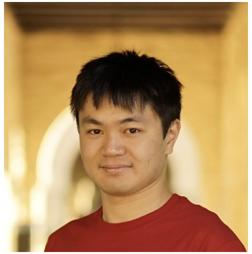
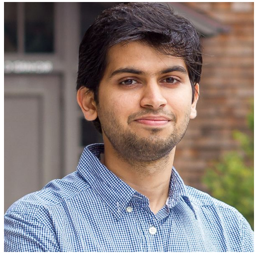
REvolveR: Continuous Evolutionary Models for Robot-to-robot Policy Transfer

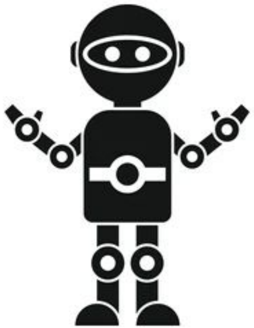
Xingyu Liu, Deepak Pathak, Kris M. Kitani

Carnegie Mellon University
The Robotics Institute

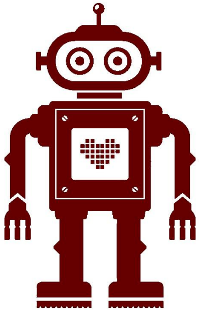
ICML
International Conference
On Machine Learning

Carnegie Mellon University
The Robotics Institute

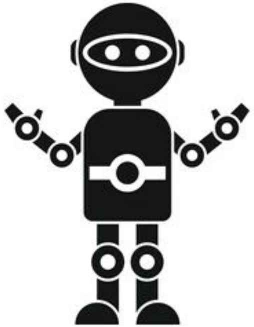
Problem Formulation: **Robot-to-robot Policy Transfer**

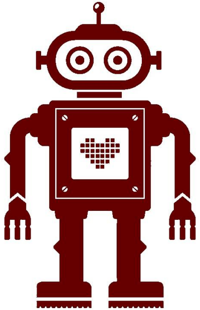


π_{teacher}



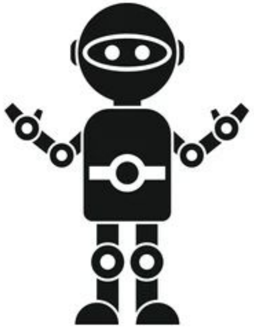
Problem Formulation: **Robot-to-robot Policy Transfer**

