Learning to Control Self-Assembling Morphologies
Generalization via Modularity

Deepak Pathak*

Chris Lu*

Trevor Darrell

Phillip Isola

Alyosha Efros

* equal contribution

Berkeley UNIVERSITY OF CALIFORNIA

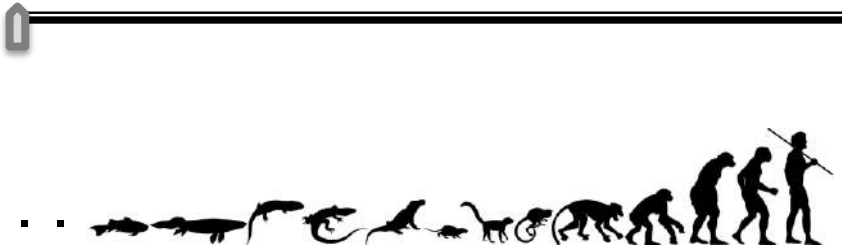
NeurIPS 2019

How do we train a robot?

- Multiple tasks
- Expert demonstrations
- Rewards, labels
- …

- Self-supervision
- Curious exploration
- Learning “common sense”
- …

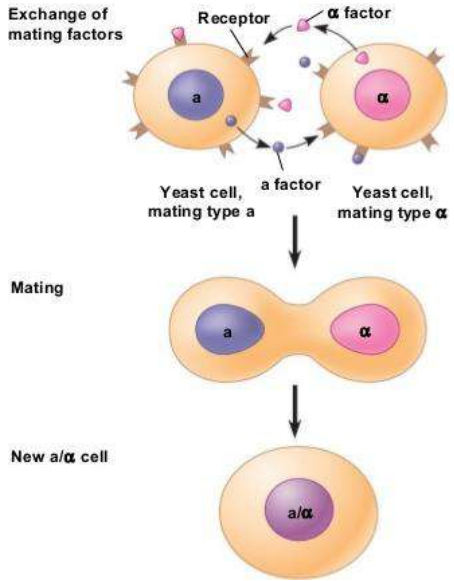
- Multiple tasks
- Expert demonstrations
- Rewards, labels
- …



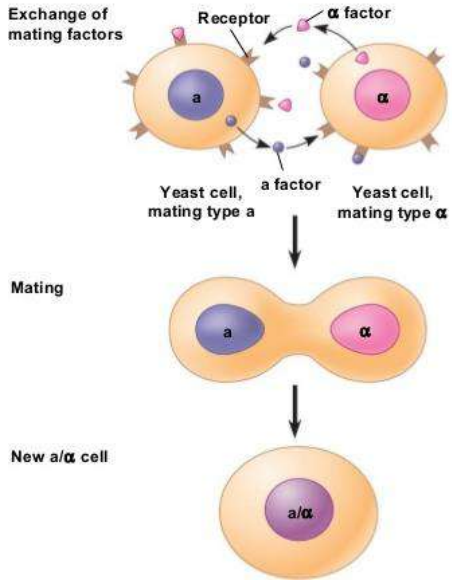
... even earlier?

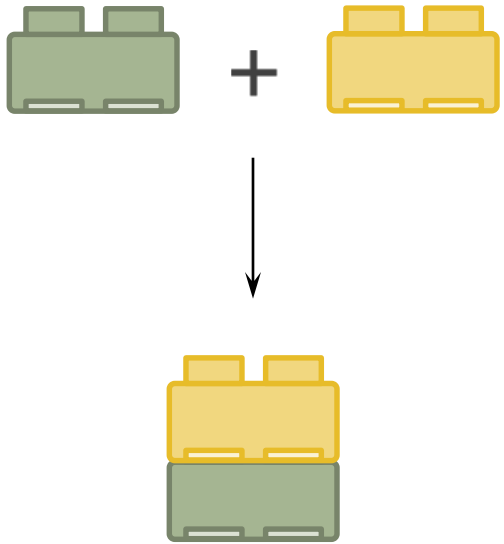
Single to Multicellular

Single to Multicellular *competition → collaboration*



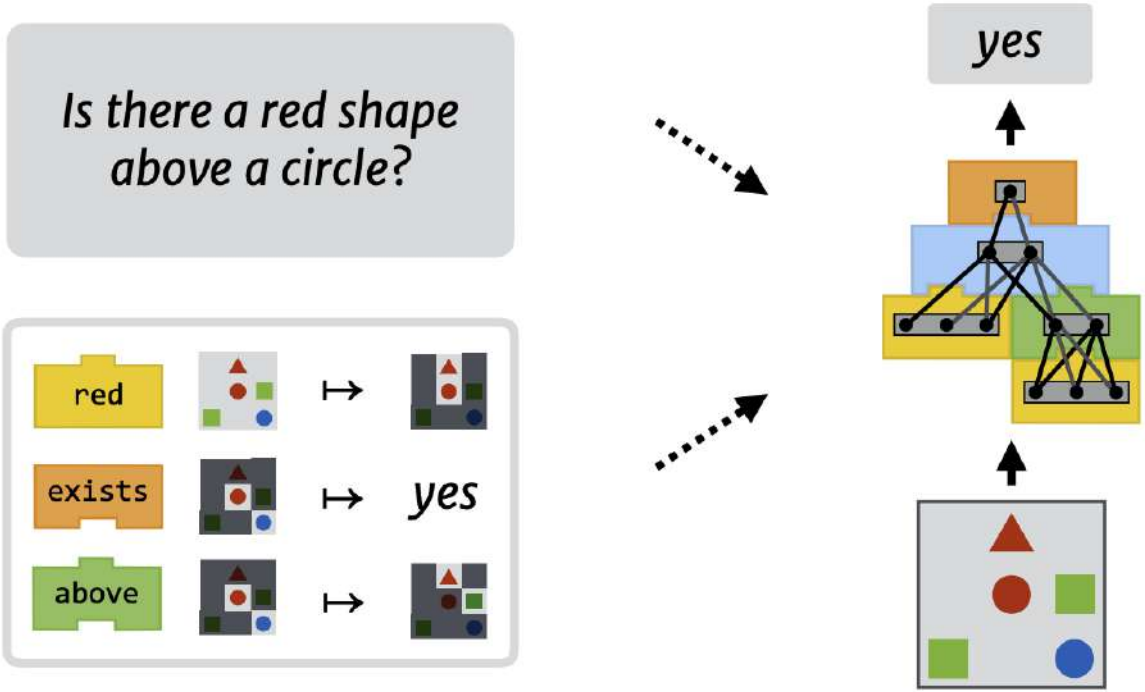
Single to Multicellular *competition → collaboration*





shared objective

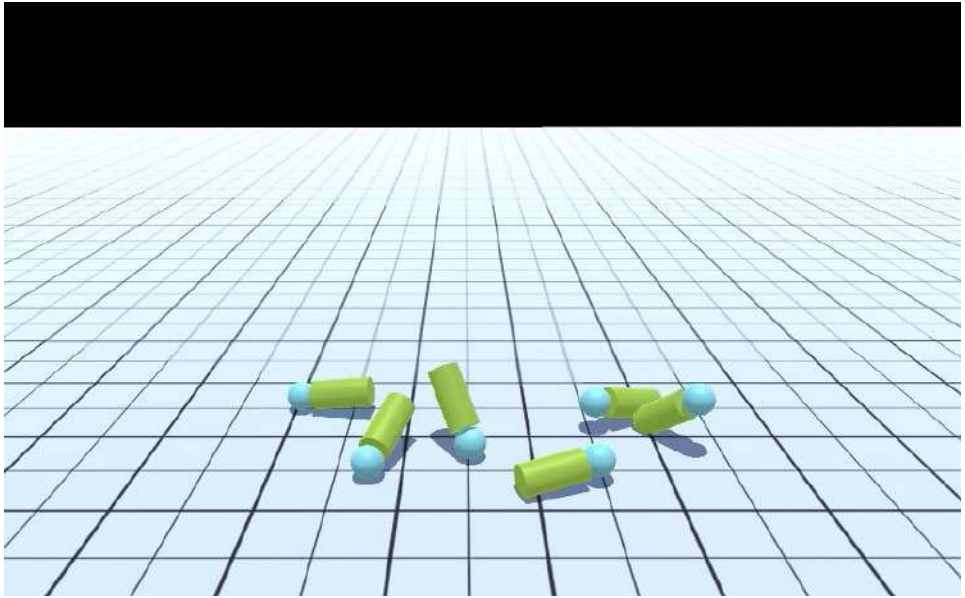
Compositionality has been useful in language …



