2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.01447

Local-guided Global: Paired Similarity Representation for Visual Reinforcement Learning

Hyesong Choi[1], Hunsang Lee[2], Wonil Song[3], Sangryul Jeon[4], Kwanghoon Sohn[3], Dongbo Min[1†]
[1]Ewha W. University [2]Hyundai Motor Company [3]Yonsei University [4]University of Michigan

Abstract

Recent vision-based reinforcement learning (RL) methods have found extracting high-level features from raw pixels with self-supervised learning to be effective in learning policies. However, these methods focus on learning global *representations of images, and disregard* local *spatial structures present in the consecutively stacked frames. In this paper, we propose a novel approach, termed self-supervised* ***P****aired* ***S****imilarity* ***R****epresentation* ***L****earning (PSRL) for effectively encoding spatial structures in an unsupervised manner. Given the input frames, the latent volumes are first generated individually using an encoder, and they are used to capture the variance in terms of local spatial structures,* i.e., *correspondence maps among multiple frames. This enables for providing plenty of fine-grained samples for training the encoder of deep RL. We further attempt to learn the global semantic representations in the action aware transform module that predicts future state representations using action vectors as a medium. The proposed method imposes similarity constraints on the three latent volumes; transformed query representations by estimated pixel-wise correspondence, predicted query representations from the action aware transform model, and target representations of future state, guiding action aware transform with locality-inherent volume. Experimental results on complex tasks in Atari Games and DeepMind Control Suite demonstrate that the RL methods are significantly boosted by the proposed self-supervised learning of paired similarity representations.*

1. Introduction

Deep reinforcement learning (RL) has been an appealing tool for training agents to solve various tasks including complex control and video games [12]. While most approaches have focused on training RL agent under the assumption that compact state representations are readily available, this assumption does not hold in the cases where raw visual observations (*e.g.* images) are used as inputs for training the deep RL agent. Learning visual features from raw pixels only using a reward function leads to limited performance and low sample efficiency.

To address this challenge, a number of deep RL approaches [1, 10, 38, 40, 43, 44, 46] leverage the recent advance of self-supervised learning which effectively extracts high-level features from raw pixels in an unsupervised fashion. In [38, 46], they propose to train the convolutional encoder for pairs of images using a contrastive loss [24, 50]. For training the RL agent, given a query and a set of keys consisting of positive and negative samples, they minimize the contrastive loss such that the query matches with the positive sample more than any of the negative samples [38, 46]. While the parameters of the query encoder are updated through backpropagation using the contrastive loss [50], the parameters of the key encoder are computed with an exponential moving average (EMA) of the query encoder parameters. The output representations of the query encoder are passed to the RL algorithm for training the agent. These approaches have shown compelling performance and high sample efficiency on the complex control tasks when compared to existing image-based RL approaches [31, 33, 51].

While these approaches can effectively encode the *global* semantic representations of images with the self-supervised representation learning, there has been no attention on the *local* fine-grained structures present in the consecutively stacked images. Our key observation is that spatial deformation, *i.e.*, the change in terms of the spatial structures across the consecutive frames, can provide plenty of local samples for training the RL agent. Establishing dense correspondence [19, 34, 39, 42, 55], which has been widely used for various tasks such as image registration and recognition in computer vision, can be an appropriate tool in modeling the local spatial deformation.

In this work, we propose a novel approach, termed self-supervised **P**aired **S**imilarity **R**epresentation **L**earning (PSRL), that learns representations for deep RL by effectively encoding the spatial structures in a self-supervised

This work was partly supported by IITP grant funded by the Korea government (MSIT) (No.RS-2022-00155966, AI Convergence Innovation Human Resources Development (Ewha Womans University)) and the Mid-Career Researcher Program through the NRF of Korea (NRF-2021R1A2C2011624). † Corresponding author: dbmin@ewha.ac.kr

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.01447

fashion. The query representations generated from an encoder are used to predict the correspondence maps among the input frames. A correspondence aware transform is then applied to generate future representations. We further extend our framework by introducing the concept of future state prediction, originally used for action planning in RL [8, 11], into the proposed action aware transform in order to learn temporally-consistent global semantic representations. The proposed method is termed 'Paired Similarity' as it encodes both local and global information of agent observations. More structured details of the terms are provided in the supplementary material due to lack of space. To learn the proposed paired similarity representation, we impose similarity constraints on the three representations; transformed query representations by the estimated pixel-wise correspondence, predicted query representations from the action aware transform module, and target representations of future state. When applying the paired similarity constraint, the prediction and projection heads of global similarity constraint are shared with the local constraint head, inducing locality-inherent volume to guide the global prediction. Finally, the well-devised paired similarity representation is then used as input to the RL policy learner.

We evaluate the proposed method with two challenging benchmarks including Atari 2600 Games [31, 51] and DMControl Suite [48], which are the common benchmarks adopted to evaluate the performance of recent sample-efficient deep RL algorithms. The proposed method competes favorably compared to the state-of-the-arts in 13 out of 26 environments on Atari 2600 Games and in 4 out of 6 tasks on DMControl Suite, in terms of cumulative rewards per episode.

We highlight our contributions as follows.

- While prior approaches place emphasis only on encoding global representations, our method takes advantage of spatial deformation to learn local fine-grained structures together, providing sufficient supervision for training the encoder of deep RL.
- We propose to impose the paired similarity constraints for visual deep RL by guiding the global prediction heads with locality-inherent volume.
- We introduce the action aware transform module to self-supervised framework to learn temporally-consistent instance discriminability by using action as a medium.

