Does Self-supervised Learning Really Improve Reinforcement Learning from Pixels?

Xiang Li[1] Jinghuan Shang[1] Srijan Das[1,2] Michael S. Ryoo[1]

Department of Computer Science
[1]Stony Brook University [2]UNC Charlotte

Does Self-supervised Learning Really Improve Reinforcement Learning from Pixels?

No

No*

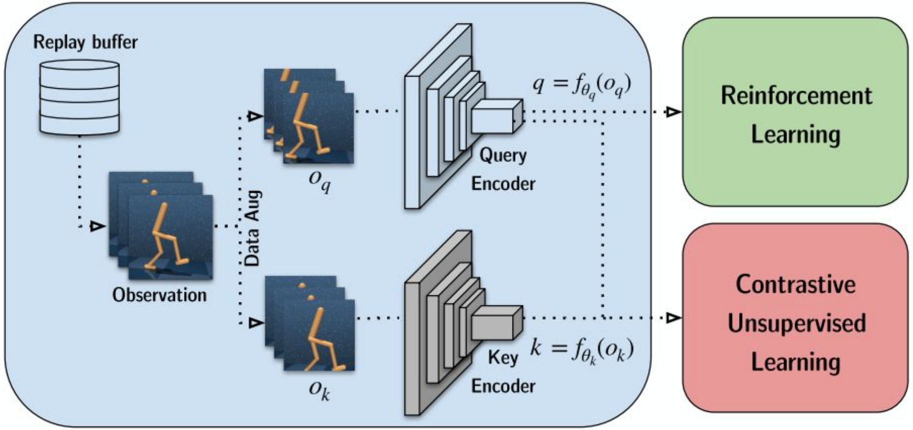
* The **existing SSL framework** for RL fails to bring meaningful improvement over the baselines when the **same amount of data and augmentation** is used

Previous works on SSL + RL

Apply **RL update** and **SSL update** in one batch **alternatingly**

SAC+AE[1]
```
for each batch:
    SAC update
    RAE[2] update
```

CURL[3]
```
for each batch:
    SAC update
    MoCo[4] update
```



[3]

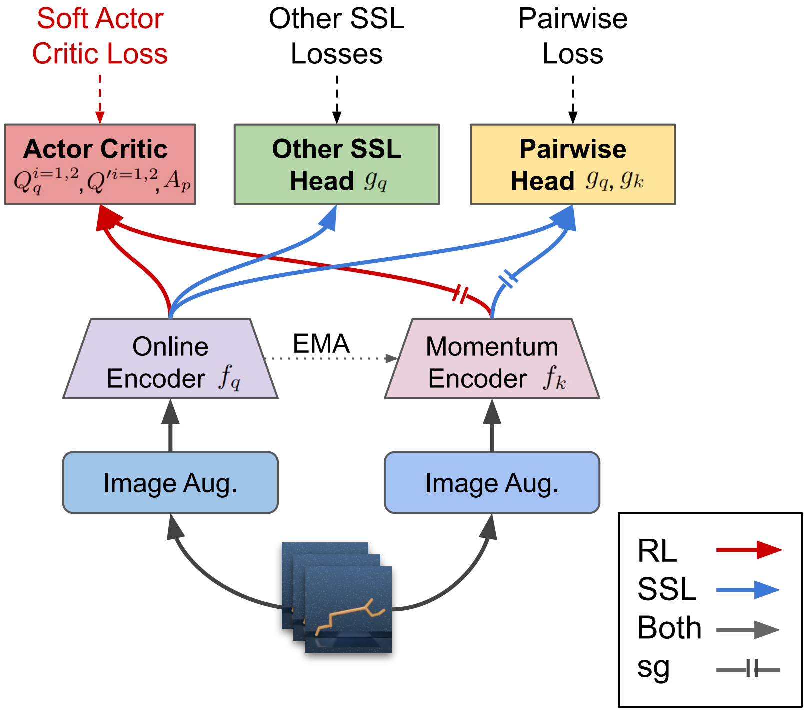
[1] Denis Yarats, et al. Improving sample efficiency in model-free reinforcement learning from images. AAAI, 2021.
[2] Ghosh, Partha, et al. From variational to deterministic autoencoders. ICLR, 2020.
[3] Michael Laskin, et al. Curl: Contrastive unsupervised representations for reinforcement learning. ICML, 2020.
[4] He, Kaiming, et al. Momentum contrast for unsupervised visual representation learning.CVPR. 2020.