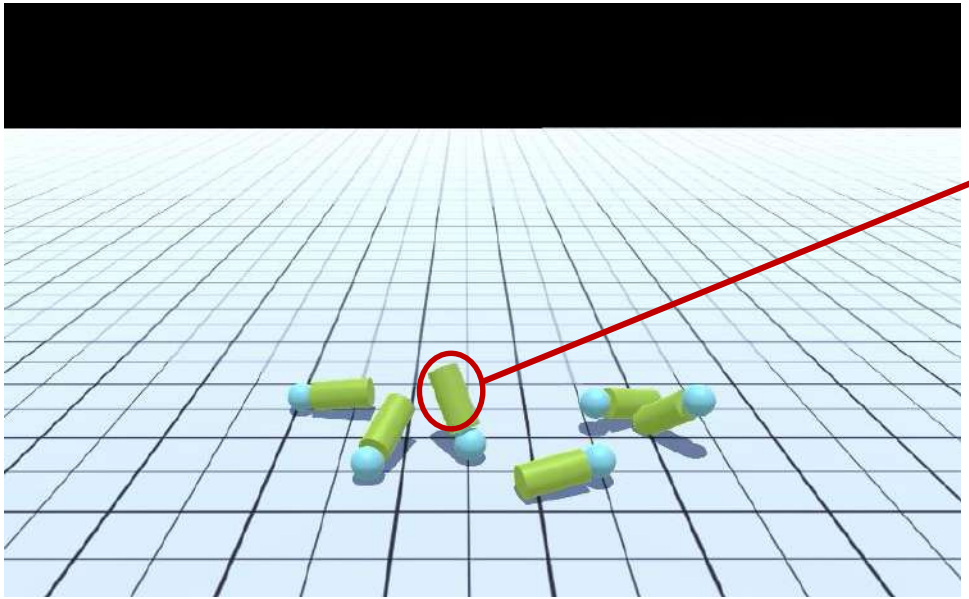
[Andreas et. al. 2016]

How to implement compositionality in hardware?

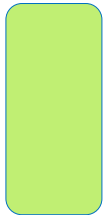
Modular Co-evolution of Control and Morphology



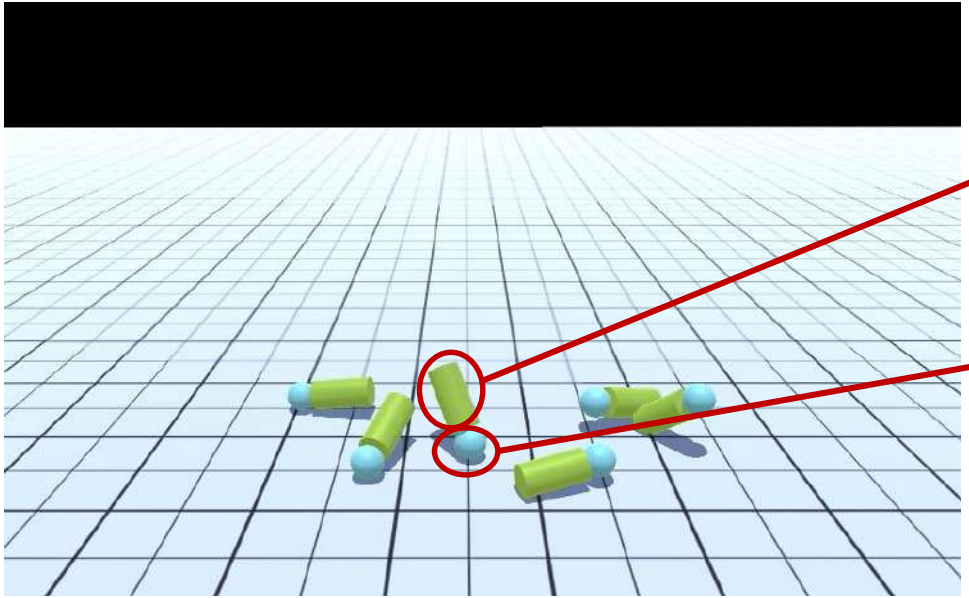
Modular Co-evolution of Control and Morphology



Cylindrical Limb

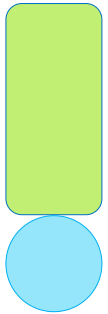


Modular Co-evolution of Control and Morphology

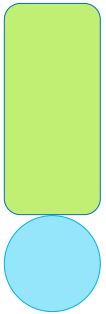


Cylindrical Limb

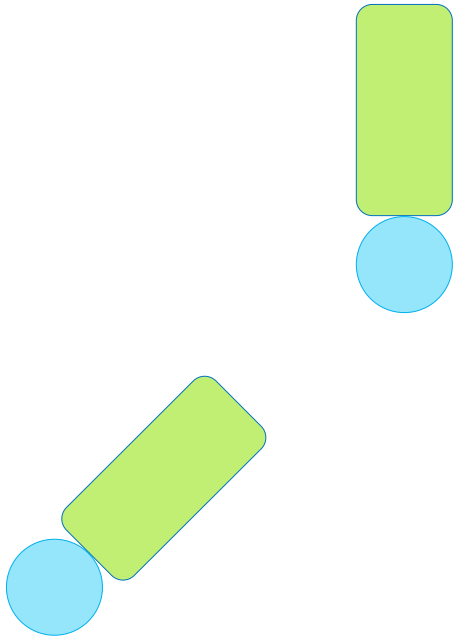
Configurable Motor Joint



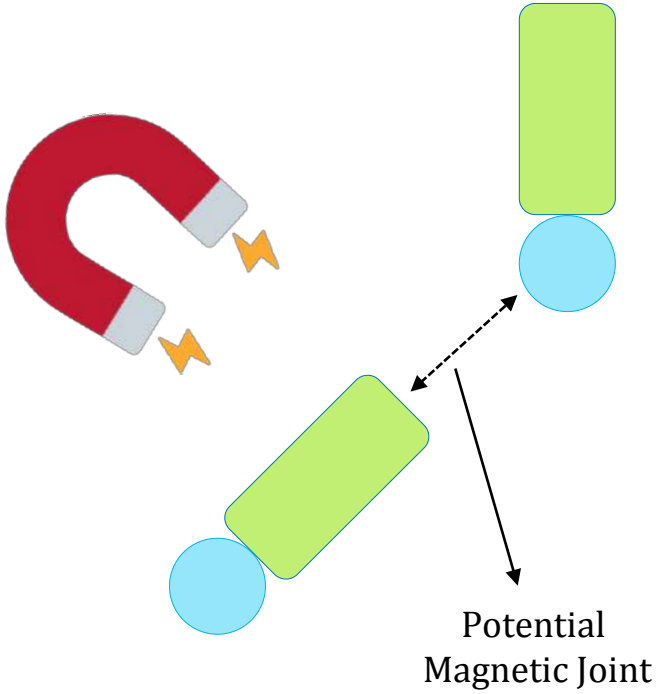
Modular Co-evolution of Control and Morphology



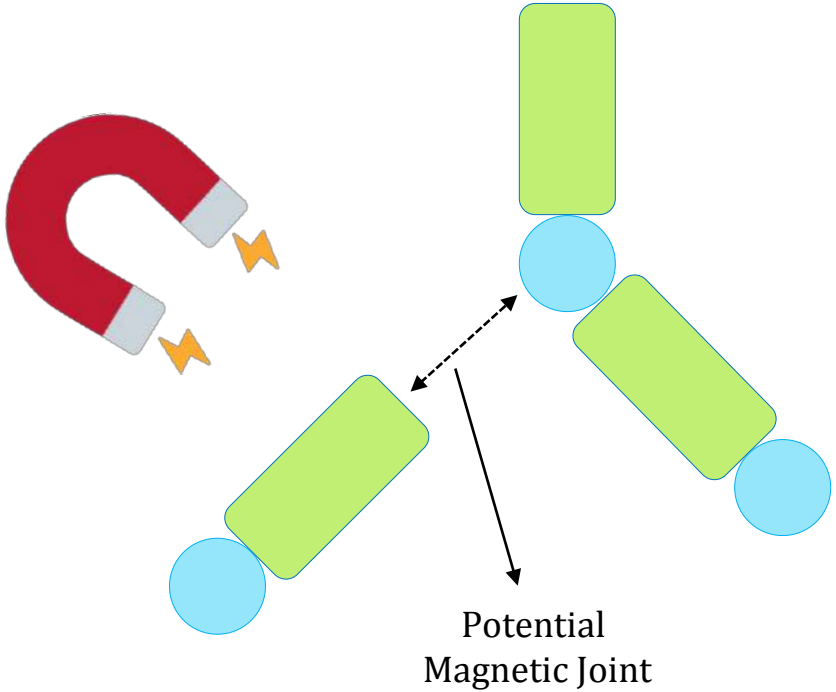
Modular Co-evolution of Control and Morphology



