Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, Marc G. Bellemare

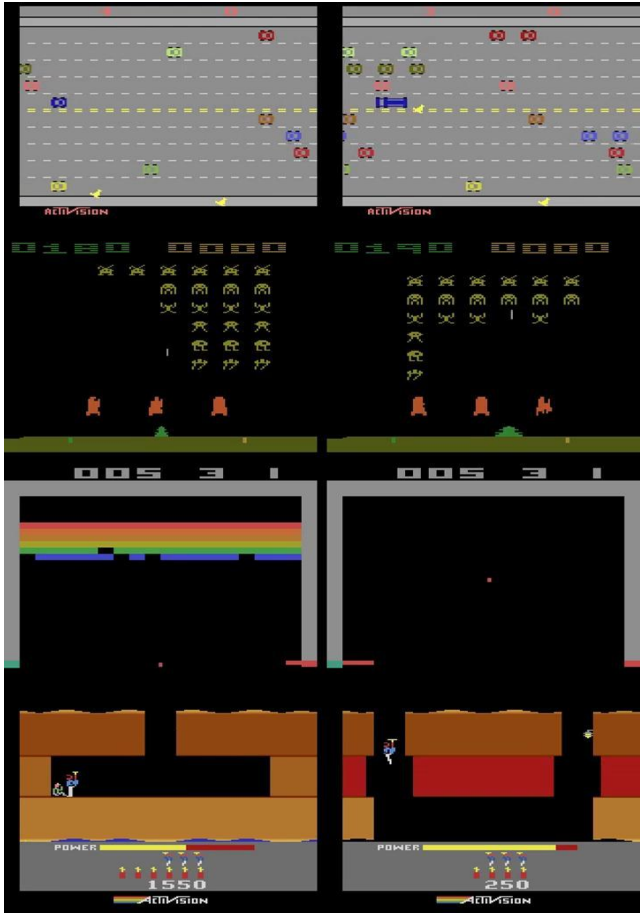



Google Research

[Machado et al., JAIR 2018]

Aspiration I

Agents should "do well" in environment(s) *semantically* similar to training environments.

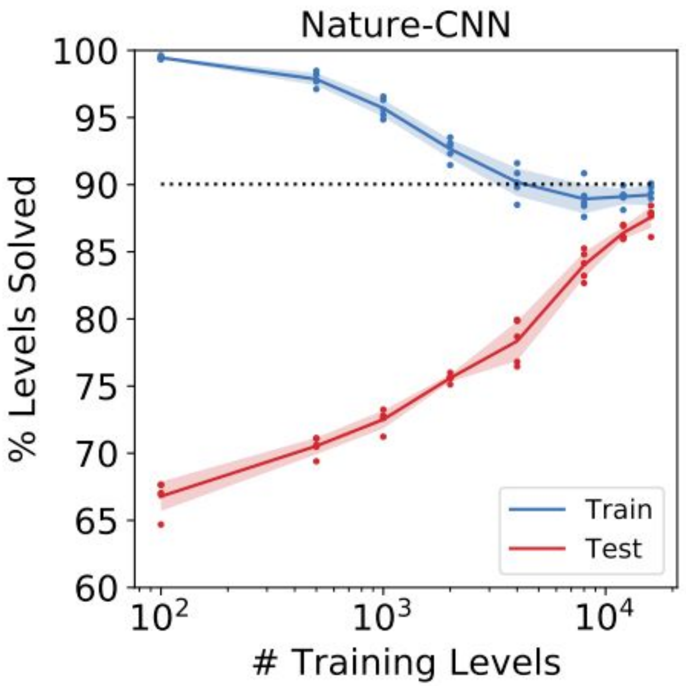


Slide adapted from Marlos C. Machado

[Cobbe et al., ICML 2019]

Aspiration II

Train agents that can generalize from a "few" environments rather than hundreds or thousands of environments.

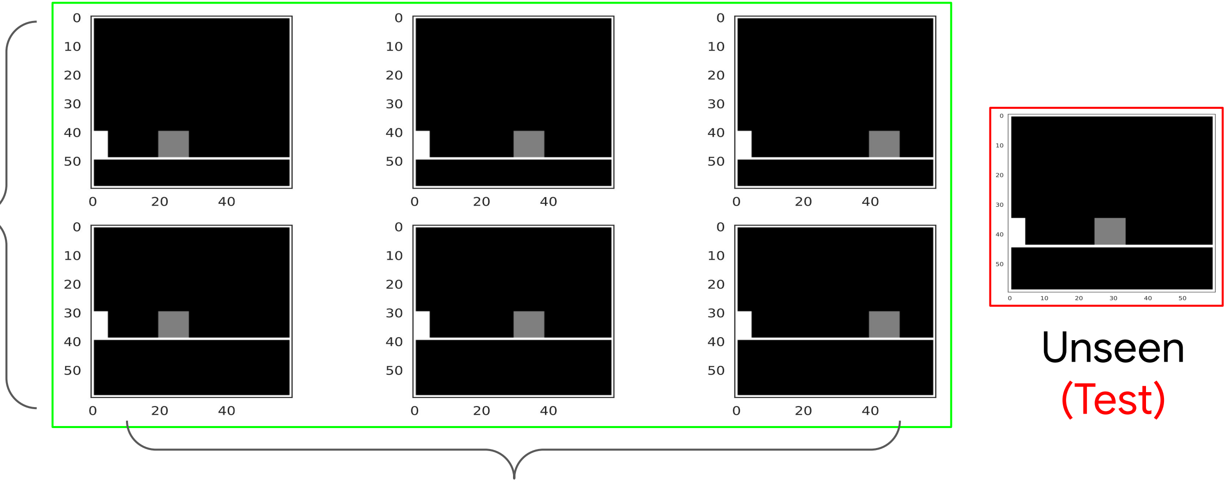


Generalization Performance

Setup: Generalization in RL

- Learn using finite tasks sampled from distribution $\mathcal{D}$
- Evaluate performance on “unseen” tasks in $\mathcal{D}$

Different floor heights (Train)



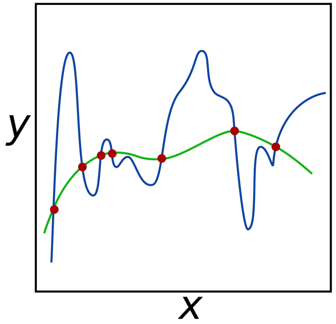
Unseen (Test)

Different obstacle positions (Train)

Jumping Task from Pixels

Prior Work on Generalization

Adapted from supervised learning, *e.g.* :



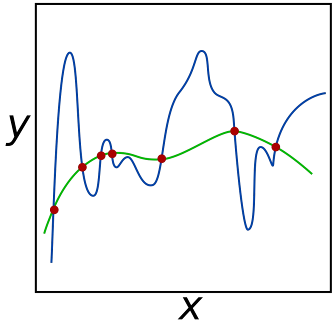
Regularization
(ℓ2-reg., Dropout, Noise Injection)

1. Farebrother, Jesse, et al. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123*, 2018
2. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. Quantifying generalization in reinforcement learning. ICML, 2019
3. Igl, Maximilian, et al. "Generalization in reinforcement learning with selective noise injection and information bottleneck. NeurIPS, 2019

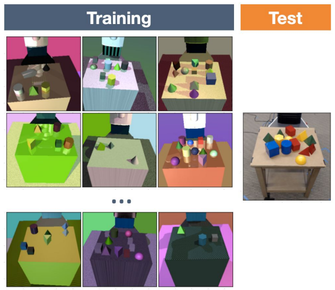
Prior Work on Generalization

Google Research

Adapted from supervised learning, *e.g.* :



Regularization (ℓ2-reg., Dropout, Noise Injection)

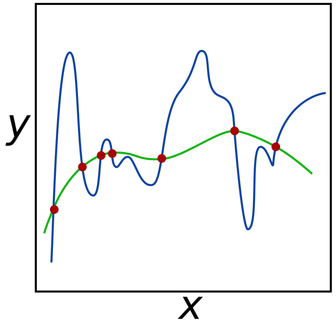


Domain Randomization

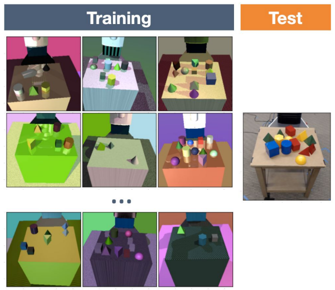
1. Farebrother, Jesse, et al. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123*, **2018**
2. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. Quantifying generalization in reinforcement learning. ICML, **2019**
3. Igl, Maximilian, et al. "Generalization in reinforcement learning with selective noise injection and information bottleneck. NeurIPS, **2019**
4. **Tobin, Josh, et al. "Domain randomization for transferring deep neural networks from simulation to the real world." IROS, 2017**

Prior Work on Generalization

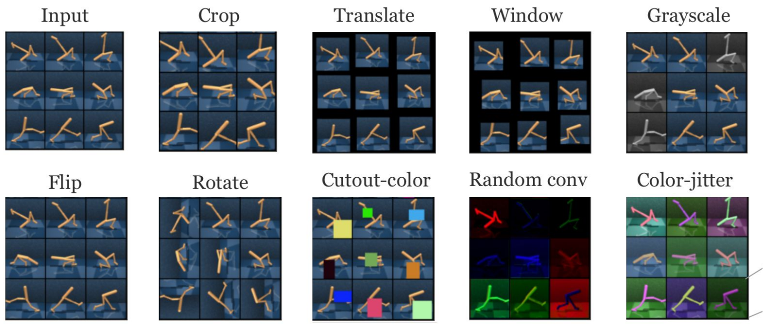
Adapted from supervised learning, *e.g.* :



Regularization
(ℓ2-reg., Dropout,
Noise Injection)



Domain
Randomization



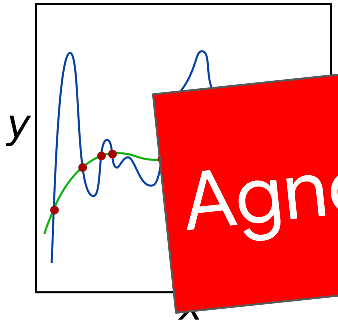
Data Augmentation
(RandConv, RAD, DrQ ..)

