NerveNet: Learning Structured Policy with Graph Neural Networks

Wang, Tingwu, et al. ICLR2018

Present by Kibeom Kim

Interdisciplinary Program in Neuroscience

Seoul National University

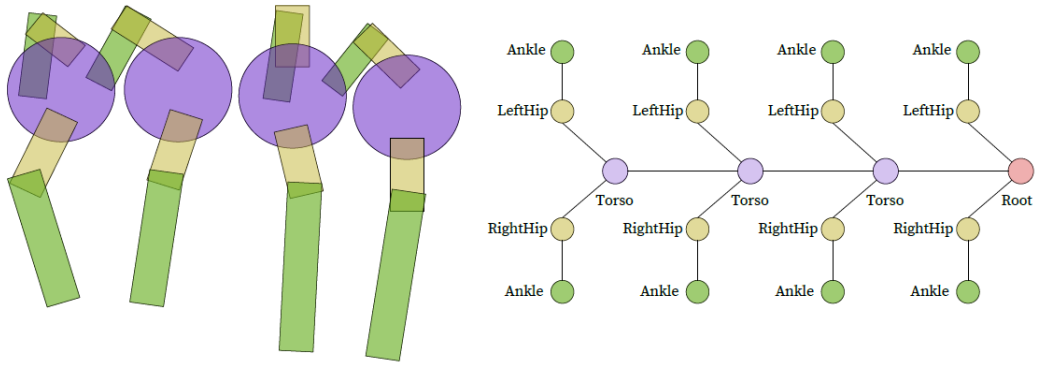
NerveNet: Learning Structured Policy with GNN

- Typically use MLP to learn the agent's policy in RL
  - Concatenation of observations from the environment
  - This leads to longer training times, requiring more data

- To exploit the body structure of an agent, and physical dependencies that naturally exist in such agents.

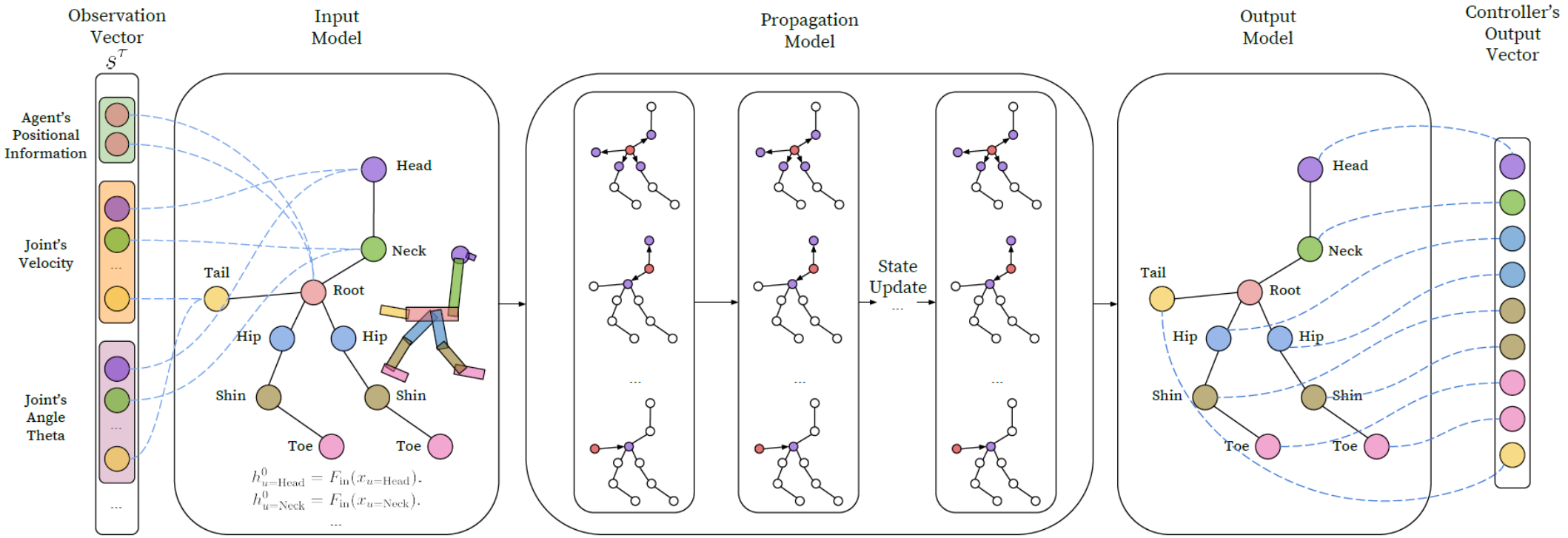
NerveNet: Learning Structured Policy with GNN