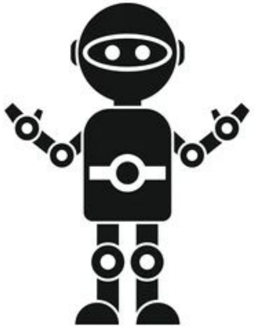




π_{teacher} $\xrightarrow{?}$ π_{student}

Existing Imitation Learning / Learning from Demonstration

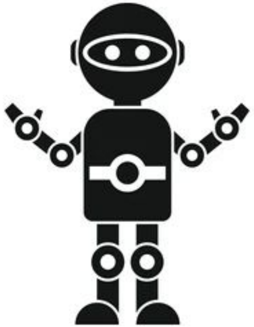




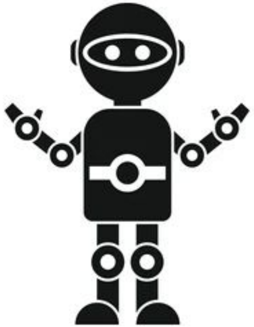
π_{teacher}

$$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$$

Existing Imitation Learning / Learning from Demonstration



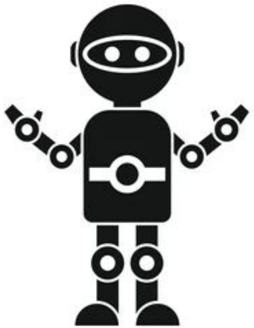
π_{teacher}

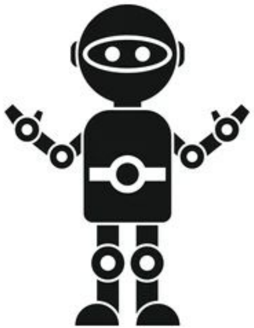


π_{student}

$$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$$

Existing Imitation Learning / Learning from Demonstration

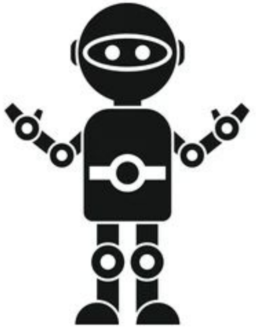


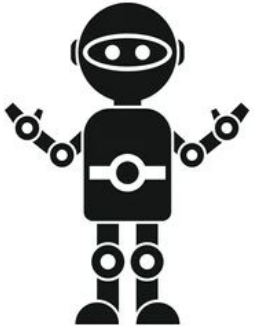


$\pi_{\text{teacher}}(a|s)$ ← Behavior Cloning → $\pi_{\text{student}}(a|s)$

$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$

Existing Imitation Learning / Learning from Demonstration



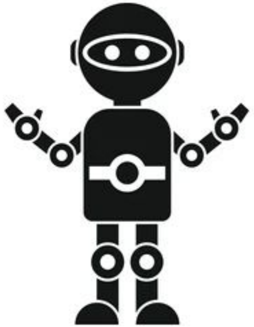


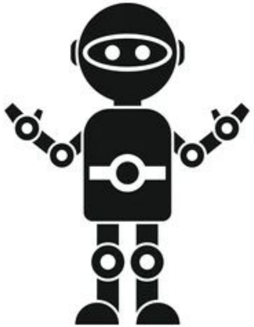
$\pi_{\text{teacher}}(a|s)$ ← Behavior Cloning → $\pi_{\text{student}}(a|s)$

$T_{\text{teacher}}(\pi_{\text{teacher}}(s), s)$ ← SOIL, SAIL → $T_{\text{student}}(\pi_{\text{student}}(s), s)$

$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$

Existing Imitation Learning / Learning from Demonstration





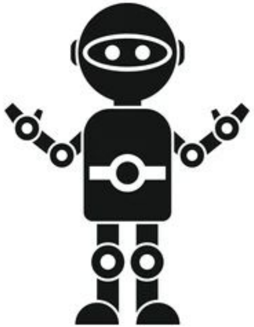
$\pi_{\text{teacher}}(a|s) \longleftarrow$ Behavior Cloning $\longrightarrow \pi_{\text{student}}(a|s)$

$T_{\text{teacher}}(\pi_{\text{teacher}}(s), s) \longleftarrow$ SOIL, SAIL $\longrightarrow T_{\text{student}}(\pi_{\text{student}}(s), s)$

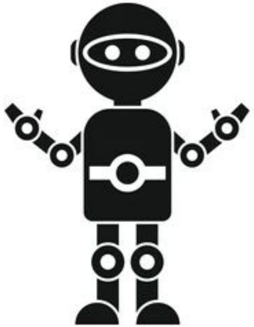
$R_{\text{teacher}}(\pi_{\text{teacher}}(s), s) \longleftarrow$ Inverse RL $\longrightarrow R_{\text{student}}(\pi_{\text{student}}(s), s)$

$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$

Existing Imitation Learning / Learning from Demonstration



Core Assumption:
MDP Transition Dynamics
T_{teacher} and T_{student} are Same or Similar



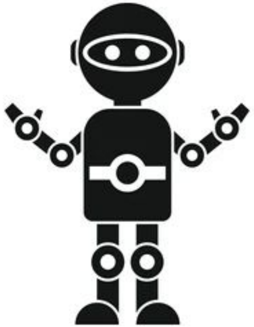
$\pi_{\text{teacher}}(a|s)$ ⟵ Behavior Cloning ⟶ $\pi_{\text{student}}(a|s)$

$T_{\text{teacher}}(\pi_{\text{teacher}}(s), s)$ ⟵ SOIL, SAIL ⟶ $T_{\text{student}}(\pi_{\text{student}}(s), s)$