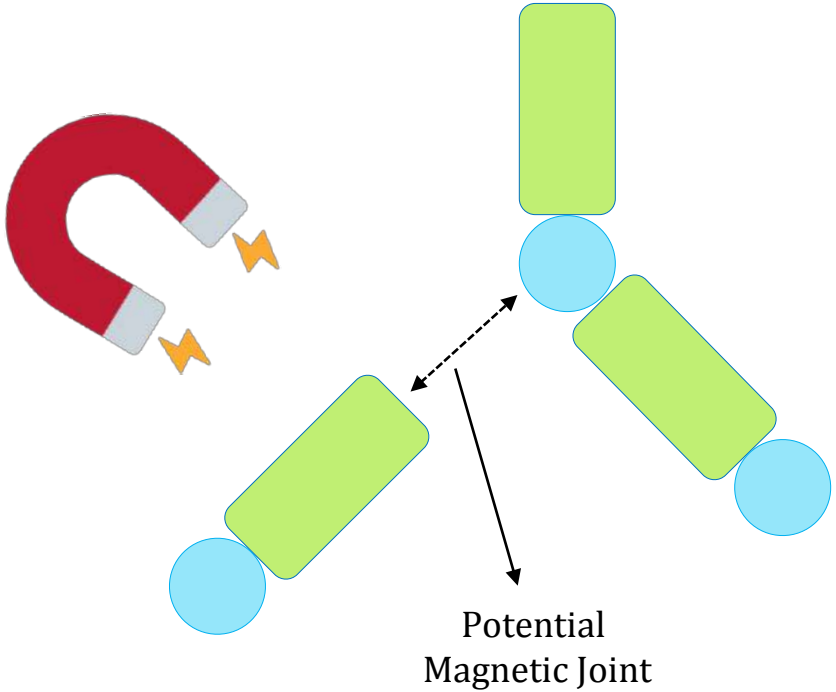
Modular Co-evolution of Control and Morphology



Modular Co-evolution of Control and Morphology



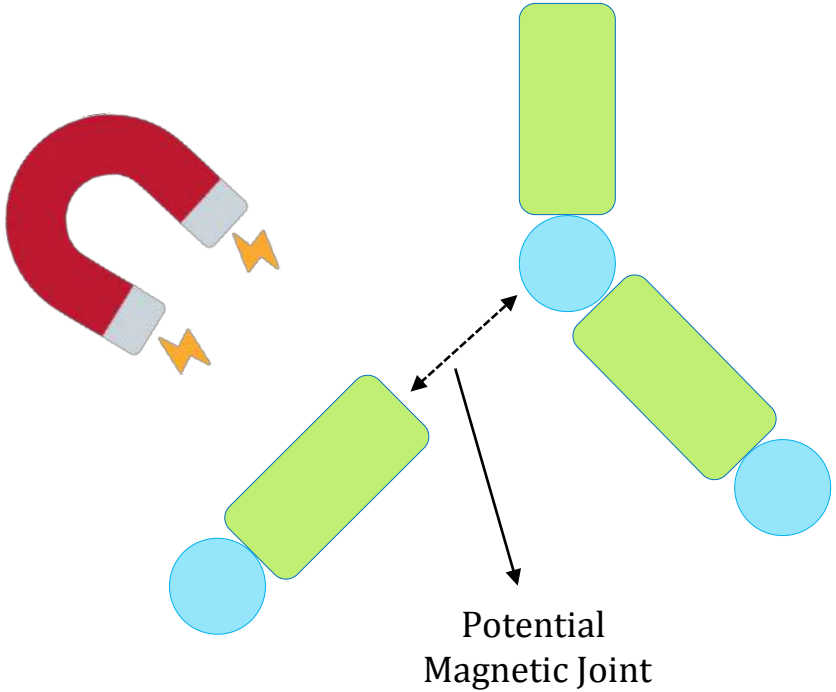
Modular Co-evolution of Control and Morphology



Acts as single agent upon joining

Rewards are shared!

Modular Co-evolution of Control and Morphology

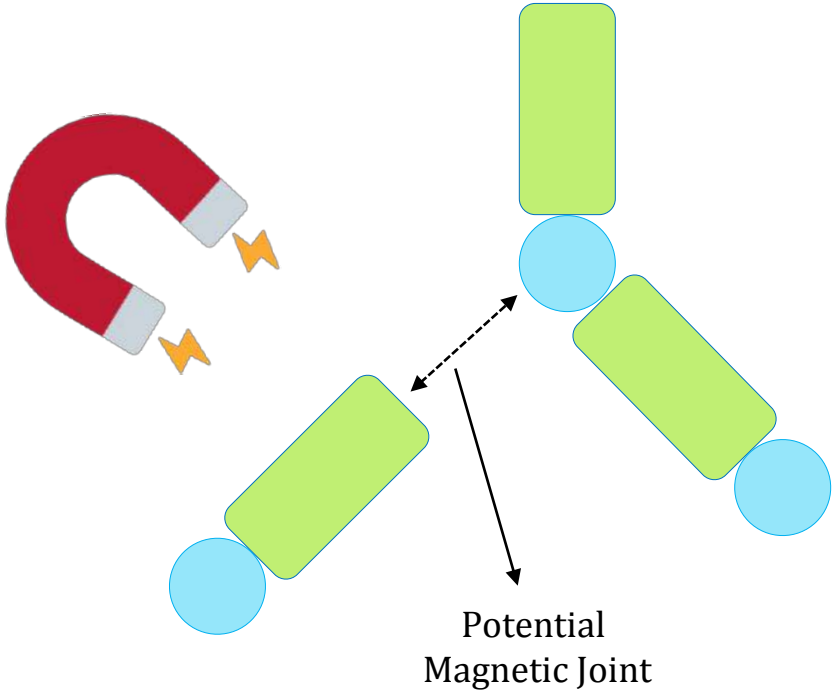


Acts as single agent upon joining

Rewards are shared!

