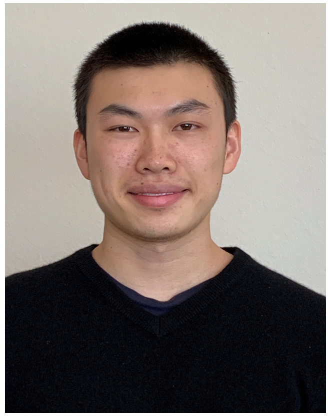
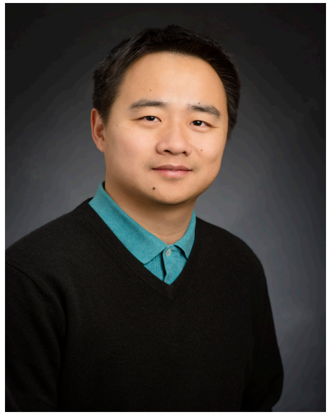
Mutual Information Based Knowledge Transfer Under State-Action Dimension Mismatch


Michael Wan


Tanmay Gangwani

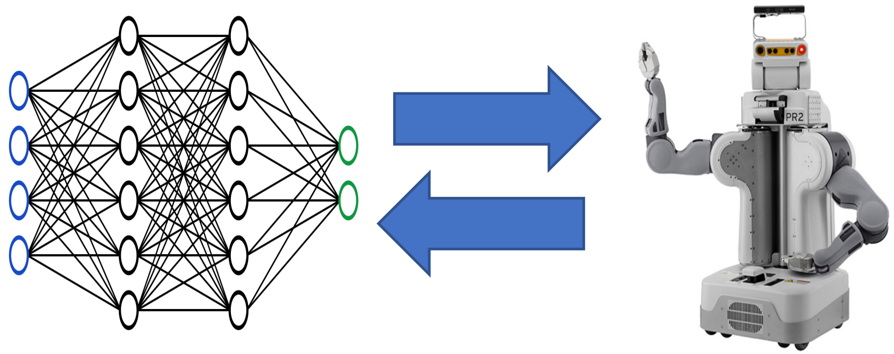

Jian Peng

UAI 2020

Motivation

- Reinforcement Learning is sample-inefficient: often millions of samples required to learn a good policy

- Environment interaction may be expensive
  - True of real-world robotics. Risk of causing damage to hardware/surroundings



Policy

Environment

Transfer Learning in RL

- Assume access to a pre-trained teacher policy
  - Trained in **source MDP** with state space $\boldsymbol{S_{src}}$, action space $\boldsymbol{A_{src}}$
  - Teacher policy network $\pi_{\theta'}$, value network $V_{\psi'}$

- Train student policy in **target MDP** with state space $\boldsymbol{S_{targ}}$, action space $\boldsymbol{A_{targ}}$
  - Student policy network π_{θ}, value network V_{ψ}

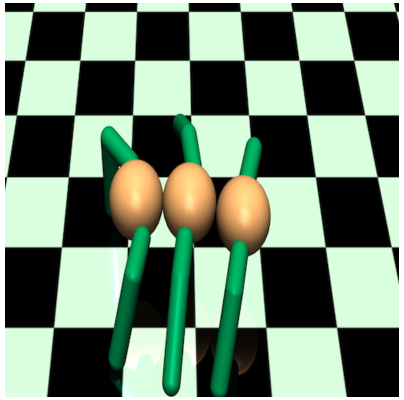
- Ideally, teacher should help accelerate the student learning

Transfer Learning – MDP Mismatch

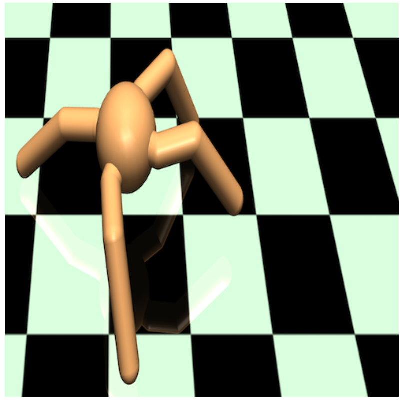
- Prior work has considered setting where $S_{src} = S_{targ}$, $A_{src} = A_{targ}$
- What if $S_{src} \neq S_{targ}$, $A_{src} \neq A_{targ}$?

