Reinforcement Learning with Augmentation Invariant Representation: A Non-contrastive Approach

Nasik Muhammad Nafi William Hsu

Department of Computer Science, Kansas State University

GenPlan Workshop at NeurIPS 2023

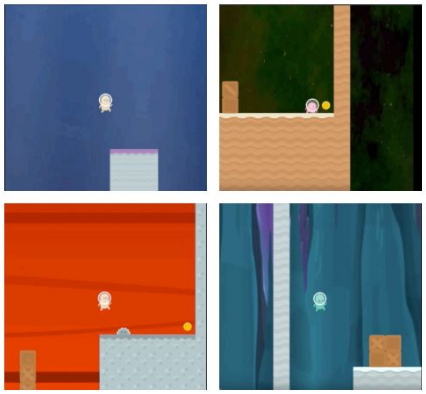
KANSAS STATE UNIVERSITY

NEURAL INFORMATION PROCESSING SYSTEMS

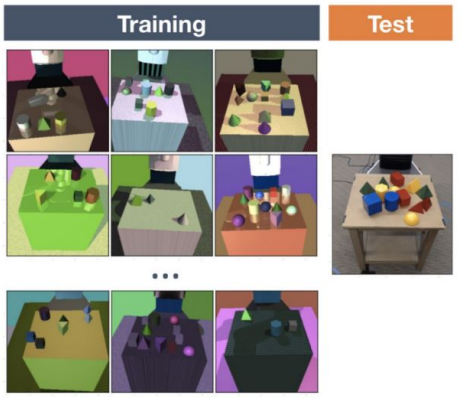
Generalization in RL

- Earlier RL agents were trained and tested on the same setting. Overfitting was appreciated implicitly!
- Generalization to similar but unseen scenarios/dynamics/tasks.

Traditional ALE environments



Coinrun Environment

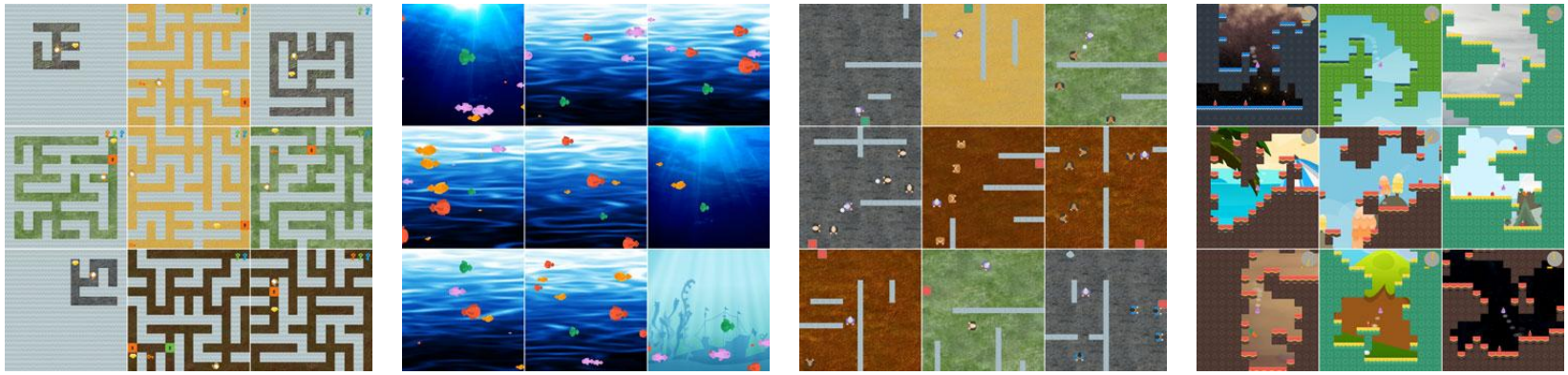


Object Manipulation

Environments to evaluate RL generalization

Problem Statement

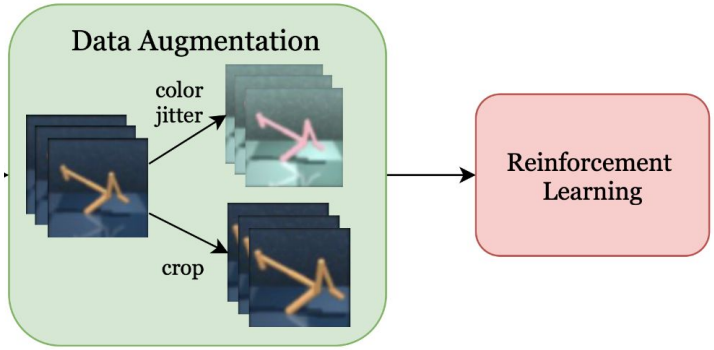
- **Zero-shot generalization** to unseen levels of procedurally generated environments.
- Training on a limited number of levels sampled from the distribution $\mathcal{D}$ while testing on the full distribution.



