

Learning to Control Self-Assembling Morphologies

A Study of Generalization via Modularity

Deepak Pathak* Chris Lu* Trevor Darrell Phillip Isola Alexei A. Efros

UC Berkeley (* equal contribution)

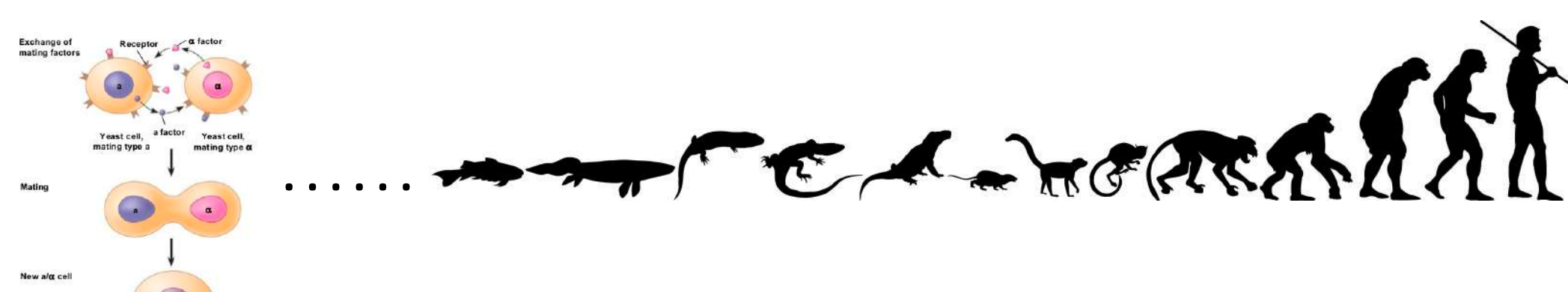
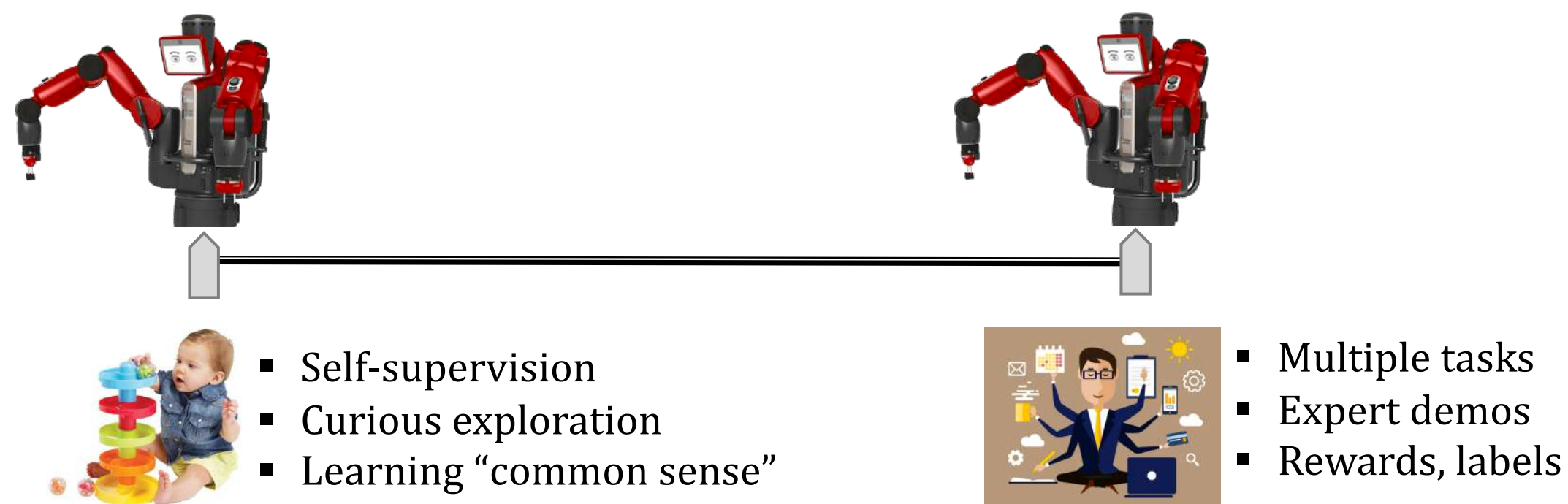


Code Released!