- *How to handle difference in state space?*
- *How to handle difference in action space?*

Teacher (6-legged Centipede)
State dimension : 139
Action dimension : 16



Transfer Learning



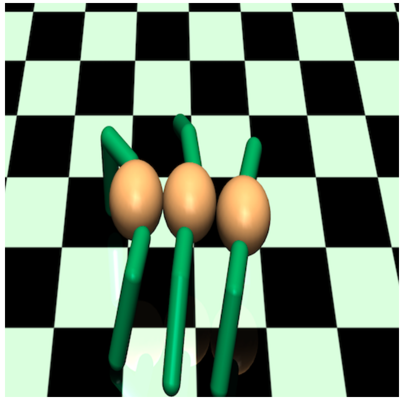
Student (4-legged Ant)
State dimension : 111
Action dimension : 8

Transfer Learning – MDP Mismatch

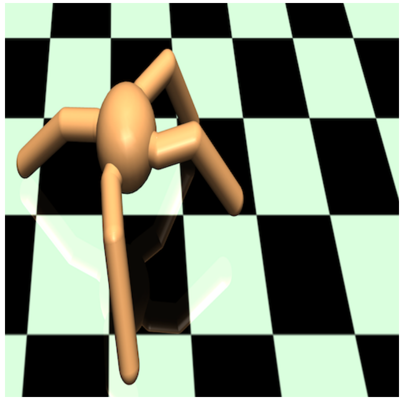
- **How to handle difference in state space?**
  - An embedding space learned through a network
  - This network acts a conduit between $\boldsymbol{S_{targ}}$ and $\boldsymbol{S_{src}}$

- How to handle difference in action space?

Teacher (6-legged Centipede)
State dimension : 139
Action dimension : 16



Transfer Learning



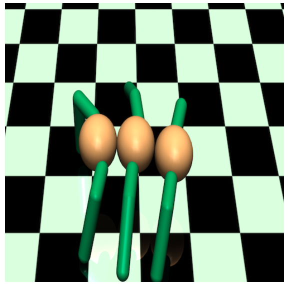
Student (4-legged Ant)
State dimension : 111
Action dimension : 8

Embedding Space

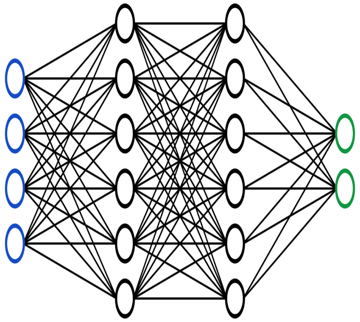
- Introduce learned embedding space
  - Parameterized by encoder function $\phi(\cdot)$
  - Defined as $S_{emb} \coloneqq \{\phi(s) \mid s \in S_{targ}\}$

- Dimension of embedding space must match state-space dimension of source MDP: $|S_{emb}| = |S_{src}|$

- Can now utilize the teacher networks for knowledge transfer!



s_{targ}



Encoder ϕ

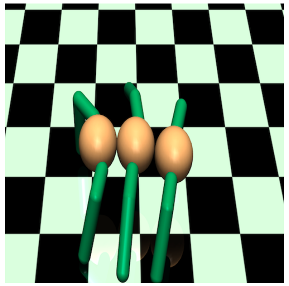
$\phi(s_{targ})$

Pass this to teacher networks to extract useful information

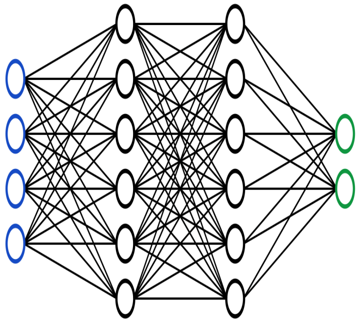
Embedding Space Desiderata

- ***Desired property 1**: Embeddings must be task aligned*
  - Embedding parameters should be updated to maximize the cumulative discounted rewards in the target MDP

