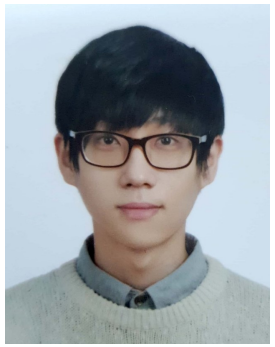
Local-guided Global: Paired Similarity Representation for Visual Reinforcement Learning

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

<WED-PM-259>

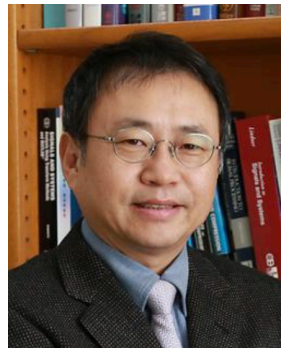
Hyesong Choi
Ewha W. University



Hunsang Lee
Hyundai Motors

Hunsang Lee
Yonsei University

Hunsang Lee
University of Michigan



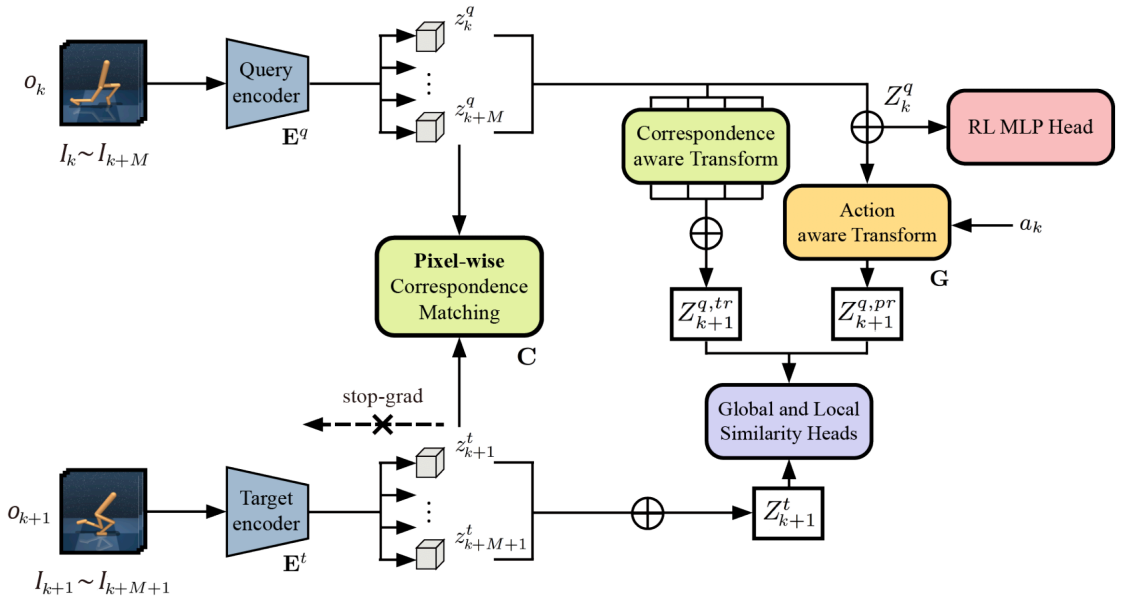
Kwanghoon Sohn
Yonsei University

Dongbo Min
Ewha W. University

Brief Summary

Self-supervised Paired Similarity Representation Learning (PSRL)

- Effectively encodes spatial structures for vision-based reinforcement learning
- Propose to impose the paired similarity constraints for visual deep RL by guiding the global prediction heads with locality-inherent volume