1. Farebrother, Jesse, et al. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123*, **2018**
2. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. Quantifying generalization in reinforcement learning. ICML, **2019**
3. Igl, Maximilian, et al. "Generalization in reinforcement learning with selective noise injection and information bottleneck. NeurIPS, **2019**
4. Tobin, Josh, et al. "Domain randomization for transferring deep neural networks from simulation to the real world." *IROS, **2017***
5. **Lee, Kimin, et al. "Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning." ICLR. 2019**
6. **Kostrikov, Ilya, et al. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020**
7. **Laskin, Mischa, et al.. Reinforcement Learning with Augmented Data. NeurIPS, 2020**

Prior Work on Generalization

Adapted from supervised learning, e.g. :

Agnostic to Sequential Structure in RL!



Regularization
(ℓ2-reg., Dropout,
Noise Injection)

Domain
Randomization

Data Augmentation
(RandConv, RAD, DrQ ..)

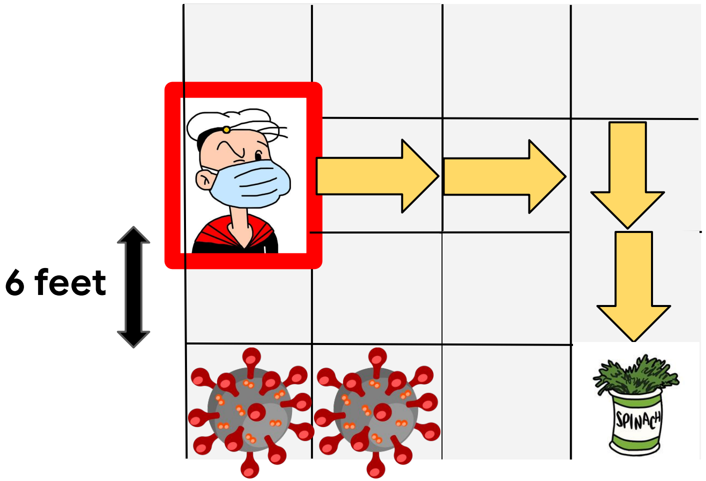
1. Farebrother, Jesse, et al. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123*, **2018**
2. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. Quantifying generalization in reinforcement learning. ICML, **2019**
3. Igl, Maximilian, et al. "Generalization in reinforcement learning with selective noise injection and information bottleneck. NeurIPS, **2019**
4. Tobin, Josh, et al. "Domain randomization for transferring deep neural networks from simulation to the real world." *IROS, **2017***
5. Lee, Kimin, et al. "Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning." ICLR. 2019
6. Kostrikov, Ilya, et al. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020
7. Laskin, Mischa, et al.. Reinforcement Learning with Augmented Data. NeurIPS, 2020

This work ...

Learn representations that encode
“**behavioral similarity**” across states!

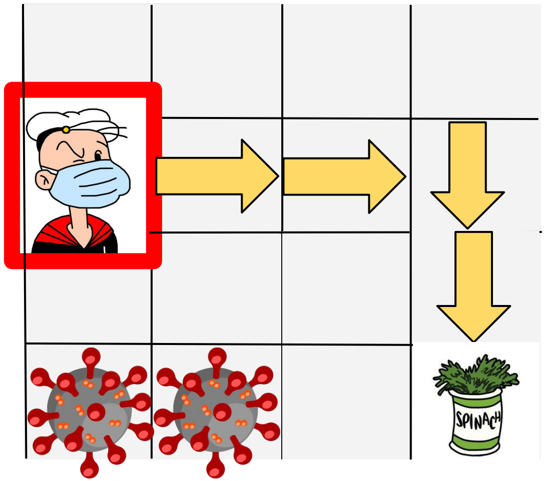
This work ...

Learn representations that encode
"**behavioral similarity**" across states!

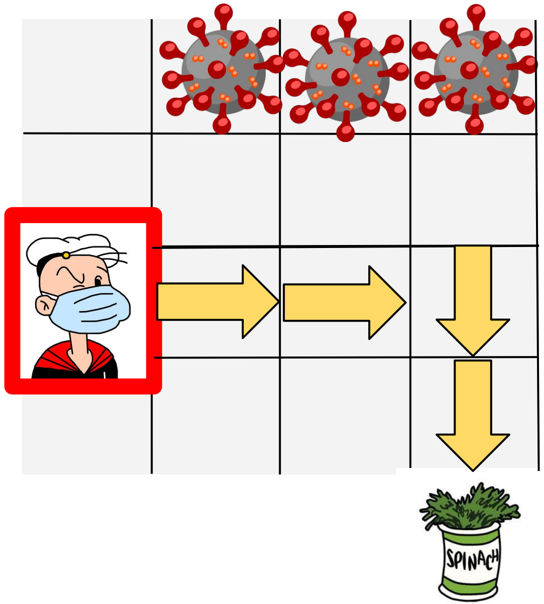


This work ...

Learn representations that encode "**behavioral similarity**" across states!

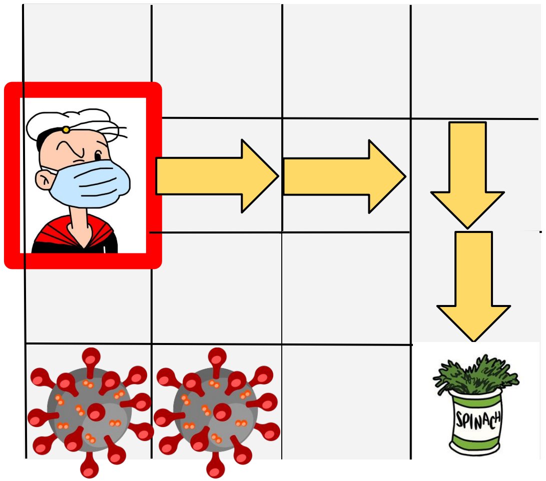


"Similar" States

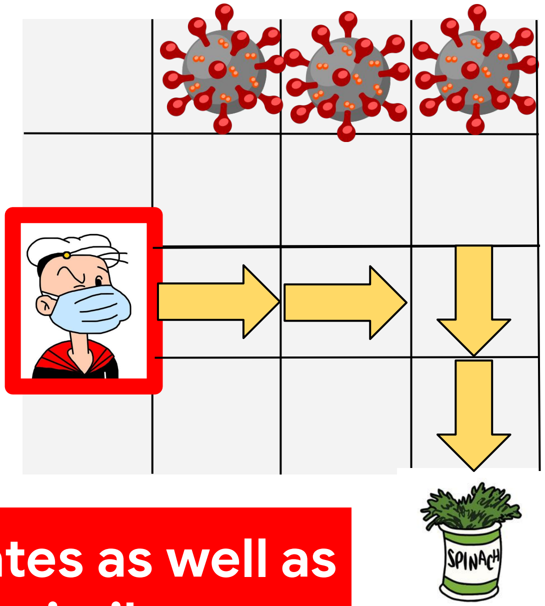


This work ...

Learn representations that encode "**behavioral similarity**" across states!



"Similar" States



Actions in current states as well as future states are similar.

Defining Behavioral Similarity

- Our metric builds on **bisimulation** metrics.
- Two states are bisimular if they have **similar expected rewards** and **dynamics**.

Ferns, Norm, Prakash Panangaden, and Doina Precup. "Metrics for Finite Markov Decision Processes." *UAI*. Vol. 4. 2004.