[https:// pathak22.github.io/modular-assemblies/](https://pathak22.github.io/modular-assemblies/)

How to train a robot to act?

Software starts simple, but hardware remains fixed & complex!



single → multicellular
(i.e. *competition* → *collaboration*)

Compositionality is useful in language (Andreas et.al. 2016). However, parser is fixed → tractable.

How to implement **compositionality in hardware?**

Modular Co-evolution of Control and Morphology



➤ **Input** = Local Sensory State

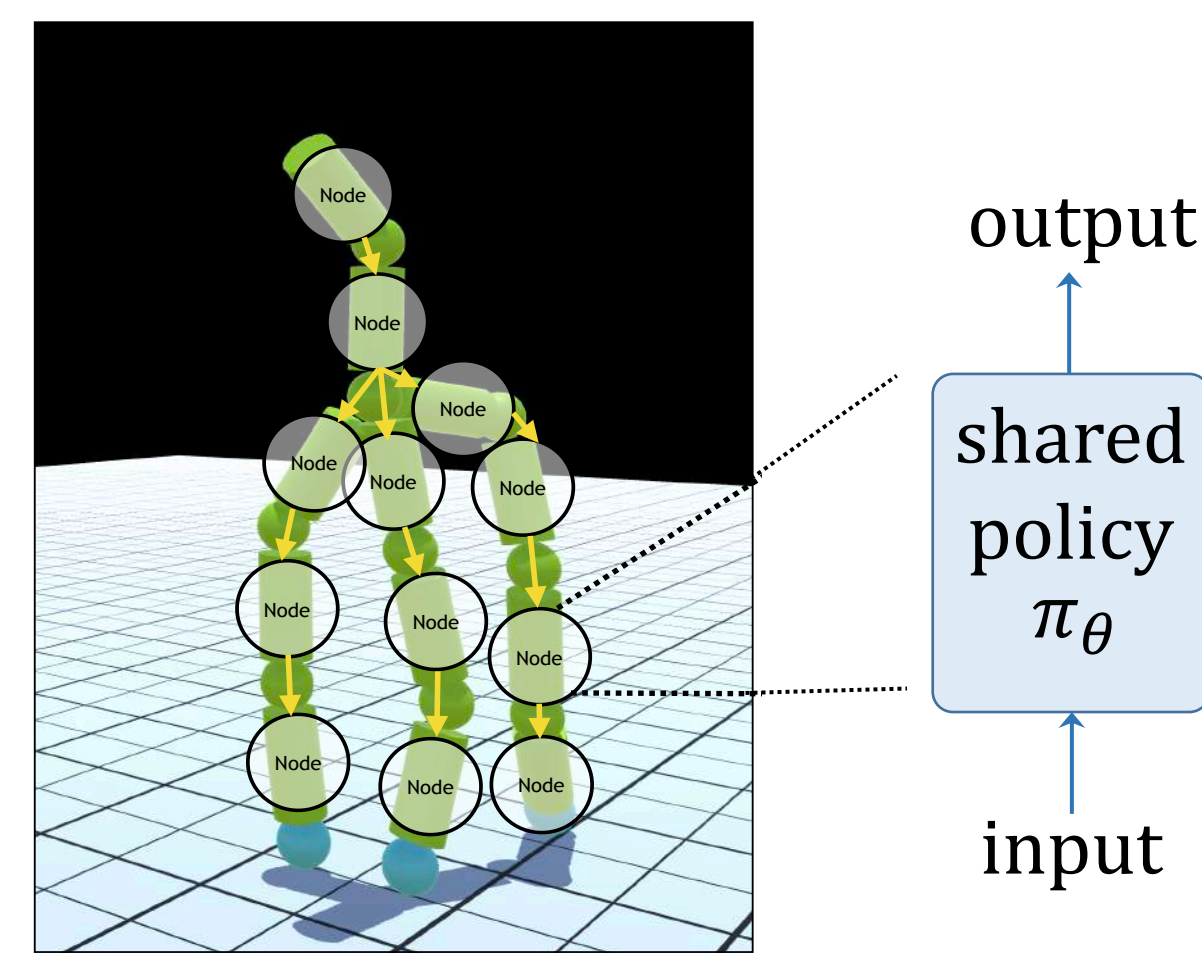
➤ **Output** = Torques, Link, Unlink

Acts as single agent upon joining.

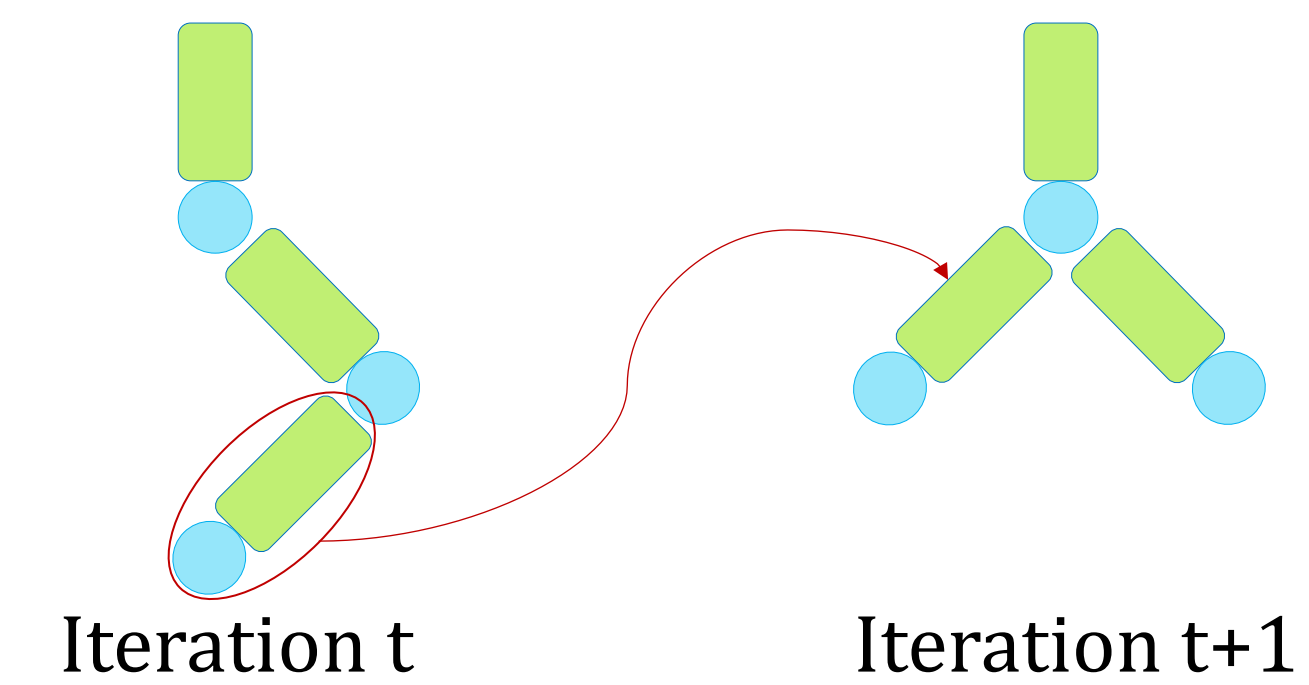
Rewards are shared!