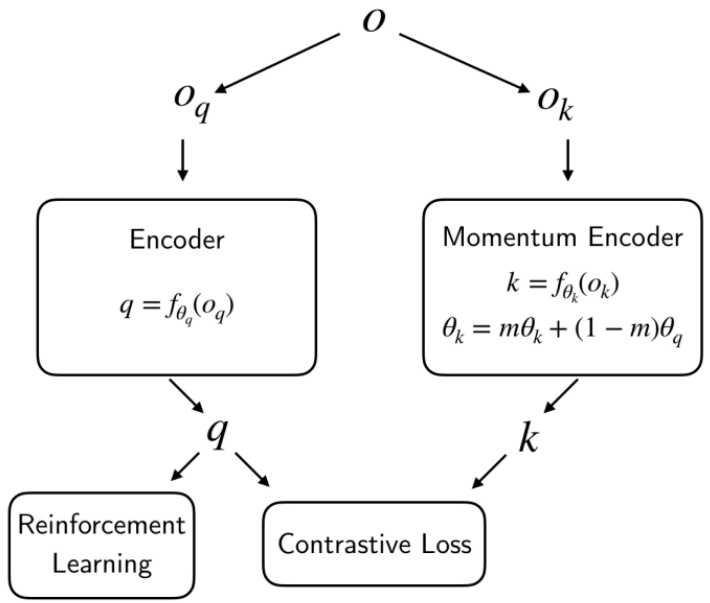
1. Introduction

1. Deep reinforcement learning (RL)

- Training agents to solve various tasks including complex control and video games
- 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

1. Introduction

2. Self-supervised learning approaches

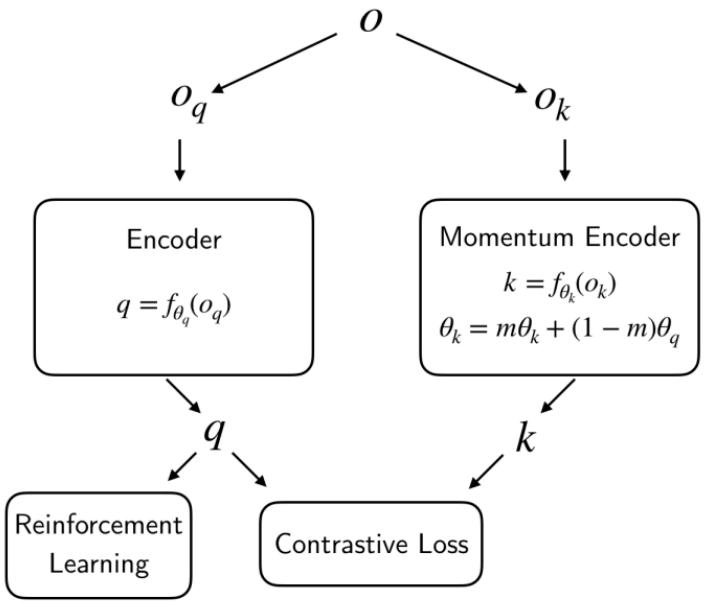


1. Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre C.t., and R. Devon Hjelm. "Unsupervised state representation learning in atari." NeurIPS, 2019.
2. Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. "Learning actionable representations from visual observations." In IEEE/RSJ IROS, 2018.
3. Michael Laskin, Aravind Srinivas, and Pieter Abbeel. "CURL: contrastive unsupervised representations for reinforcement learning." ICML, 2020.
4. Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R. Devon Hjelm. "Deep reinforcement and infomax learning." NeurIPS, 2020.
5. Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. "Data-efficient reinforcement learning with self-predictive representations." ICLR, 2021.
6. 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 ICRA, 2018.
7. Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. "Decoupling representation learning from reinforcement learning." ICML, 2020.

- A number of deep RL approaches leverage the recent advance of self-supervised learning which effectively extracts high-level features from raw pixels

1. Introduction

2. Self-supervised learning approaches



1. Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre C.t., and R. Devon Hjelm. "Unsupervised state representation learning in atari." NeurIPS, 2019.
2. Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. "Learning actionable representations from visual observations." In IEEE/RSJ IROS, 2018.
3. Michael Laskin, Aravind Srinivas, and Pieter Abbeel. "CURL: contrastive unsupervised representations for reinforcement learning." ICML, 2020.
4. Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R. Devon Hjelm. "Deep reinforcement and infomax learning." NeurIPS, 2020.
5. Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. "Data-efficient reinforcement learning with self-predictive representations." ICLR, 2021.
6. 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 ICRA, 2018.
7. Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. "Decoupling representation learning from reinforcement learning." ICML, 2020.