- ***Desired property 2:** Embeddings must have high correlation with input states in target MDP*
  - We propose to maximize a lower bound to the mutual information (MI) between state s_{targ} and embedding $\phi(s_{targ})$



s_{targ}



Encoder ϕ

$\phi(s_{targ})$

Pass this to teacher networks to extract useful information

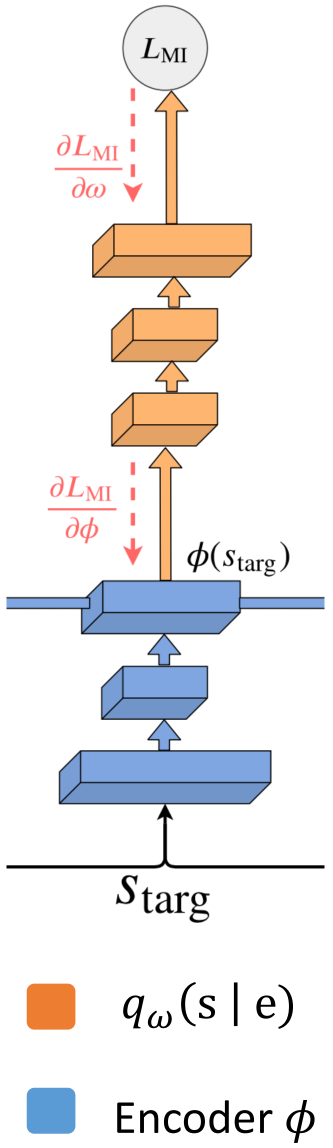
Embeddings with Mutual Information Maximization

- Lower bound to MI between state s and embedding e[1]:

$$\boldsymbol{\mathcal{H}}(\boldsymbol{s}) + \mathbb{E}_{\boldsymbol{s},\, \boldsymbol{e}}[\mathbf{log}\, \boldsymbol{q}_{\boldsymbol{\omega}}(\boldsymbol{s} \mid \boldsymbol{e})]$$

- $q_\omega(\mathrm{s} \mid \mathrm{e})$: variational distribution
  - Neural network that outputs mean of a multivariate Gaussian
  - Learned diagonal covariance matrix

- Loss: $L_{MI}(\phi, \omega) = -\mathbb{E}_{s \sim \rho_{\pi_\theta}}[\log q_\omega(s \mid \phi(s))]$
  - ρ_{π_θ}: state-visitation distribution
  - Entropy $\mathcal{H}(s)$ is constant *w.r.t.* encoder parameters ϕ and variational parameters ω, so we can omit it



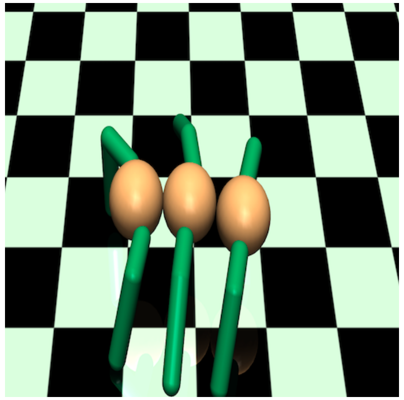
[1] Variational information maximization for neural coding, Felix Agakov and David Barber.

Transfer Learning – MDP Mismatch

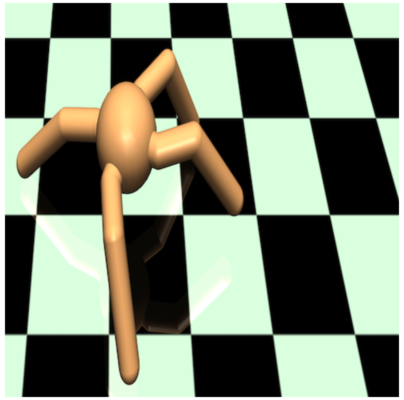
- How to handle difference in state space?
  - An embedding space learned through a network
  - This network acts a conduit between S_{targ} and S_{src}