2. Related Work

Self-supervised Representation Learning: The self-supervised representation learning aims to learn general features from large-scale unlabeled images or videos without expensive data annotations. The contrastive methods have achieved state-of-the-art performance in the self-supervised representation learning [2,4,6,7,15,24,25,27,49,50,54]. The contrastive learning aims to bring positive samples closer while separating negative samples from each other [20]. Wu *et al.* [54] formulate the contrastive learning as a non-parametric classification problem at the instance level, and propose to learn visual features with the memory bank and noise contrastive estimation (NCE) [16, 41]. The method in [50] proposes a probabilistic contrastive loss, called InfoNCE, for inducing representations by leveraging positive and negative samples. The InfoNCE loss has widely been adopted in [6, 24, 25, 49]. Chen *et al.* [6] present a simple framework for contrastive self-supervised learning without specialized architecture [2, 25] or memory bank [54], but it requires a large batch size for using enough negative samples when computing the InfoNCE loss [50]. He *et al.* [24] propose to build a dynamic dictionary with a queue to avoid the use of large batches when collecting negative samples, and also uses the moving averaged (momentum) encoder for target data (positive and negative samples of query data). Grill *et al.* [15] use the momentum encoder to produce representations of the targets as a means of stabilizing the bootstrap step. This enables for learning the representations with only positive samples, which are generated by data augmentation, for a given query without the need to carefully set up negative samples. The method in [7] further extends this idea by using only stop-gradient operation without using the momentum update. Hjelm *et al.* [27] propose Deep InfoMax (DIM) that learns representations by maximizing mutual information between the input and learned features from deep networks. This was extended in [2] by maximizing mutual information between features extracted from multiple images of a shared context, *e.g.*, augmented images. While these approaches focuses on learning global representations of a single image, our method proposes to learn paired similarity representations for effectively encoding the spatial structures in the consecutive images.

Self-supervised Representation Learning in Deep RL: Representation learning is crucial for RL algorithms to learn policies with high-dimensional visual observations. Contrastive learning has been used to extract desired latent representations of visual observations used in the RL algorithms. For training robot agents, Sermanet *et al.* [44] present the time-contrastive networks (TCN) that train viewpoint-invariant representations using a metric learning such that multiple viewpoints of the same scene are encouraged to be close, while negative images taken from a different timestep are separated. This work was extended in [10] by embedding multiple frames at each timestep for learning task-agnostic representations such as position and velocity attributes in continuous control tasks. In [40], a new objective based on DIM [27] was presented for adapting to RL algorithms. In [1], the representations for RL algorithms are learned by maximizing mutual

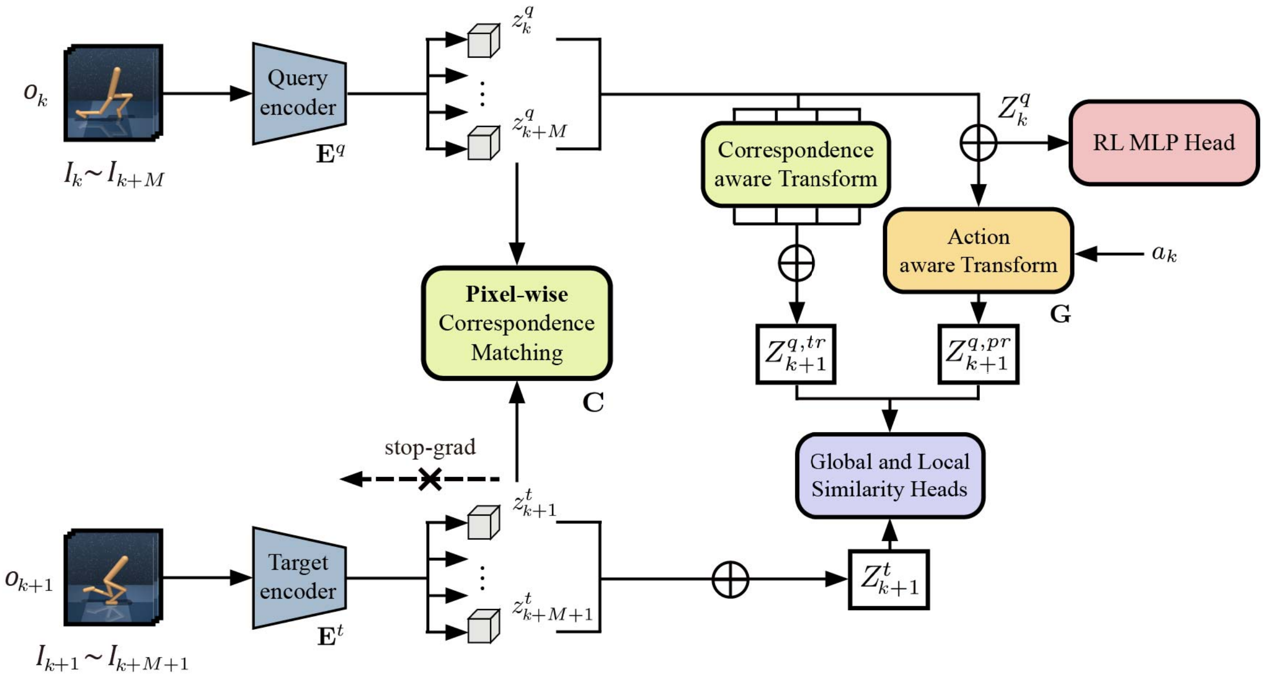

Figure 1. Overall framework of the PSRL method: Multiple representations generated by the query and target encoders are used to infer a set of pixel-wise correspondence maps. The transformed representation $Z_{k+1}^{q,tr}$ is produced using an inverse warping with the set of pixel-wise correspondence maps. The action aware transform module $\mathbf{G}$ with an action a_k predicts the future representation $Z_{k+1}^{q,pr}$. The proposed method imposes paired similarity constraints on the three latent volumes, $Z_{k+1}^{q,tr}$, $Z_{k+1}^{q,pr}$ and Z_{k+1}^{t}, guiding global prediction with local spatial structure. The target encoder and projection heads are updated using the stop-gradient operation. The encoder representation Z_k^q is used as an input in the RL algorithm. In our work, Rainbow DQN [51] ($M = 3$) and SAC [17] ($M = 2$) are used as RL algorithms.