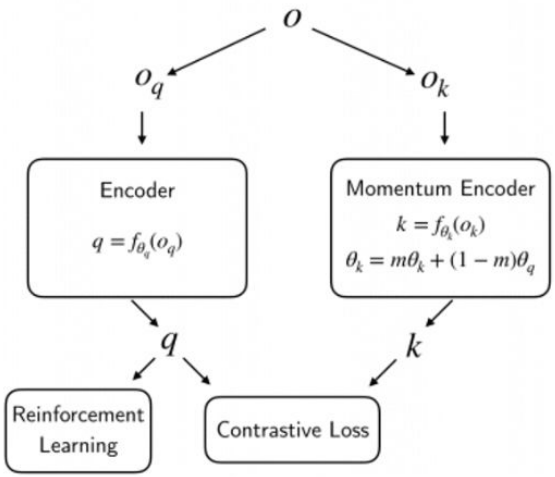
Heist | Bigfish | Dodgeball | Jumper

Example of Procgen Benchmark

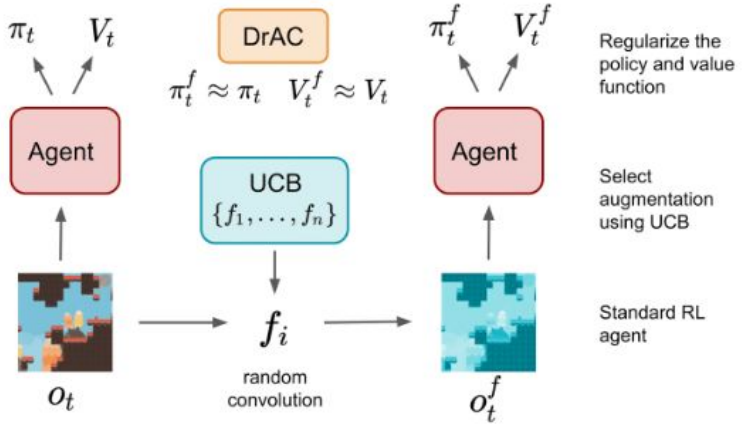
Approaches to Use Data Augmentation in RL



RAD (Laskin et al., 2020)



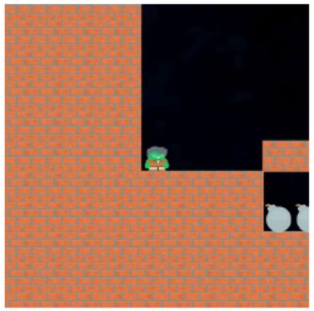
CURL (Laskin et al., 2020)



UCB-DrAC (Raileanu et al., 2021)

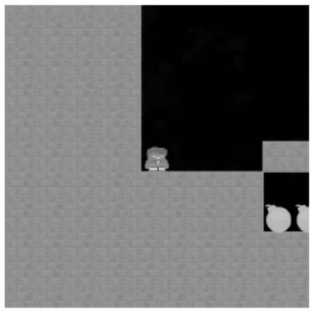
Motivation

- Representations of similar states may get pushed apart in the latent space if contrastive learning is used.