Notation

- $\mathcal{D}$: Distribution over environments with action space **A**
- Environments $\mathbf{M}_{\mathcal{X}} \sim \mathcal{D}$, $\mathbf{M}_{\mathcal{Y}} \sim \mathcal{D}$ with state spaces $\mathcal{X}$, $\mathcal{Y}$
- $\mathbf{M}_{\mathcal{X}} \rightarrow$ Optimal Policy $\boldsymbol{\pi}^*_{\mathcal{X}}$, Dynamics $\mathbf{P}_{\mathcal{X}}$, Rewards $\mathbf{R}_{\mathcal{X}}$
- $\mathcal{S} = \mathcal{X} \cup .. \cup \mathcal{Y}$. Union of state spaces of environments in $\mathcal{D}$
- **Union MDP:** State Space $\mathcal{S}$, Dynamics **P**, Rewards **R**
- $\mathbf{P}^{\boldsymbol{\pi}}$, $\mathbf{R}^{\boldsymbol{\pi}}$: Dynamics and rewards induced by $\boldsymbol{\pi}$

π-Bisimulation Metric

$$d_\pi(x, y) = \underbrace{|R^\pi(x) - R^\pi(y)|}_{\text{Reward Difference}} + \gamma \underbrace{\mathcal{W}_1(d_\pi)(P^\pi(\cdot \mid x), P^\pi(\cdot \mid y))}_{\text{Long-term discounted future reward difference}}$$

[1] Castro, Pablo Samuel. "Scalable methods for computing state similarity in deterministic markov decision processes." *AAAI, 2020.*

Google Research

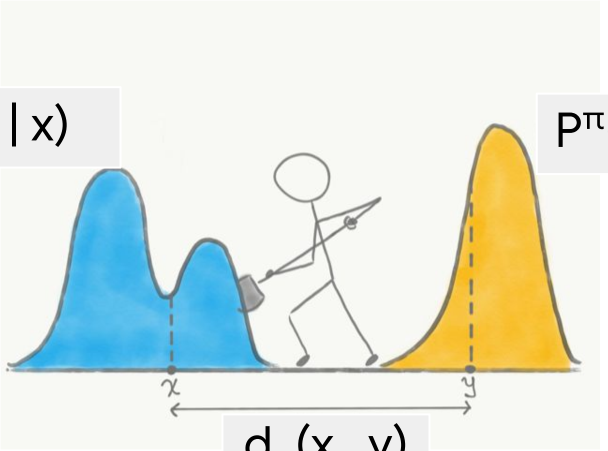
π-Bisimulation Metric

$$d_\pi(x, y) = |R^\pi(x) - R^\pi(y)| + \gamma \mathcal{W}_1(d_\pi)(P^\pi(\cdot \,|\, x), P^\pi(\cdot \,|\, y))$$

Next State Distributions

Wasserstein-1 distance under d_π

$P^\pi(. \,|\, x)$



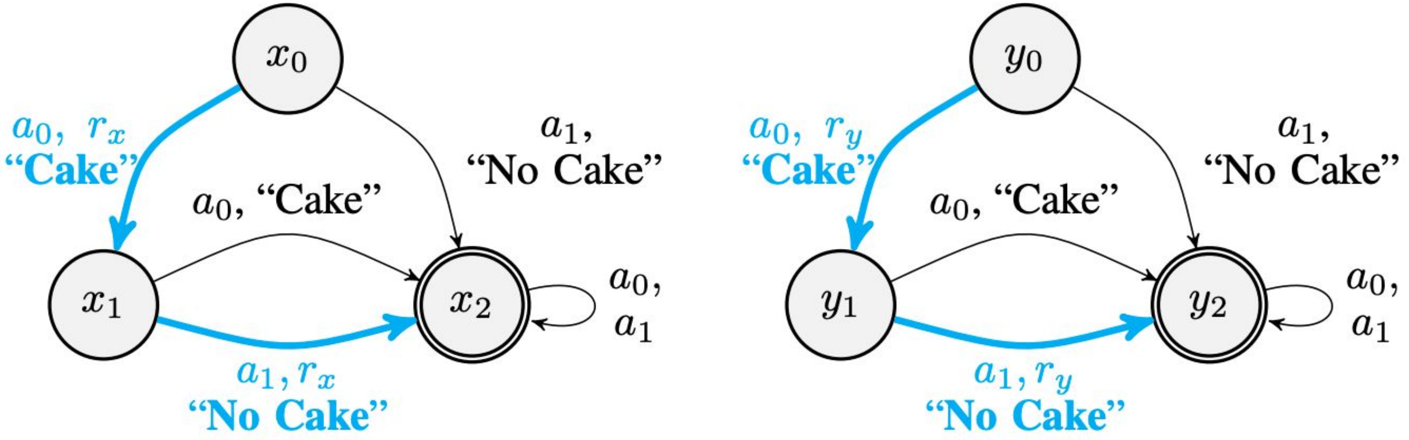
$P^\pi(. \,|\, y)$

$d_\pi(x , y)$

Minimal cost of transporting probability mass between 2 distributions under the base metric d_π

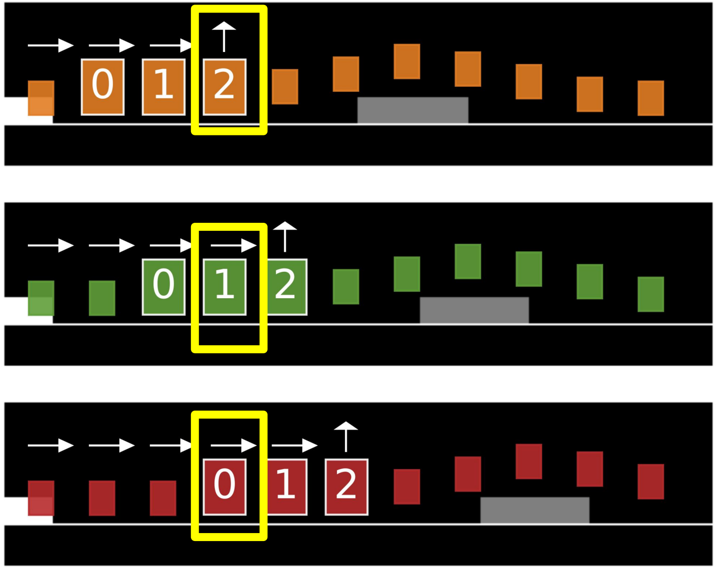
Google Research

Problem #1
Similar Behavior, **Different** Rewards



Bisimilarity(x_0, y_0) > Bisimilarity(x_0, y_1)

Problem #2
Different Behavior, **Similar** Rewards



+1 reward at each step

Expected rewards are same for states 2 1 0

Policy Similarity Metric (PSM)

$$d_\pi(x, y) = \underbrace{|R^\pi(x) - R^\pi(y)|}_{\textbf{Reward Difference}} + \gamma \mathcal{W}_1(d_\pi)(P^\pi(\cdot \,|\, x), P^\pi(\cdot \,|\, y)$$

Replace

$$d^*(x, y) = \underbrace{\textsc{Dist}\big(\pi^*(x), \pi^*(y)\big)}_{\textbf{Policy Difference}} + \gamma \mathcal{W}_1(d^*)\big(P^{\pi^*}(\cdot \,|\, x), P^{\pi^*}(\cdot \,|\, y)\big)$$

Google Research

Policy Similarity Metric (PSM)

$$\underbrace{d^*(x, y) = \textsc{Dist}\big(\pi^*(x), \pi^*(y)\big)}_{\text{Local Optimal Behavior Difference}} + \gamma \underbrace{\mathcal{W}_1(d^*)\big(P^{\pi^*}(\cdot \,|\, x), P^{\pi^*}(\cdot \,|\, y)\big)}_{\text{Long-term Optimal Behavior Difference}}$$

Local Optimal Behavior Difference

How far into the future?

Long-term Optimal Behavior Difference

PSM (Deterministic Environments)

$$d^*(x, y) = \textsc{Dist}\big(\pi^*(x), \pi^*(y)\big) + \gamma d^*(x', y')$$

$$= \underbrace{\underbrace{\textsc{Dist}\big(\pi^*(x), \pi^*(y)\big)}_{\text{One-step optimality difference}} + \gamma \textsc{Dist}\big(\pi^*(x'), \pi^*(y')\big)}_{\text{Two-step discounted optimality difference}} + \gamma^2 d^*(x'', y'')$$

Google Research

PSM for generalization

- Given *d**, how well can we transfer optimal policy on $M_{\mathcal{X}}$ to $M_{\mathcal{Y}}$?
- For each y in $M_{\mathcal{Y}}$, pick state in $\mathcal{X}$ closest to y based on PSM, i.e.,

$$\underbrace{\tilde{\pi}(y) = \pi^*(\tilde{x}_y)}_{\text{Transfer Policy}} \text{ where } \underbrace{\tilde{x}_y = \arg\min_{x \in \mathcal{X}} d^*(x, y)}_{\text{Nearest Neighbor}}$$

PSM for generalization

Google Research

- Given *d**, how well can we transfer optimal policy on $M_{\mathcal{X}}$ to $M_{\mathcal{Y}}$?
- For each y in $M_{\mathcal{Y}}$, pick state in $\mathcal{X}$ closest to y based on PSM, i.e.,

$$\tilde{\pi}(y) = \pi^*(\tilde{x}_y) \text{ where } \tilde{x}_y = \arg\min_{x \in \mathcal{X}} d^*(x, y)$$

Transfer Policy

Nearest Neighbor

Theorem 1. *[Bound on policy transfer] For any $y \in \mathcal{Y}$, let $Y_y^t \sim P^{\tilde{\pi}}(\cdot \,|\, Y_y^{t-1})$ define the sequence of random states encountered starting in $Y_y^0 = y$ and following policy $\tilde{\pi}$. We have:*

$$\mathbb{E}_{Y_y^t}\left[\sum_{t \geq 0} \gamma^t TV\left(\tilde{\pi}(Y_y^t), \pi^*(Y_y^t)\right)\right] \leq \frac{1+\gamma}{1-\gamma} d^*(\tilde{x}_y, y)\,.$$