- Input = *Local* Sensory State
- Output = Torques, Link, Unlink

Modular Co-evolution of Control and Morphology

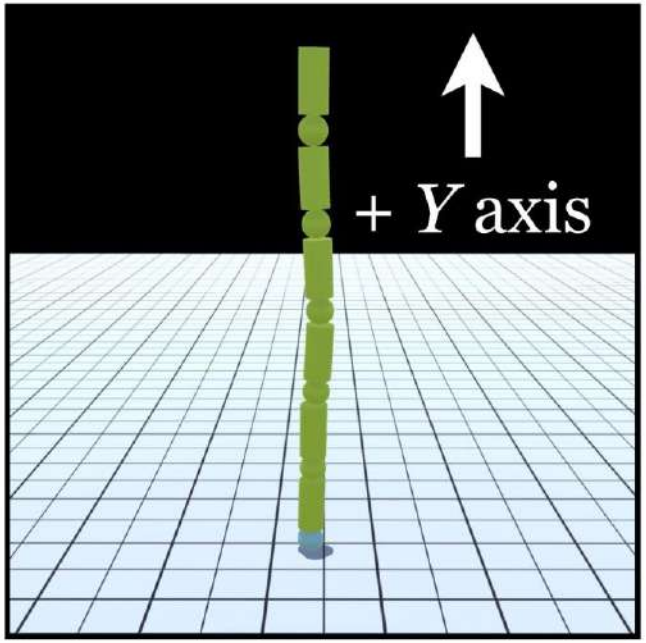


Acts as single agent upon joining

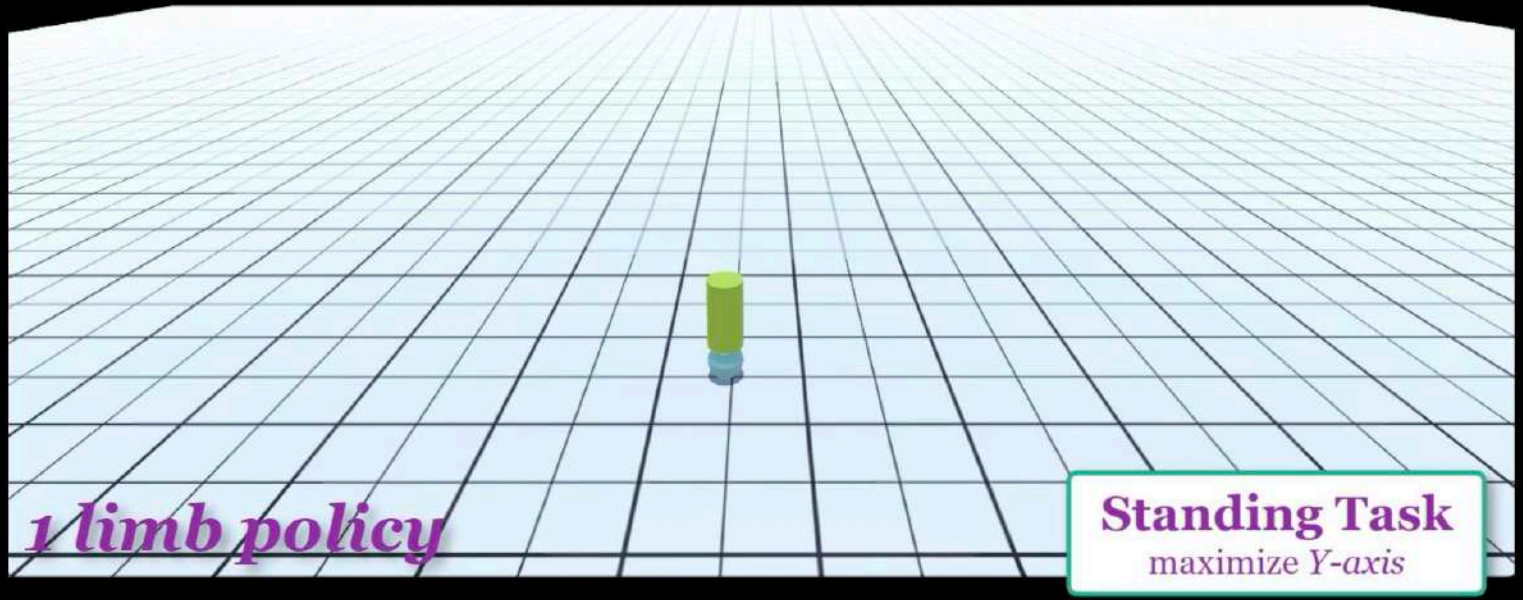
Rewards are shared!

- Input = *Local* Sensory State
- Output = Torques, Link, Unlink

Consider the task of “standing up” …

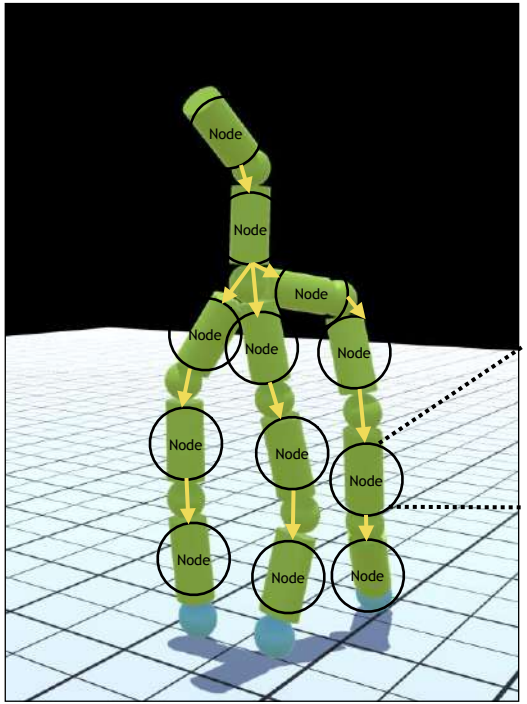


Vanilla Reinforcement Learning

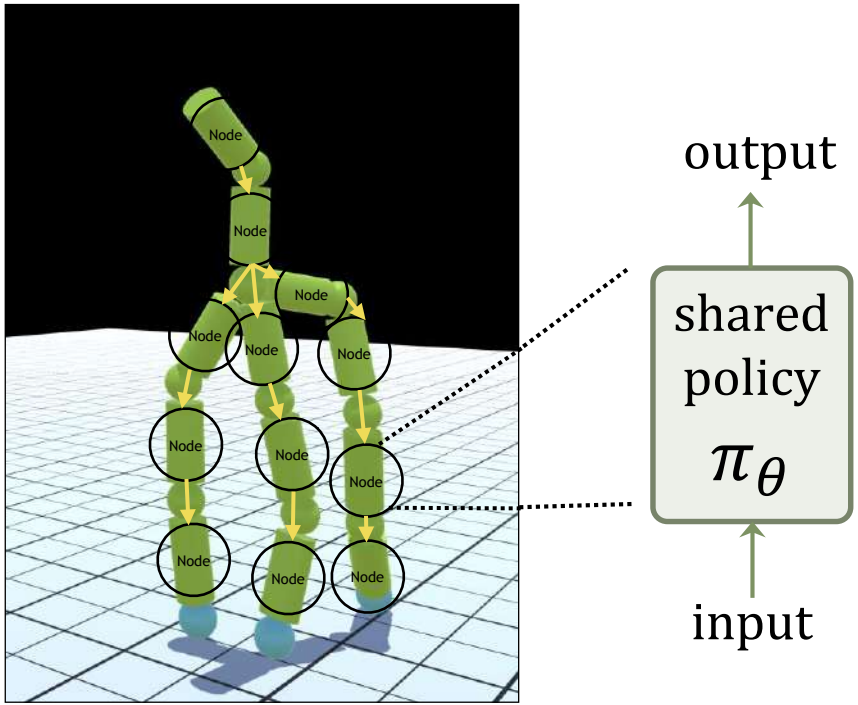


How to learn compositional controllers?

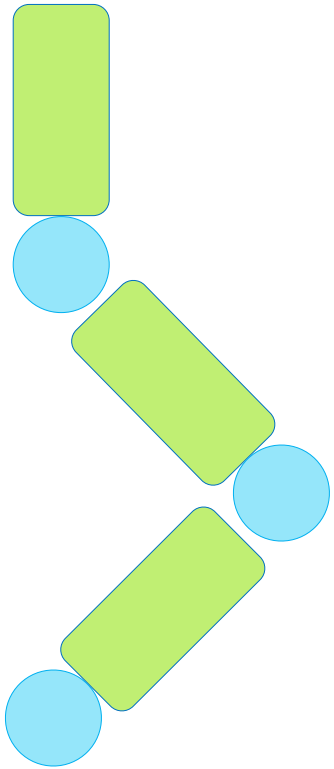
Idea: Shared policy network across limbs



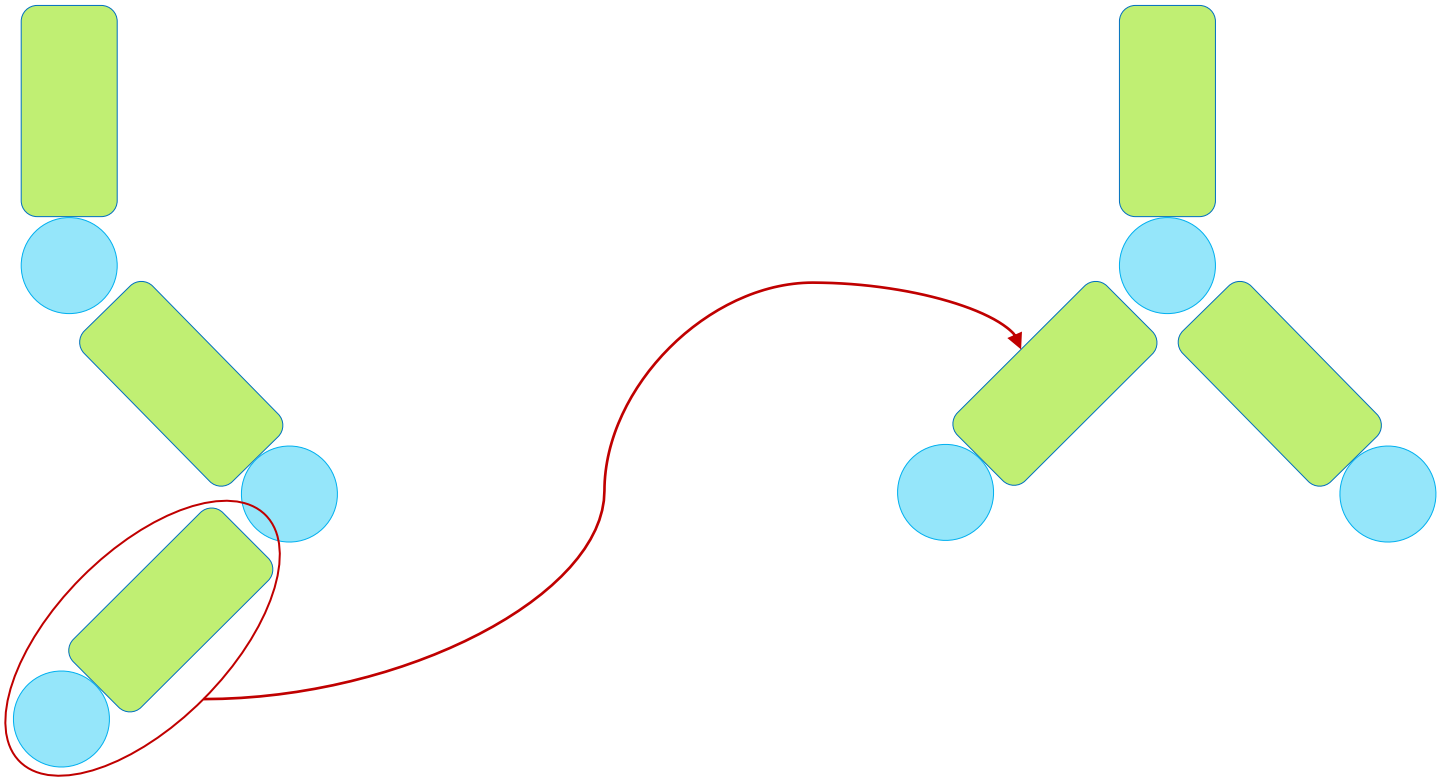
Idea: Shared policy network across limbs