Dynamic Graph Networks (DGN)

Idea 1: Separate policy for each limb with shared params



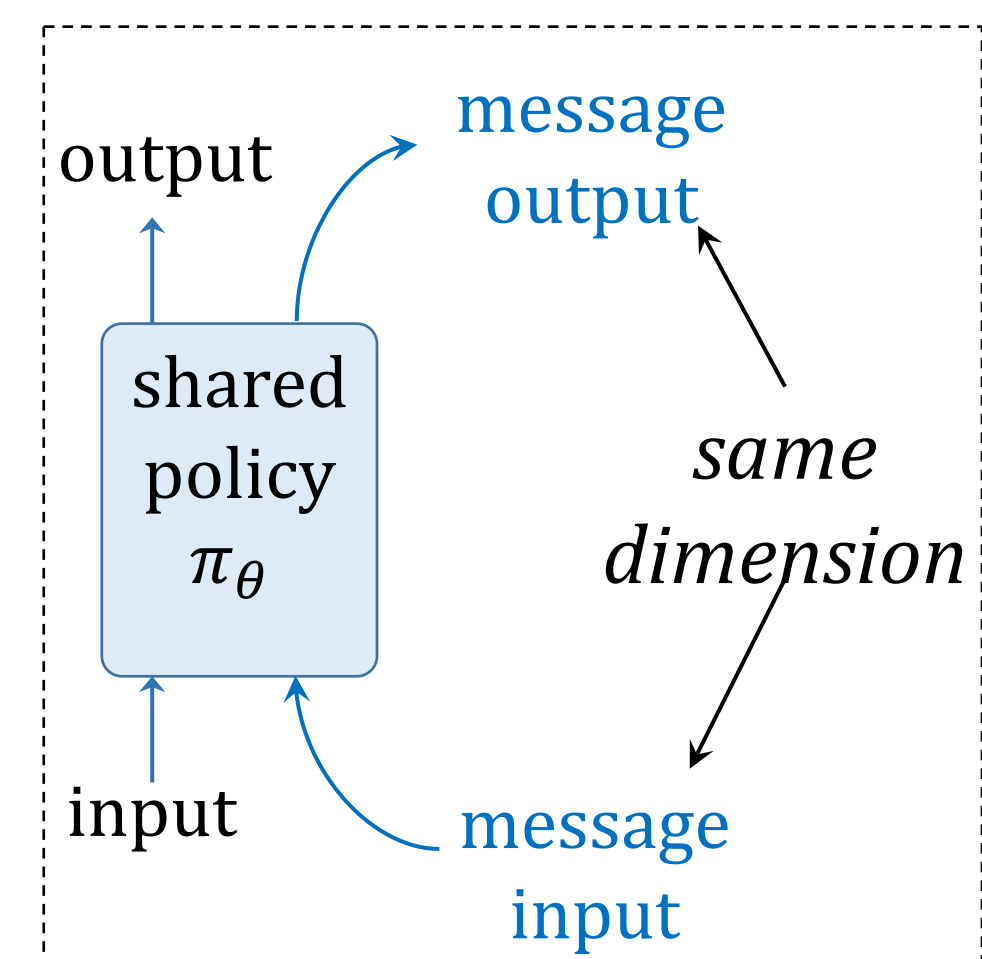
How to adapt when hardware changes?



Adapt by Finetuning? ❌

Too slow as shape changes every iteration!