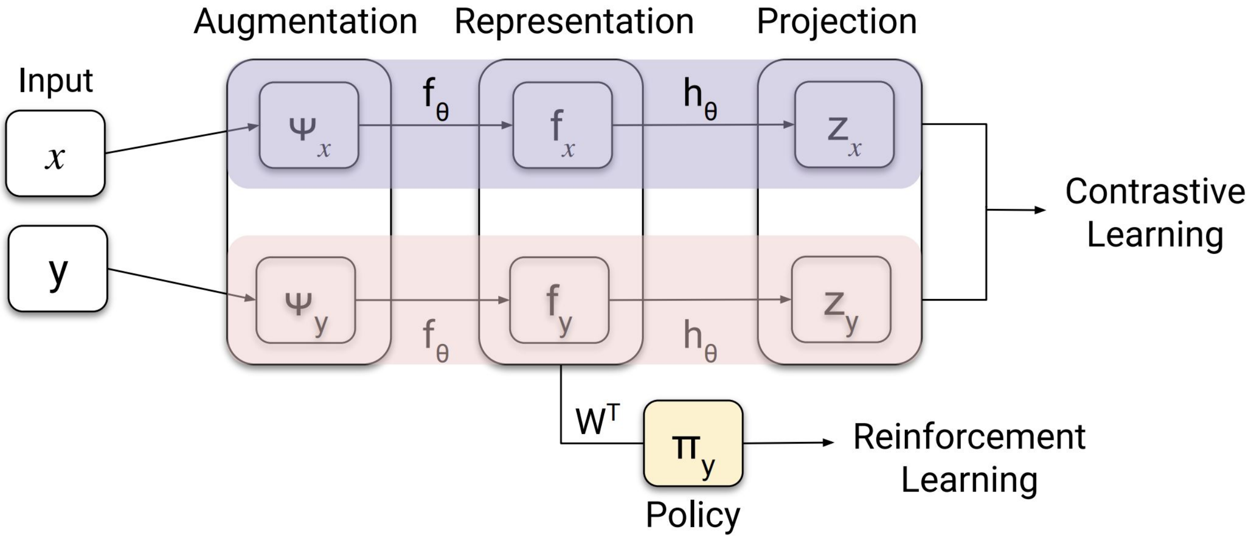
Google Research

Representations that encode PSM

- To achieve good generalization, we learn **policy similarity embeddings** (PSEs) that encode PSM
- We adapt **SimCLR**[1], a popular contrastive method for learning embeddings of image inputs.

1. Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." *ICML* (2020).

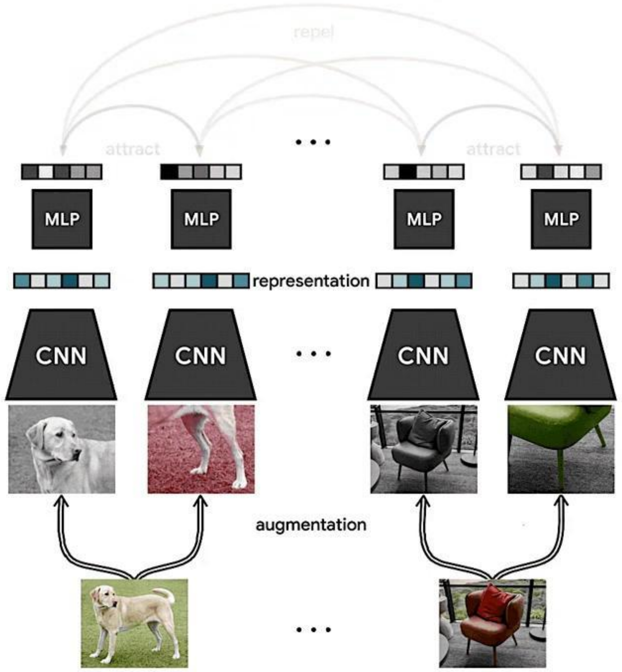
Policy Similarity Embeddings (PSEs)



Learn representations that put together states in which the agent's long-term optimal behavior is similar.

A quick summary of SimCLR

Google Research



Maximize "positive pair" similarity

$$-\log \frac{\exp(\lambda s_\theta(x, y))}{\exp(\lambda s_\theta(x, y)) + \sum_{x' \in \mathcal{X}' \setminus \{x\}} \exp(\lambda s_\theta(x', y))}$$

Minimize "negative pair" similarity

Contrastive Metric Embeddings (CMEs)

Google Research

Nearest Neighbor

$$\boxed{\tilde{x}_y = \arg\min_{x \in \mathcal{X}} d^*(x, y)}$$

Maximize "positive pair" similarity

$$\ell_\theta(\tilde{x}_y, y; \mathcal{X}') = -\log \frac{\Gamma(\tilde{x}_y, y)\exp(\lambda s_\theta(\tilde{x}_y, y))}{\Gamma(\tilde{x}_y, y)\exp(\lambda s_\theta(\tilde{x}_y, y)) + \sum_{x' \in \mathcal{X}' \setminus \{\tilde{x}_y\}} (1 - \Gamma(x', y))\exp(\lambda s_\theta(x', y))}$$

Minimize "negative pair" similarity

Google Research

Contrastive Metric Embeddings (CMEs)

Nearest Neighbor

$$\boxed{\tilde{x}_y = \arg\min_{x \in \mathcal{X}} d^*(x, y)}$$

Maximize "positive pair" similarity

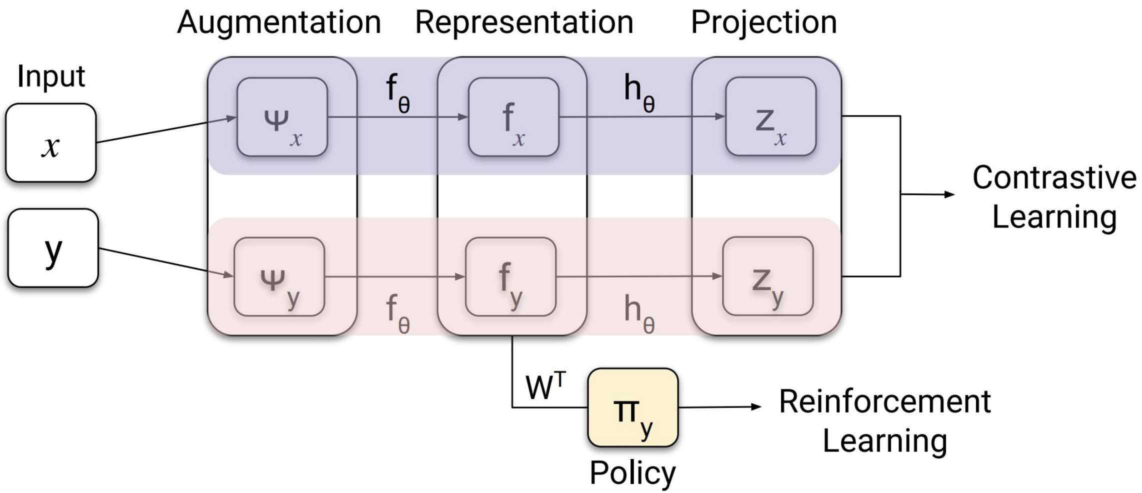
$$\ell_\theta(\tilde{x}_y, y; \mathcal{X}') = -\log \frac{\Gamma(\tilde{x}_y, y)\exp(\lambda s_\theta(\tilde{x}_y, y))}{\underbrace{\Gamma(\tilde{x}_y, y)}_{\text{Soft Similarity Score}}\exp(\lambda s_\theta(\tilde{x}_y, y)) + \sum_{x' \in \mathcal{X}' \setminus \{\tilde{x}_y\}} (1 - \Gamma(x', y))\exp(\lambda s_\theta(x', y))}$$

Minimize "negative pair" similarity

Soft Similarity Score

$$\boxed{\Gamma(x, y) = \exp(-d(x, y)/\beta)}$$

Policy Similarity Embeddings (PSEs)



Policy Similarity Embeddings = Policy Similarity Metric + CMEs

Jumping Task from Pixels: A Case Study

Combes, Remi Tachet des, Philip Bachman, and Harm van Seijen. "Learning Invariances for Policy Generalization." *arXiv preprint arXiv:1809.02591* (2018).

Jumping Task from Pixels [des Combes et al, 2018]