- **Problem of recent approaches**: Encode the ***global*** semantic representations of images, there has been no attention on the ***local*** fine-grained structures

- **Our key observation**: **Spatial deformation** can provide plenty of local samples!

1. Introduction

3. Proposed Method (PSRL)

- **Paired similarity constraints**

  - We impose the paired similarity constraints for visual deep RL by guiding the global prediction heads with locality-inherent volume.

  - We impose similarity constraints on the three representations:
    1) transformed query representations by the estimated pixel-wise correspondence
    2) predicted query representations from the action aware transform module
    3) target representations of future state

- **Correspondence aware transform**

  Correspondence aware transform is applied to generate future representations, which has been widely used for various tasks such as image registration and recognition to model the local spatial deformation.

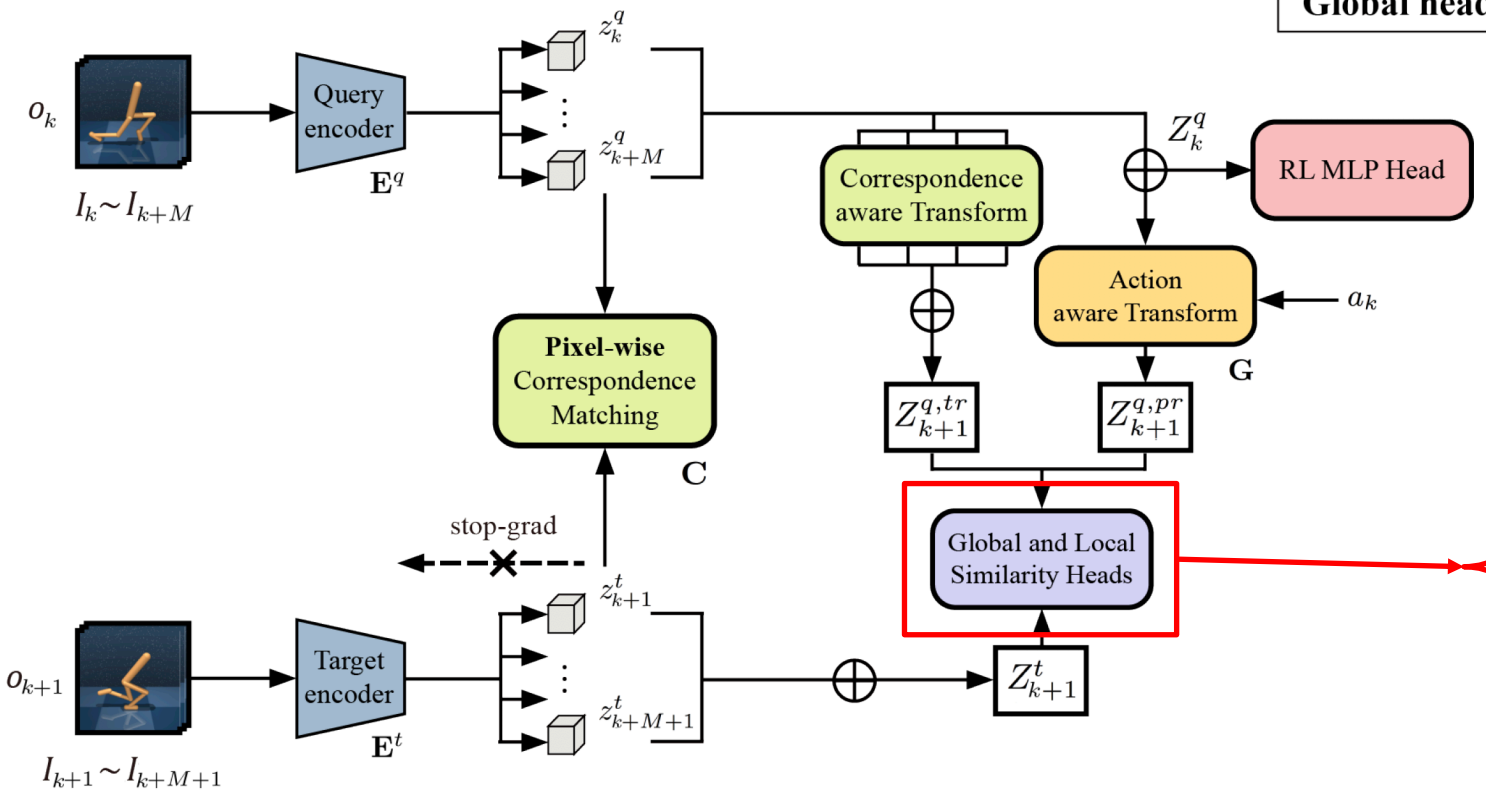
- **Action aware transform**

  We extend our framework by introducing the concept of future state prediction into the proposed action aware transform.

2. Proposed Method – Overview

- Overall framework of the PSRL method

	Correspondence Aware Transform (CAT)	Action Aware Transform (AAT)
Local head	$\mathcal{L}_1$ (3D volume)	$\mathcal{L}_1$ (3D volume)
Global head	$\mathcal{L}_s$ (1D global vector)	$\mathcal{L}_s$ (1D global vector)



1) CAT

- generates the next feature via warping using a correspondence map.

2) AAT

- predicts the next feature using the current action.

3) $\mathcal{L}_1$ loss

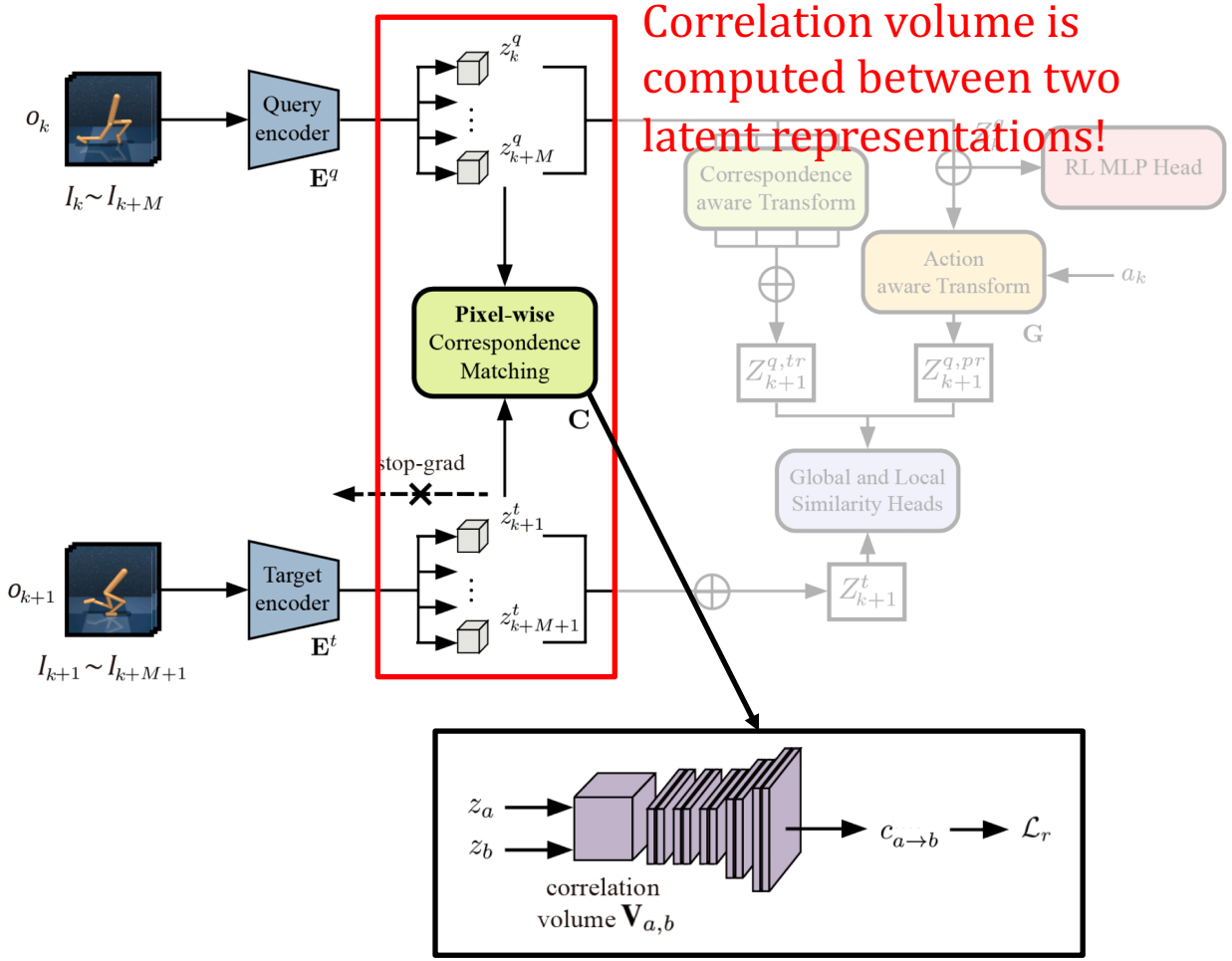
- encodes local information as it operates in pixel units.

