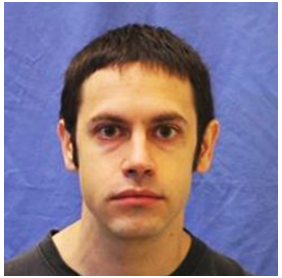
Fast Adaptation via Policy-Dynamics Value Functions

Roberta Raileanu
NYU

Max Goldstein
NYU



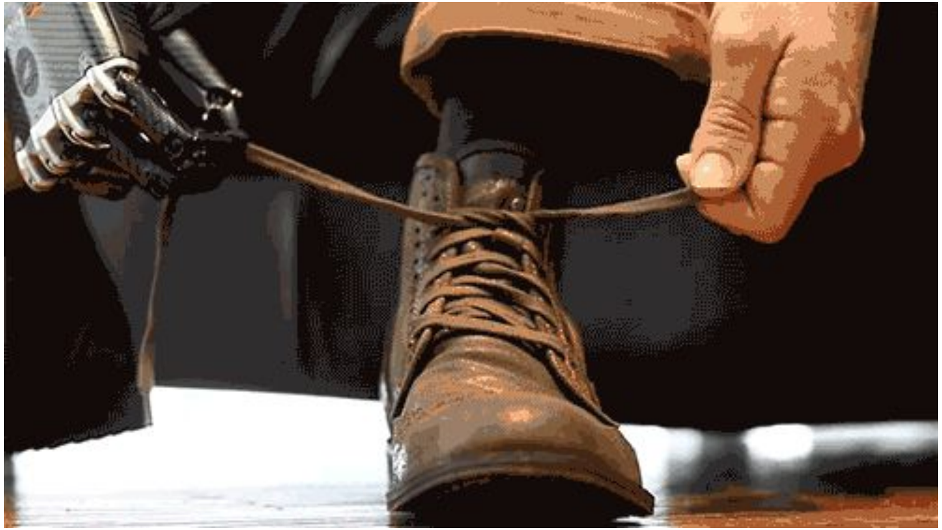
Arthur Szlam
FAIR

Rob Fergus
NYU

NYU

ICML 2020

Dynamics Often Change in the Real World



How can agents rapidly **adapt**
to changes in the environment's **dynamics**?

Learn a **General Value Function** in the
Space of Policies and Dynamics

Policy-Dynamics Value Function (PD-VF)

Value Function → Total Future Reward → Fixed $\pi, \mathcal{T}$

$$V^{\pi}(s) = \mathbb{E}\left[R_t \mid S_t = s, A_t \sim \pi, S_{t+1} \sim \mathcal{T}\right]$$

Policy-Dynamics Value Function → Total Future Reward → $\pi \in \Pi, \mathcal{T}_d \in \mathcal{T}_{\mathcal{D}}$

$$W(s, \pi, d) = \mathbb{E}\left[R_t \mid S_t = s, A_t \sim \pi, S_{t+1} \sim \mathcal{T}_d\right]$$

Fast Adaptation to New Dynamics

Family of Environments $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$

Each Environment has a Different Transition Function $\mathcal{T}_d(s'|s,a) \in \mathcal{T}$ $\quad d$ unobserved

Train on a Family of Different but Related Dynamics $d \sim \mathcal{D}_{train}$

Test on New Dynamics $d \sim \mathcal{D}_{test} \quad \mathcal{D}_{test} \neq \mathcal{D}_{train}$