Joint Learning Framework for RL + SSL

NEURAL INFORMATION PROCESSING SYSTEMS

We **extend** the existing framework to a **general** joint learning framework

```
for each batch:
    Image augmentation
    SAC update
    SSL update
```



SSL Loss options

1. **Pairwise Learning (6 methods)**

   - BYOL, DINO, SimSiam, CURL-w-actor…

2. **Transformation Awareness (2 methods)**

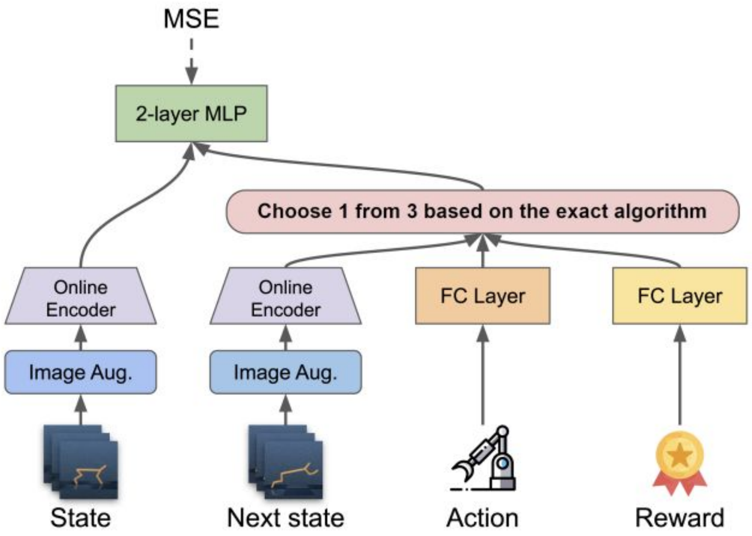
   - RotationCLS, ShuffleCLS

3. **Reconstruction (2 methods)**

   - AutoEncoder, MAE

4. **RL Context Prediction (9 methods)**

   - Predict Action, Predict Reward, Predict Future, …



General RL context prediction

Evolving Losses for RL + SSL

Combine multiple self-supervised losses

$$\mathcal{L}_{\text{Combo}} = \sum_{i=1}^{N_l} w_i \cdot \mathcal{L}_i$$

Inspired by ELo[1], we use **evolutionary search** to find the optimal :

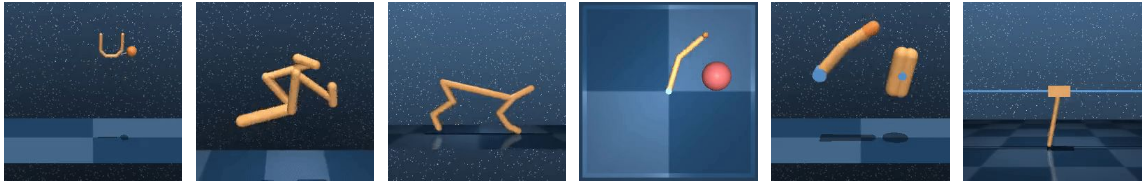
- **Loss combination**
  - CURL, DINO, Predict-FR, Extract-AR, AutoEncoder, RotationCLS, …

- **Magnitude of image augmentation**
  - With Online branch, with EMA branch

[1] AJ Piergiovanni, et al. Evolving losses for unlabeled video representation learning. CVPR, 2020