information [27] across spatially and temporally distinct features of an encoder of visual observations. [43] leverage the self-supervised learning [15] for imposing the similarity constraint between self-predictive and target representations. Srinivas *et al.* [38] introduce Contrastive Unsupervised representations for Reinforcement Learning (CURL) that learns the representations from visual inputs using the InfoNCE loss [50]. Stooke *et al.* [46] present Augmented Temporal Contrast (ATC) using image augmentations and InfoNCE loss [50] for representation learning, and decouples it from policy learning. From a different perspective, [23] propose to adapt the policy network through self-supervised representation learning in unseen environments where it is difficult to predict changed rewards. Our method imposes the similarity constraint on the fine-grained dynamics information as well as the global semantic representations in an self-supervised manner, thus providing plenty of supervision for training the encoder of deep RL.

Visual Correspondence Learning: Visual correspondence estimation [19, 34, 42, 55] is a long-standing research in the computer vision community. It aims to establish a pair of corresponding pixels between two (or more) views taken under different locations (stereo matching) or timestep (optical flow). Recent methods for stereo matching [5, 57, 58] and optical flow estimation [9, 28, 47] have been advanced largely thanks to the expressive power of deep networks. Though both approaches share a similar objective of finding corresponding pixels across views, the optical flow is known to be effective for encoding temporal motion trajectories, while the stereo matching is tailored to predicting 3D depth map in the scene. The commonly used architecture for two-frame correspondence estimation involves the feature map extraction of two frames, correlation volume computation, a series of convolutions for refinement, and regression. Some unsupervised learning approaches have attempted to infer correspondence maps with an image reconstruction loss for imposing the constraint that corresponding pixels should have similar intensities. Note that the image reconstruction loss has also been used for self-supervised monocular depth estimation [13, 14] and stereo matching [53]. In our work, we present the self-supervised correspondence estimation network that learns fine-grained dynamics information from the consecutive frames used in the RL algorithms.

3. Method

We consider the Markov Decision Process (MDP) setting where an agent interacts with environments in a sequence of observations, actions, and rewards. We denote o_k, a_k, and r_k as the observation, the action of the agent, and the reward received at timestep k. Since our method is a general framework that leverages the representation learning for training the RL agent, it can be combined with any RL algorithm. Following the state-of-the-arts RL approaches [38, 43, 46] using the self-supervised learning, we adopt the Soft Actor Critic (SAC) method [17] for continuous control task in DeepMind Control Suite benchmark, and Rainbow DQN [51] for discrete control task in Atari Games. The proposed self-supervised paired similarity representation learning (PSRL) is used as an auxiliary task for training RL agents.

3.1. Self-supervised Correspondence Estimation

We start with how to generate the locality-inherent representations for capturing spatial deformations from the consecutively stacked frames in a self-supervised manner. An instance used by the model-free off-policy RL algorithms [17, 51] is a stack of images, not a single image. Given an input raw observation $o_k = \{I_k, ..., I_{k+M}\}$ where I_k is an image at timestep k, the latent encoder features $e_k = \{z_k, ..., z_{k+M}\}$ are first generated by applying an encoder individually to each of the input observations o_k. Note that $z \in \mathbb{R}^{h\times w\times d}$ is a 3-D volume with a spatial resolution $h \times w$ and a feature dimension d. We apply query encoder and target encoder to o_k and o_{k+1}, respectively, and denote the output of the query encoder $\mathbf{E}^q$ as z^q, and the output of the target encoder $\mathbf{E}^t$ as z^t. While the existing methods [1, 10, 38, 40, 43, 44, 46] feeds the stacked frames to the encoder at once, which can be viewed as an early fusion [32], our method generates the set of the latent representations individually with the encoder. Later, they are fused using 1×1 convolutional layer in a manner similar to a late fusion [45].

The set of representations is used to predict the spatial deformations, *i.e.*, correspondence maps between two consecutive frames. We compute a correlation volume $\mathbf{V}_{a,b} \in \mathbb{R}^{h\times w\times r^2}$ using a dot product between two latent representations z_a and z_b [9] as follows:

$$\mathbf{V}_{a,b}(u, v, \delta) = < z_a(u+\delta), z_b(v+\delta) >, \tag{1}$$

where u and v represent 2D feature position in z_a and z_b, $\delta \in [-\bar{r}, \bar{r}]$, and $\bar{r}$ indicates the kernal size for computing correlation, $r = 2\bar{r}+1$. Computing the patch similarity in (1) for all combinations of u and v (totally, $h^2 \cdot w^2$ times) causes a huge amount of computation. Thus, the maximum displacement for computing the patch similarity is limited for $v \in \mathcal{N}(u)$ where $\mathcal{N}(u)$ represents neighboring pixels of u within pre-defined search range.

