Decoupling Value and Policy for Generalization in Reinforcement Learning

Roberta Raileanu
NYU

Rob Fergus
NYU

ICML 2021

NYU

AI Habitat, Savva et al. 2019

Source: aihabitat.org

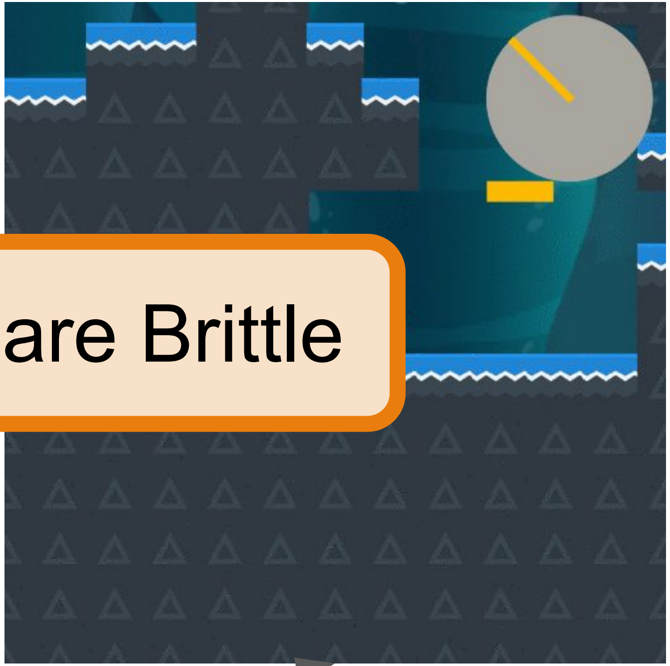
Learn to Solve a Task in **Any** Scenario by Training on a **Limited** Number of Task Instances