- **How to handle difference in action space?**

Teacher (6-legged Centipede)
State dimension : 139
Action dimension : 16



Transfer Learning



Student (4-legged Ant)
State dimension : 111
Action dimension : 8

Knowledge Transfer

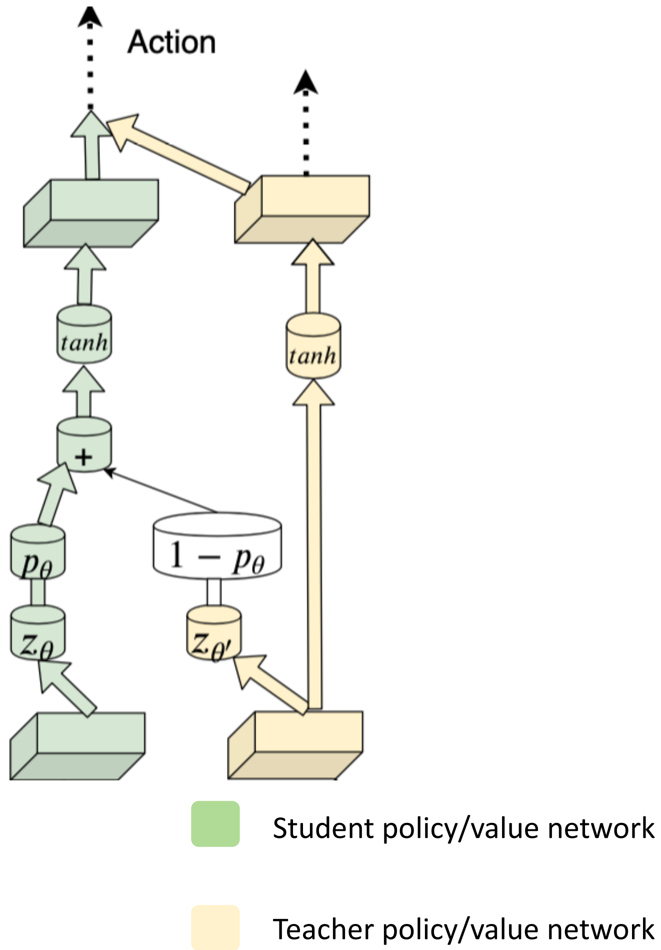
- Idea: **Augment representations of student with teacher representations**[1]

- Feed current state $s_{targ} \in S_{targ}$ into student networks, embedding $\phi(s_{targ})$ into teacher networks

- Weighted linear combination at each layer j:

$$h^j_{\pi_\theta} = \sigma(p^j_\theta z^j_\theta + (1 - p^j_\theta) z^j_{\theta'})$$
$$h^j_{V_\psi} = \sigma(p^j_\psi z^j_\psi + (1 - p^j_\psi) z^j_{\psi'})$$

<u>Notation</u>
σ: activation function
z: pre-activations
p: mixing weights

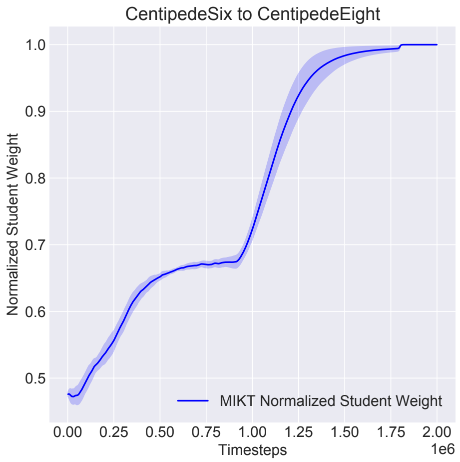


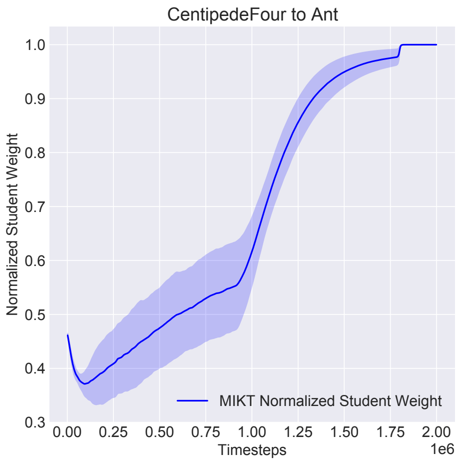
1. *Knowledge Flow: Improve Upon Your Teachers, Liu et al.*

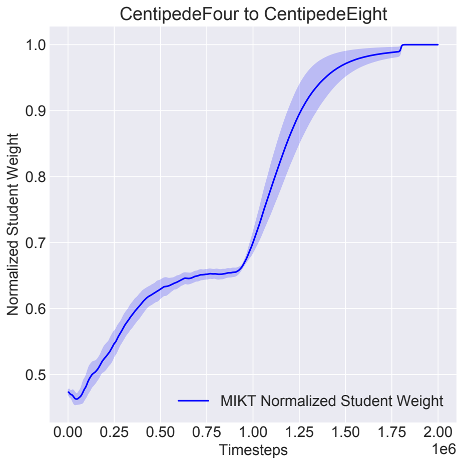
Mixing Weights

- Lower values indicate heavier dependence on teacher representations

- Encourage student to be **independent** of the teacher by end of training[1]
    - $L_{coupling} = -\frac{1}{N_\pi}\sum_{j=1}^{N_\pi}\log(p_\theta^j) - \frac{1}{N_V}\sum_{j=1}^{N_V}\log(p_V^j)$