Idea 2: Neural Networks as reusable LEGO Blocks



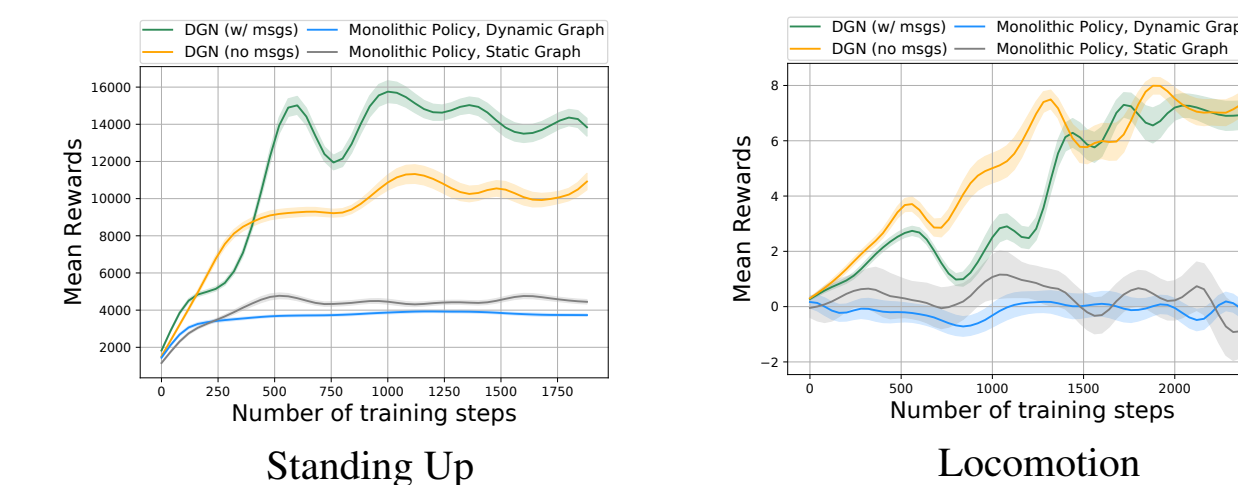
Message Passing:
Children pass messages to parents for communication.