AI Habitat, Savva et al. 2019

Source: aihabitat.org

Train Environment

Test Environment



Current Agents are Brittle

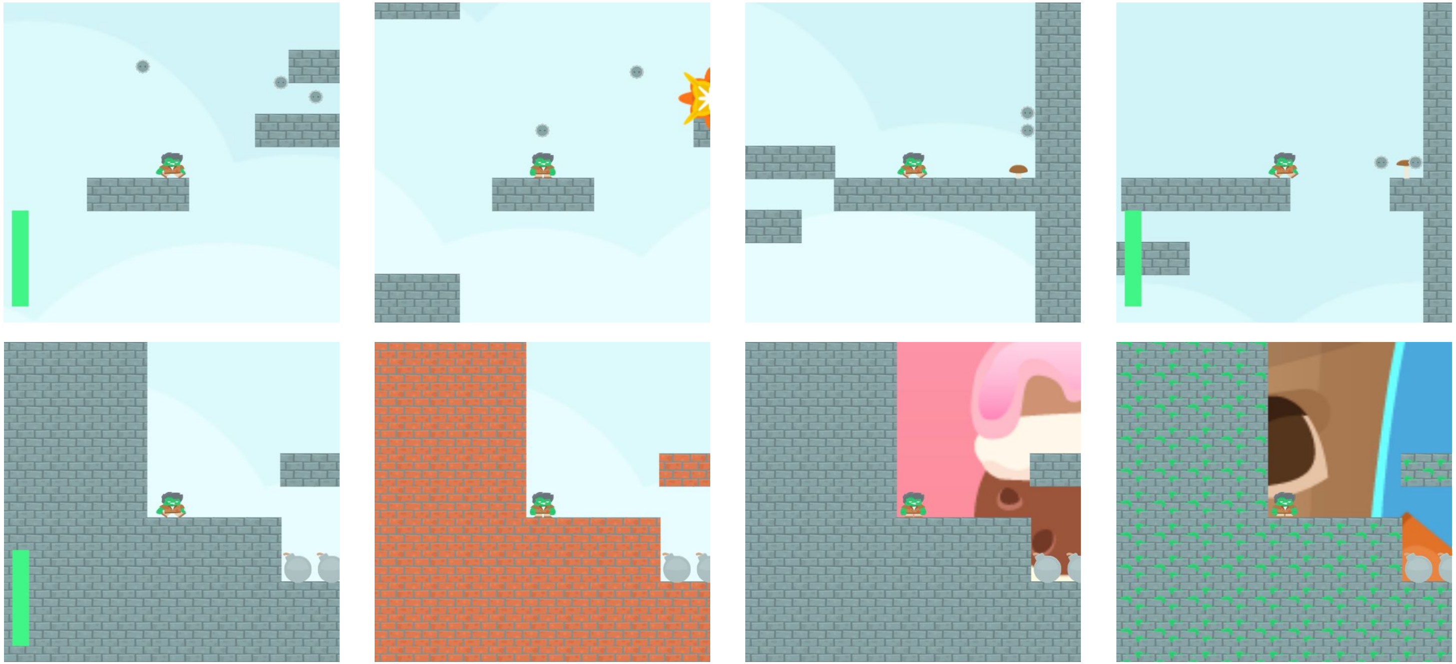
Different Backgrounds

Problem Setting: Family of POMDPs

Same action space and reward function, different dynamics

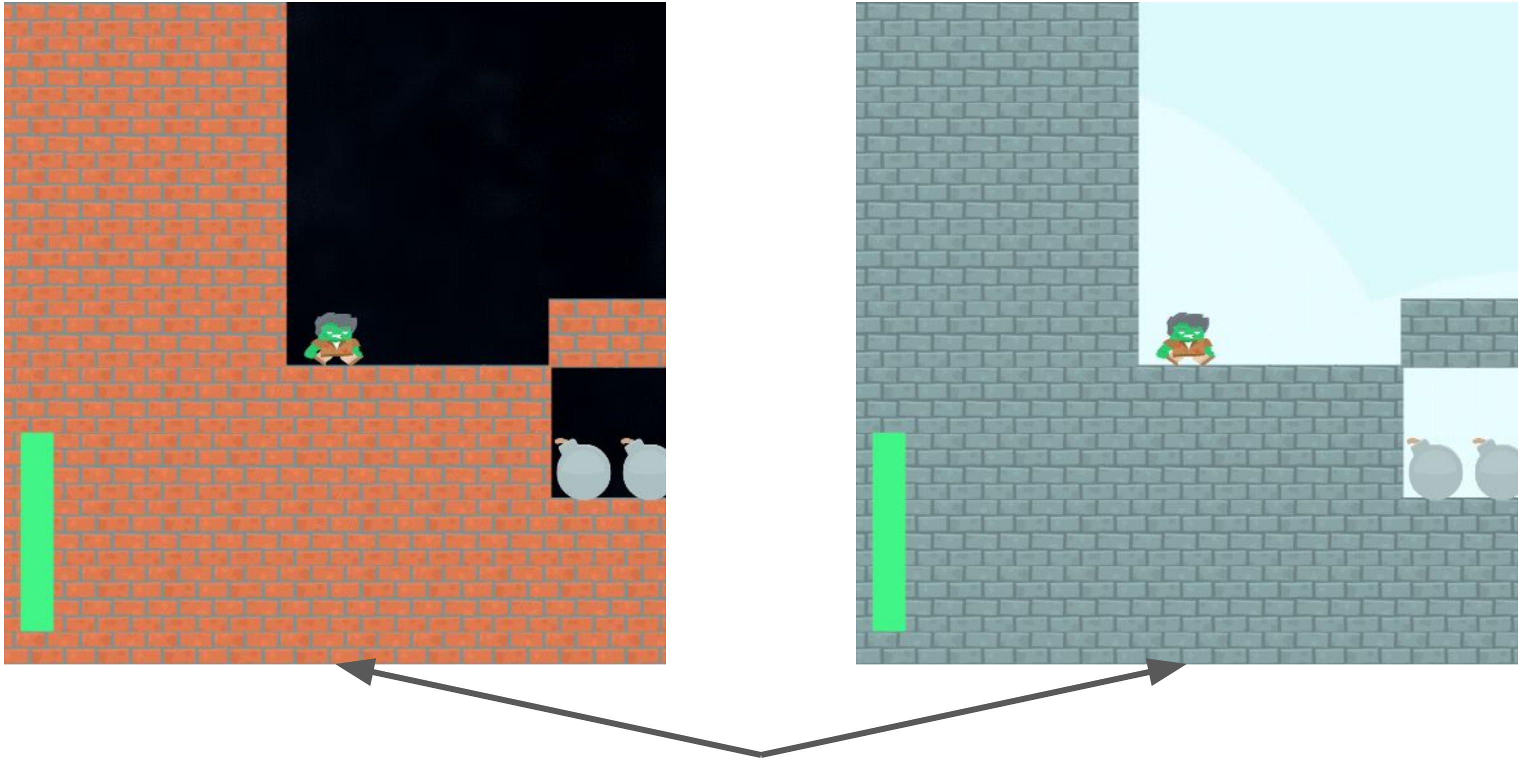
Different States

Different Observations



Train on a small number of environments and test on the full distribution

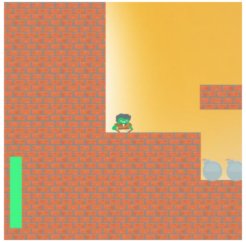
Generalizing to New Task Instances



Different Episode Lengths

Common Network for the Policy and Value

$\boldsymbol{s}$



