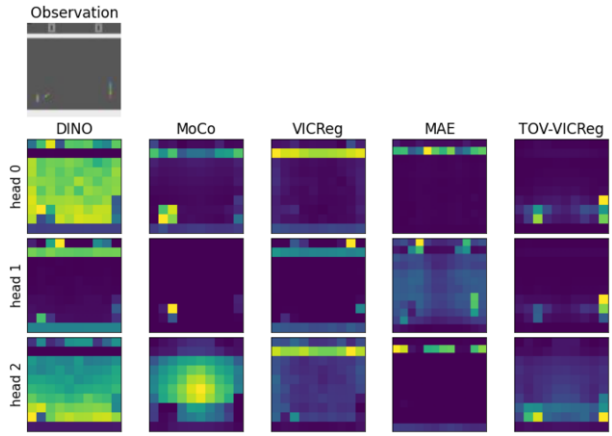

Figure 4. Attention maps produced by the pretrained ViTs. We fed a pretrained ViT with an observation from the game Pong and obtained the attention maps from the three heads in the last block.

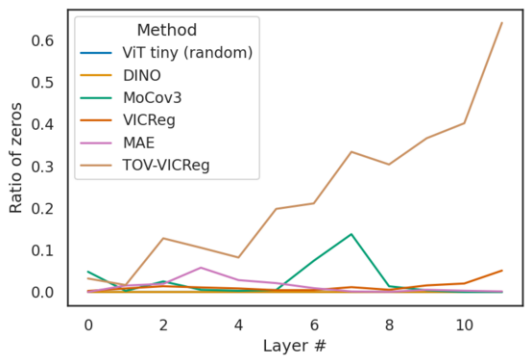

Figure 5. The ratio of zeros of the representation vector after the activation of the MLP across the different layers of Vision Transformer.

9 Discussion & Conclusion

In this work, we presented a study of ViT for vision-based deep reinforcement learning using self-supervised pretraining, and proposed a simple self-supervised learning method that extends VICReg to better capture temporal relations between consecutive observations. This type of approach has seen successes in natural language processing [17, 10], and computer vision [40] and we believe that similar approaches in RL have the potential to unlock new levels of performance never achieved before [6]. With this work, we hope to contribute to the growing body of work on self-supervised learning for RL and to provide important insights to the community on the importance of exploring dimensions where observations are similar and a better understanding of the representations learned during the self-supervised pretraining.

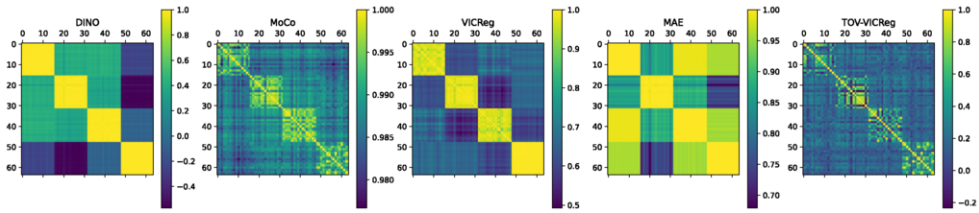

Figure 6. Similarity matrices of the representations computed by MoCo, DINO, VICReg, MAE and TOV-VICReg respectively. There are a total of 64 data points, from 4 different games: Alien, Breakout, MsPacman, and Pong, where from 0-15 are from Alien, 16-31 are from Breakout and so forth.

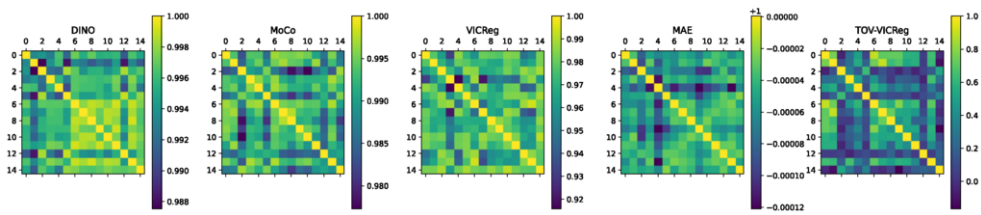

Figure 7. Similarity matrices of the representations computed by MoCo, DINO, VICReg, MAE and TOV-VICReg respectively, of observations from MsPacman.