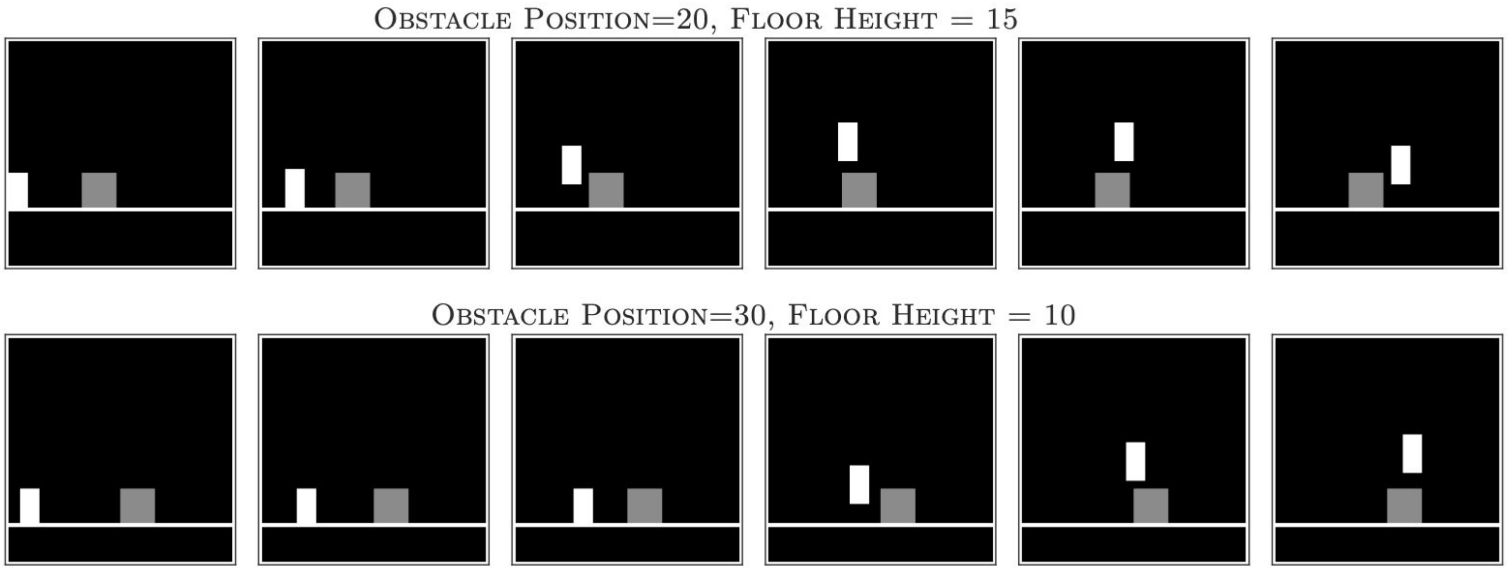
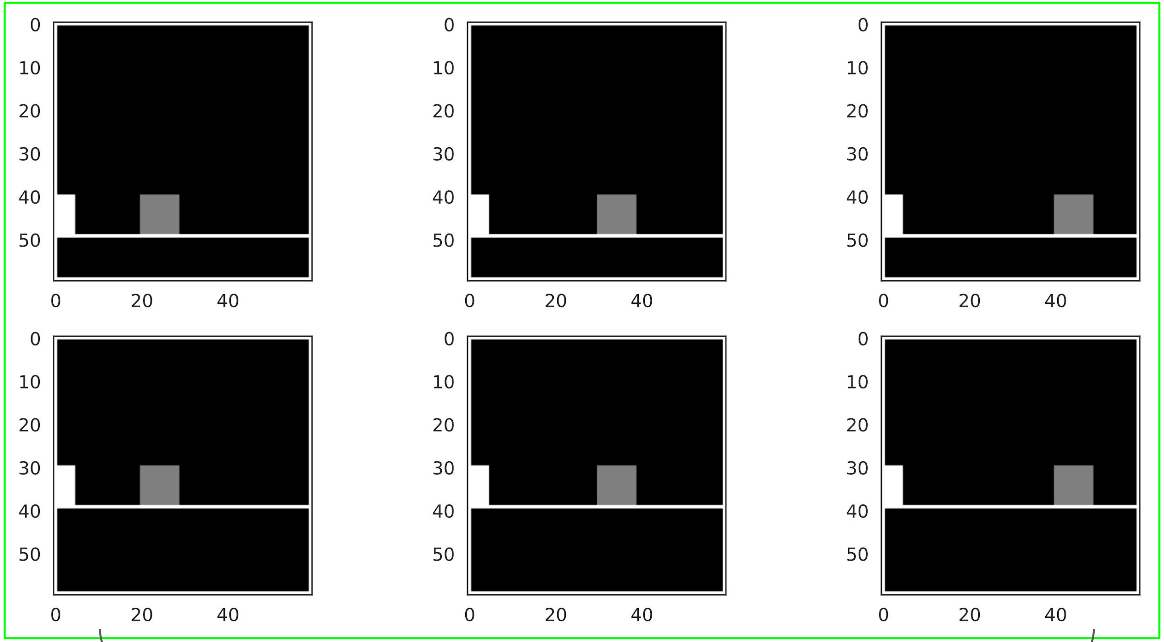


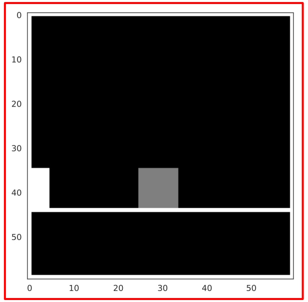
Figure G.1: Optimal trajectories on the jumping tasks for two different environments. Note that the optimal trajectory is a sequence of *right* actions, followed by a single *jump* at a certain distance from the obstacle, followed by *right* actions.

Jumping Task from Pixels [des Combes et al, 2018]

Different floor heights (train)



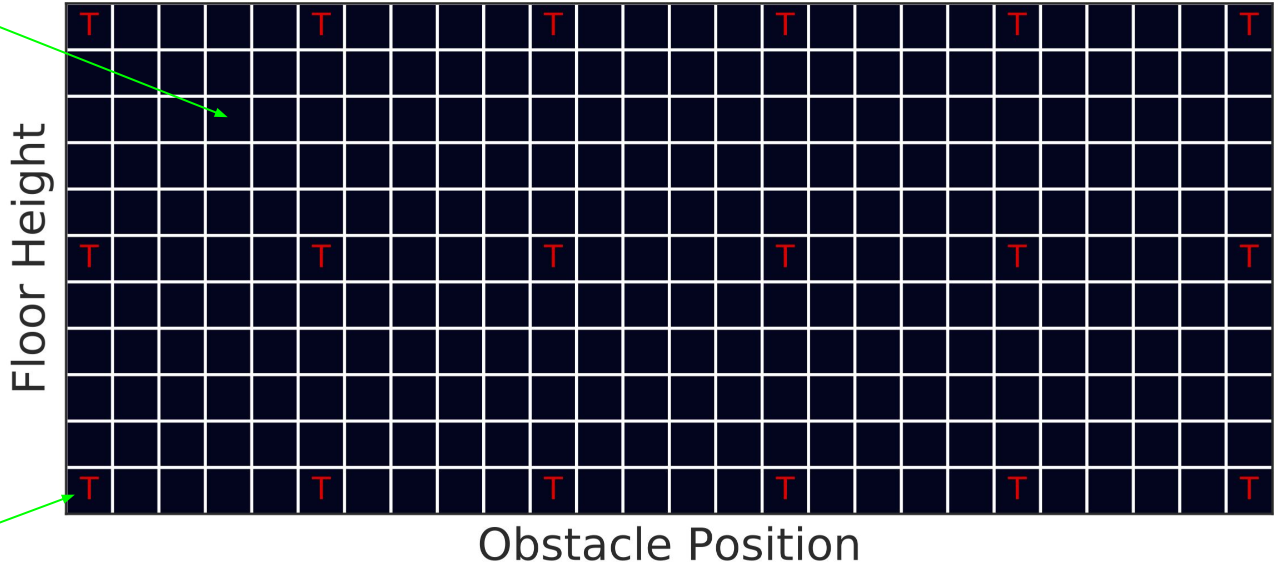
Different obstacle positions (train)



Unseen (test)

Experiment Setup

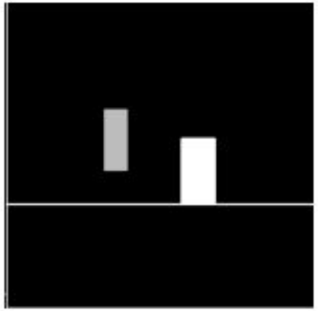
Each cell is a different jumping task.



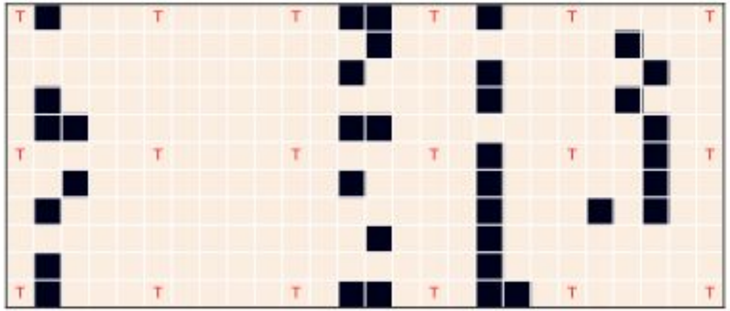
Training Task

“Wide” Grid

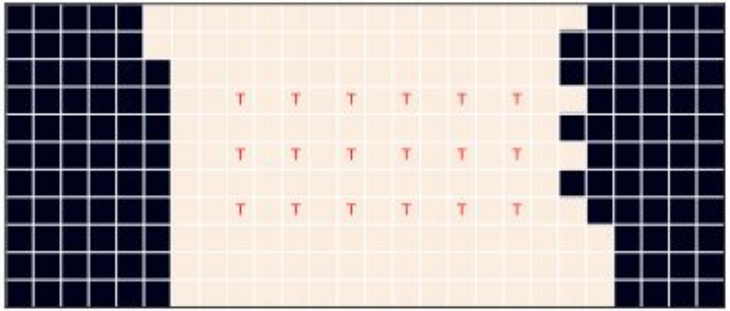
Grid Configurations



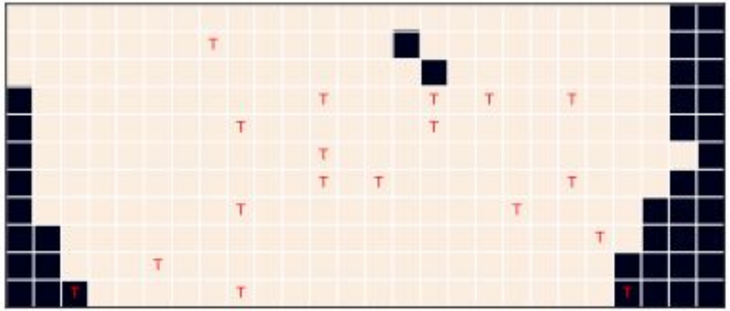
(a) Jumping task



(b) “Wide” grid



(c) “Narrow” grid



(d) Random grid

Generalization on Jumping Task without Data Augmentation

Google Research

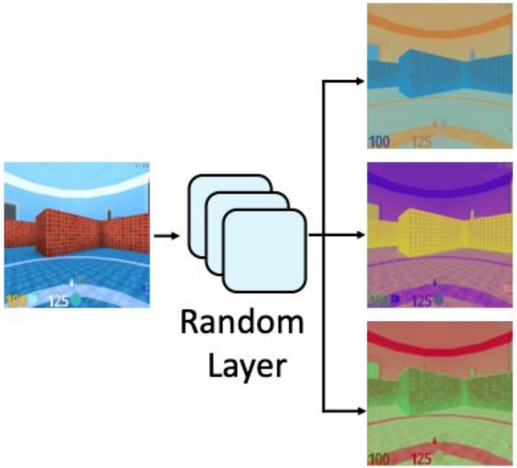
% of test environments solved (average over 100 seeds)