How to adapt when morphology changes?

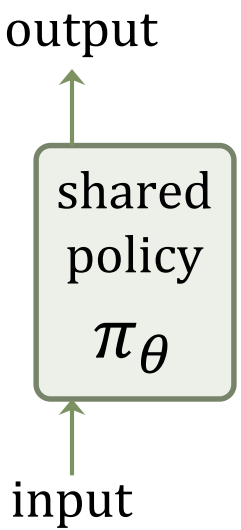


How to adapt when morphology changes?

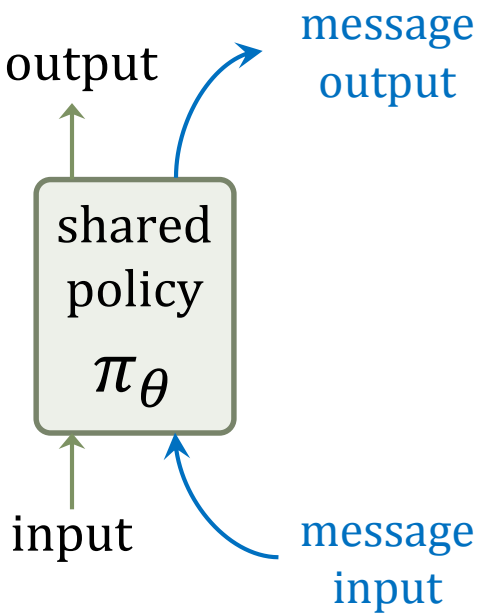


Network as reusable LEGO Blocks

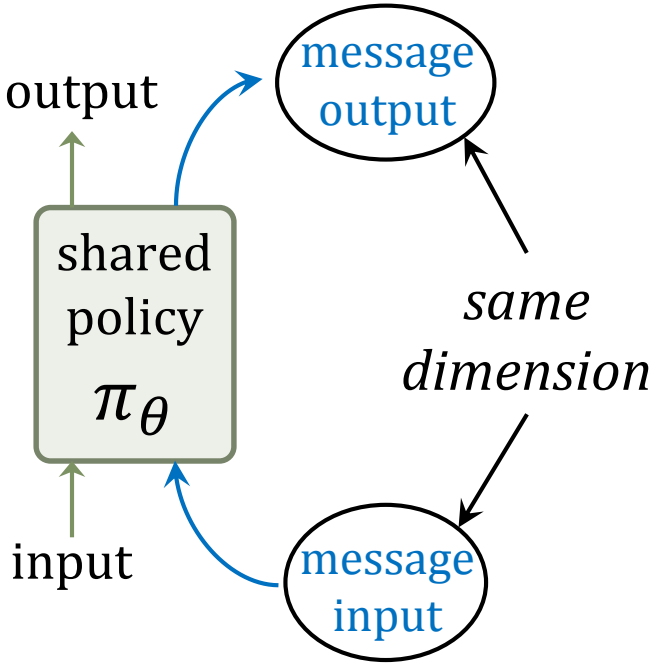
Network as reusable LEGO Blocks



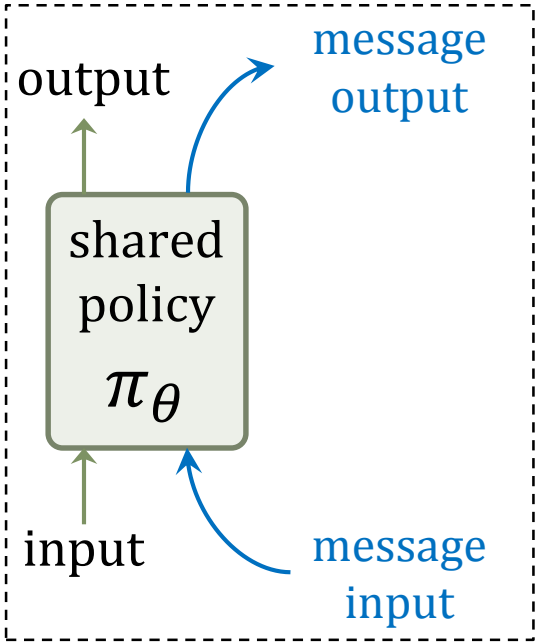
Network as reusable LEGO Blocks



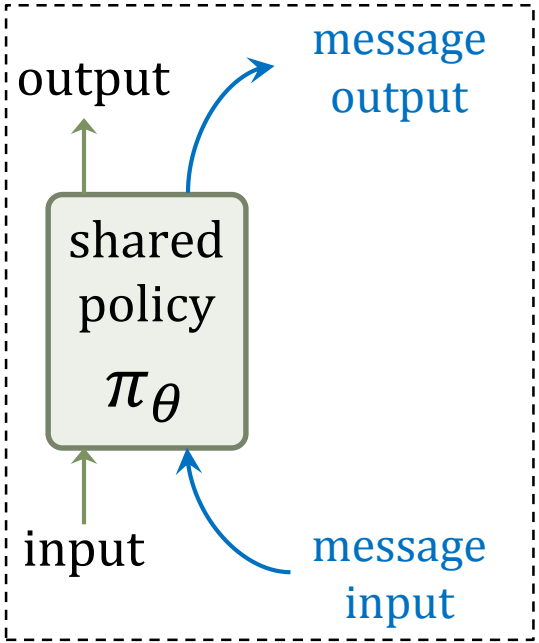
Network as reusable LEGO Blocks

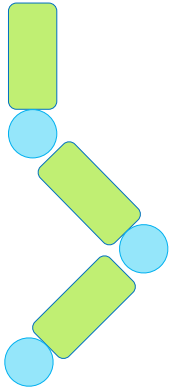


Network as reusable LEGO Blocks