Our results showed that pretraining a Vision Transformer using SOTA self-supervised learning methods is effective in improving the data-efficiency of RL agents and improving the performance of a linear probe of the encoder in an imitation learning task. Additionally, we showed that exploring the temporal relations between consecutive observations can further improve the results, as the encoder pretrained with TOV-VICReg was the best performing in both experiments. We used several metrics to assess if the pretrained encoders suffered from representational collapse and found that all methods were effective in avoiding this problem and that TOV-VICReg shows the best results, especially in the use of the representation vector dimensions and low level of correlation between variables of the representation vector. Moreover, our analysis of the representations shows that the best-performing encoders were also the ones with richer representations in the cosine similarity matrix, more focused attention maps and a higher sparsity. Sparsity in particular is a property that has been exploited to achieve better inference time and memory usage in the deployment of large Transformers [29, 25] and which can be important to successfully deploy reinforcement learning agents in real-world applications that use much more capable models.

Considering the impact of the temporal dimension on our results we believe that future work may improve these results by exploring other dimensions, like, semantics and behavior. Another type of evaluation where this kind of approach has the potential to be extremely important and which we do not explore in this work is the generalization to unseen tasks, which can be different in different dimensions like observation and state.

Using models such as the Vision Transformer which are predominantly larger than the models commonly used in RL and therefore slower to train was a great challenge during this work and we believe it is necessary to push for a change in some of the common practices in the field of reinforcement learning. Firstly, when training agents online it is necessary to use paralyzed environments like EnvPool [54] instead of running in a single processing environment. Additionally, as the SOTA progresses to larger and more robust models we will need environments that are capable to evaluate such capabilities. While ALE is still a valid option we believe that environments specifically designed for RL, like Procgen [15], are preferable and are, in our opinion, still missing. In the context of pretrained models for reinforcement learning, we have found the linear probing evaluation task an extremely valuable approach to evaluate the quality of the pretrained models in a fast and informative way. For those reasons, we find the adoption of such evaluations and the development of new ones of great value for advancing the field.

Lastly, while our best pretrained encoder was only able to match the sample efficiency of a Nature CNN we were able to achieve a good improvement in comparison to the non-pretrained Vision Transformer. The ability to use larger models, with millions of parameters, that are as sample efficient as some of the most popular CNN-based models (like Nature CNN or Impala ResNet), with thousands of parameters, can open the door to using Deep RL in even more complex problems where smaller models tend to struggle, without losing sample-efficiency.

10 Acknowledgments

We acknowledge the financial support provided by the Recovery and Resilience Fund towards the Center for Responsible AI project (Ref. C628696807-00454142), the Foundation for Science and Technology (FCT) through the Project PRELUNA - PTDC/CCI-INF/4703/2021 and also the multiannual financing of the Foundation for Science and Technology (FCT) for INESC-ID (Ref. UIDB/50021/2020).

References

[1] Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare, 'Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning', *arXiv:2101.05265*, (March 2021).

[2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi, 'An Optimistic Perspective on Offline Reinforcement Learning', *arXiv:1907.04543*, (June 2020).

[3] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare, 'Deep Reinforcement Learning at the Edge of the Statistical Precipice', in *Advances in neural information processing systems*, (2021).

[4] Unaiza Ahsan, Rishi Madhok, and Irfan Essa, 'Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition', in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 179–189. IEEE, (2019).

[5] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm, 'Unsupervised State Representation Learning in Atari', Technical Report arXiv:1906.08226, arXiv, (November 2020).