- Bodies of most robots and animals have a discrete graph structure.
- NerveNet first propagates information over the structure of the agent and then predict actions for different parts of the agent.
- Contribution:
  - Policy Network as GNN
  - Transfer learning ability



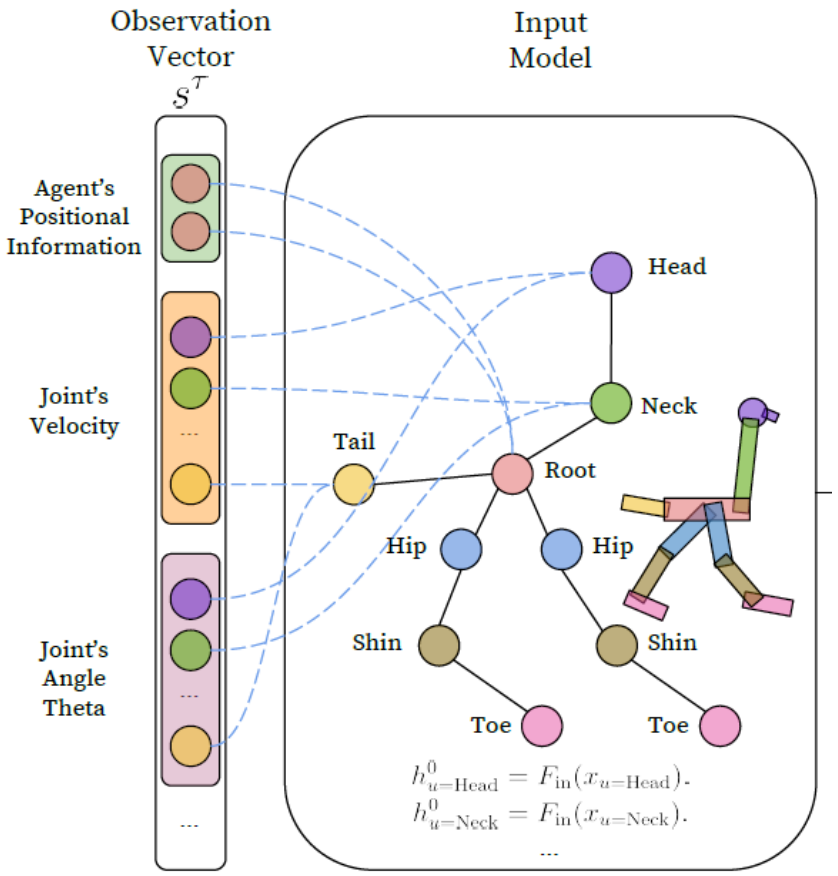
Visualization of the graph structure of *CentipedeEight*