The correlation volume is fed into a series of convolutions followed by the refinement layers, producing a correspondence map $c_{a\rightarrow b} \in \mathbb{R}^{h\times w\times 2}$ from I_a to I_b. As PSRL is a fully self-supervised framework, the correspondence estimation module $\mathbf{C}$ is trained by self-supervised loss L_r as follows:

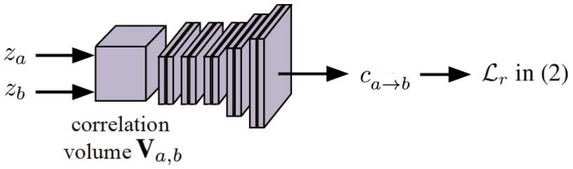

Figure 2. Correspondence matching block in Figure 1: Self-supervised correspondence estimation module $\mathbf{C}$ including the correlation volume, convolutions, and refinement layers.

$$\mathcal{L}_r(c_{a\rightarrow b}) = \sum_p |I_a(p) - I_b(p + c_{a\rightarrow b})| + \mathcal{L}_{reg}, \tag{2}$$

where $I(p)$ indicates an intensity at the pixel corresponding to 2D feature position p. For computing the loss $\mathcal{L}_r$, we resize I_a and I_b to the size of the latent representations, $h\times w$. We additionally use the Charbonnier regularization loss $\mathcal{L}_{reg}$ [3] for producing spatially smooth correspondence maps. In Figure 1, we denote 'correspondence matching' block as the self-supervised correspondence estimation module $\mathbf{C}$ including the correlation volume computation, the series of convolutions, and the refinement layers as in Figure 2.

3.2. Paired Similarity Representation Learning

Figure 1 illustrates the overall architecture of the proposed PSRL approach. Following the prior work on the self-supervised learning [7, 15, 24], we use the query encoder $\mathbf{E}^q$ with the parameters θ^q and the target encoder $\mathbf{E}^t$ with the parameters θ^t for encoding the query observation o_k and the target observation o_{k+1}, respectively. While the parameters θ^q of the query encoder are updated through back-propagation, the parameters θ^t of the target encoder are updated with the query encoder parameters θ^q using a stop-gradient operation [7] as $\theta^t \leftarrow \theta^q$.

Pixel-wise Correspondence Learning and Correspondence Aware Transform (CAT): By minimizing (2), we first compute a set of $M+1$ *external* correspondence maps $\{c^{ext}_{k+i+1\rightarrow k+i} | i = 0, ..., M\}$ with the self-supervised correspondence estimation module $\mathbf{C}$ such that

$$c^{ext}_{k+i+1\rightarrow k+i} = \mathbf{C}(z^t_{k+i+1}, z^q_{k+i}) \qquad \text{for } \ i = 0, ..., M. \tag{3}$$

Note that the external correspondence map is predicted from the target feature z^t_{k+i+1} to the query feature z^q_{k+i}. Then, we transform the query features $e^q_k = \{z^q_k, ..., z^q_{k+M}\}$ into the future state via the inverse warping [30] using $M+1$ external correspondence maps. The transformed query features $\{z^{q,tr}_{k+1}, ..., z^{q,tr}_{k+M+1}\}$ are then fused using 1×1 convolution, producing the transformed query representation $Z^{q,tr}_{k+1}$ at the timestep $k+1$.

As an additional exploitation of predicted volumes, we can also predict *internal* correspondence maps within the

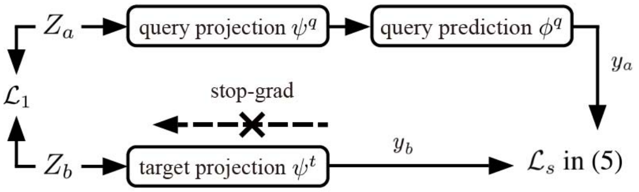
Z_a
query projection ψ^q
query prediction ϕ^q
y_a
$\mathcal{L}_1$
stop-grad
Z_b
target projection ψ^t
y_b
$\mathcal{L}_s$ in (5)

Figure 3. Global and local similarity heads in Figure 1: Similarity module consisting of the query projection and prediction heads and the target projection heads. When applying the paired similarity constraint, the heads of global similarity constraint are shared with the local constraint head, inducing locality-inherent volume to guide the global prediction.

query features $e_k^q = \{z_k^q, ..., z_{k+M}^q\}$ as $c_{a \rightarrow b}^{int} = \mathbf{C}(z_a^q, z_b^q)$. Various combinations of a and b are possible for computing the internal correspondence maps, and we choose to compute a single correspondence map $c_{k \rightarrow k+M}^{int}$. We found that this is an appropriate choice in terms of computational efficiency and accuracy as the external correspondence maps are already used to impose the structural similarity constraint between multiple frames, and is effective in dealing with the case where the external spatial difference between two consecutive frames is relatively small. More details are presented in the supplementary material. The loss function $\mathcal{L}_c$ for computing the internal and external correspondence maps is given as

$$\mathcal{L}_c = \mathcal{L}_r(c_{k \rightarrow k+M}^{int}) + \sum_{i=0}^{M} \mathcal{L}_r(c_{k+i+1 \rightarrow k+i}^{ext}). \tag{4}$$

To measure the similarity between the transformed query representation $Z_{k+1}^{q,tr}$ and the target representation Z_{k+1}^t which is the fusion of target encoder features $e_{k+1}^t = \{z_{k+1}^t, ..., z_{k+M+1}^t\}$, we use two projection heads and one predictor. We project the two representations $Z_{k+1}^{q,tr}$ and Z_{k+1}^t into a smaller latent space by passing them into the query projection head ψ^q with parameters ξ^q and the target projection head ψ^t with parameters ξ^t, and also apply an additional query prediction head ϕ^q to the query projection. The target projection head parameters ξ^t are updated with the stop-gradient operation as in the target encoder, *i.e.*, $\xi^t \leftarrow \xi^q$. The prediction loss $\mathcal{L}_s$ is computed using the cosine similarity between the transformed query representation $y_{k+1}^{q,tr} = \phi^q(\psi^q(Z_{k+1}^{q,tr}))$ and the observed target representation $y_{k+1}^t = \psi^t(Z_{k+1}^t)$, such that

$$\mathcal{L}_s(y_1, y_2) = -\frac{< y_1, y_2 >}{\|y_1\|_2 \|y_2\|_2}. \tag{5}$$

In Figure 3, we depict the module consisting of the query projection and prediction heads and the target projection heads.