Training Recipe

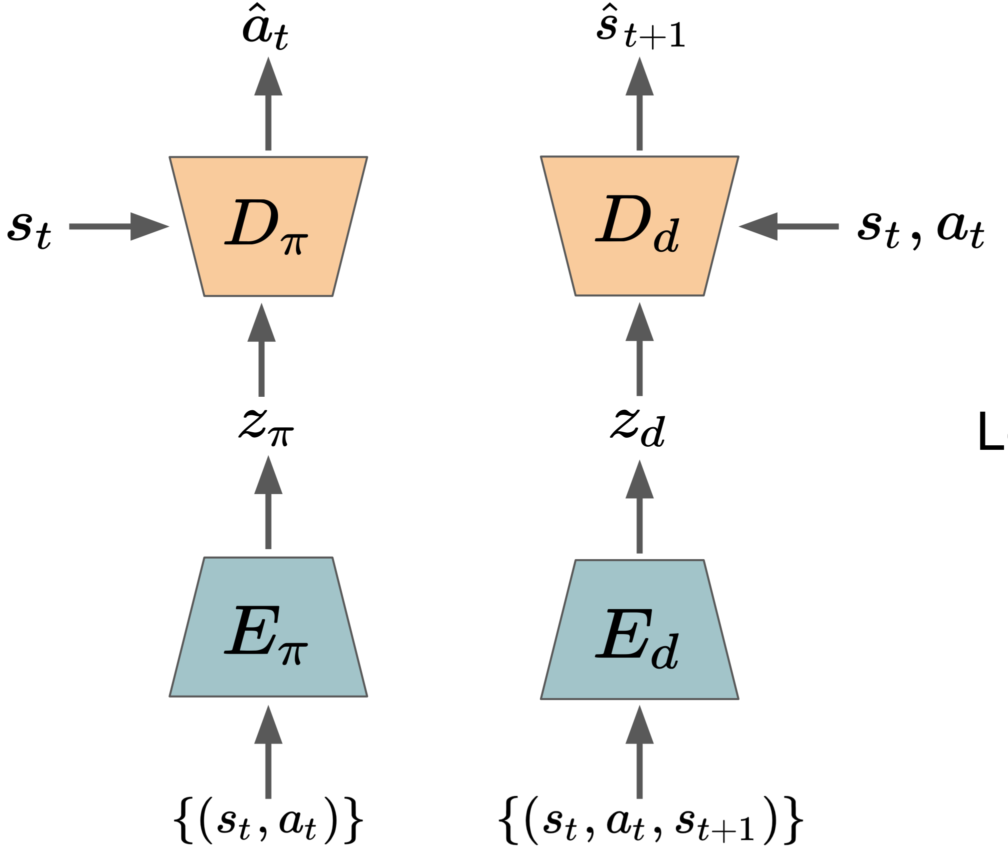
1. Reinforcement Learning Phase
   - train individual policies on each training environment

2. Self-Supervised Learning Phase
   - Learn policy and dynamics embeddings using collected the trajectories

3. Supervised Learning Phase
   - Learn a value function for this space of policies and environments

4. Evaluation Phase
   - Infer the dynamics of a new environment using ≤ 4 steps
   - Find the policy that maximizes the learned value function