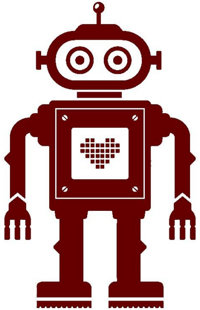
$R_{\text{teacher}}(\pi_{\text{teacher}}(s), s)$ ⟵ Inverse RL ⟶ $R_{\text{student}}(\pi_{\text{student}}(s), s)$

$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$

Existing Imitation Learning / Learning from Demonstration



But What if
MDP Transition Dynamics
T_{teacher} and T_{student} are **Very Different**?



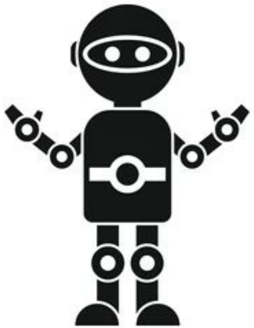
$\pi_{\text{teacher}}(a|s)$ ← Behavior Cloning → $\pi_{\text{student}}(a|s)$

$T_{\text{teacher}}(\pi_{\text{teacher}}(s), s)$ ← SOIL, SAIL → $T_{\text{student}}(\pi_{\text{student}}(s), s)$

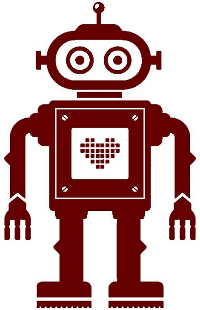
$R_{\text{teacher}}(\pi_{\text{teacher}}(s), s)$ ← Inverse RL → $R_{\text{student}}(\pi_{\text{student}}(s), s)$

$\{(s_{\text{teacher}}, a_{\text{teacher}}, s'_{\text{teacher}})\}$

Existing Imitation Learning / Learning from Demonstration



But What if
MDP Transition Dynamics
T_{teacher} and T_{student} are **Very Different**?



$\pi^*_{\text{teacher}}(a|s)$

$T_{\text{teacher}}(\pi^*_{\text{teacher}}(s), s)$

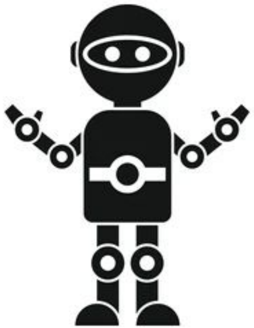
$R_{\text{teacher}}(\pi^*_{\text{teacher}}(s), s)$

$\pi^*_{\text{student}}(a|s)$

$T_{\text{student}}(\pi^*_{\text{student}}(s), s)$

$R_{\text{student}}(\pi^*_{\text{student}}(s), s)$

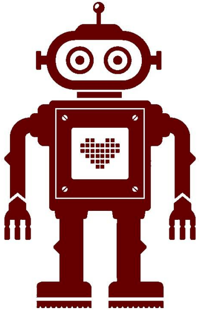
Existing Imitation Learning / Learning from Demonstration



But What if

MDP Transition Dynamics

T_{teacher} and T_{student} are **Very Different**?