[6] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune, 'Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos', *arXiv:2206.11795*, (June 2022).

[7] Adrien Bardes, Jean Ponce, and Yann Lecun, 'Vicreg: Variance-invariance-covariance regularization for self-supervised learning', in *ICLR 2022-10th International Conference on Learning Representations*, (2022).

[8] Marc G. Bellemare, Will Dabney, and Rémi Munos, 'A distributional perspective on reinforcement learning', in *Proceedings of the 34th International Conference on Machine Learning*, eds., Doina Precup and Yee Whye Teh, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458, (06–11 Aug 2017).

[9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, 'OpenAI Gym', *arXiv:1606.01540*, (June 2016).

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, 'Language Models are Few-Shot Learners', in *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, (2020).

[11] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, 'Unsupervised Learning of Visual Features by Contrasting Cluster Assignments', in *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924, (2020).

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, 'Emerging Properties in Self-Supervised Vision Transformers', in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, (2021).

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, 'Improved Baselines with Momentum Contrastive Learning', *arXiv:2003.04297*, (March 2020).

[14] Xinlei Chen, Saining Xie, and Kaiming He, 'An Empirical Study of Training Self-Supervised Vision Transformers', *arXiv:2104.02057*, (August 2021).

[15] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman, 'Leveraging Procedural Generation to Benchmark Reinforcement Learning', *arXiv:1912.01588*, (July 2020).

[16] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun, 'ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases', *arXiv:2103.10697*, (June 2021).

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv:1810.04805*, (May 2019).

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', in *International Conference on Learning Representations*, (September 2020).

[19] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent, 'Why Does Unsupervised Pre-training Help Deep Learning?', in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 201–208, (March 2010).

[20] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar, 'Minedojo: Building open-ended embodied agents with internet-scale knowledge', in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, (2022).

[21] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He, 'A large-scale study on unsupervised spatiotemporal representation learning', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3299–3309, (2021).

[22] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg, 'Noisy Networks For Exploration', (February 2018).

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko, 'Bootstrap your own latent: A new approach to self-supervised Learning', *arXiv:2006.07733*, (September 2020).

[24] Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar, 'Bootstrap Latent-Predictive Representations for Multitask Reinforcement Learning', *arXiv:2004.14646*, (April 2020).

[25] Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention, 2021.

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[27] 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*, pp. 9729–9738, (2020).

[28] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver, 'Rainbow: Combining Improvements in Deep Reinforcement Learning', *arXiv:1710.02298*, (October 2017).

[29] Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Lukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva, 'Sparse is enough in scaling transformers', in *Advances in Neural Information Processing Systems*, eds., A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, (2021).

[30] Joshua Knights, Ben Harwood, Daniel Ward, Anthony Vanderkop, Olivia Mackenzie-Ross, and Peyman Moghadam, 'Temporally coherent embeddings for self-supervised video representation learning', in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8914–8921. IEEE, (2021).

[31] Ilya Kostrikov, Denis Yarats, and Rob Fergus, 'Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels', *arXiv:2004.13649*, (March 2021).

[32] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas, 'Reinforcement Learning with Augmented Data', in *Advances in Neural Information Processing Systems*, volume 33, pp. 19884–19895, (2020).

[33] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang, 'Unsupervised representation learning by sorting sequences', in *Proceedings of the IEEE international conference on computer vision*, pp. 667–676, (2017).

[34] Xiang Li, Jinghuan Shang, Srijan Das, and Michael S. Ryoo, 'Does Self-supervised Learning Really Improve Reinforcement Learning from Pixels?', *arXiv:2206.05266*, (June 2022).

[35] Guillaume Lorre, Jaonary Rabarisoa, Astrid Orcesi, Samia Ainouz, and Stephane Canu, 'Temporal contrastive pretraining for video action recognition', in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 662–670, (2020).