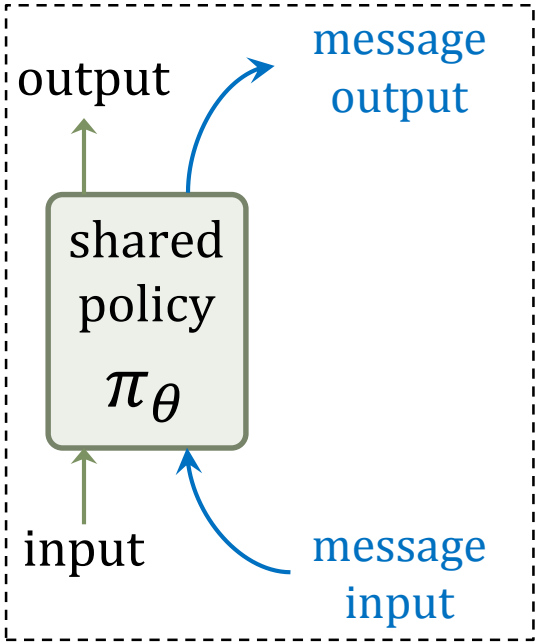


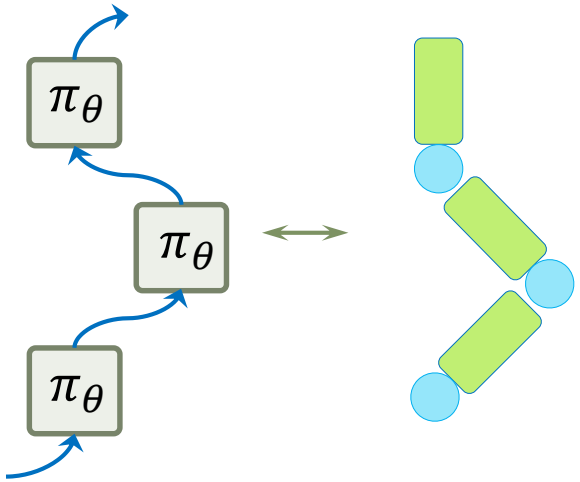
Network as reusable LEGO Blocks



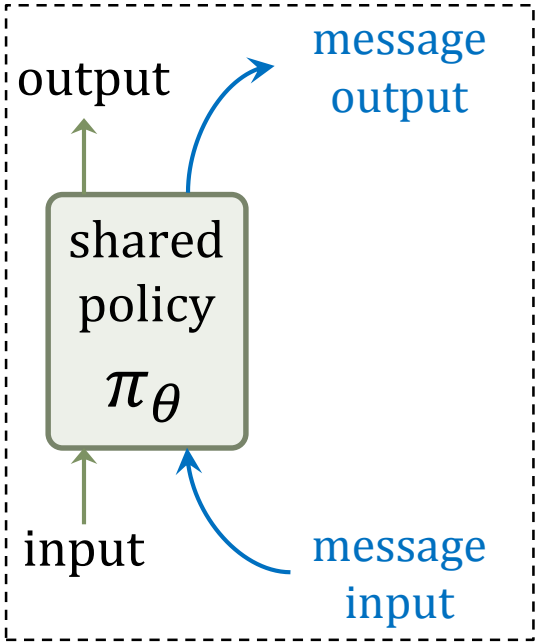


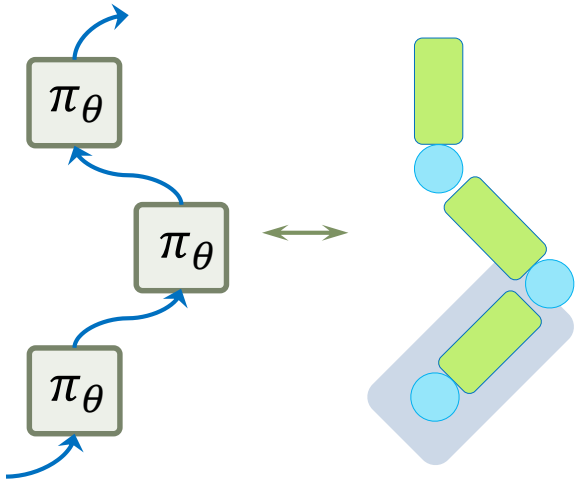
Network as reusable LEGO Blocks



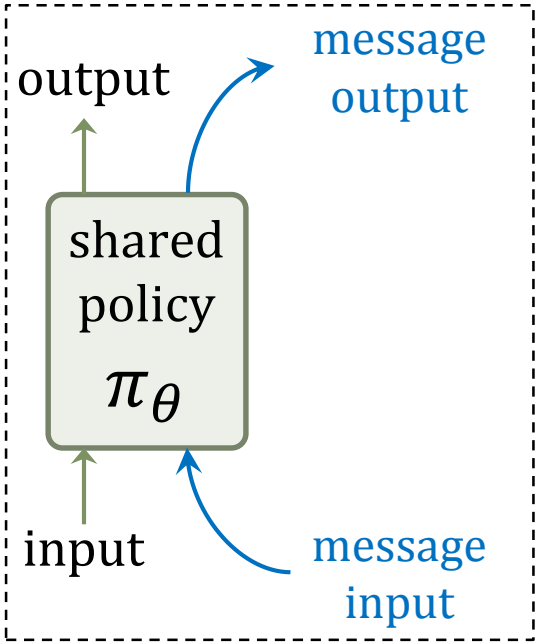


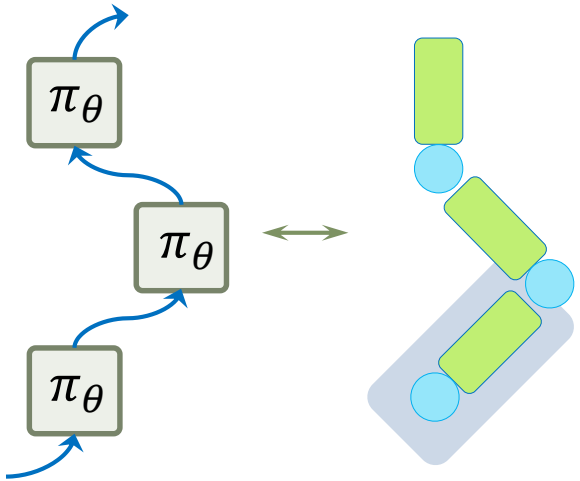
Network as reusable LEGO Blocks

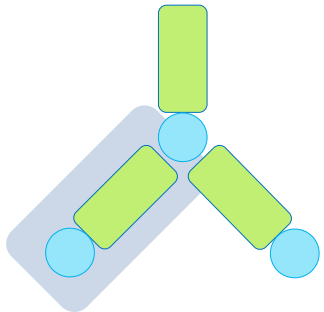




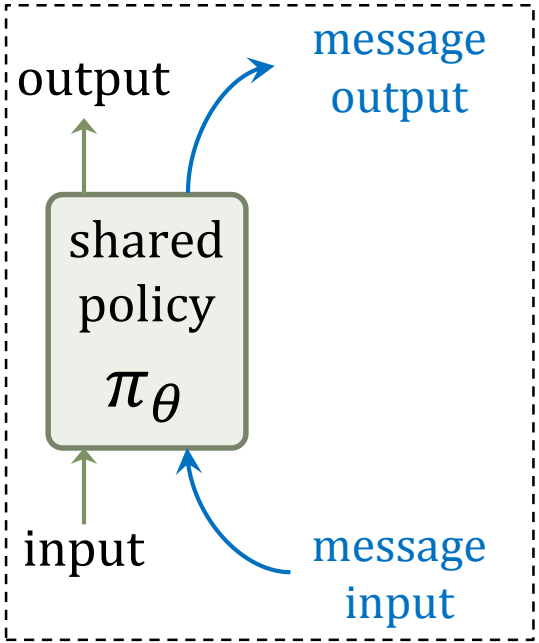
Network as reusable LEGO Blocks