NerveNet: Learning Structured Policy with GNN



Example of NerveNet

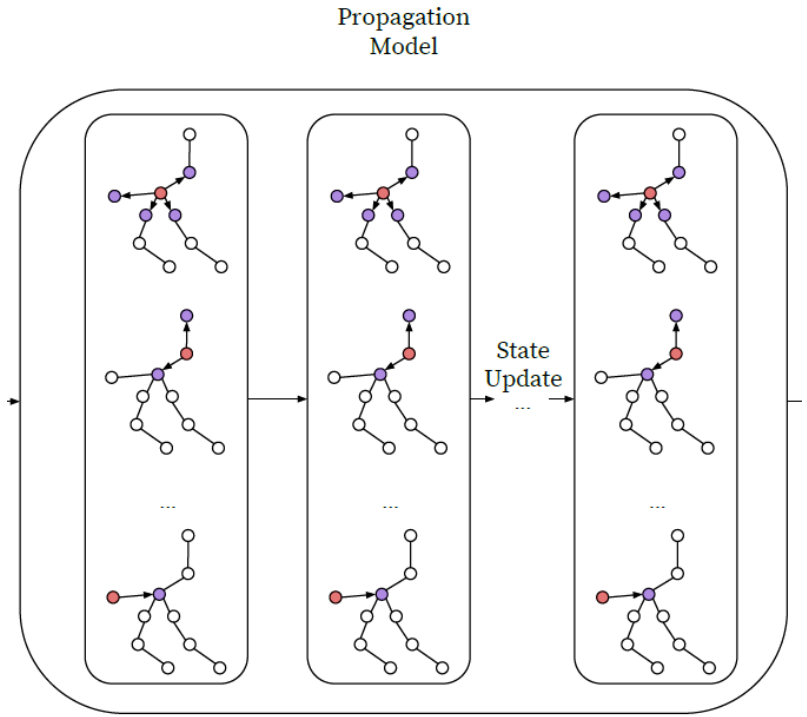
NerveNet: Learning Structured Policy with GNN



Input model of NerveNet

- Input Model
  - State vector $h_u^0 = F_{in}(x_u)$
    - x_u is the elements of observation vector corresponding to node u.
    - F_{in} is MLP.
    - Zero padding if observation size is different

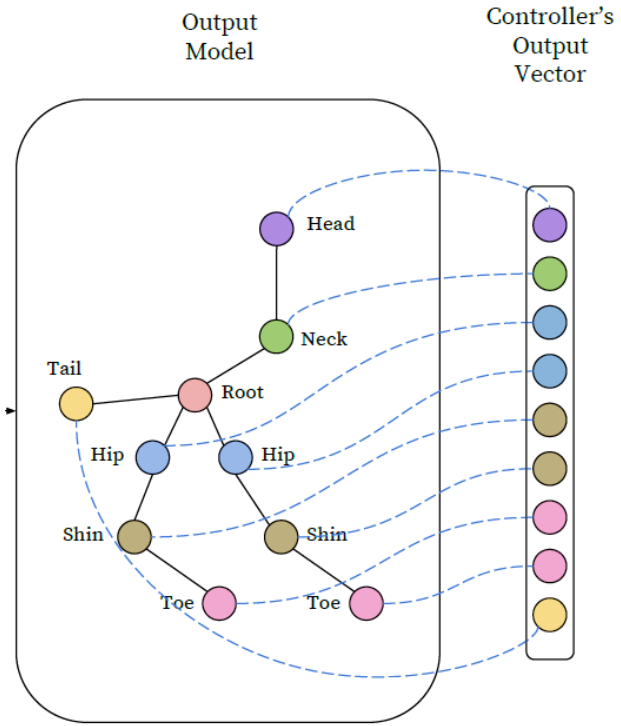
NerveNet: Learning Structured Policy with GNN



Propagation model of NerveNet

- Propagation Model
  - Message Computation $m_{(u,v)}^t = m_{c_{(u,v)}}(h_u^t)$
    - $c_{(u,v)}$ indicates that edges of the same edge type share the same instance of the message function. $m_{c_{(u,v)}}$ is MLP.
  - Message Aggregation $\bar{m}_u^t = A(\{h_v^t | v \in \mathcal{N}_{in}(u)\})$
    - $A()$ is the aggregation function which may be a summation, average or max-pooling function.
  - States Update $h_u^{t+1} = U_{p_u}(h_u^t, \bar{m}_u^t)$
    - p_u is nodes of the same node type share the same instance of the update function. U_{p_u} is LSTM or GRU.

