Learning to Control Self-Assembling Morphologies
Generalization via Modularity

Deepak Pathak*
Chris Lu*
Trevor Darrell
Phillip Isola
Alyosha Efros

* equal contribution

NeurIPS 2019
How do we train a robot?
- Multiple tasks
- Expert demonstrations
- Rewards, labels
- ...

Image: A robot arm and a cartoon character with multiple arms, symbolizing multiple tasks.
Self-supervision
Curious exploration
Learning “common sense”

- Multiple tasks
- Expert demonstrations
- Rewards, labels
... even earlier?
Single to Multicellular
Single to Multicellular competition $\rightarrow$ collaboration
Single to Multicellular

competition $\rightarrow$ collaboration

shared objective
Compositionality has been useful in language ...
How to implement compositionality in hardware?
Modular Co-evolution of Control and Morphology
Modular Co-evolution of Control and Morphology
Modular Co-evolution of Control and Morphology

Cylindrical Limb

Configurable Motor Joint
Modular Co-evolution of Control and Morphology
Modular Co-evolution of Control and Morphology
Modular Co-evolution of Control and Morphology

Potential Magnetic Joint
Modular Co-evolution of Control and Morphology

Potential Magnetic Joint
Modular Co-evolution of Control and Morphology

Acts as single agent upon joining

Rewards are shared!

Potential
Magnetic Joint
Modular Co-evolution of Control and Morphology

Acts as single agent upon joining
Rewards are shared!

- **Input** = *Local* Sensory State
- **Output** = Torques, Link, Unlink
Modular Co-evolution of Control and Morphology

Acts as single agent upon joining
Rewards are shared!

- **Input** = *Local* Sensory State
- **Output** = Torques, Link, Unlink
Consider the task of “standing up” ...
Vanilla Reinforcement Learning

1 limb policy

Standing Task
maximize Y-axis
How to learn compositional controllers?
Idea: Shared policy network across limbs
Idea: Shared policy network across limbs
How to adapt when morphology changes?
How to adapt when morphology changes?
Network as reusable LEGO Blocks
Network as reusable LEGO Blocks

\[ \pi \theta \]

shared policy

output

input
Network as reusable LEGO Blocks

Input

\[ \pi_\theta \]

Message input

Message output

Output

Shared policy
Network as reusable LEGO Blocks

- Shared policy: $\pi_\theta$
- Output
- Same dimension: message
- Input
- Message input
Network as reusable LEGO Blocks

\[ \pi \theta \]
Network as reusable LEGO Blocks

\[ \pi_\theta \]

shared policy

input

message input

output

message output
Network as reusable LEGO Blocks

shared policy $\pi_\theta$

input

output

message

output

message

input
Network as reusable LEGO Blocks

shared policy $\pi_\theta$

output

message output

input

message input
Network as reusable LEGO Blocks

shared policy $\pi_\theta$

input

message output

output

message input
Network as reusable LEGO Blocks

shared policy \( \pi_\theta \)

input  message input

output message output

cut
Network as reusable LEGO Blocks

- Shared policy: $\pi_\theta$
- Input and output messages
- Cut and paste
Network as reusable LEGO Blocks

adaptation by conditioning

cut and paste
Dynamic Graph Networks

Network as reusable LEGO Blocks

output

message output

shared policy

$\pi_\theta$

input

message input

conditioning

cut and paste
BTW, basically curriculum learning but in hardware
How well does it generalize?
Generalization w/o Fine-tuning

twice as many limbs

Standing Task
maximize Y-axis
a bit crazy... is it even possible in real world?
Self-Assembling Robots in the Real World

[Mark Yim’s Lab at UPenn]

[Daniela Rus's Lab at MIT]

Also: [Modular Snake Robot – Howie Choset’s Lab at CMU]
code & data at
https://people.eecs.berkeley.edu/~pathak/

Poster # 197
Today (Tues)!!

Thank You!