[36] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert, 'Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification', in *Computer Vision – ECCV 2016*, eds., Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Lecture Notes in Computer Science, pp. 527–544, Cham, (2016).

[37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, 'Playing Atari with Deep Reinforcement Learning', *arXiv:1312.5602*, (December 2013).

[38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie,

Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, 'Human-level control through deep reinforcement learning', *Nature*, **518**(7540), 529–533, (February 2015).

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, 'Pytorch: An imperative style, high-performance deep learning library', in *Neural Information Processing Systems*, (2019).

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, 'Learning Transferable Visual Models From Natural Language Supervision', in *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, (July 2021).

[41] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, 'Prioritized Experience Replay', in *ICLR (Poster)*, (January 2016).

[42] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman, 'Data-Efficient Reinforcement Learning with Self-Predictive Representations', *arXiv:2007.05929*, (May 2021).

[43] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville, 'Pretraining Representations for Data-Efficient Reinforcement Learning', in *Advances in Neural Information Processing Systems*, volume 34, pp. 12686–12699, (2021).

[44] Aravind Srinivas, Michael Laskin, and Pieter Abbeel, 'CURL: Contrastive Unsupervised Representations for Reinforcement Learning', *arXiv:2004.04136*, (September 2020).

[45] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin, 'Decoupling Representation Learning from Reinforcement Learning', in *Proceedings of the 38th International Conference on Machine Learning*, pp. 9870–9879, (July 2021).

[46] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid, 'Videobert: A joint model for video and language representation learning', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7464–7473, (2019).

[47] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning, second edition: An Introduction*, November 2018.

[48] Tianxin Tao, Daniele Reda, and Michiel van de Panne, 'Evaluating Vision Transformer Methods for Deep Reinforcement Learning from Pixels', *arXiv:2204.04905*, (May 2022).

[49] Hado van Hasselt, Arthur Guez, and David Silver, 'Deep Reinforcement Learning with Double Q-Learning', in *Thirtieth AAAI Conference on Artificial Intelligence*, (March 2016).

[50] Hado P van Hasselt, Matteo Hessel, and John Aslanides, 'When to use parametric models in reinforcement learning?', in *Advances in Neural Information Processing Systems*, volume 32, (2019).

[51] Guido Van Rossum and Fred L. Drake, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA, 2009.

[52] Han Wang, Erfan Miahi, Martha White, Marlos C. Machado, Zaheer Abbas, Raksha Kumaraswamy, Vincent Liu, and Adam White, 'Investigating the Properties of Neural Network Representations in Reinforcement Learning', *arXiv:2203.15955*, (March 2022).

[53] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas, 'Dueling Network Architectures for Deep Reinforcement Learning', in *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1995–2003, (June 2016).

[54] Jiayi Weng, Min Lin, Shengyi Huang, Bo Liu, Denys Makoviichuk, Viktor Makoviychuk, Zichen Liu, Yufan Song, Ting Luo, Yukun Jiang, Zhongwen Xu, and Shuicheng YAN, 'Envpool: A highly parallel reinforcement learning environment execution engine', in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, (2022).

[55] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[56] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang, 'Self-supervised spatiotemporal learning via video clip order prediction', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10334–10343, (2019).

[57] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye, 'Video playback rate perception for self-supervised spatio-temporal representation learning', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6548–6557, (2020).

[58] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto, 'Reinforcement Learning with Prototypical Representations', in *Proceedings of the 38th International Conference on Machine Learning*, pp. 11920–11931, (July 2021).

[59] Albert Zhan, Philip Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin, 'A Framework for Efficient Robotic Manipulation', *arXiv:2012.07975*, (December 2020).

[60] Victor Zhong, Jesse Mu, Luke Zettlemoyer, Edward Grefenstette, and Tim Rocktäschel, 'Improving policy learning via language dynamics distillation', in *Advances in Neural Information Processing Systems*, eds., Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, (2022).