NerveNet: Learning Structured Policy with GNN



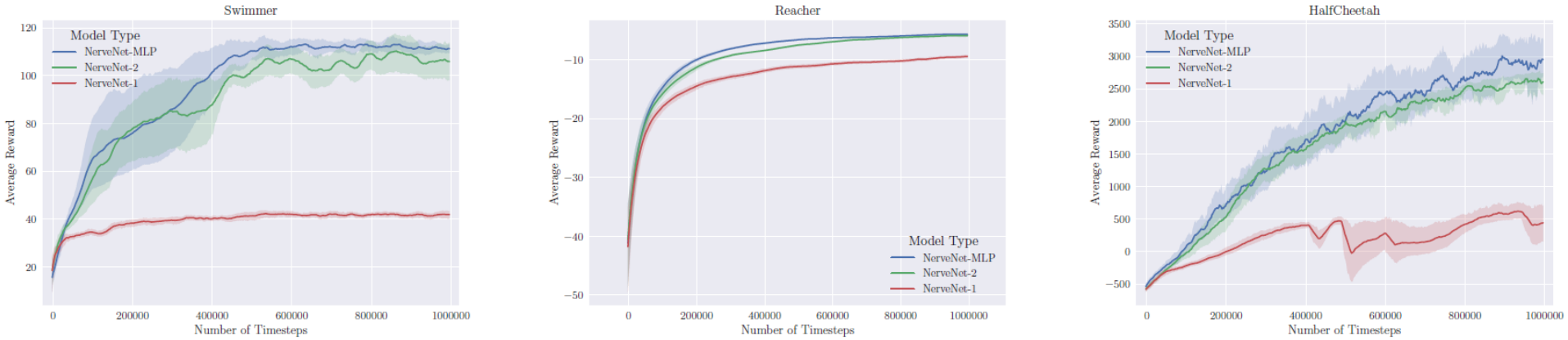
Output model of NerveNet

- Output Model
  - $\mu_{u\in\mathcal{O}} = O_{q_u}(h_u^T)$
    - $\mu_{u\in\mathcal{O}}$ is the mean value for action applied on each actuator.
    - O_{q_u} is MLP.
  - Policy as Gaussian distribution
    - $\pi_\theta(a^\tau|s^\tau) = \prod_{u\in\mathcal{O}} \pi_{\theta,u}(a_u^\tau|s^\tau) = \prod_{u\in\mathcal{O}} \frac{1}{\sqrt{2\pi\sigma_u^2}} e^{(a_u^\tau-\mu_u)^2/(2\sigma_u^2)}$
    - $a^\tau \in \mathcal{A}$ is the output action, and $\sigma_u$is the variable standard deviation for each action

NerveNet: Learning Structured Policy with GNN

- Learning Algorithm
  - PPO (Proximal Policy Optimization)

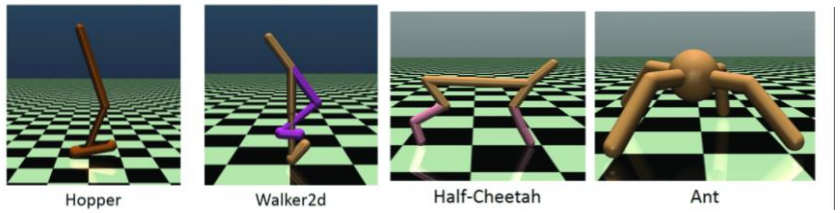
- Value Network
  - NerveNet-MLP - GNN as the policy network, MLP as the value network
  - NerveNet-2 – GNN as the policy network, another GNN as the value network
  - NerveNet-1 – GNN as both policy and value network