4) $\mathcal{L}_s$ loss

- encodes global information as it operates after converting 3D volume into a global vector.

2. Proposed Method – Correspondence Estimation

Self-supervised Correspondence Estimation



1. Given an input raw observation o_k and o_{k+1}, we apply query encoder $\mathbf{E}^q$ and target encoder $\mathbf{E}^t$ to generate the latent features z^q and z^t.

2. We compute a correlation volume $\mathbf{V}_{a,b}$ using a dot product between two latent representations as follows:

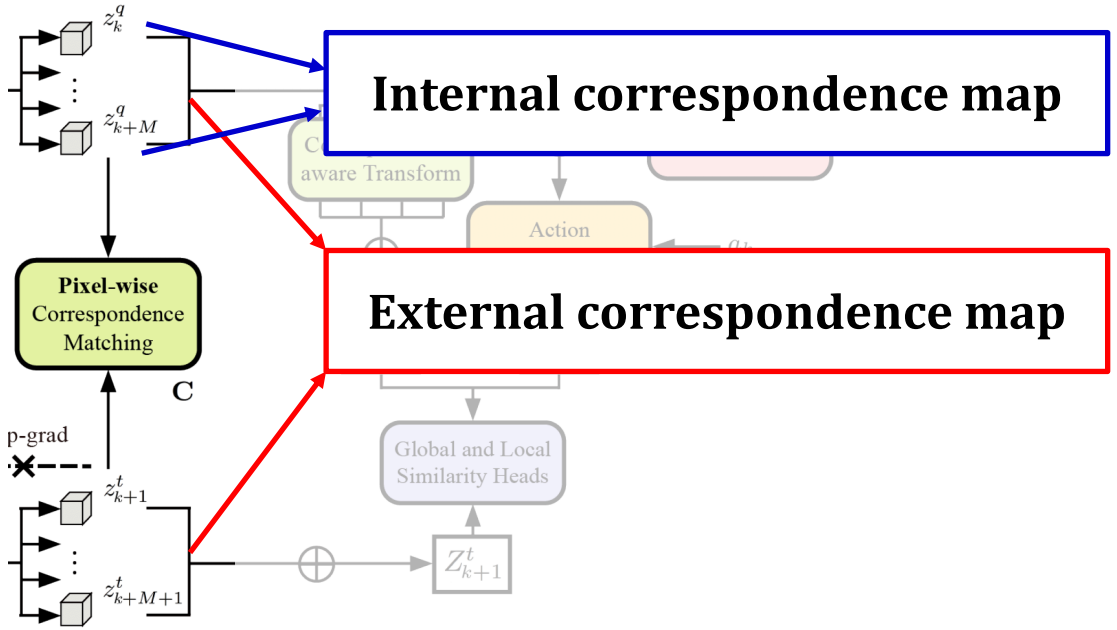
$$\mathbf{V}_{a,b}(u, v, \delta) = \langle z_a(u + \delta), z_b(v + \delta) \rangle$$