Action Aware Transform (AAT): We further extend our method by leveraging an action aware transform module conditioned on an action. We generate the query representation Z_k^q by applying 1×1 convolution to the query features $\{z_k^q, ..., z_{k+M}^q\}$ and then feed it into the convolutional prediction model $\mathbf{G}$. Then, we use a single next prediction $Z_{k+1}^{q,pr} = \mathbf{G}(Z_k^q, a_k)$ from the query representation Z_k^q. The predicted global query representation $Z_{k+1}^{q,pr}$ is fed into the query projection head ψ^q and the query prediction head ϕ^q such that $y_{k+1}^{q,pr} = \phi^q(\psi^q(Z_{k+1}^{q,pr}))$. Note that, $Z_{k+1}^{q,pr}$ is a 3-dimensional representation and it becomes 1-dimensional vector, $y_{k+1}^{q,pr}$, after passing the heads. The prediction loss is also computed using the cosine similarity loss $\mathcal{L}_s(y_{k+1}^{q,pr}, y_{k+1}^t)$.

We measure the paired similarity loss L_{sim} between the three representations $y_{k+1}^{q,tr}$, $y_{k+1}^{q,pr}$, and y_{k+1}^t as

$$\begin{aligned}\mathcal{L}_{sim} = \mathcal{L}_s(y_{k+1}^{q,tr}, y_{k+1}^t) + \mathcal{L}_s(y_{k+1}^{q,pr}, y_{k+1}^t) \\ + \mathcal{L}_1(Z_{k+1}^{q,tr}, Z_{k+1}^t) + \mathcal{L}_1(Z_{k+1}^{q,pr}, Z_{k+1}^t).\end{aligned} \tag{6}$$

We also include pixel-level $\mathcal{L}_1$ loss on the original spatial latent space to guide the semantic loss with additional pixel-level similarity. Note that when applying $\mathcal{L}_{sim}$, the projection and the prediction heads of global similarity constraint are shared with the local constraint head, inducing the locality-inherent volume generated from the correspondence to guide the global prediction process. Finally, the query representation Z_k^q is fed into the deep RL algorithm.

Final Loss: The final loss function is summarized as

$$\mathcal{L}_{total} = \mathcal{L}_c + \alpha \mathcal{L}_{sim} + \mathcal{L}_{RL}(Z_k^q), \tag{7}$$

where $\mathcal{L}_{RL}(Z_k^q)$ indicates the loss of the RL algorithm which uses Z_k^q as an input. α is a hyper-parameter that balances the loss function. We summarize the overall method in Algorithm 1.

3.3. Implementation Details

Self-supervised Correspondence Estimation Module: The input image I_i is of 84×84 for Atari Games and DeepMind Control (DMControl) Suites. The query and target encoders generates $z_i^q, z_{i+1}^t \in \mathbb{R}^{7 \times 7 \times 64}$ $(i = k, ..., k+3)$ for Atari Games and $z_i^q, z_{i+1}^t \in \mathbb{R}^{32 \times 32 \times 32}$ $(i = k, ..., k+2)$ for DMControl Suites, respectively. The search window for computing the correlation volume $\mathbf{V}$ is 6×6 for Atari games and DMControl Suites. The correlation volume goes through 3×3 convolution layers 3 times. The decoder is then applied to provide a dense correspondence map. The decoder includes three un-convolutional layers, consisting of un-pooling and convolution, and the coarser correspondence maps and encoder feature maps are concatenated into each

Algorithm 1: Self-Supervised Paired Similarity Representation Learning (PSRL)

```
$\mathbf{E}^q$, $\mathbf{E}^t$: Query encoder, Target encoder
$\psi^q$, $\psi^t$: Query projection head, Target projection head
$\phi^q$: Query prediction head
Initialize replay buffer and network parameters.
while Training do
    (1) PSRL
    Generate $z^q_{k+i}, z^t_{k+i+1}$ with $\mathbf{E}^q$, $\mathbf{E}^t$ for $i = 0, ..., M$.
    Generate query representation $Z^q_k$ by fusing a set of query
      features $z^q_{k+i}$ for $i = 0, ..., M$.
    Generate target representation $Z^t_{k+1}$ by fusing a set of target
      features $z^t_{k+i+1}$ for $i = 0, ..., M$.
    (1-1) Correspondence Learning
    Learn external and internal correspondences with (4).
    (1-2) Correspondence Aware Transform
    Generate transformed query representation $Z^{q,tr}_{k+1}$ with
      external correspondence $c^{ext}_{k+i+1 \rightarrow k+i}$ for $i = 0, ..., M$.
    (1-3) Action Aware Transform
    Generate predicted query representation $Z^{q,pr}_{k+1}$ from $Z^q_k$
      using action aware transform model $\mathbf{G}$.

    (2) Training
    $Z^q_k$ goes into RL MLP head.
    Compute global and local similarities of (6) as in Figure 3.
    Optimize the networks by minimizing (7).
    Update parameters of $\mathbf{E}^t$ and $\psi^t$ with $\mathbf{E}^q$ and $\psi^q$.
end
```

un-convolutional layer.

Action aware transform Model: The action aware transform model includes two convolutional layers interweaved with ReLU and batch normalization [29], with the current representations Z^q_k and the action a_k of one-hot vector taken to each location being fed to the first convolutional layer.

Other Details: The query and target projection heads, ψ^q and ψ^t, are implemented as the multi-layer perceptron (MLP). For the query prediction head ϕ^q, we reuse the first linear layer of the RL head. We used $\alpha = 5$ in (7) to balance the weight of the losses. More details are presented in the supplementary material.