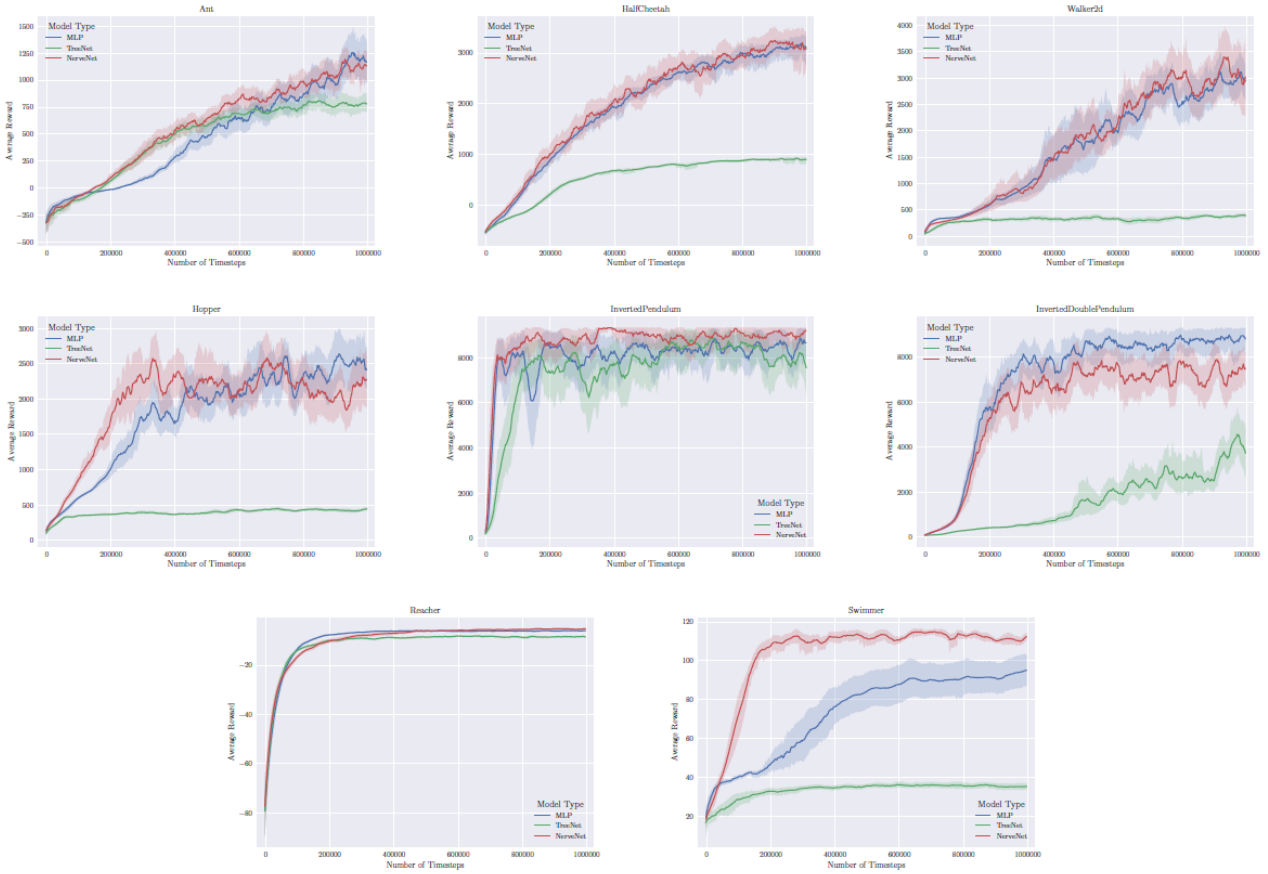
Results of several variants of NerveNet

Experiments

- Comparison on standard benchmark
  - 8 continuous control benchmarks
    - MuJoCo simulation