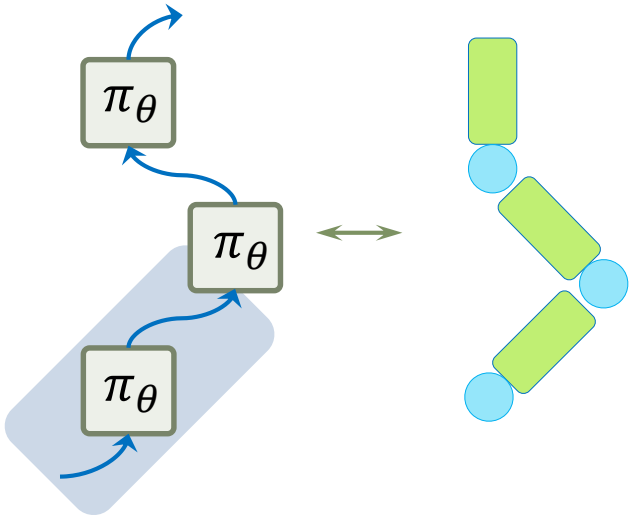




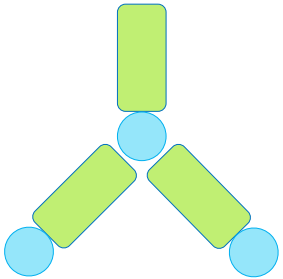


Network as reusable LEGO Blocks

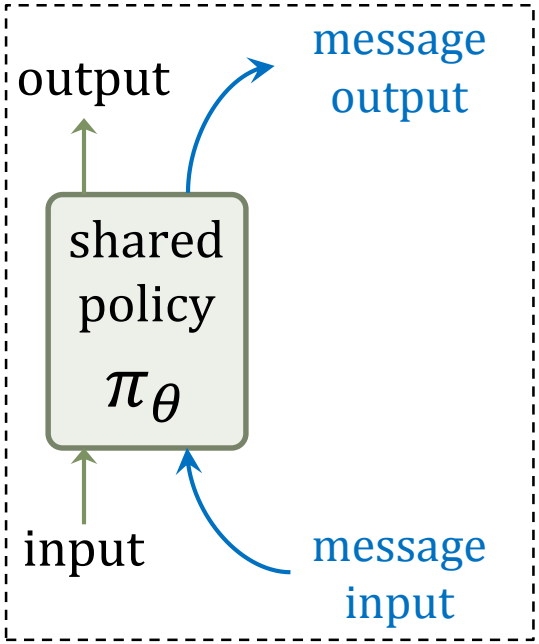


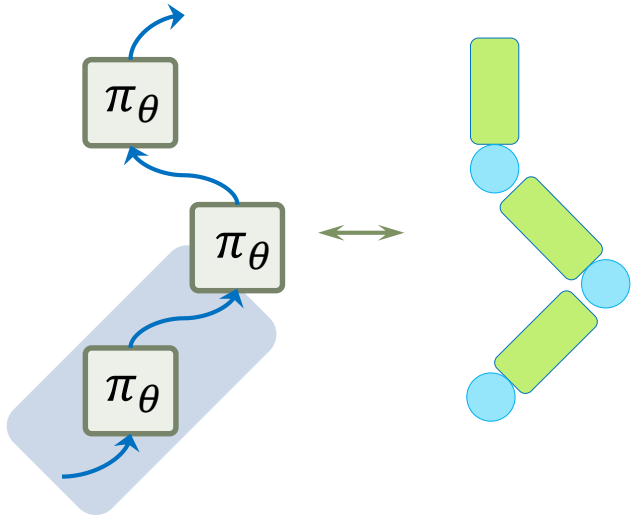


cut

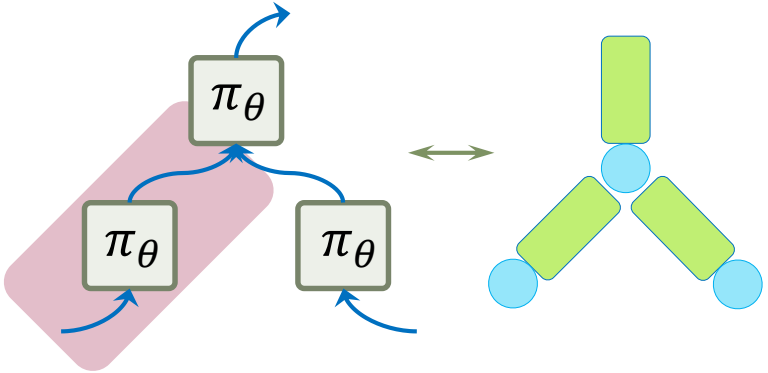


Network as reusable LEGO Blocks

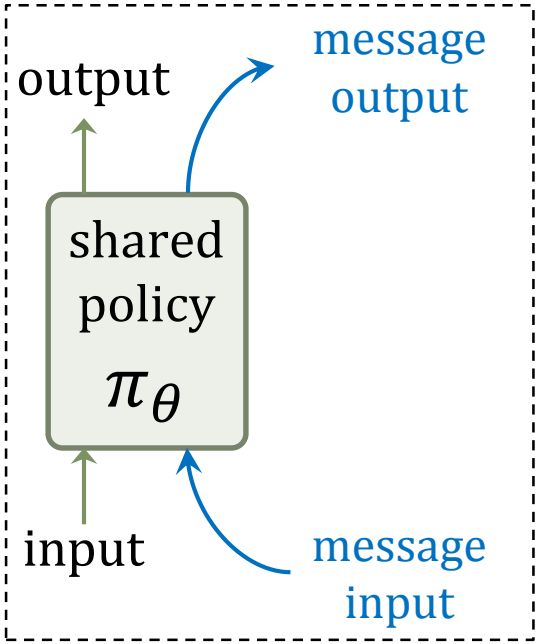




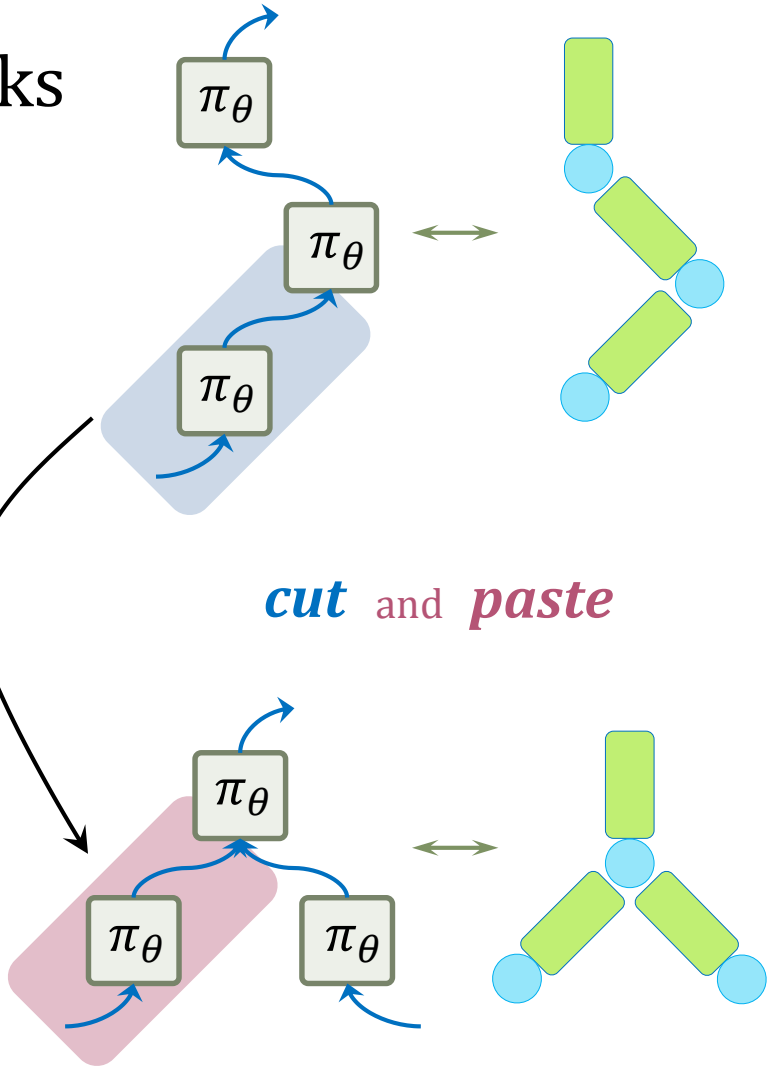
cut and ***paste***



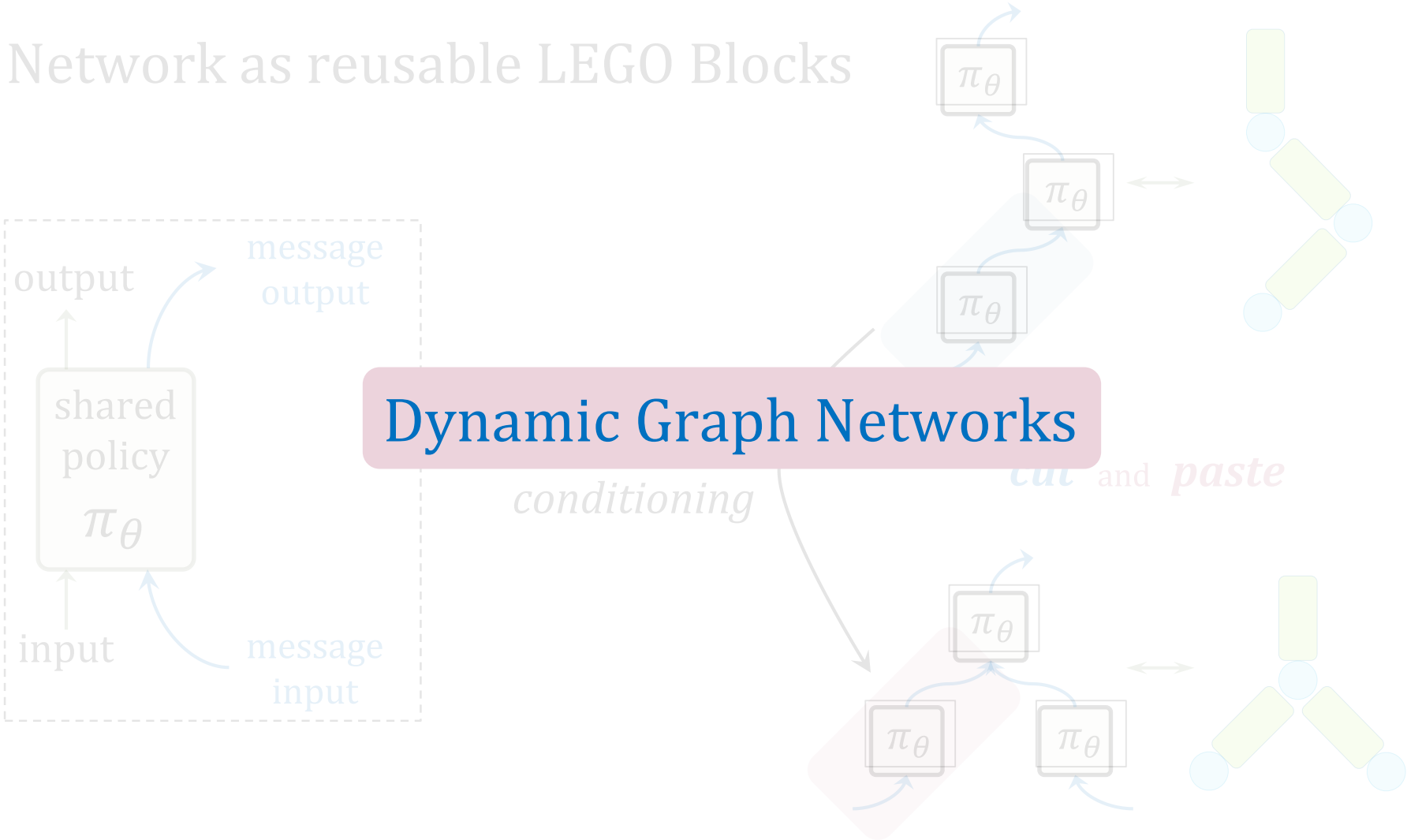
Network as reusable LEGO Blocks



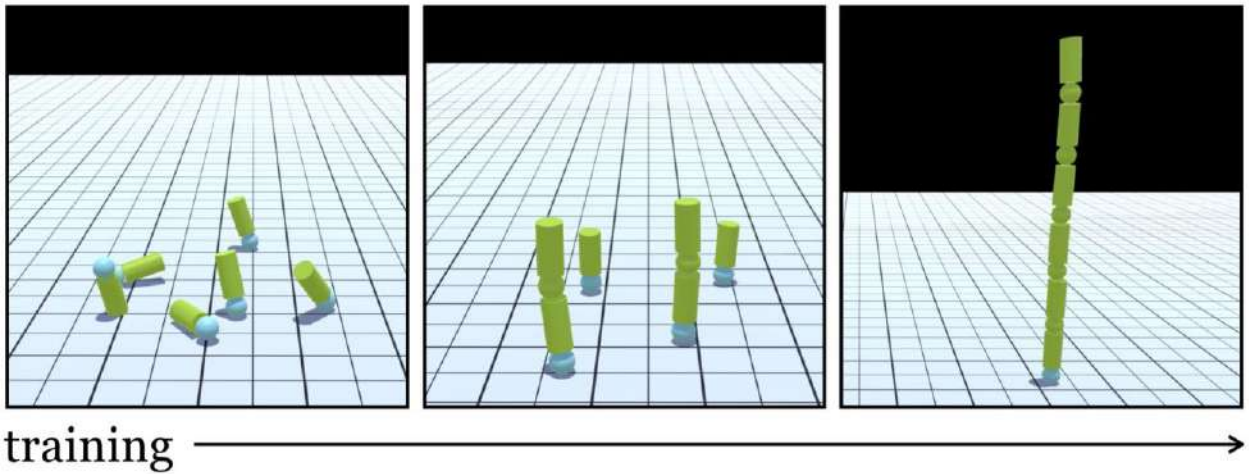