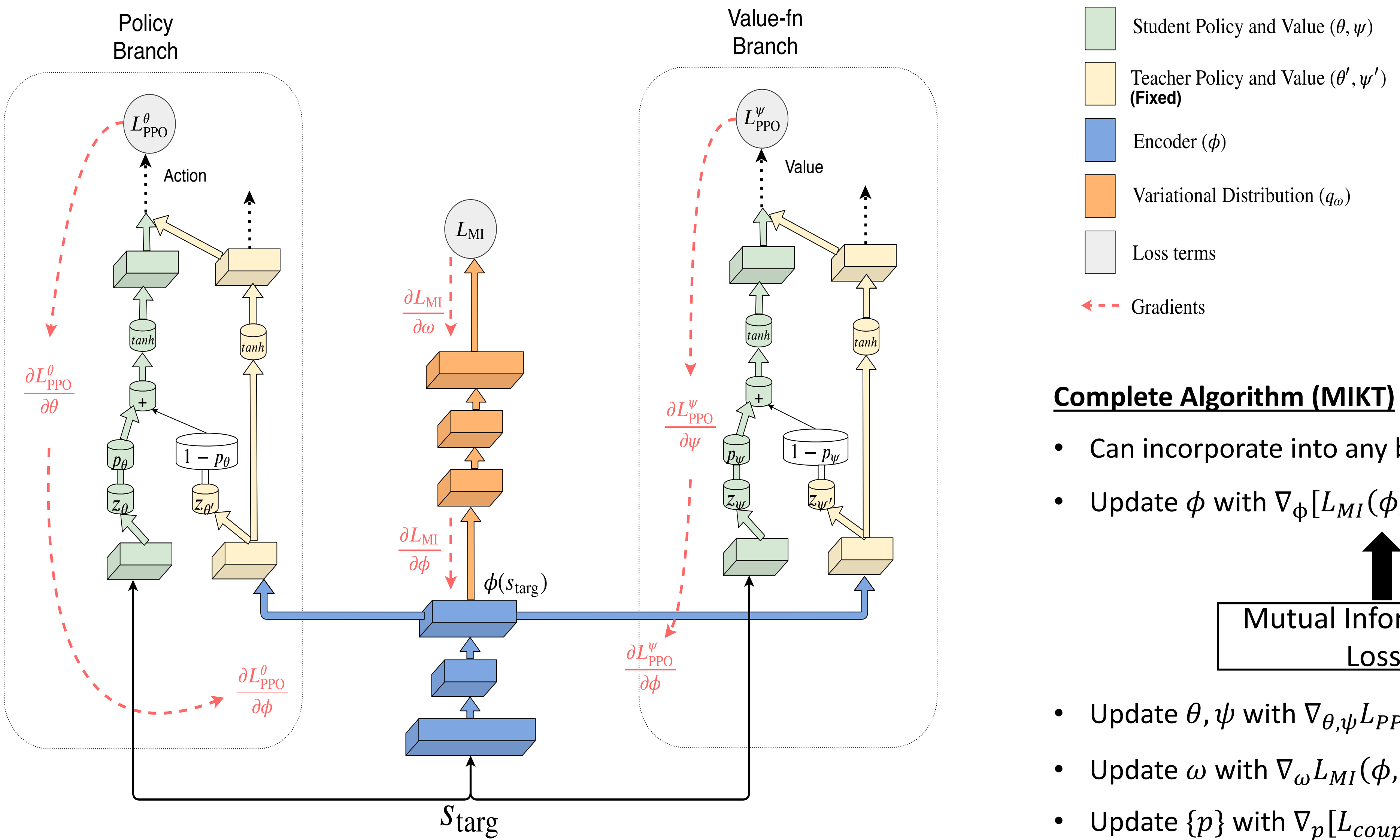






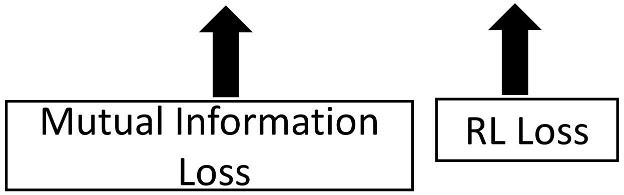
1. Knowledge Flow: Improve Upon Your Teachers, Liu et al.