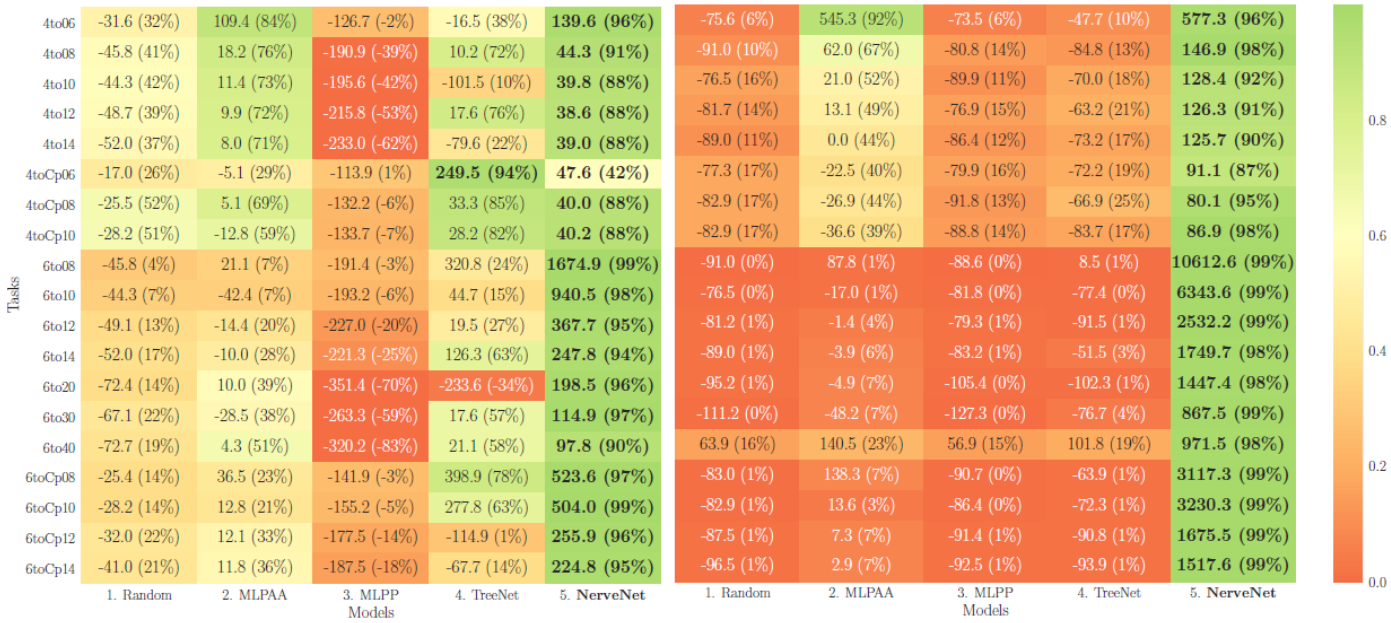
    - TreeNet
      - singly rooted depth-1 tree
      - Aggregates the information and then feeds the state vector to the output model



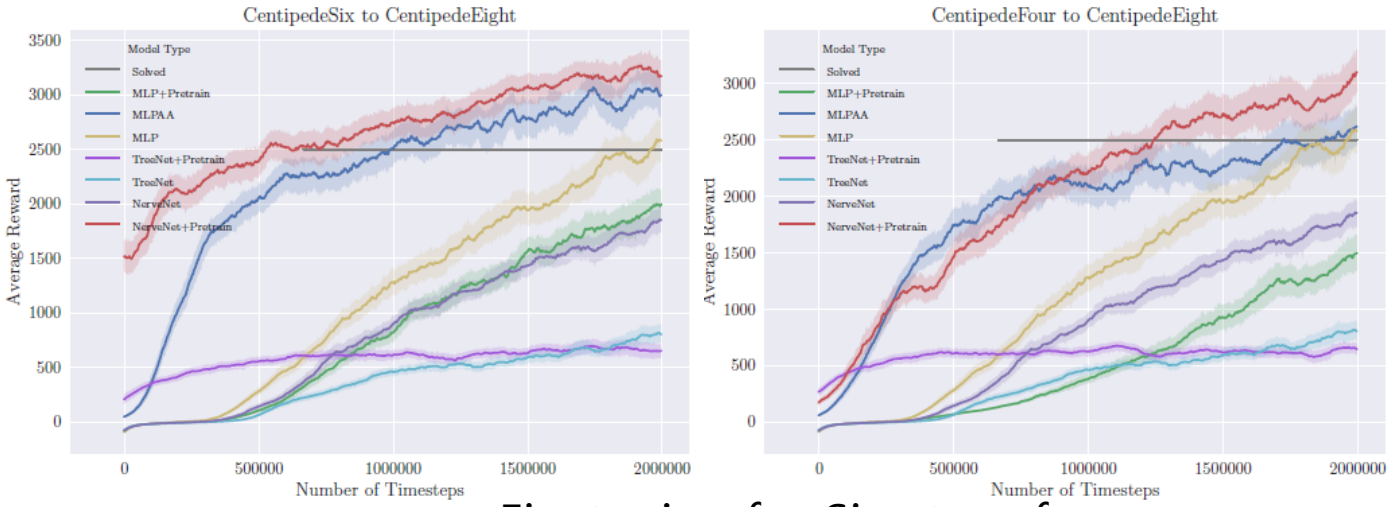
Results of MLP, TreeNet and NerveNet on 8 continuous control benchmarks from the gym