adaptation by conditioning



cut and ***paste***


Network as reusable LEGO Blocks
output
message output
shared policy
π_θ
input
message input
π_θ
π_θ
π_θ
Dynamic Graph Networks
conditioning
cut and paste
π_θ
π_θ
π_θ

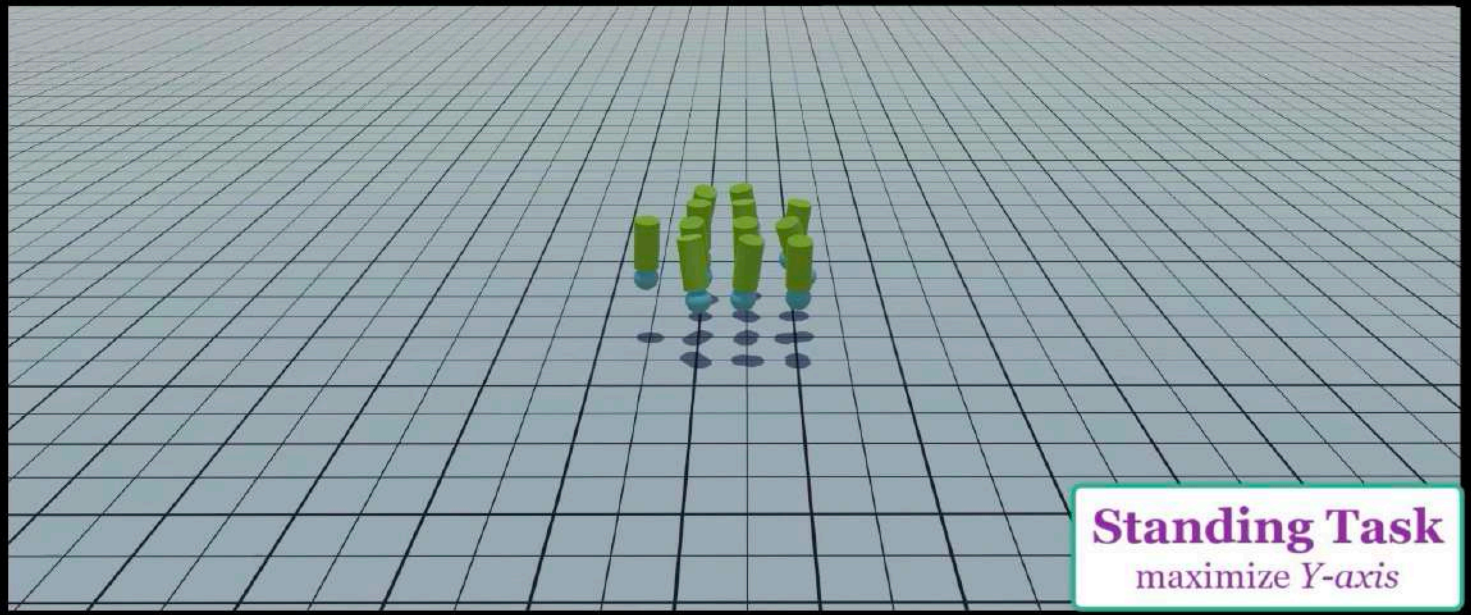
BTW, basically curriculum learning but in hardware



How well does it generalize?

Generalization w/o Fine-tuning

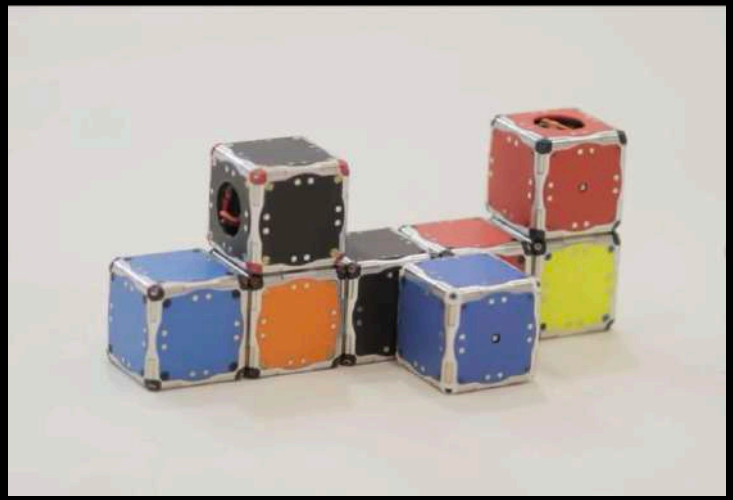
twice as many limbs



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!