Mutual Information based Knowledge Transfer



Complete Algorithm (MIKT)

- Can incorporate into any base RL algorithm; we choose PPO
- Update ϕ with $\nabla_\phi[L_{MI}(\phi,\omega) + L_{PPO}(\theta,\psi,\theta',\psi',\phi)]$

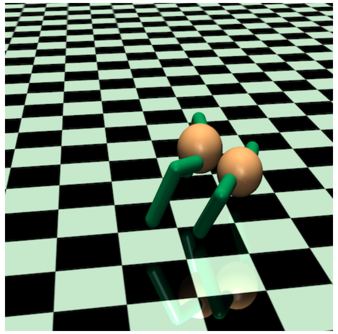
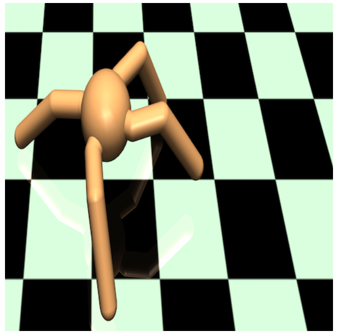
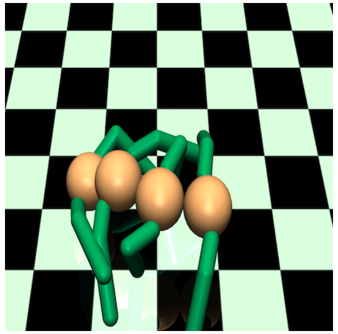
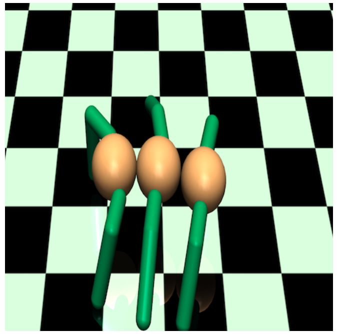


- Update θ, ψ with $\nabla_{\theta,\psi} L_{PPO}(\theta,\psi,\theta',\psi',\phi)$
- Update ω with $\nabla_\omega L_{MI}(\phi,\omega)$
- Update $\{p\}$ with $\nabla_p[L_{coupling} + L_{PPO}]$

Experimental Setup

- MuJoCo locomotion tasks (Ant, Centipede[1])
- Centipede tasks differ in number of legs and disability
  (Cp variants have some legs disabled)

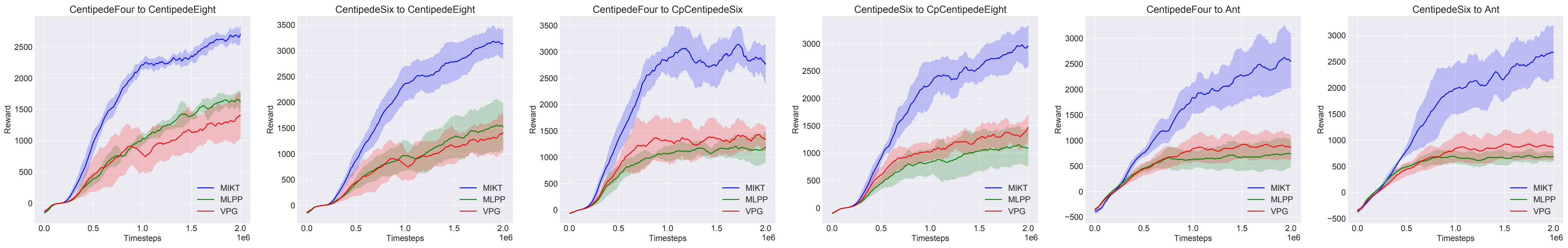
Environment	State Dimension	Action Dimension
CentipedeFour	97	10
CentipedeSix	139	16
CentipedeEight	181	22
CpCentipedeSix	139	12
CpCentipedeEight	181	18
Ant	111	8