4. Experimental Results

4.1. Evaluation on Atari Games

To compare the performance of the proposed method with state-of-the-arts, we chose Atari 2600 Games introduced in [31, 51] where only 100K environment steps, corresponding to two hours of gameplay experiences, are available for training data. This sample-efficient setup, which uses much less environment steps than the standard setup of 50,000K environment steps, has been adopted for evaluating the performance of recent sample-efficient deep RL algorithms [31, 33, 38, 43, 51]. We compared our results with various RL algorithms including SimPLe [31] which learns to infer its own latent representations for Atari, Data-Efficient Rainbow (DER) [51] which modifies the Rainbow hyperparameters for improving the sample efficiency, OTRainbow [33] which is an over-trained version of the Rainbow for the sample efficiency, CURL [38] which proposes the use of image augmentation with the contrastive loss [50] for self-supervised representation learning, DrQ [36] which uses the modest image augmentation to improve the sample efficiency, and SPR [43] which trains an agent to predict its own latent state representations into the future. Following the experimental setup on the above-mentioned approaches, we evaluated on 26 environments of Atari 2600 games by measuring the average return after 100K interaction steps. We trained our method with 10 random seeds, similar to other methods.

As shown in Table 1, the proposed method (PSRL) achieved the best performance on **13** out of **26** environments. CURL [38] recorded the highest mean in 7 games out of 26, and SPR [43] recorded the highest mean in 11 games out of 26. PSRL has the highest mean in 13 games out of 26. It can be interpreted that the performance increase of the PSRL is not small by considering the quantitative aspects of these games. Also, among the 13 games in which PSRL has an edge, in particular, in 8 games (Alien, Assault, Gopher, Jamesbond, Krull, Kung Fu Master, Ms Pacman, and Seaquest), PSRL records a remarkably higher performance compared to other methods. Even the performance of certain games is high enough to match that of humans. This is because the proposed method of capturing the local-global spatial structure is able to derive an effective representation from the images of the specific Atari Games with various movements.

However, PSRL may not be effective for some games. In particular, PSRL did not perform well in the task 'Pong' in Atari Games [31]. The biggest reason for this is that there are too few discriminative spatial structures available in the game images. Therefore, we can be sure that our representation learning method, which effectively captures the spatial structure, will work particularly well for data with much more complex structural features. In other words, while most of the simple methods suffer from training with data with complex structural features, PSRL can be a good substitute for addressing this.

4.2. Evaluation on DMControl Suite

Various approaches including ours have been benchmarked on the DMControl Suite where the agent operates from pixels to evaluate challenging visual continuous control tasks [48]. We compared our results with State-SAC which supposes that the agent has access to low-level state based features, Pixel-SAC [18] which directly operates from pixels, SAC+AE [56] which uses a joint learning of SAC with

Table 1. Quantitative evaluation with state-of-the-arts on the 26 Atari games [31] after 100K time steps using 10 random seeds: Numbers in bold represent 1^{st} ranking. PSRL achieves the best performance on **13** out of **26** environments. We compared results with SimPLe [31], Data-Efficient Rainbow (DER) [51], OverTrained Rainbow (OTRainbow) [33], CURL [38], DrQ [36], and SPR [43].

Game	Human	Random	Rainbow	SimPLe	DER	OTRainbow	CURL	DrQ	SPR	PSRL
Alien	7127.7	227.8	318.7	616.9	739.9	824.7	558.2	771.2	801.5	**1030.1**
Amidar	1719.5	5.8	32.5	88.0	**188.6**	82.8	142.1	102.8	176.3	114.3
Assault	742.0	222.4	231.0	527.2	431.2	351.9	600.6	452.4	571.0	**708.3**
Asterix	8503.3	210.0	243.6	**1128.3**	470.8	628.5	734.5	603.5	977.8	959.3
Bank Heist	753.1	14.2	15.55	34.2	51.0	182.1	131.6	168.9	**380.9**	95.8
BattleZone	37187.5	2360.0	2360.0	5184.4	10124.6	4060.6	14870.0	12954.0	16651.0	**16688.0**
Boxing	12.1	0.1	-24.8	9.1	0.2	2.5	1.2	6.0	35.8	**35.9**
Breakout	30.5	1.7	1.2	16.4	1.9	9.8	4.9	16.1	17.1	**17.5**
ChopperCommand	7387.8	811.0	120.0	1246.9	861.8	1033.3	1058.5	780.3	974.8	**1251.2**
Crazy Climber	35829.4	10780.5	2254.5	**62583.6**	16185.3	21327.8	12146.5	20516.5	42923.6	42544.0
Demon Attack	1971.0	152.1	163.6	208.1	508.0	711.8	817.6	**1113.4**	545.2	884.0
Freeway	29.6	0.0	0.0	20.3	**27.9**	25.0	26.7	9.8	24.4	24.8
Frostbite	4334.7	65.2	60.2	254.7	866.8	231.6	1181.3	331.1	**1821.5**	776.9
Gopher	2412.5	257.6	431.2	771.0	349.5	778.0	669.3	636.3	715.2	**920.3**
Hero	30826.4	1027.0	487.0	2656.6	6857.0	6458.8	6279.3	3736.3	**7019.2**	3977.3
Jamesbond	302.8	29.0	47.4	125.3	301.6	112.3	471.0	236.0	365.4	**471.4**
Kangaroo	3035.0	52.0	0.0	323.1	779.3	605.4	872.5	940.6	**3276.4**	1580.0
Krull	2665.5	1598.0	1468.0	4539.9	2851.5	3277.9	4229.6	4018.1	3688.9	**4958.3**
Kung Fu Master	22736.3	258.5	0.0	17257.2	14346.1	5722.2	14307.8	9111.0	13192.7	**17759.5**
Ms Pacman	6951.6	307.3	67.0	1480.0	1204.1	941.9	1465.5	960.5	1313.2	**1597.3**
Pong	14.6	-20.7	-20.6	**12.8**	-19.3	1.3	-16.5	-8.5	-5.9	-8.2
Private Eye	69571.3	24.9	0.0	58.3	97.8	100.0	**218.4**	-13.6	124.0	158.0
Qbert	13455.0	163.9	123.46	1288.8	1152.9	509.3	1042.4	854.4	669.1	**1290.3**
Road Runner	7845.0	11.5	1588.46	5640.6	9600.0	2696.7	5661.0	8895.1	**14220.5**	3175.7
Seaquest	42054.7	68.4	131.69	683.3	354.1	286.9	384.5	301.2	583.1	**734.9**
Up N Down	11693.2	533.4	504.6	3350.3	2877.4	2847.6	2955.2	3180.8	**28138.5**	4263.8