Data Augmentation	Method	Grid Configuration (%)		
		"Wide"	"Narrow"	Random
✗	Dropout and ℓ_2 reg.	17.8 (2.2)	10.2 (4.6)	9.3 (5.4)
	Bisimulation Transfer[4]	17.9 (0.0)	**17.9** (0.0)	30.9 (4.2)
	PSEs	**33.6** (10.0)	9.3 (5.3)	**37.7** (10.4)

4. No learning. Oracle access = Impractical!

What about Data Augmentation?

Google Research



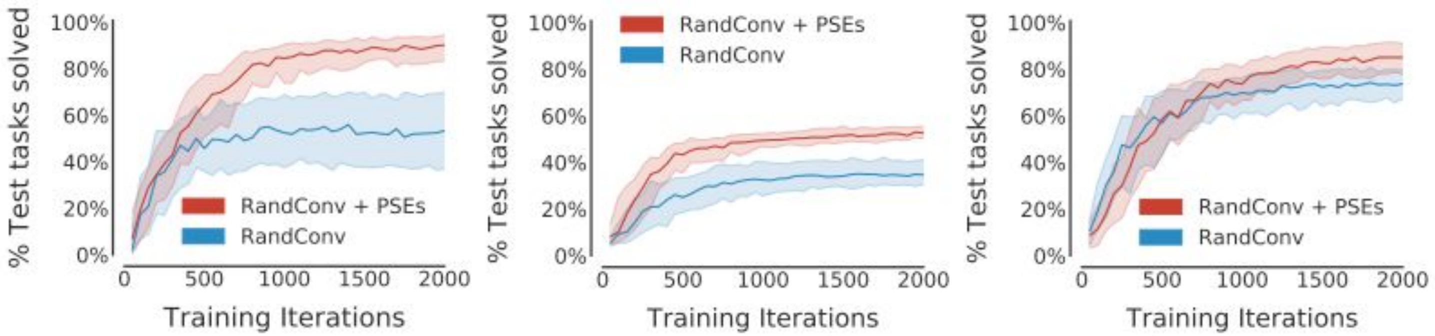
RandConv

A SOTA augmentation for generalization in RL

1. Lee, Kimin, et al. "Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning." ICLR. 2019

Google Research

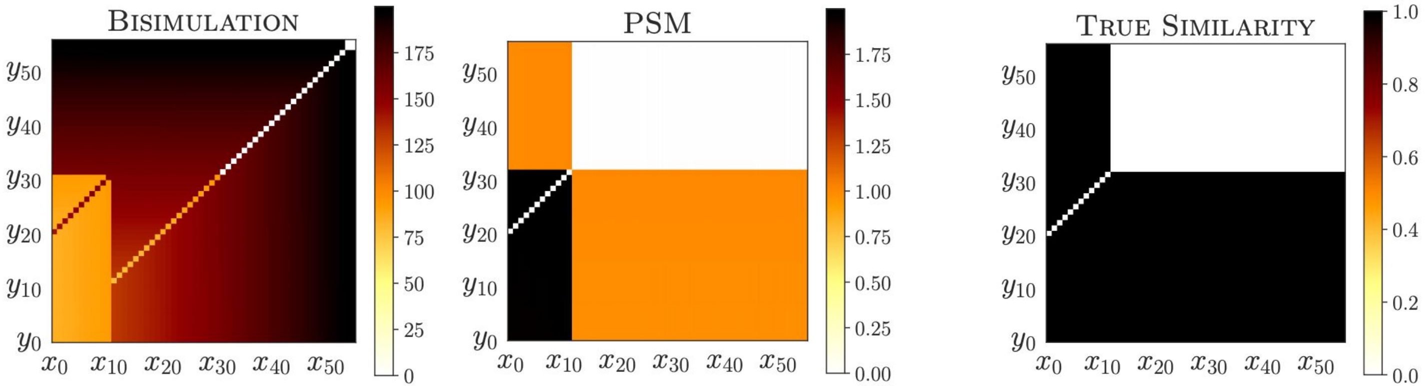
Generalization with Data Augmentation



% of test environments solved (average over 100 seeds)

Data Augmentation	Method	Grid Configuration (%)		
		"Wide"	"Narrow"	Random
✓	RandConv	50.7 (24.2)	33.7 (11.8)	71.3 (15.6)
	RandConv + Bisimulation	41.4 (17.6)	17.4 (6.7)	33.4 (15.6)
	RandConv + PSEs	**87.0** (10.1)	**52.4** (5.8)	**83.4** (10.1)

Visualizing Similarity Metrics on Jumping Task

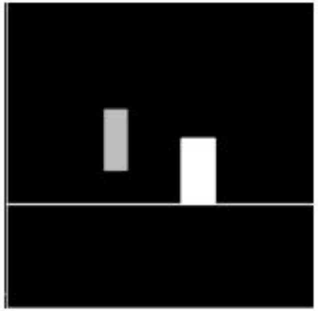


(a) Bisimulation metric

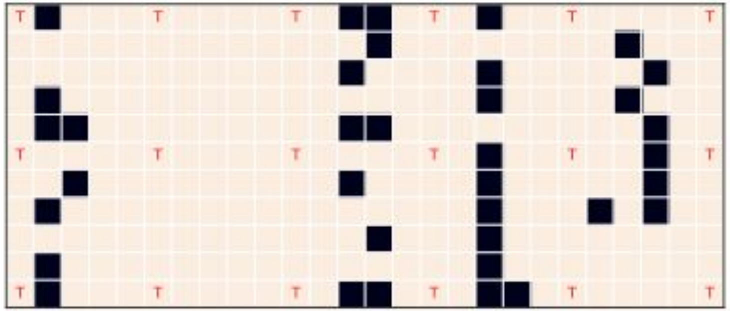
(b) Policy Similarity metric

(d) True State Similarity

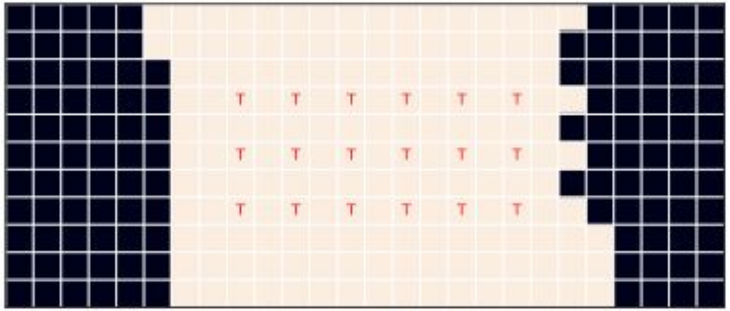
What does the generalization looks like?