$\boldsymbol{\theta}$

$V(s)$

$\pi(a|s)$

Without gradients from the value function, the policy struggles to learn

Policy-Value Asymmetry

Semantically Identical, Visually Different

Same Optimal Policy

Different Optimal Values

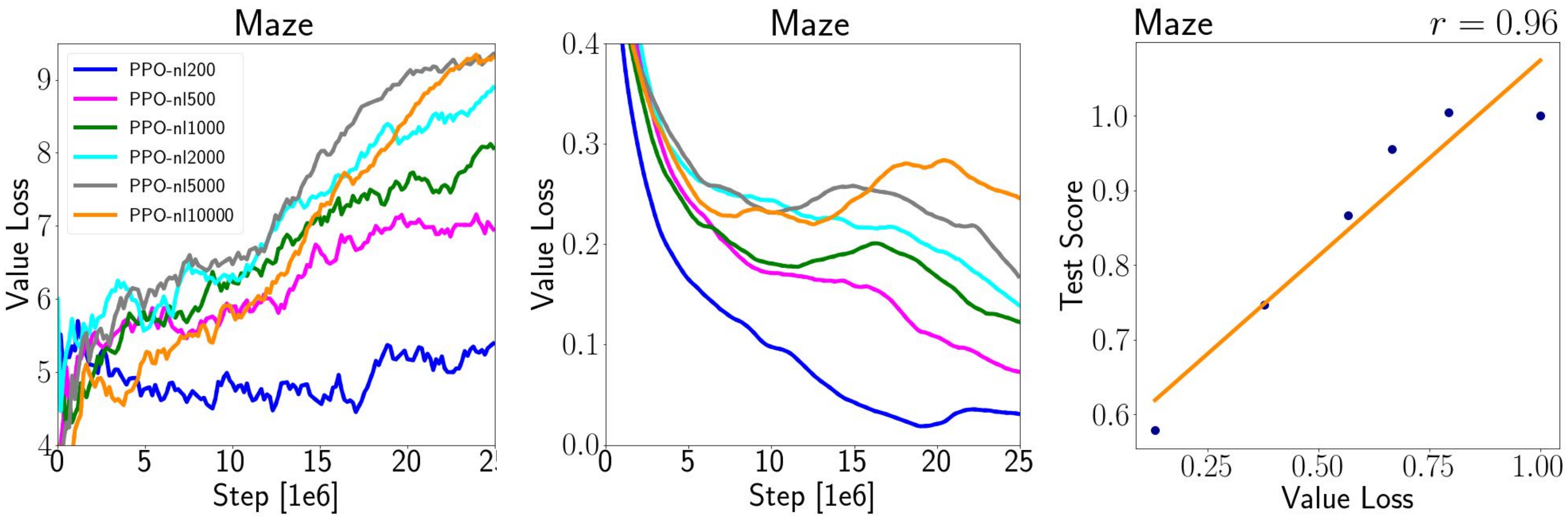
$V_1(s_0) = 7.9$ Short Episode

$V_2(s_0) = 6.1$ Long Episode

Need level-specific features to accurately estimate the value

Using a common representation for the policy and value can lead to overfitting

Trade-off between Generalization and Value Loss



Counterintuitive finding: models with **good generalization** have **high value loss**