Experiments

- Structure Transfer Learning
  - Size transfer
    - To train a model with a small size agent and apply learned model to larger size
    - NerveNet+Pretrain
      - Large agent has repetitive structure
    - MLPAA
      - Learning large-agent model and setting the remaining weight to be zero
    - MLP, **Solved**
    - MLP+Pretrain, NerveNet



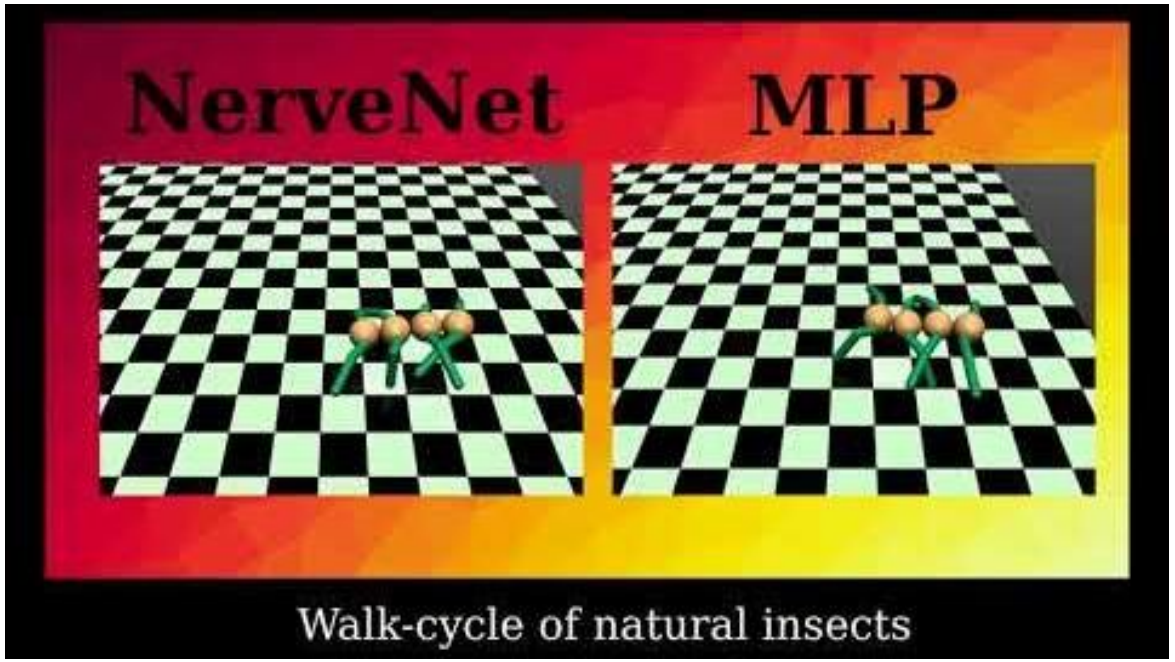
(a) Zero-shot average reward. (b) Zero-shot average running-length.