Evaluation

Continuous Action Space: 6 envs from DMControl[1] + 1 real robot env + more



bs=512, env step=100k, frame stack=3, 10 seeds each

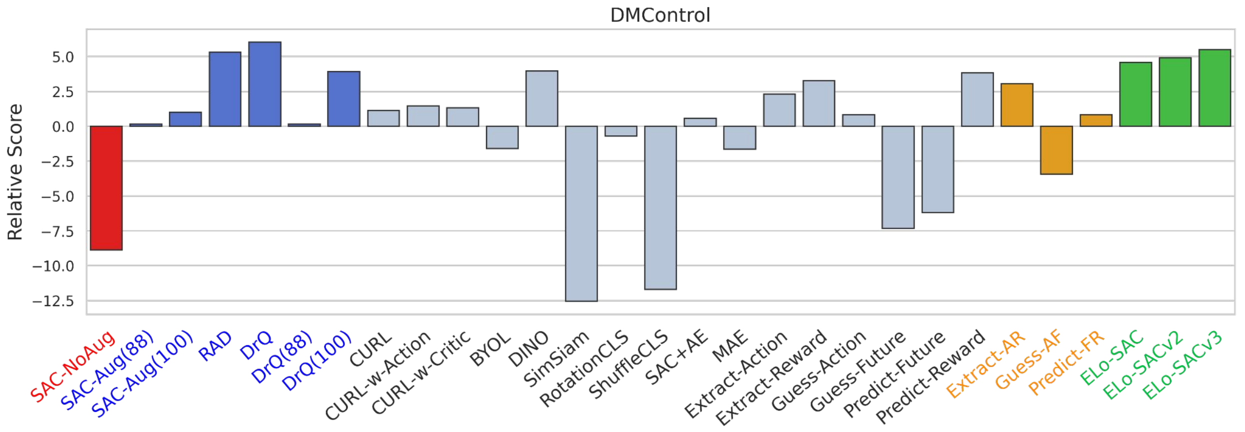
5 seeds

Discrete Action Space: 7 envs from Atari[2]

bs=32, env step=400k, frame stack=4, 20 seeds each

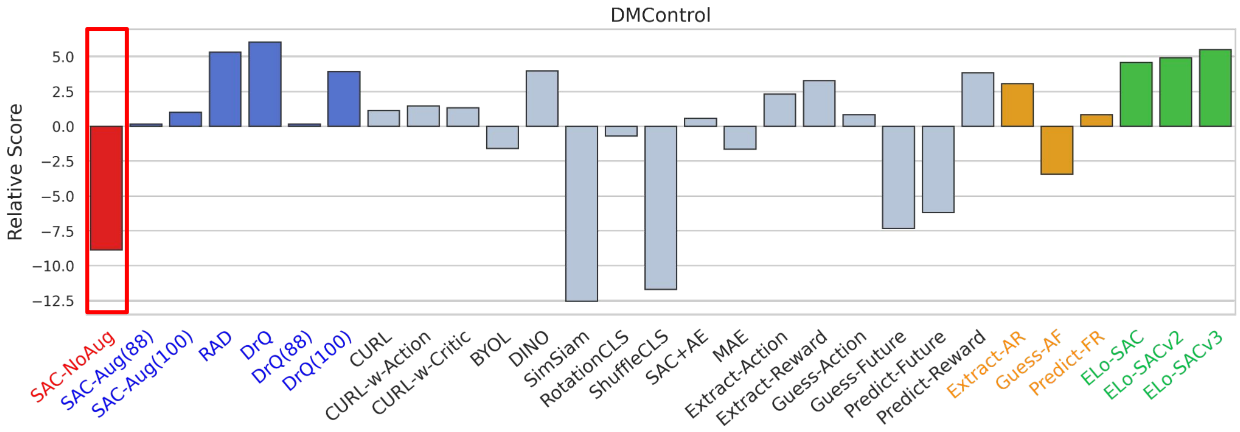
[1] Yuval Tassa, et al. Deepmind control suite. arXiv preprint arXiv:1801.00690, 2018.
[2] Marc G Bellemare, et al. The arcade learning environment: An evaluation platform for general agents. JAIR, 2013.