Advantage Function

$$A^{\pi}(s_t, a_t) := Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

$$Q^{\pi}(s_t, a_t) := \mathbb{E}_{\pi}\left[\sum_{l=0}^{H-t} \gamma^l r_{t+l} | s_t = s, a_t = a\right]$$

$$V^{\pi}(s_t) := \mathbb{E}_{\pi}\left[\sum_{l=0}^{H-t} \gamma^l r_{t+l} | s_t = s\right]$$

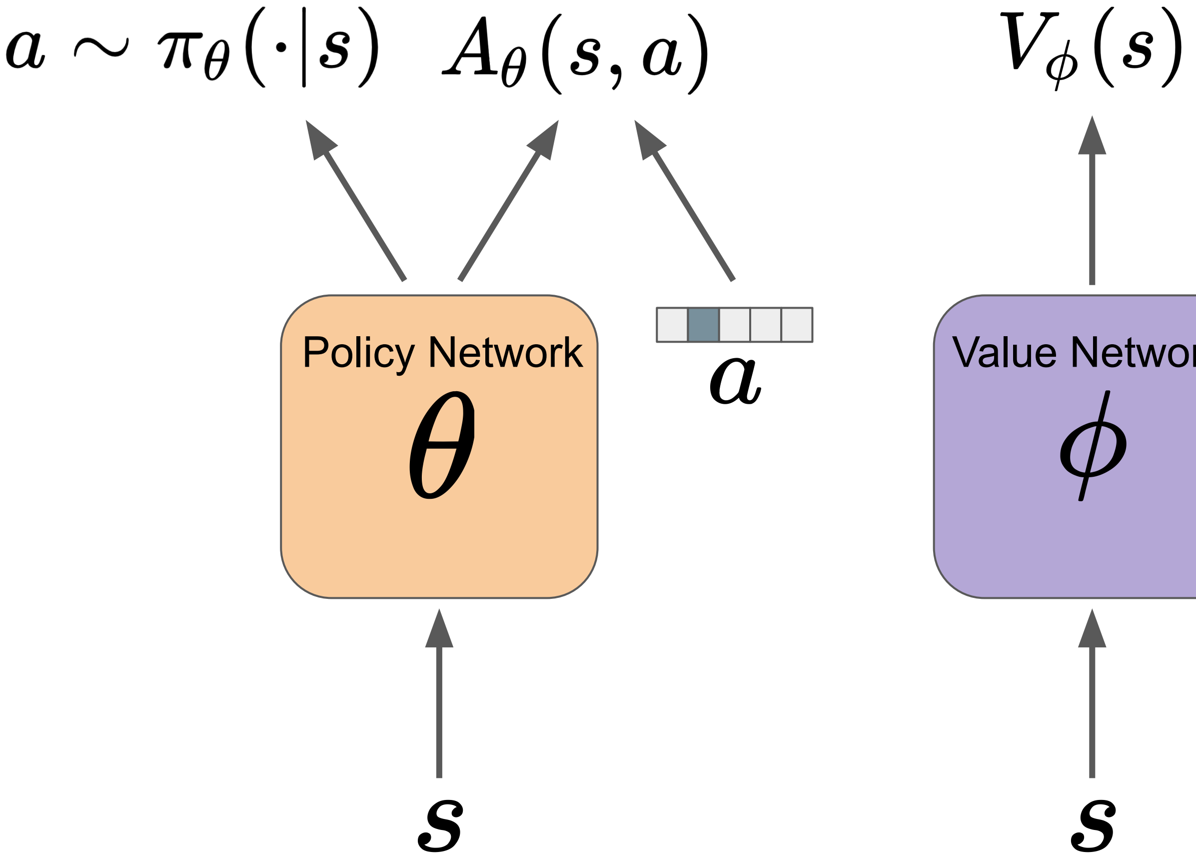
$\boxed{A_1(s_0, a_0) = -0.1}$

$\boxed{A_2(s_0, a_0) = -0.1}$

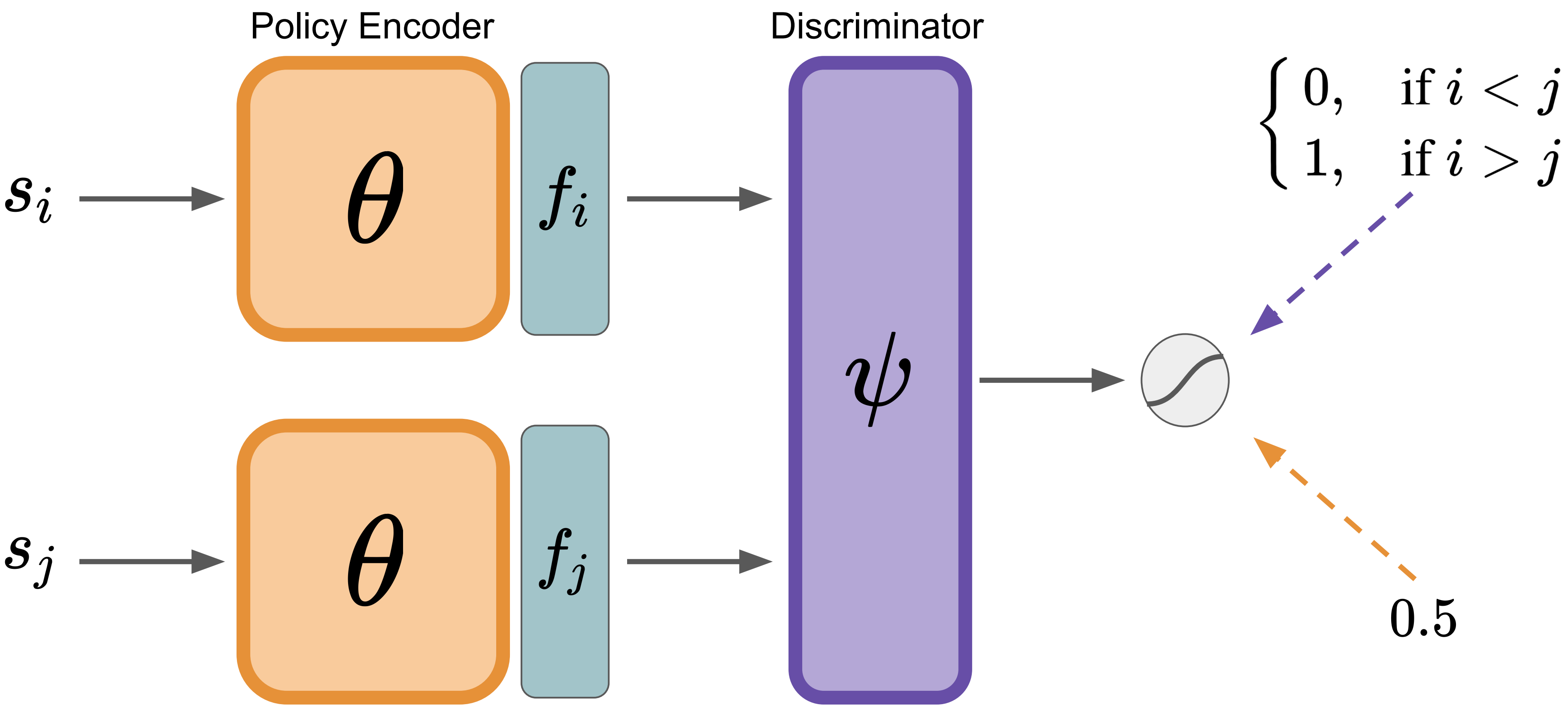
Same Advantages

The advantage function is less prone to overfitting than the value function

Decoupled Advantage Actor-Critic (DAAC)



Invariant Decoupled Advantage Actor-Critic (IDAAC)