Finetuning for Size transfer

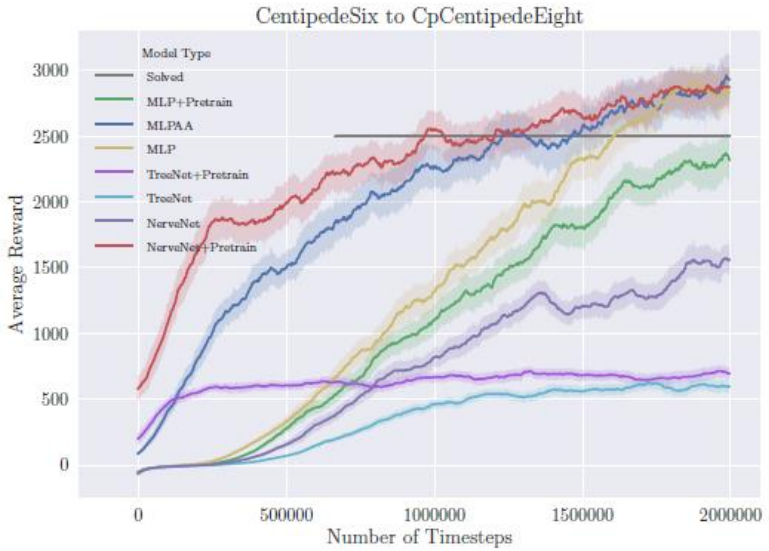
Experiments

- Size transfer
  - 6 to 8 legs
    - MLP+Pretrain
    - TreeNet
    - NerveNet

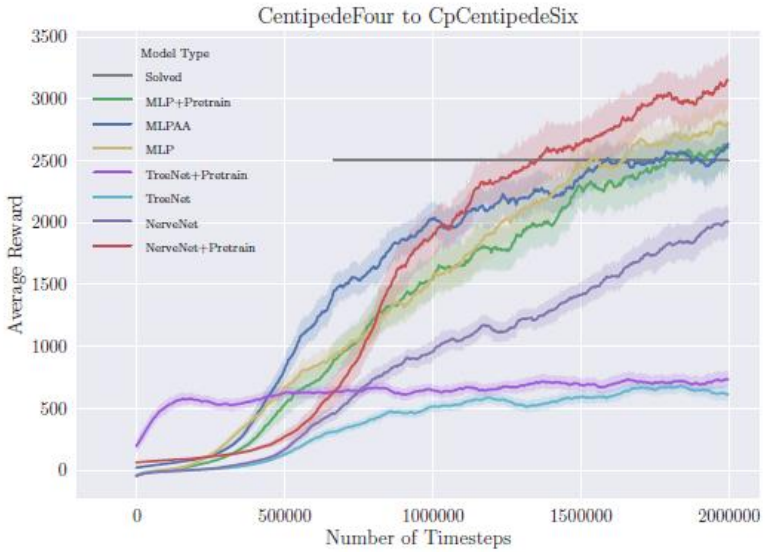


Experiments

- Structure Transfer Learning
  - Disability transfer
    - Learn a model for the original agent and then apply it some components disabled
      - two back legs are disabled
    - If overfits, disabling some components of the agent might bring catastrophic performance degradation.
    - NerveNet+Pretrain, MLPAA, MLP, Solved



(c)



(d)

Experiments

- Multi-task experiments
  - Constrain the Walker multi-task learning
  - Aim to test the ability to control multiple agents using one unified network.

Results of Multi-task learning. We train the networks simultaneously on five different tasks from Walker.

My Opinion on this paper

- Pros
  - Structured Policy
  - Transfer learning experiments

- Cons
  - Only for graph structured agents
  - No significant difference compared to MLP


Thanks!