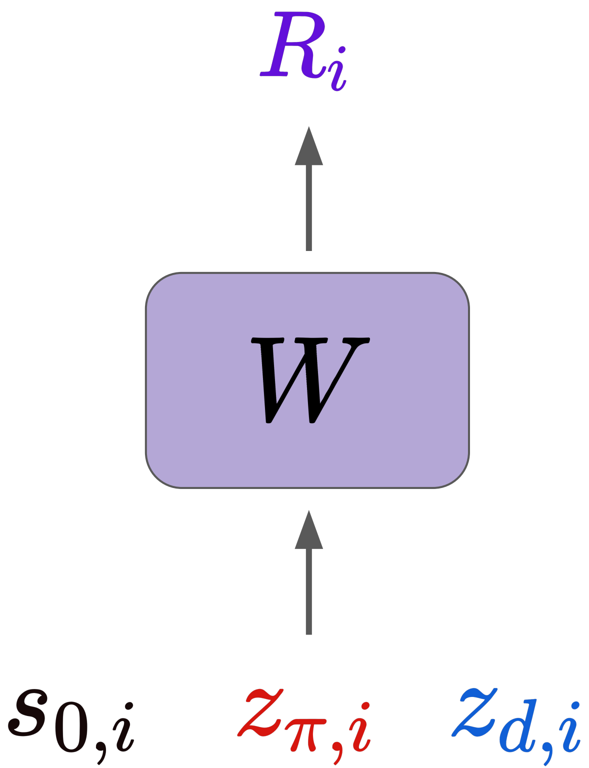
Learning Policy and Dynamics Embeddings



Learn Policy Embedding z_π

Learn Dynamics Embedding z_d

Learning the Policy-Dynamics Value Function



Training the Policy-Dynamics Value Function

$$R = W(s_0, z_\pi, z_d; \psi)$$

Evaluation Phase

$$z_\pi^\star = argmax_{z_\pi} W(s_0, z_\pi, z_d)$$

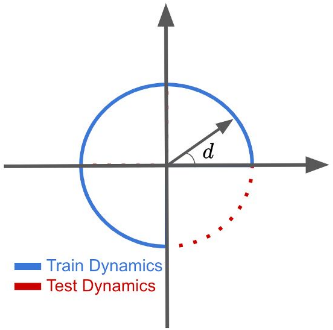
$$W(s_0, z_\pi, z_d) = z_\pi^T A(s_0, z_d; \psi)\, z_\pi$$

Closed-form solution: **top singular vector** of A’s SVD decomposition

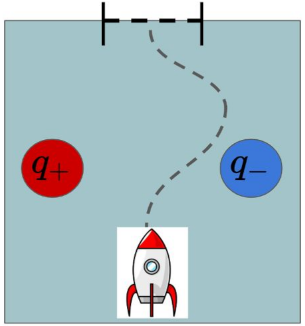
$z_\pi^\star$ **Optimal Policy Embedding (OPE)**