Results: DMControl



	Augmentation	# of SSL
Red	N	0
Blue	Y	0
Gray	Y	1
Orange	Y	2
Green	Y	6

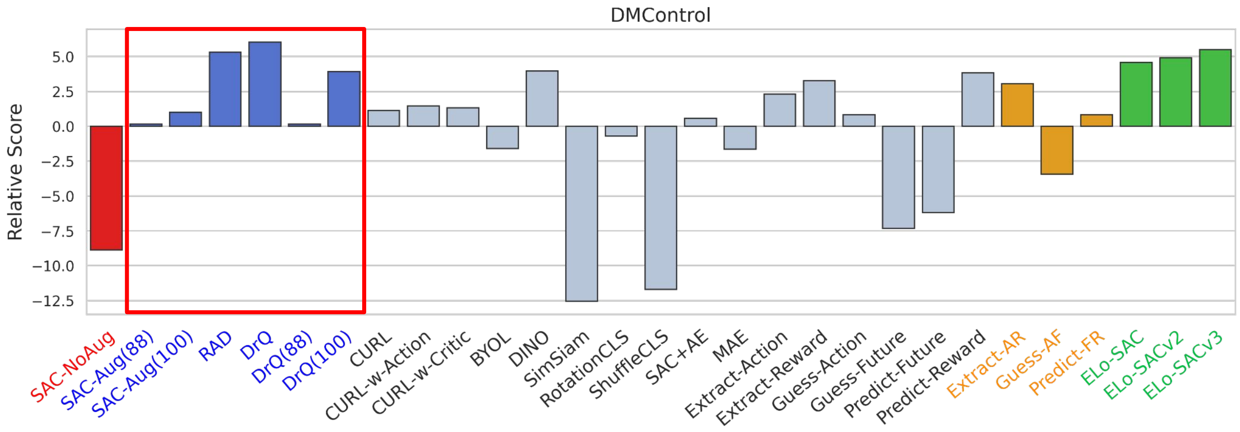
Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