Table 2. Quantitative evaluation of mean and standard deviation with state-of-the-arts on the DMControl suite [48] after 100K time steps and 500K time steps using 10 random seeds. Numbers in bold represent 1^{st} ranking, and PSRL achieves the best performance on **4** out of **6** environments for 500K time steps. We compared results with state-based SAC and pixel-based SAC [18], SAC+AE [56], Dreamer [21], PlaNet [22], CURL [38], RAD [37], and DrQ [36].

100K step scores	State SAC	Pixel SAC	SAC+AE	Dreamer	PlaNet	CURL	RAD	DrQ	PSRL
Finger, Spin	811±46	179±66	740±64	341±70	136±216	767±56	856±73	**901±104**	882±132
Cartpole, Swingup	835±22	419±40	311±11	326±27	297±39	582±146	828±27	759±92	**849±63**
Reacher, Easy	746±25	145±30	274±14	314±155	20±50	538±233	**826±219**	601±213	621±202
Cheetah, Run	616±18	197±15	267±24	235±137	138±88	299±48	**447±88**	344±67	398±71
Walker, Walk	891±82	42±12	394±22	277±12	224±48	403±24	504±191	**612±164**	595±104
Ball in Cup, Catch	746±91	312±63	391±82	246±174	0±0	769±43	840±179	913±53	**922±60**
500K step scores	State SAC	Pixel SAC	SAC+AE	Dreamer	Planet	CURL	RAD	DrQ	PSRL
Finger, Spin	923±21	179±166	884±128	796±183	561±284	926±45	947±101	938±103	**961±121**
Cartpole, Swingup	848±15	419±40	735±63	762±27	475±71	841±45	863±9	868±10	**895±39**
Reacher, Easy	923±24	145±30	627±58	793±164	210±390	929±44	**955±71**	942±71	932±41
Cheetah, Run	795±30	197±15	550±34	570±253	305±131	518±28	**728±71**	660±96	686±80
Walker, Walk	948±54	42±12	847±48	897±49	351±58	902±43	918±16	921±45	**930±75**
Ball in Cup, Catch	974±33	312±63	794±58	879±87	460±380	959±27	974±12	963±9	**988±54**

β-VAE [26], VAE [35], and regularized autoencoder [52], Dreamer [21] and PlaNet [22] which learn a latent space world model, CURL [38] which uses image augmentation with the contrastive loss [50], RAD [37] and DrQ [36] which demonstrate that data augmentation can greatly improve the performance of model-free RL algorithms and achieve state-of-the-art performance on DMControl Suite. We trained our method with 10 random seeds, and the results with 5 random seeds are provided in the supplementary material.

Table 2 demonstrates that the self-supervised paired similarity representations of PSRL achieved best performance on **4** out of **6** environments for 500K time steps including Cartpole Swingup, Reacher Easy, Walker Walk and Ball in Cup Catch. In general, the performance at 500K steps after most methods converge is widely adopted for the evaluation. When compared to the performance improvement rate of other methods, the performance increase of PSRL is significant. The performance at 100K steps is usually based on when most methods do not converge. In Table 2, the performance of PSRL recorded the highest mean in 2 tasks out of 6 tasks at 100k steps, and RAD [37] and DrQ [36] also recorded the highest mean in 2 tasks out of 6 tasks, respectively. It can be interpreted that RAD [37], DrQ [36] and PSRL are the three methods with the highest convergence speed.

4.3. Ablation Study

Impact of Losses: Table 3 measured the average performance over 10 random seeds according to the combinations of several losses on DMControl Suite [48] with 500K time

Table 3. To study the impact of several losses, we measured the average performance over 10 random seeds according to the combinations of losses on DMControl Suite [48] with 500K time steps. Refer to section 4.3 for 'C', 'C+T', 'P', and 'C+P'.

500K step scores	C	C+T	P	C+P	C+T+P (PSRL)
Finger, Spin	729±110	757±100	711±64	768±112	**961±121**
Cartpole, Swingup	819±38	876±19	793±15	868±27	**895±39**
Reacher, Easy	857±45	901±29	904±58	922±31	**932±41**
Cheetah, Run	501±73	586±40	**691±104**	690±73	686±80
Walker, Walk	770±87	879±67	822±53	876±41	**930±75**
Ball in Cup, Catch	849±42	953±26	848±107	951±20	**988±54**

Table 4. To study the impact of various data augmentation, we measured the average performance over 10 random seeds according to the data augmentation on DMControl Suite [48] with 500K time steps.

500K step scores	PSRL + no aug	PSRL + crop	PSRL + translation
Finger, Spin	932±115	915±91	**961±121**
Cartpole, Swingup	**895±39**	837±16	872±51
Reacher, Easy	**932±41**	833±87	930±83
Cheetah, Run	635±74	611±59	**686±80**
Walker, Walk	914±30	**930±75**	886±51
Ball in cup, Catch	962±14	**988±54**	946±42

steps.