Environments

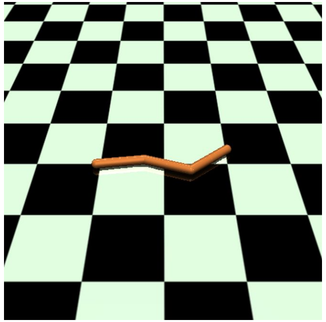
Continuous Dynamics



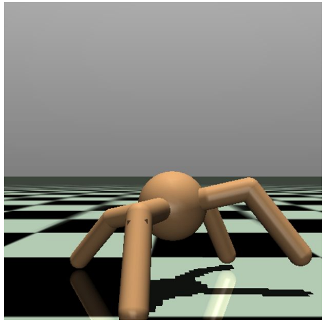
Spaceship



Swimmer

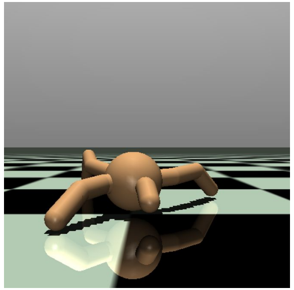


Ant-Wind

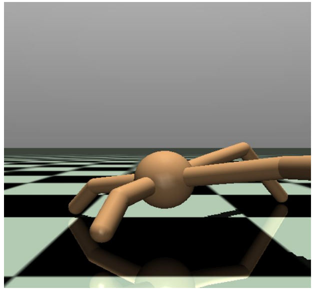


Discrete Dynamics

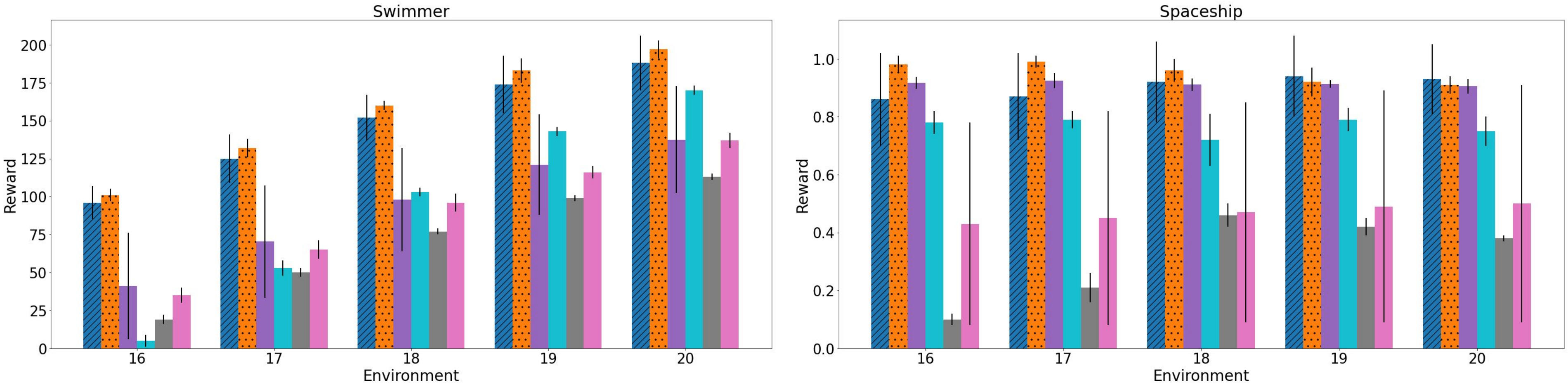
Ant-Legs

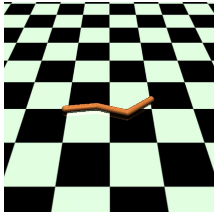


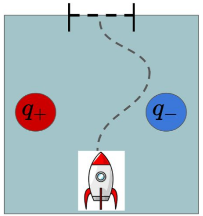
Ant-Legs



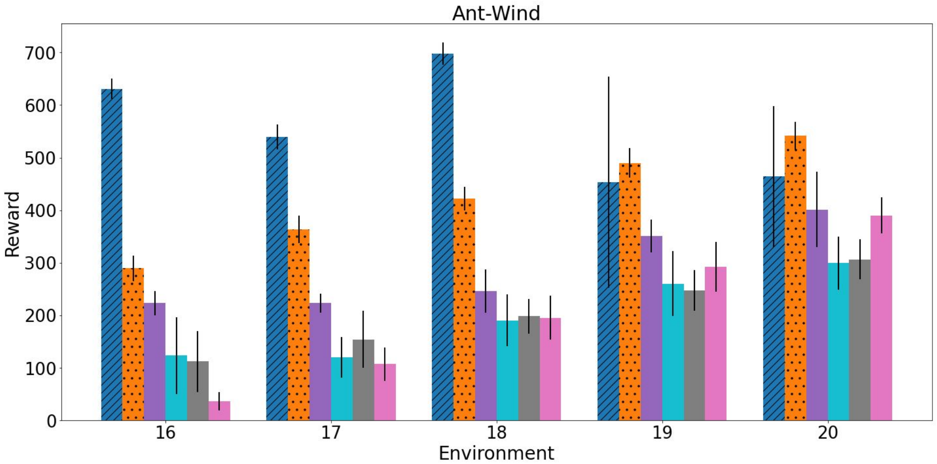
Evaluation on Unseen Environments

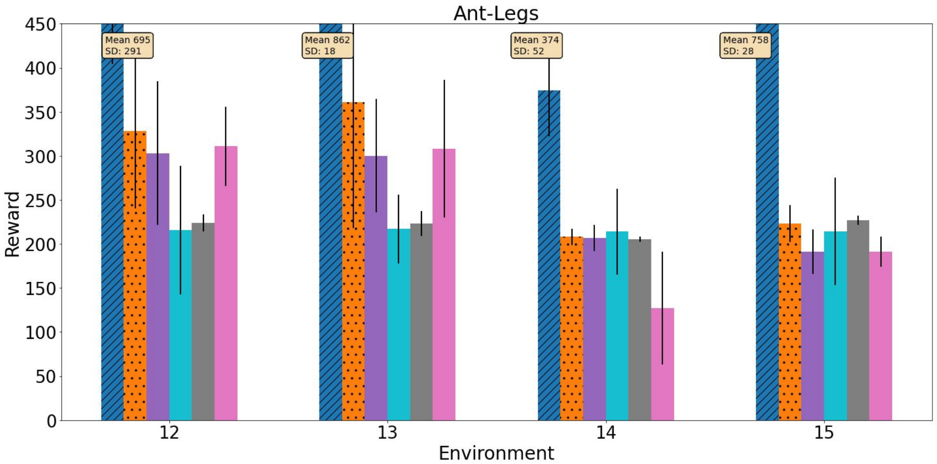


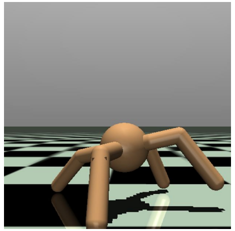


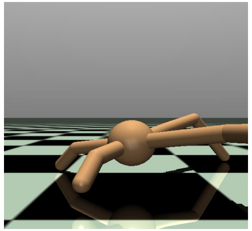


Evaluation on Unseen Environments