Current Observation

Augmented



Positive

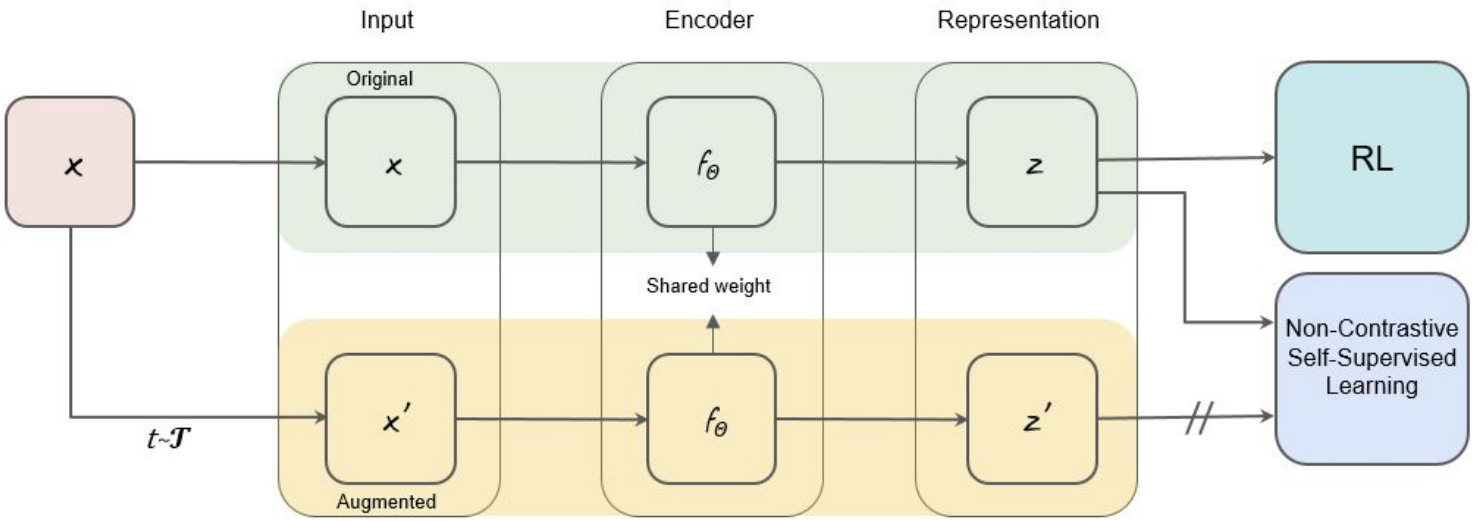
Negative

From different episode

Same spot of the road in different weather condition
From *CARLA* simulator

RL with Augmentation Invariant Representation

- Seeks to learn similar latent representation for augmented observations.
- Alternates optimization between non-contrastive loss and RL objective.



Overall Algorithm: RAIR

Algorithm 1 RAIR: **R**einforcement learning with **A**ugmentation **I**nvariant **R**epresentation

Hyperparameters: total number of updates N, replay buffer size T, minibatch size M, encoder network params ϕ, params for actor-critic heads θ, image transformation t_r with parameters v_i.
for $n = 1, \ldots, N$ **do**
  Collect $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{T}$ using $\pi(\theta)$
  for $j = 1, \ldots, \frac{T}{M}$ **do**
    $\{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{M} \sim \mathcal{D}$
    for $i = 1, \ldots, M$ **do**
      $v_i \sim \mathcal{H}$
      $z_i \leftarrow f_\phi(s_i)$
      $\hat{z}_i \leftarrow f_\phi(t_r(s_i; v_i))$
    end for
    $J(\phi) = \frac{1}{M} \sum_{i=1}^{M} 1 - cos_s(z_i, \hat{z}_i)$
    Train encoder with $J(\phi)$
    $J_{PPO}(\theta, \phi) \leftarrow J_\pi + \alpha_v L_V - \alpha_\pi S_\pi$
    Train full actor-critic network with $J_{PPO}(\theta, \phi)$
  end for
end for

Evaluation on Procgen Benchmark: Crop

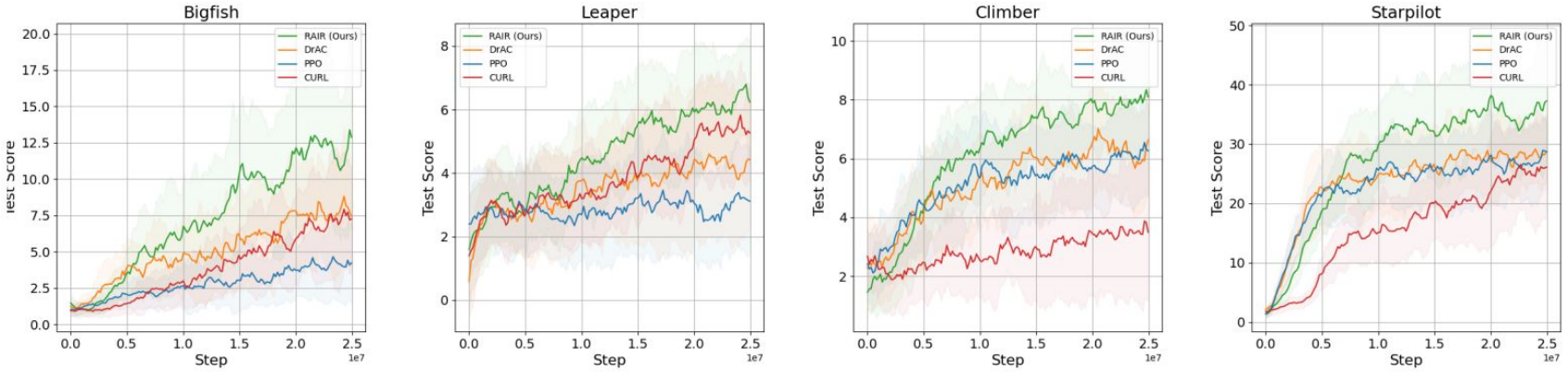


Figure: Performance of RAIR (ours) vs. standard PPO, DrAC, and CURL for ***crop*** data augmentation in Procgen environments. Mean and standard deviations are calculated over 5 trials with different seeds.