SAC-NoAug: Vanilla SAC w/o augmentation or SSL

Results: DMControl

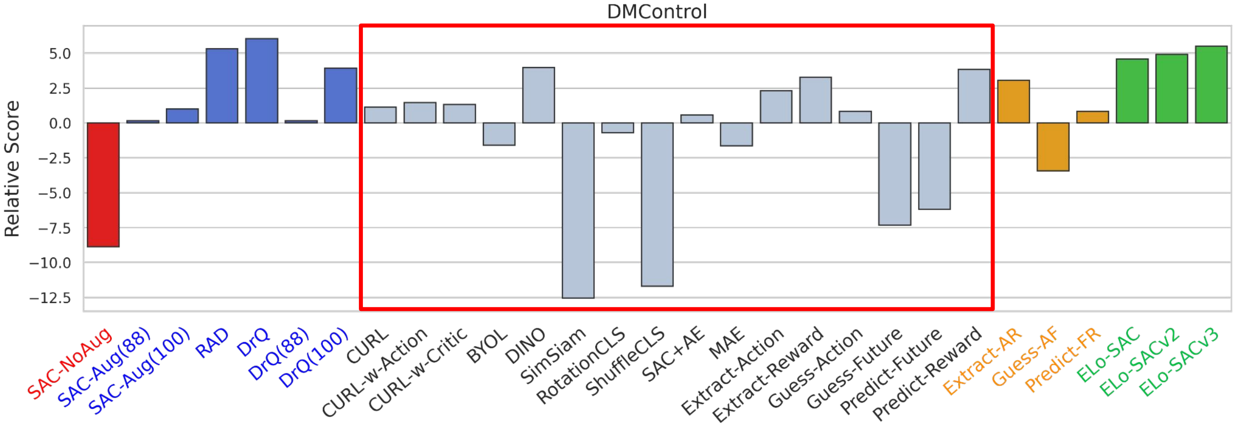


	Augmentation	# of SSL
Red	N	0
Blue	Y	0
Gray	Y	1
Orange	Y	2
Green	Y	6

SAC-NoAug: Vanilla SAC w/o augmentation or SSL

Blue: Augmentation w/o SSL

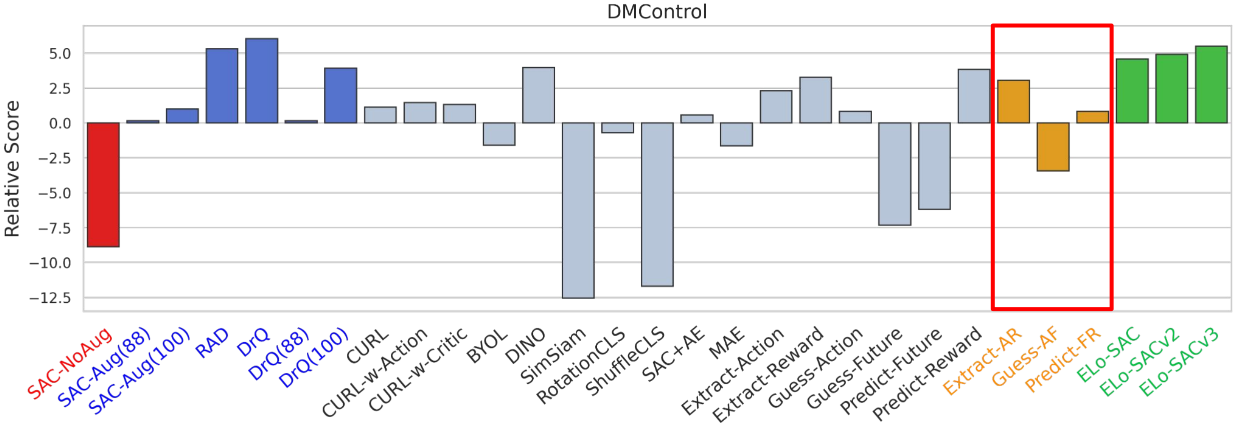
Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

SAC-NoAug: Vanilla SAC w/o augmentation or SSL
Blue: Augmentation w/o SSL
Gray: Augmentation and one SSL loss

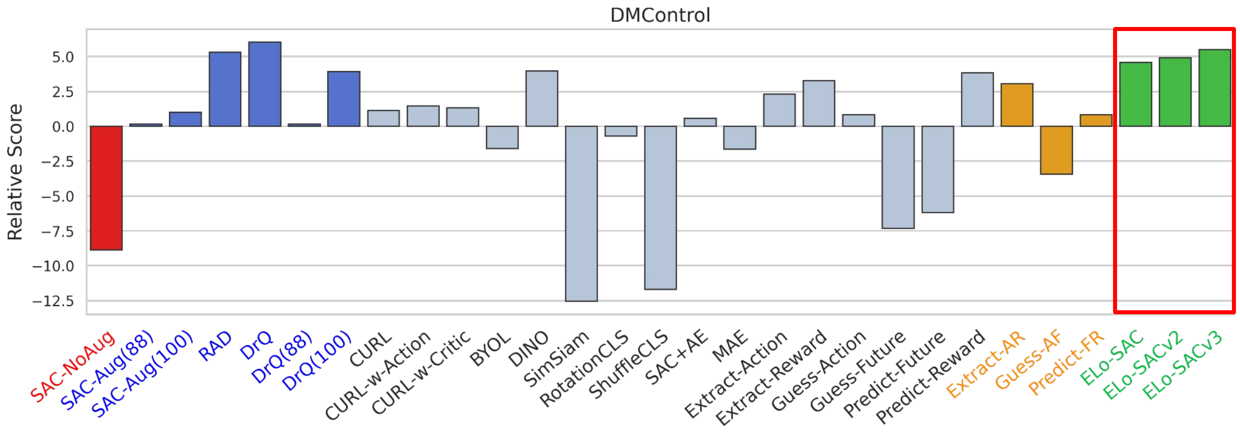
Results: DMControl



	Augmentation	# of SSL
Red	N	0
Blue	Y	0
Gray	Y	1
Orange	Y	2
Green	Y	6

SAC-NoAug: Vanilla SAC w/o augmentation or SSL
Blue: Augmentation w/o SSL
Gray: Augmentation and one SSL loss
Orange: Augmentation and two SSL losses (manually combined)

Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

SAC-NoAug: Vanilla SAC w/o augmentation or SSL

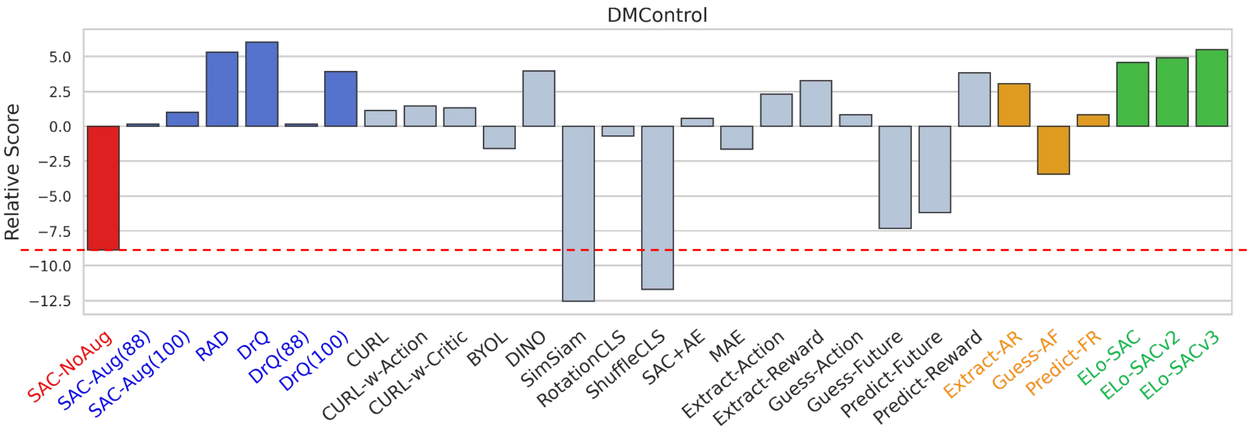
Blue: Augmentation w/o SSL

Gray: Augmentation and one SSL loss

Orange: Augmentation and two SSL losses (manually combined)

Green: Augmentation and multiple SSL losses (automatically combined)

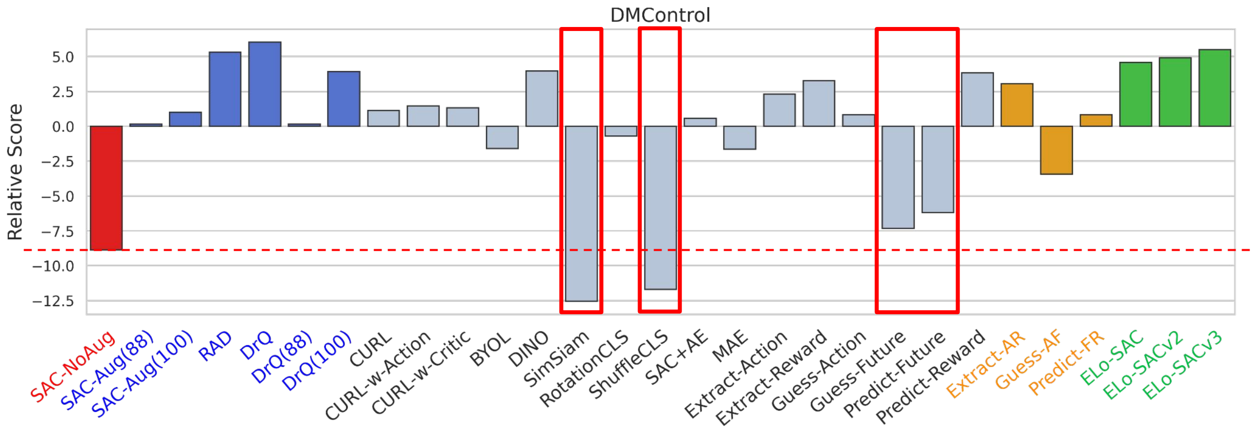
Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