- 'C' using only the correspondence estimation loss in (4)
- 'C+T' using correspondence estimation loss and similarity loss with 'T'ransformed query and target representations in (6)
- 'P' using prediction loss with 'P'redicted and target representations in (6)
- 'C+P' using the correspondence estimation loss and the prediction loss in (6).

The network trained with only 'C' produces worse performance compared to 'C+T', 'C+P' and 'C+T+P', but still produces comparable performance to state-of-the-arts, implying that even without the transform and prediction model, simply guiding the encoder to extract features for the correspondence prediction helps the RL agent to perform better. The performance of 'C+T' and 'C+P' is similar, but 'C+T' has slightly better performance on step scores with smaller standard deviations. This implies that the transformed query representation to the future state using the estimated correspondence is capable of providing as useful supervision as the predicted representation that uses an action aware transform model. The performance was further boosted, when using 'C+T+P' altogether (PSRL). To measure only the impact of each loss, data augmentation was not performed.

Impact of Data Augmentation: To study the impact of data augmentation when used with the proposed method, we measured the average performance over 10 random seeds according to the data augmentation on DMControl Suite [48]. In Table 4, we evaluated the performance of the proposed method when used with crop and translation proposed in RAD [37].

Slightly different from the result presented in RAD [37], Cartpole Swingup and Reacher Easy achieved the best performance when no augmentation was used, Finger Spin and Cheetach Run obtained the best performance for translation, and Walker Walk and Ball in cup Catch showed the best performance for crop. Since PSRL learns correspondence in an end-to-end manner with RL algorithm, it is analyzed that the results are different from those of RAD [37].

5. Discussion and Conclusion

We have presented the self-supervised paired similarity representation learning, termed PSRL, to encode global and local spatial structures in an unsupervised manner. The correspondence maps inferred by the proposed method offer plenty of supervision for learning the fine-grained latent representations, and also compute transformed predictions at future frame. PSRL achieves state-of-the-art performance on Atari benchmark with 100K steps and DMControl Suites with 100K/500K steps. We have shown the importance of learning the paired similarity representations in improving the performance and sample-efficiency of image-based RL algorithms. We hope this can facilitate future works at various aspects for RL based on self-supervised learning. Code will be available soon.

Limitations The increase in the computational cost during training is unavoidable because PSRL additionally leverage the correspondence estimation module, but we found that the additional computational cost for training is not so significant. For training on DMControl Suite [48] up to 500K on the same GPU environment, the proposed method takes about 16 hours, whereas the state-of-the-art methods CURL [38] and SPR [43] take about 10 hours and 13 hours, respectively. Note that the original SPR paper did not provide the code implemented for DM Control Suite, so we conducted the experiments by modifying the original SPR code. Additionally, the correspondence estimation module are used only during training, and the inference process is implemented in the same manner as other methods. Therefore, **the inference time of our method is exactly the same as that of the state-of-the-arts methods** (CURL [38], SPR [43], DrQ [36]) as long as the same encoder for query images is used.

References

[1] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised state representation learning in atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8766–8779, 2019. 1, 2, 4

[2] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15509–15519, 2019. 2

[3] Jonathan T. Barron. A general and adaptive robust loss function. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4331–4339, 2019. 4

[4] 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 (NeurIPS)*, 2020. 2

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020. 2

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *CoRR*, abs/2011.10566, 2020. 2, 4

[8] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018. 2

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 3, 4

[10] Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. Learning actionable representations from visual observations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1577–1584, 2018. 1, 2, 4

[11] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 2

[12] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Found. Trends Mach. Learn.*, 11(3-4):219–354, 2018. 1

[13] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 3

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019. 3

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3, 4

[16] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 297–304, 2010. 2

[17] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pages 1856–1865, 2018. 3, 4

[18] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018. 6, 7

[19] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011. 1, 3

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

[21] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 7

[22] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019. 7

[23] Nicklas Hansen, Yu Sun, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *CoRR*, abs/2007.04309, 2020. 3

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1, 2, 4

[25] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. pages 4182–4192, 2020. 2

[26] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International*

Conference on Machine Learning, pages 1480–1490. PMLR, 2017. 7

[27] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019. 2, 3

[28] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017. 3

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015. 6

[30] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 4

[31] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 1, 2, 6, 7

[32] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014. 4

[33] Kacper Piotr Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency? 1, 6, 7

[34] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2314, 2013. 1, 3

[35] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 7

[36] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020. 6, 7, 8

[37] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 7, 8

[38] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 5639–5650, 2020. 1, 3, 4, 6, 7, 8

[39] Jiangbo Lu, Hongsheng Yang, Dongbo Min, and Minh N Do. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1854–1861, 2013. 1

[40] Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R. Devon Hjelm. Deep reinforcement and infomax learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 4

[41] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2265–2273, 2013. 2

[42] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002. 1, 3

[43] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 4, 6, 7, 8

[44] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141, 2018. 1, 2, 4

[45] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014. 4

[46] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *CoRR*, abs/2009.08319, 2020. 1, 3, 4

[47] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 3

[48] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2, 6, 7, 8

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision (ECCV)*, pages 776–794, 2020. 2

[50] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 1, 2, 3, 6, 7

[51] Hado P Van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 3, 4, 6, 7

[52] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 7

[53] Longguang Wang, Yulan Guo, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, and Wei An. Parallax attention for unsupervised stereo correspondence learning. *CoRR*, abs/2009.08250, 2020. 3

[54] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[55] Hongsheng Yang, Wen-Yan Lin, and Jiangbo Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3406–3413, 2014. 1, 3

[56] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019. 6, 7

[57] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The journal of machine learning research*, 17(1):2287–2318, 2016. 3

[58] Feihu Zhang, Victor Adrian Prisacariu, Ruigang Yang, and Philip H. S. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–194, 2019. 3