Related Work

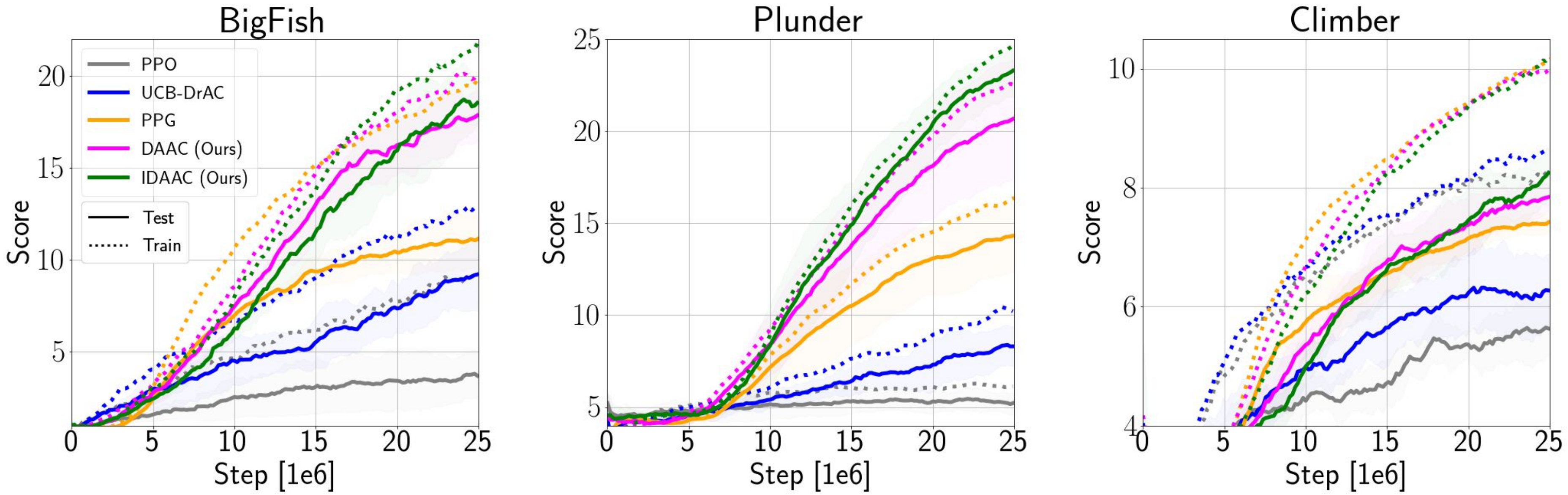
Decoupling the value and policy for sample efficiency: PPG (*Cobbe et al. 2020*)

Data Augmentation: Cobbe et al. 2018, RAND-FM (*Lee et al. 2019*), RAD (L*askin et al. 2020*), DrQ (*Kostrikov et al. 2020*), UCB-DrAC (*Raileanu et al. 2020*), Mixreg (*Wang et al. 2020*)

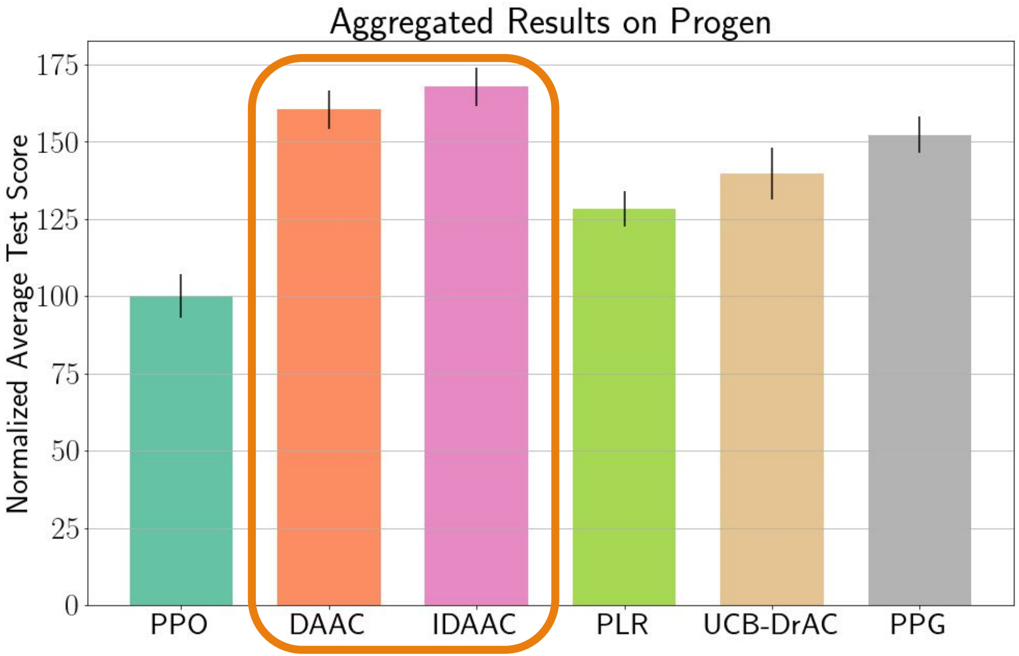
Representation Learning: information bottleneck (*Igl et al., 2019*), bisimulation metrics (*Zhang et al. 2020*), unsupervised learning (*Stooke et al., 2020*), state abstractions (*Agarwal et al. 2021*), mutual information (*Mazoure et al. 2020*)