1. Image augmentation is important, as well as how to augment

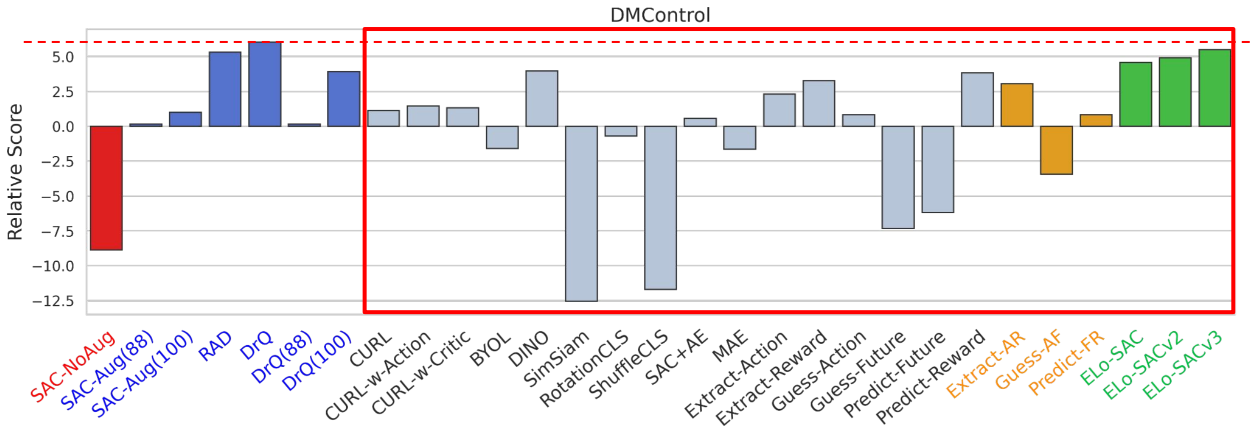
Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

1. Image augmentation is important, as well as how to augment
2. Not all SSL help

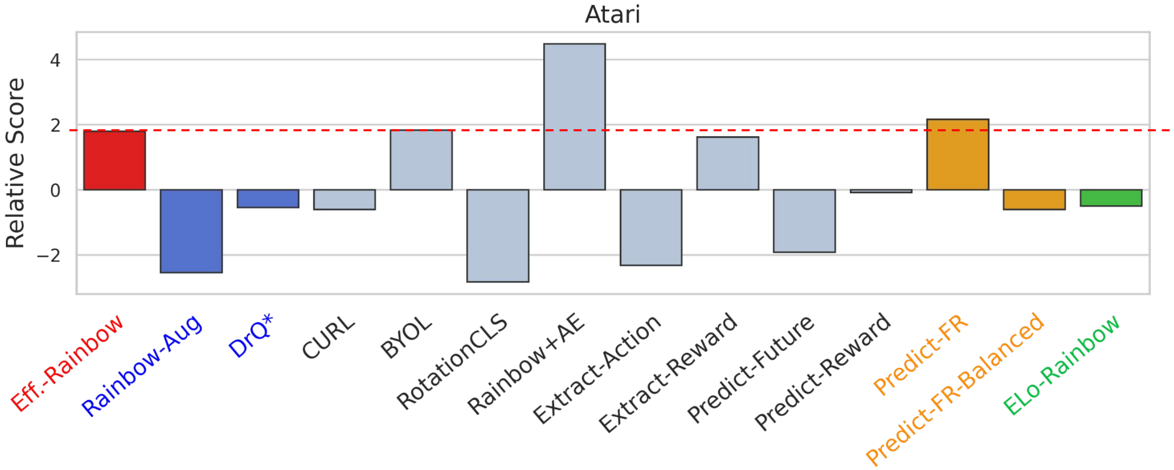
Results: DMControl



	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	6

1. Image augmentation is important, as well as how to augment
2. Not all SSL help
3. Existing SSL methods **fail** to outperform DrQ (based on augmentation)

Experiments: Atari

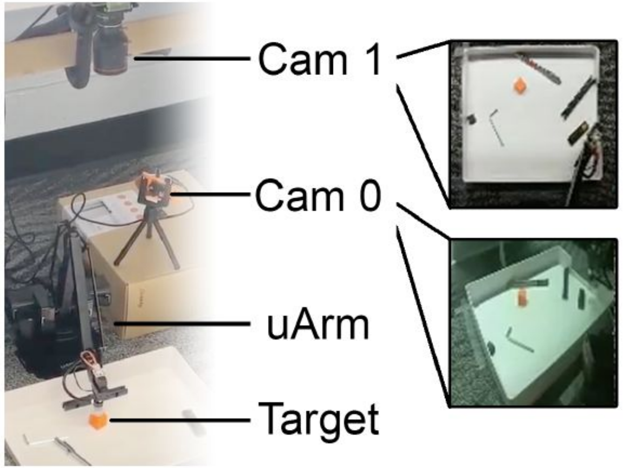


	Augmentation	# of SSL
Red	**N**	**0**
Blue	Y	**0**
Gray	Y	1
Orange	Y	2
Green	Y	5

1. Most of the SSL methods fail to bring meaningful improvements
2. Rainbow+AE performs better than other baselines in Atari, but it did not perform well in DMControl