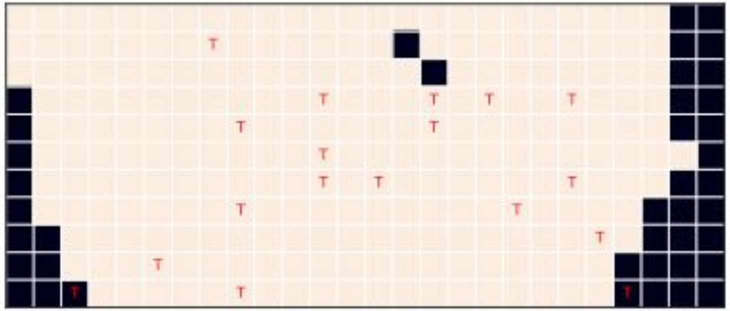
(a) Jumping task



(b) “Wide” grid

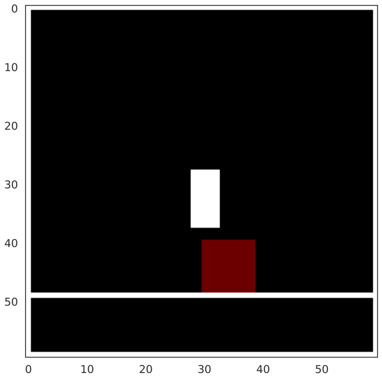


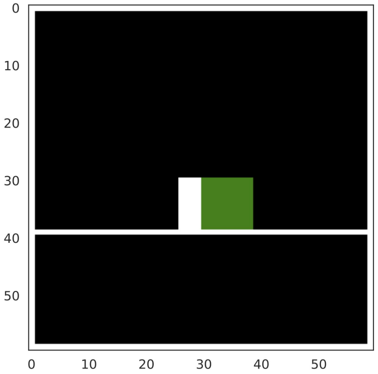
(c) “Narrow” grid



(d) Random grid

Task Dependent Invariances: Jumping Task with Colors



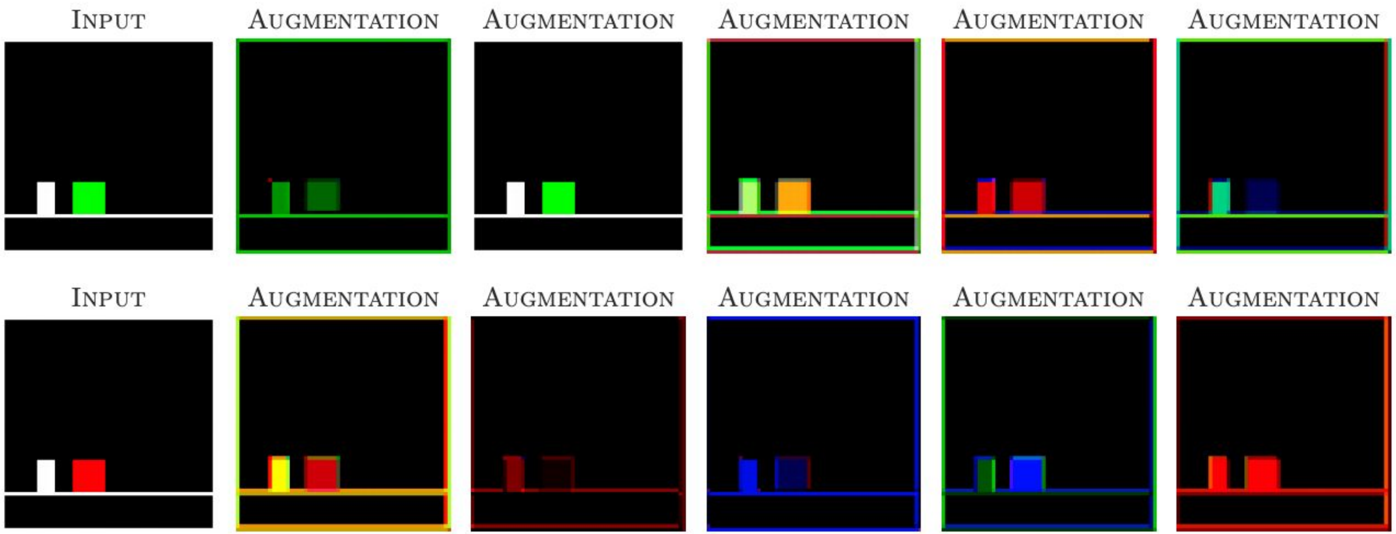


Jump over red

Strike green

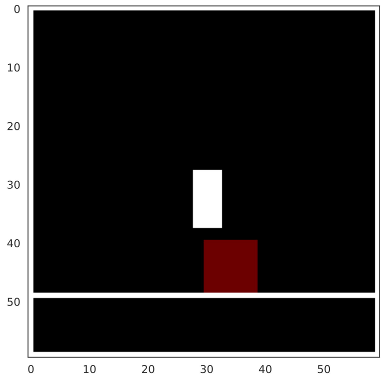
Color-dependent optimal policy.

Task Dependent Invariances: Jumping Task with Colors

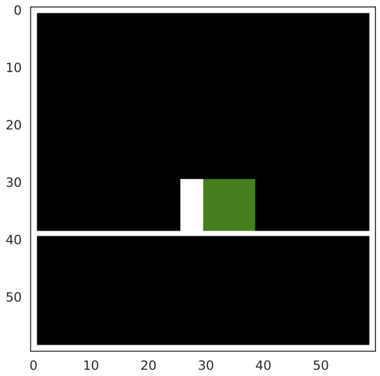


RandConv enforces color invariance.

Task Dependent Invariances: Jumping Task with Colors

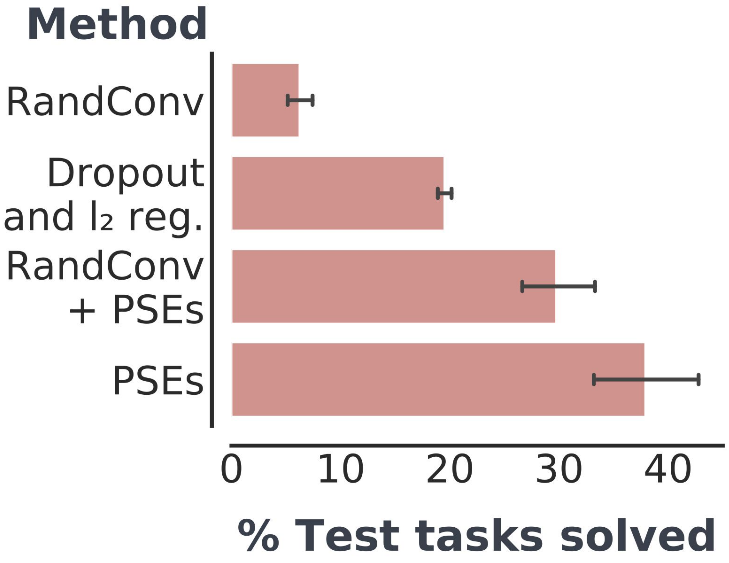


Jump over red



Strike green

Color-dependent optimal policy.



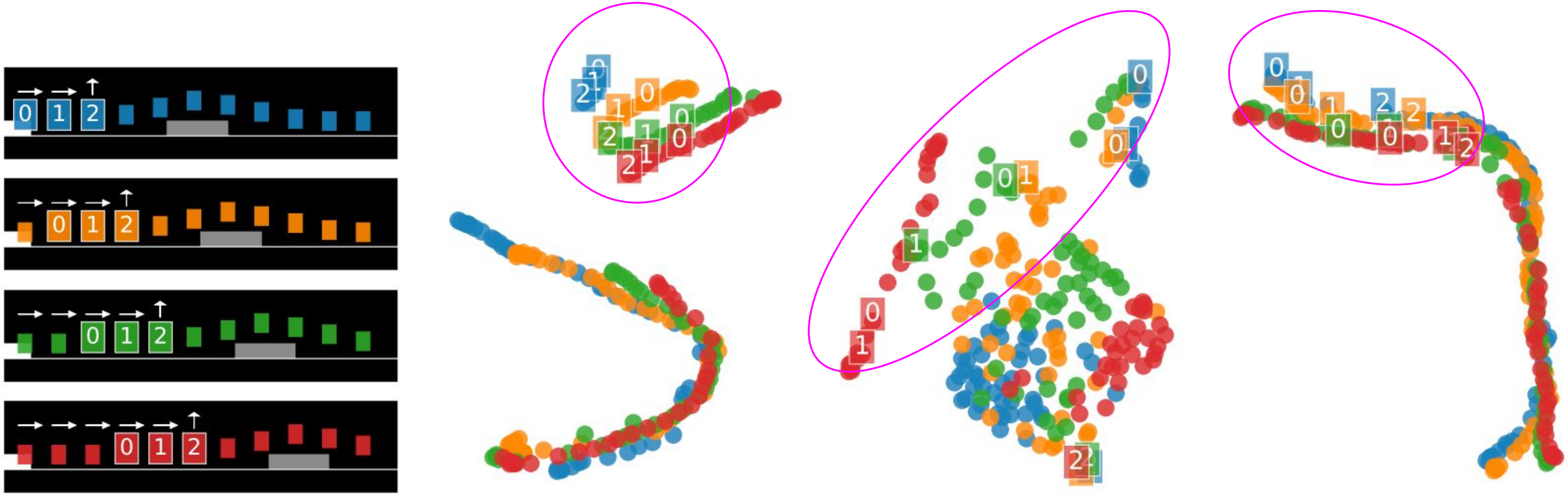
Understanding gains from PSEs

CMEs = Contrastive Metric Embeddings
PSEs = CMEs + Policy Similarity Metric

Metric / Embedding	ℓ_2-embeddings	CMEs
π^*-bisimulation	41.4 (17.6)	23.1 (7.6)
PSM	17.5 (8.4)	**87.0** (10.1)

ℓ_2-embeddings (Zhang et al., 2020)
Minimize l2-distance b/w representations to match the metric d

Visualizing learned representations

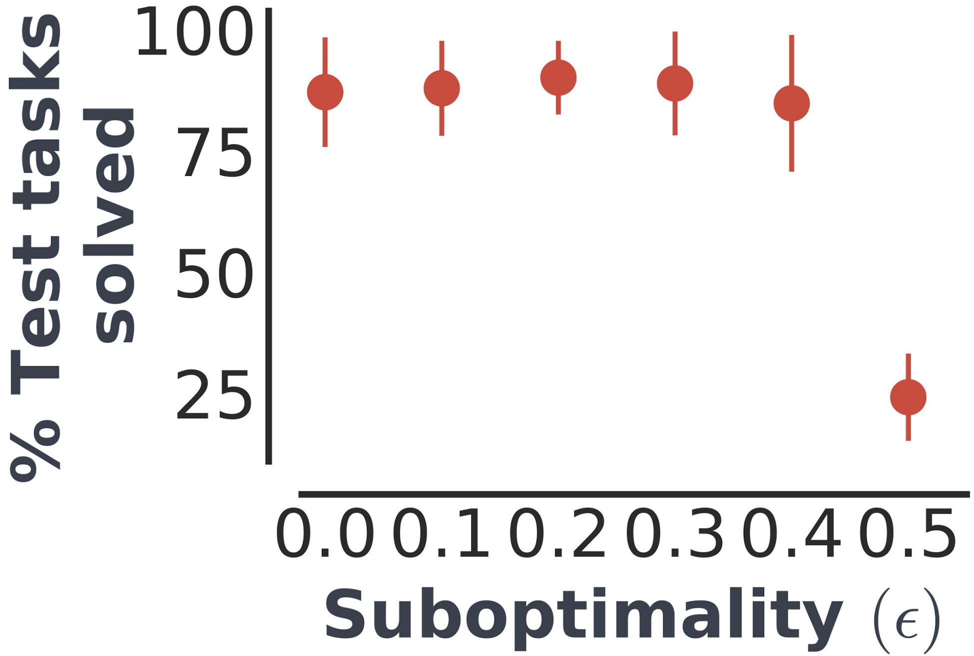


(a) Optimal Trajectories (b) PSEs (c) PSM + ℓ_2 embeddings (d) π^*-bisim. + CMEs

PSEs are robust to suboptimality!