3. The correspondence estimation module **C** is trained by self-supervised loss $\mathcal{L}_r$ as:

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

2. Proposed Method – Paired Similarity Representation Learning

Pixel-wise correspondence learning and correspondence aware transform (CAT)



1. We first compute a set of $M + 1$ ***external*** correspondence maps with the self-supervised correspondence estimation module **C** such that

$$c^{ext}_{k+i+1\rightarrow k+i} = \mathbf{C}\left(z^{t}_{k+i+1}, z^{q}_{k+i}\right) \qquad \text{for } i = 0, \dots, M.$$

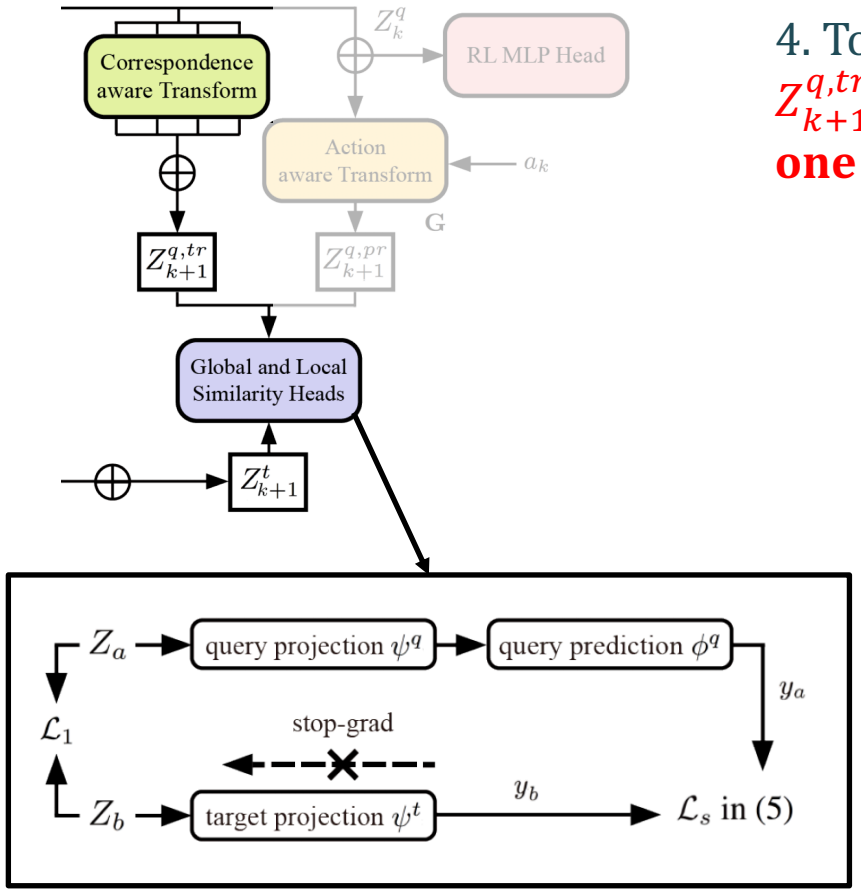
2. As an additional exploitation of predicted volumes, we also predict ***internal*** correspondence maps within the query features e^{q}_{k} as $c^{int}_{a\rightarrow b} = \mathbf{C}\left(z^{q}_{a}, z^{q}_{b}\right)$.