Other Approaches for Generalization in RL: policy distillation (*Igl et al. 2019*), automatic curricula (PLR, *Jiang et al. 2020*), etc.

Test Performance



Results on the Procgen Benchmark



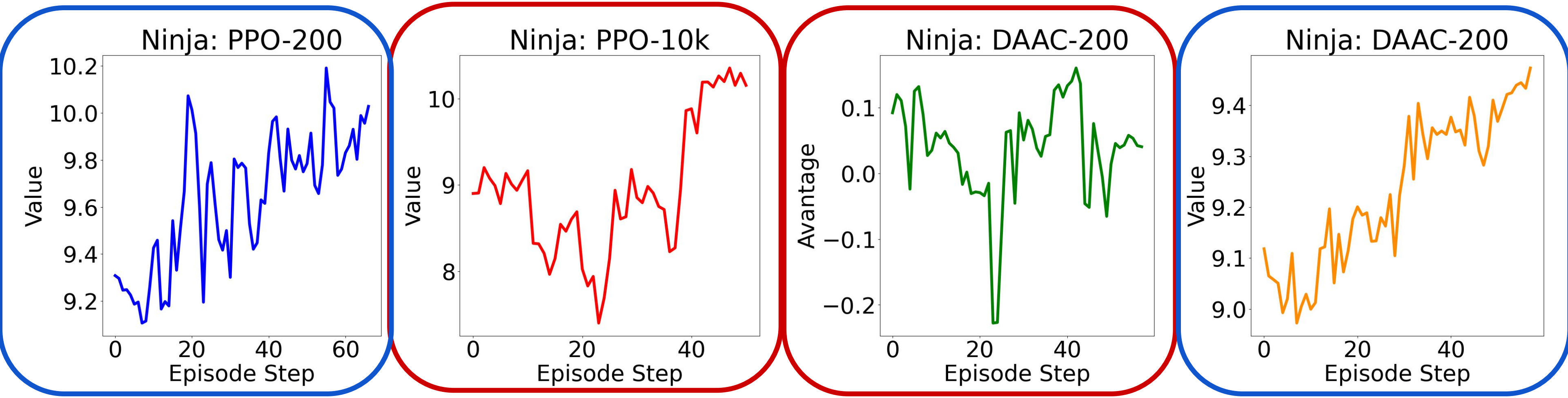
IDAAC: SOTA on Procgen and 64% better than standard RL on test environments

Good Generalization and Low Value Loss

Test Score: 5.9
Value Loss: 0.2

Test Score: 8.8
Value Loss: 0.3

Test Score: 7.3
Value Loss: 0.2



The advantage does not have a linear trend, leading to **better generalization**

By decoupling the value and policy, DAAC achieves **lower value loss**

Agent Behavior On New Environments

Takeaways

Predicting the value requires more information then learning the policy

Using a common representation for the policy and value leads to overfitting

Predicting advantage instead of value improves generalization

Inductive Bias: learn state representations invariant to the episode step

Decoupling Value and Policy for Generalization in Reinforcement Learning

Thank you!

Paper: https://arxiv.org/abs/2102.10330
Code: https://github.com/rraileanu/idaac