Google Research

Take optimal action with probability 1 - ϵ.



Ablations: Understanding gains from PSEs

Compare Embeddings

Metric / Embedding	ℓ_2-embeddings	CMEs
π^*-bisimulation	5.1 (10.0)	23.1 (7.6)
PSM	17.5 (8.4)	**87.0** (10.1)

ℓ2-embeddings (Zhang et al., 2020)

Minimize l2-distance b/w representations to match the metric d

LQR with Spurious Correlations [Song et al, 2020]

$$\begin{aligned} \text{minimize} \quad & E_{s_0 \sim \mathcal{D}} \left[\tfrac{1}{2} \sum_{t=0}^{\infty} s_t^T Q s_t + a_t^T R a_t \right], \\ \text{subject to} \quad & s_{t+1} = A s_t + B a_t, o_t = \begin{bmatrix} 0.1\, W_c \\ W_d \end{bmatrix} s_t, a_t = K o_t, \end{aligned}$$

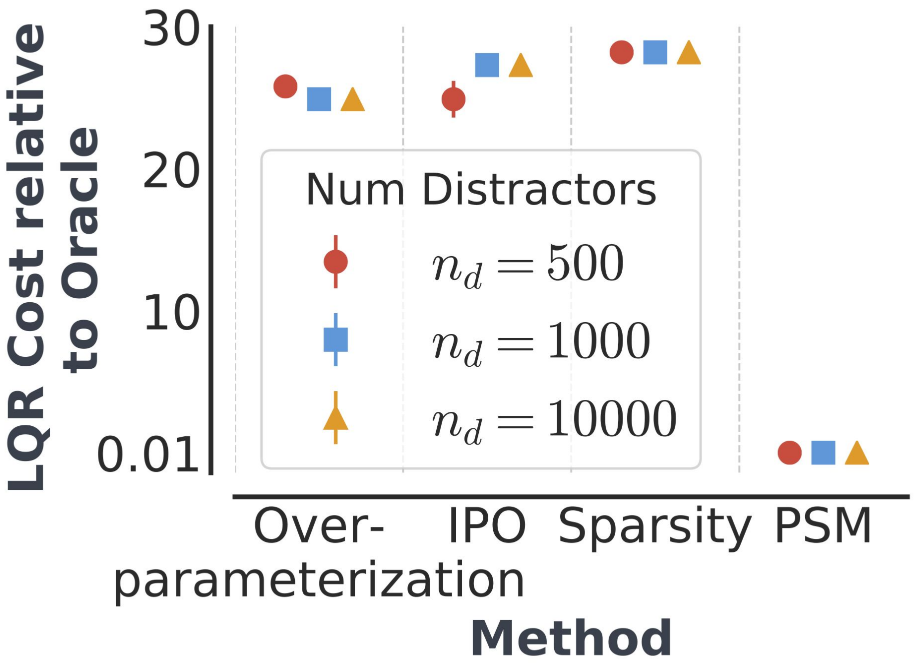
$$o_t = \begin{bmatrix} 0.1 \; W_c \\ W_d \end{bmatrix} s_t$$

W_d is domain dependent.

High-Dimensional Spurious Distractors

Song, X., Jiang, Y., Du, Y., & Neyshabur, B. (2019). Observational overfitting in reinforcement learning. *ICLR (2020).*

LQR with Spurious Correlations

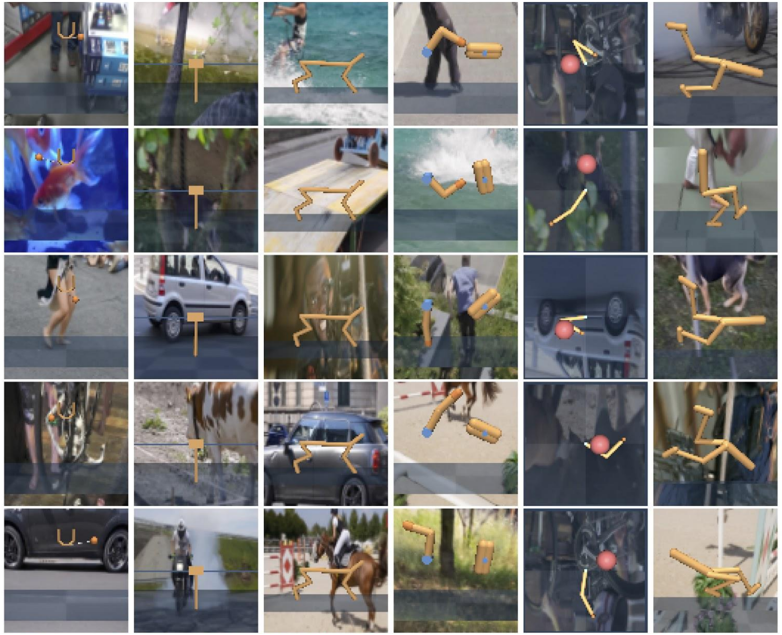


*IPO = Invariant Risk Minimization + PPO

Distracting DM Control

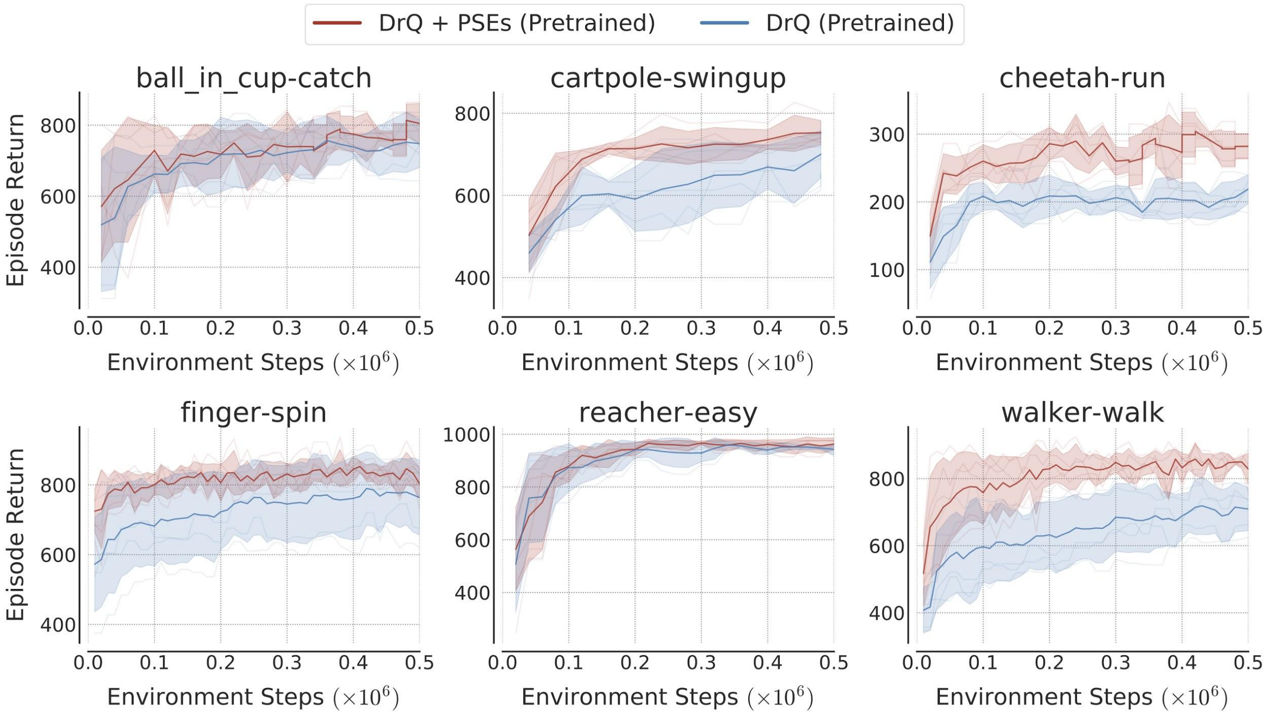
Google Research

Train Environments



Test Environments

Distracting DM Control



PSEs outperform SOTA data augmentation DrQ agent!

- Human RL literature typically thinks of state spaces structured around rewards rather than actions.

- This work shows that we expect policies to transfer rather than reward!

- **Should states be grouped by invariance to rewards or actions?**

1. Niv, Yael. "Learning task-state representations." *Nature neuroscience* 22.10 (2019): 1544-1553.
2. Gershman, Samuel J. "The successor representation: its computational logic and neural substrates." *Journal of Neuroscience* 38.33 (2018): 7193-7200.

agarwl.github.io/pse for details!

Thank You!