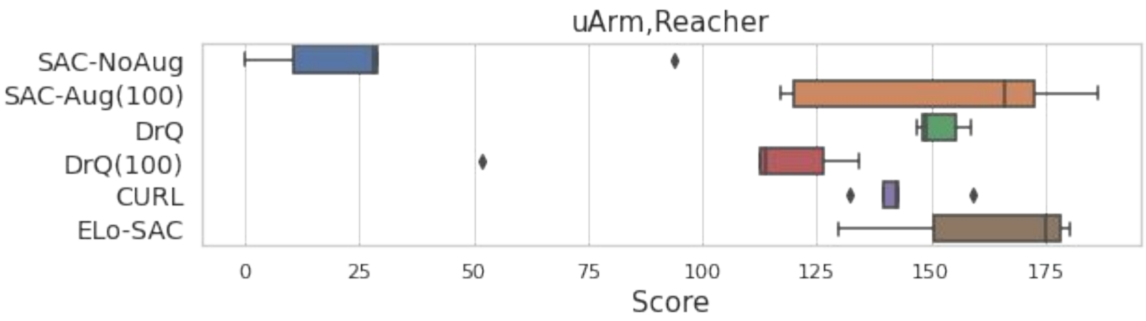
Experiments: Real Robot

Goal: move close to the target as fast as possible
Observation: images from 2 cameras
Action space: 3D robot movement



Autonomously Random Reset

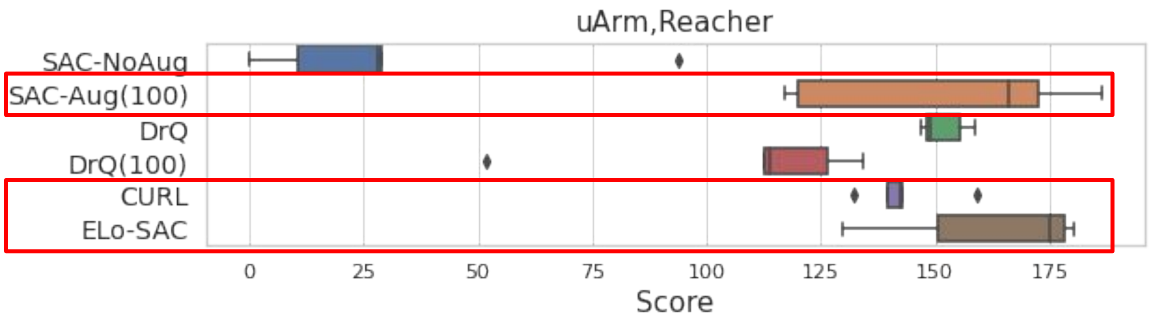
Experiments: Real Robot



SAC-NoAug: SAC w/o aug or SSL
SAC-Aug(100), DrQ, DrQ(100): Aug w/o SSL
CURL: Aug and 1 SSL loss
ELo-SAC: Aug and 6 SSL losses

1. Image augmentation is important, as well as how to augment
2. The role of SSL is usually limited

Experiments: Real Robot



SAC-NoAug: SAC w/o aug or SSL
SAC-Aug(100), DrQ, DrQ(100): Aug w/o SSL
CURL: Aug and 1 SSL loss
ELo-SAC: Aug and 6 SSL losses

1. Image augmentation is important, as well as how to augment
2. The role of SSL is usually limited

Conclusion

1. The role of SSL with the joint learning framework is **usually limited**, while the selection of image augmentation is more important
2. There is **no** golden SSL / augmentation that works for all environments

Further discussion

Ablations, representation analysis, pretraining framework, …

Please check our paper and the appendix.

Stony Brook University

CVLab@StonyBrook

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

NEURAL INFORMATION PROCESSING SYSTEMS

Thank you!

Our code is available at

https://github.com/LostXine/elo-sac