adaptation by conditioning

cut and paste

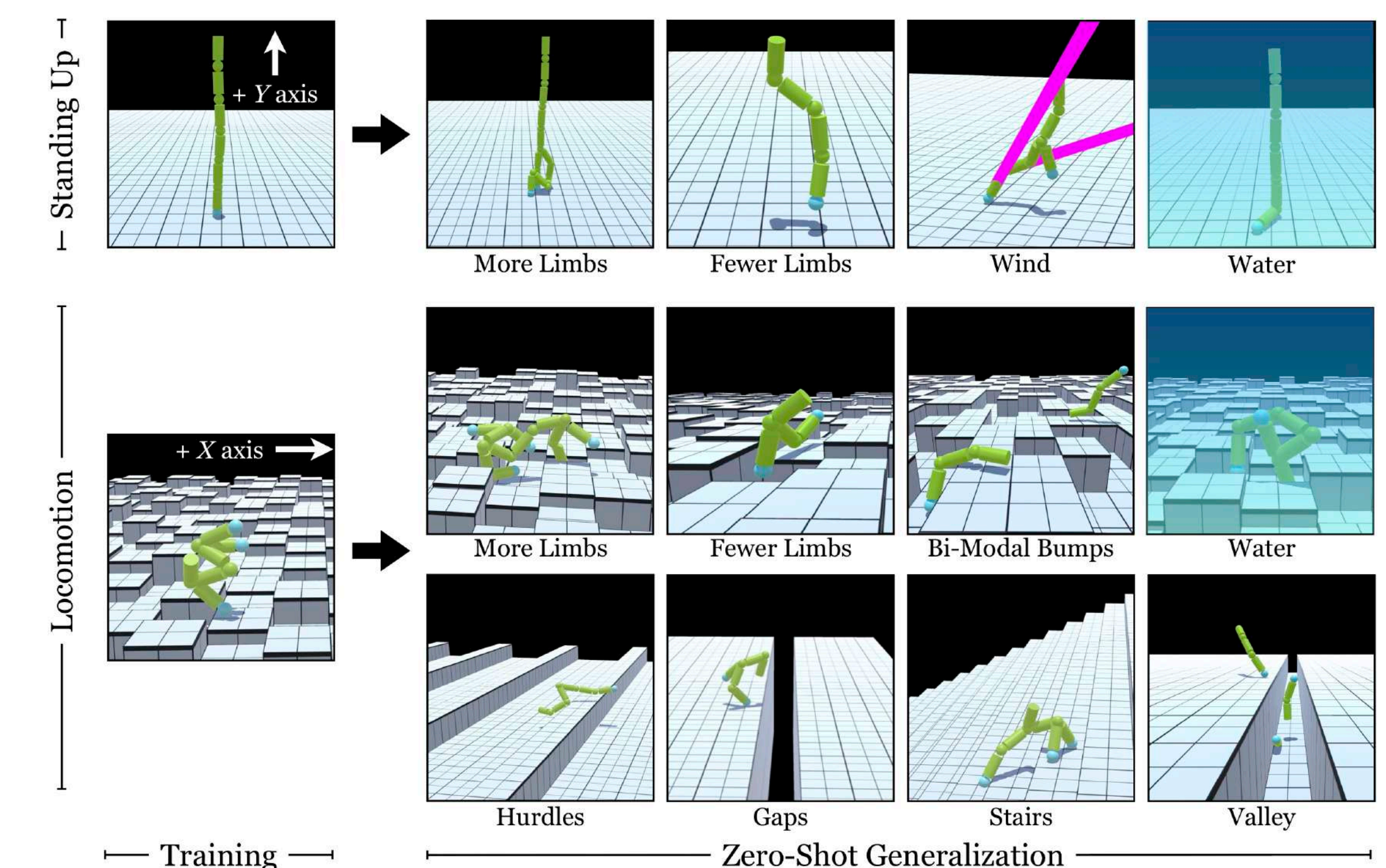
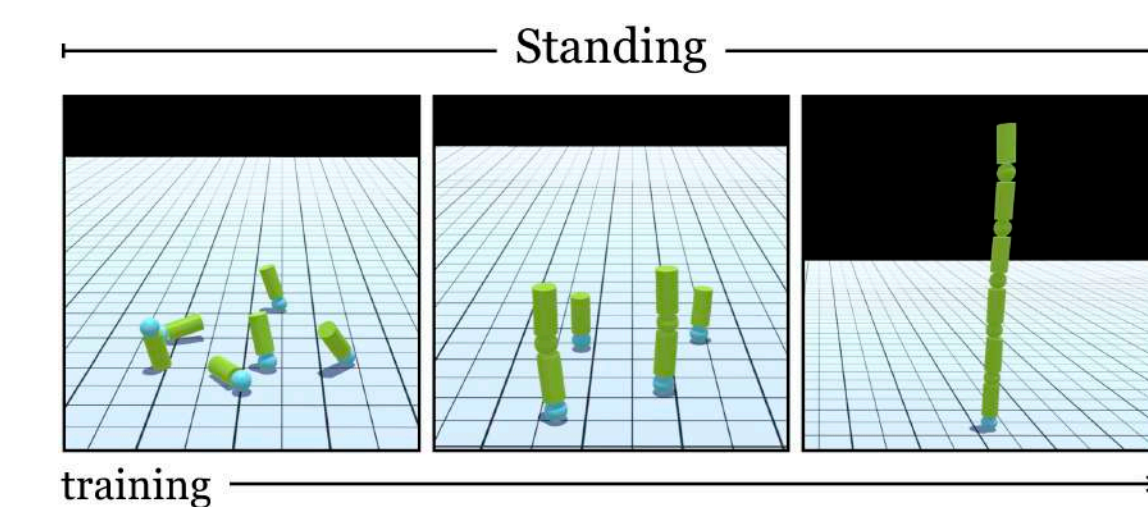
Dynamic + Graph → Dynamic Graph Nets

Generalization via Modularity



Message-passing helps!

Curriculum learning but in hardware.



Self-Assembling Robots in the Real World?

[Mark Yim's Lab at UPenn]

[Daniela Rus's Lab at MIT]

[Modular Snake Robot - Howie Choset's Lab at CMU]