3. The loss function $\mathcal{L}_c$ for computing the internal and external correspondence maps is given as

$$\mathcal{L}_c = \mathcal{L}_r\left(c^{int}_{k\rightarrow k+M}\right) + \sum_{i=0}^{M} \mathcal{L}_r\left(c^{ext}_{k+i+1\rightarrow k+i}\right)$$

2. Proposed Method – Correspondence Estimation

Pixel-wise correspondence learning and correspondence aware transform (CAT)



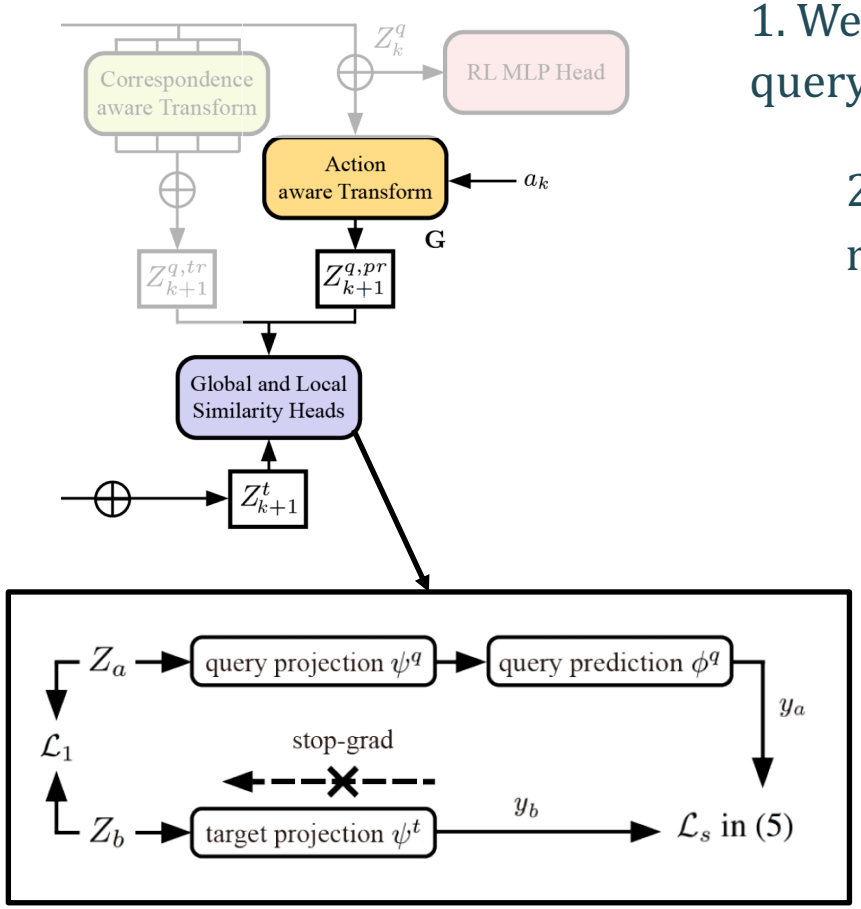
4. To measure the similarity between the transformed query representation $Z_{k+1}^{q,tr}$ and the target representation Z_{k+1}^t , we use **two projection heads and one predictor**.

5. 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

$$\mathcal{L}_s(y_1, y_2) = -\frac{\langle y_1, y_2 \rangle}{\|y_1\|_2 \|y_2\|_2}$$

2. Proposed Method – Correspondence Estimation

Action aware transform (AAT)



1. We generate the query representation Z_k^q by applying 1 X 1 convolution to the query features and then feed it into the convolutional prediction model **G**.

2. 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.

3. 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 $y_{k+1}^{q,pr} = \phi^q\left(\psi^q\left(Z_{k+1}^{q,pr}\right)\right)$, with the loss $\mathcal{L}_s(y_{k+1}^{q,pr}, y_{k+1}^t)$.

4. We measure the paired similarity loss $\mathcal{L}_{sim}$ between the three representations $y_{k+1}^{q,tr}$, $y_{k+1}^{q,pr}$, and y_{k+1}^t as

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

3. Experimental Results

1. The proposed method (PSRL) achieved the best performance on 13 out of 26 environments.
2. PSRL did not perform well in the task `Pong', because of the few discriminative spatial structures

- Evaluation on Atari Games

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

3. Experimental Results

- Evaluation on DMControl Suite

- 500K steps: after most methods converge
- 100K steps: when most methods do not converge

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**

1. The proposed method (PSRL) achieved the best performance on 4 out of 6 environments for 500K time steps.
2. RAD, DrQ and RSRL are the three methods with the highest convergence speed.