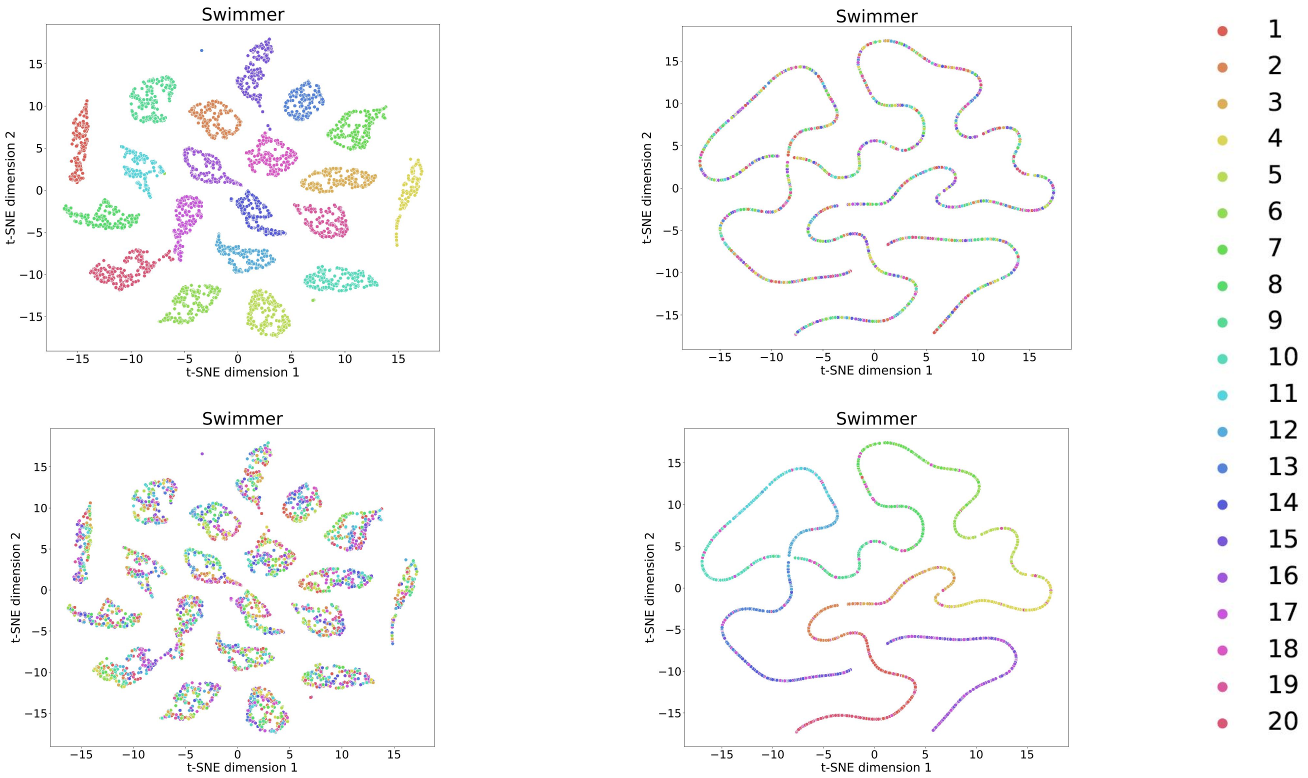


Learned Embeddings

Policy Embeddings | **Dynamics Embeddings**

Policy Color

Dynamics Color



Takeaways

Learn a value function in a space of policies and dynamics

Infer the dynamics of a new environment from only a few interactions

No need for parameter updates, long rollouts, or dense rewards to adapt

Improved performance on unseen environments

Future Work

- Reward function variation → condition W on a task embedding
- Multi-agent settings → dynamics given by the others' policies
- Continual learning
- Integrate prior knowledge / constraints
- Estimate other metrics apart from reward

Thank you!