Evaluation on Procgen Benchmark: Grayscale

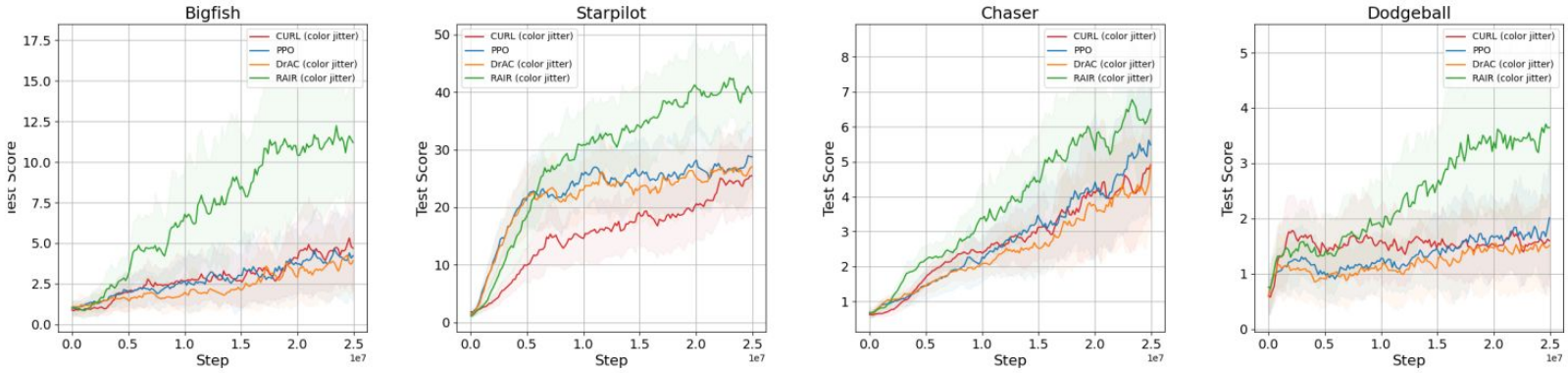


Figure: Performance of RAIR (ours) vs. standard PPO, DrAC, and CURL for ***grayscale*** data augmentation in Procgen environments. Mean and standard deviations are calculated over 5 trials with different seeds.

Evaluation on Procgen Benchmark: Color Jitter

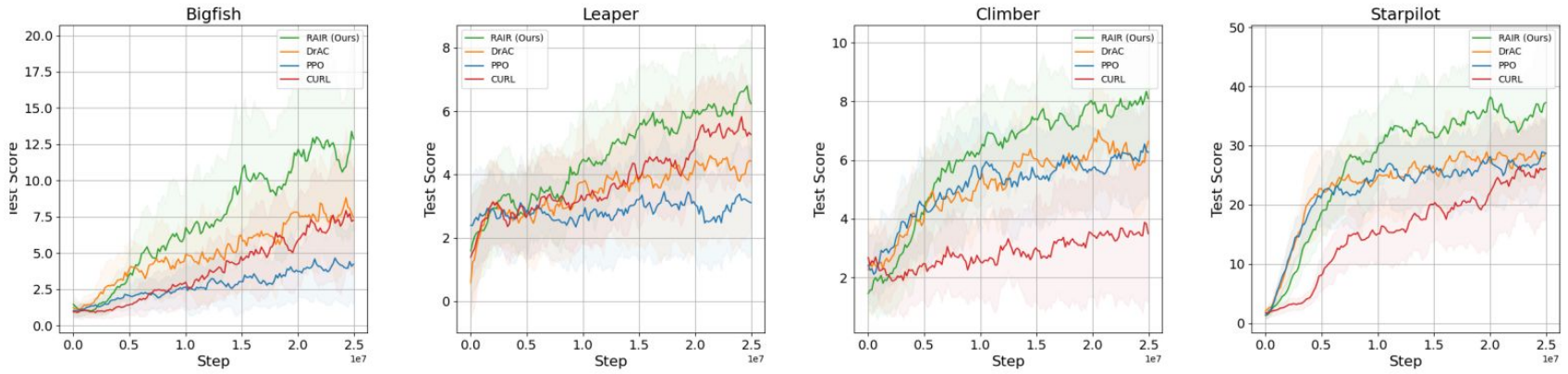


Figure: Performance of RAIR (ours) vs. standard PPO, DrAC, and CURL for ***color jitter*** data augmentation in Procgen environments. Mean and standard deviations are calculated over 5 trials with different seeds.

Takeaways

- Non-contrastive loss with only positive pairs can effectively facilitate representation learning in RL.
- RAIR can unlock the potential of different data augmentation techniques as opposed to earlier methods.
- Simple loss function like L1 loss can be used as similarity metric.
- Using non-contrastive loss as an auxiliary one is often susceptible to high variance across trials and performance degradation.
- Momentum encoder is indeed not necessary with non-contrastive loss.

For details visit

Thank You!