1. NerveNet: Learning Structured Policy with Graph Neural Networks, Wang et al.

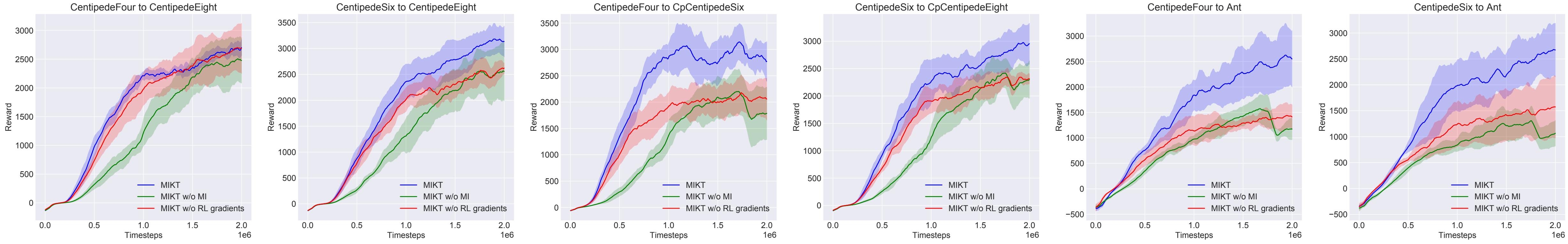
Results

- Methods
  - **MIKT (ours)**
  - VPG: PPO on target task (no transfer learning)
  - MLPP: Re-use middle layers of pre-trained network, randomly initialize input and output layers



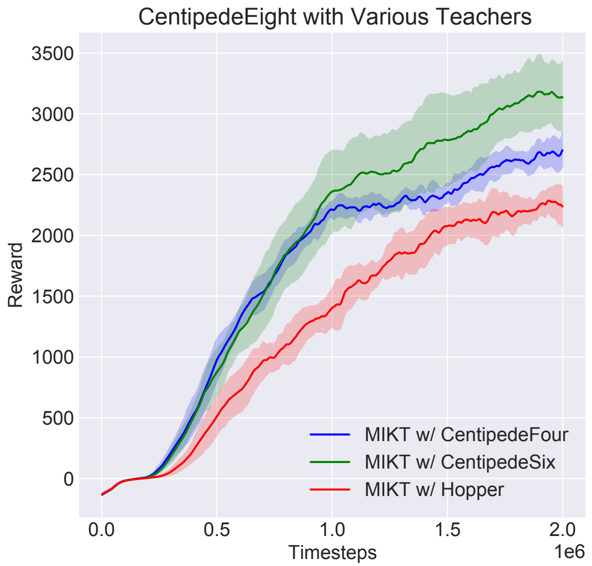
Encoder Ablation

- Are gradients from both $\{L_{PPO}, L_{MI}\}$ to the encoder beneficial?
- *MIKT w/o MI*: encoder does not receive gradients from L_{MI}
- *MIKT w/o RL gradients*: encoder does not receive gradients from L_{PPO}

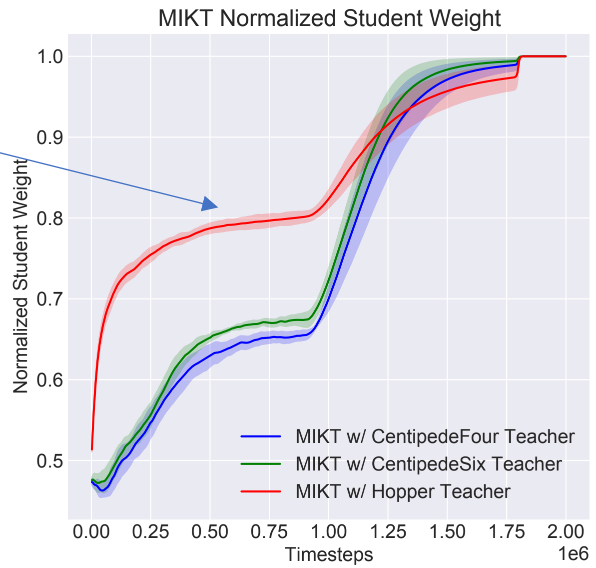


Task Similarity Ablation

- How does task similarity impact MIKT?
- We experiment with transfer from Centipede-{Four, Six} to CentipedeEight (source and target tasks are similar) and transfer from Hopper (source and target tasks are different)



Student learns to trust its own representations rather than dissimilar Hopper teacher's knowledge



Summary

- MIKT enables transfer learning between MDPs with *different state and action spaces*

- Learned embedding space
  - Mutual Information maximization: teacher representations depend on current state in the student MDP
  - Task aligned: encoder trained to maximize cumulative discounted reward

- Knowledge Transfer
  - Augment student representations with teacher representations