π_{teacher}

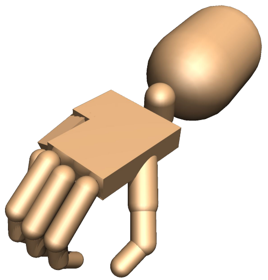
Behavior Cloning

SOIL, SAIL

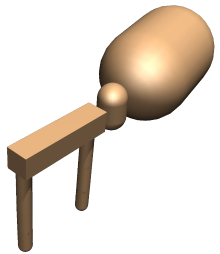
Inverse RL

π_{student}

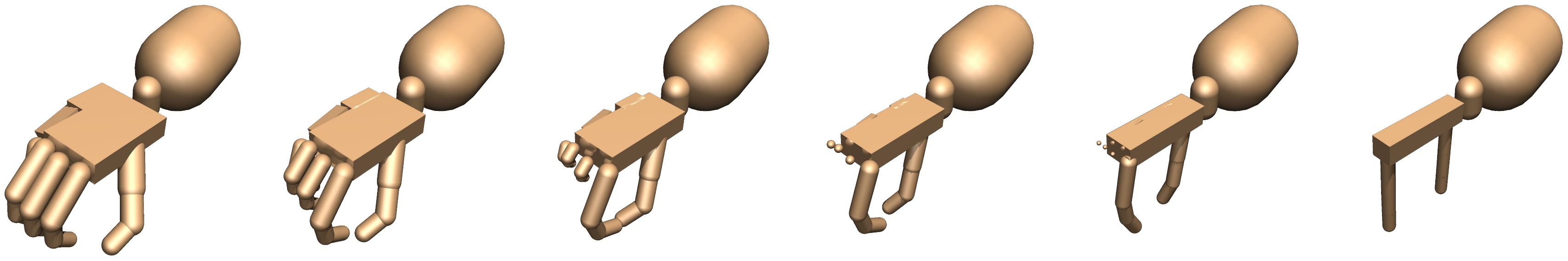
Proposed Paradigm: Continuous Robot Evolution



π_{source}



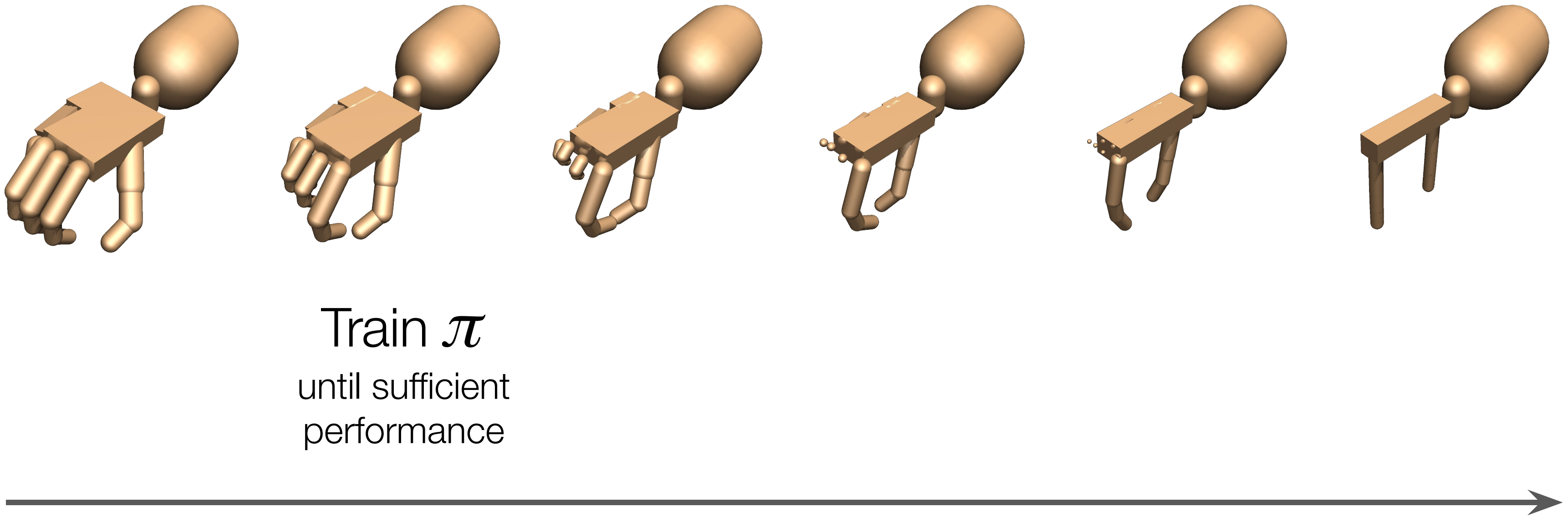
Proposed Paradigm: Continuous Robot Evolution



π_{source}

"Bridge" between source and target robots

Proposed Paradigm: Continuous Robot Evolution

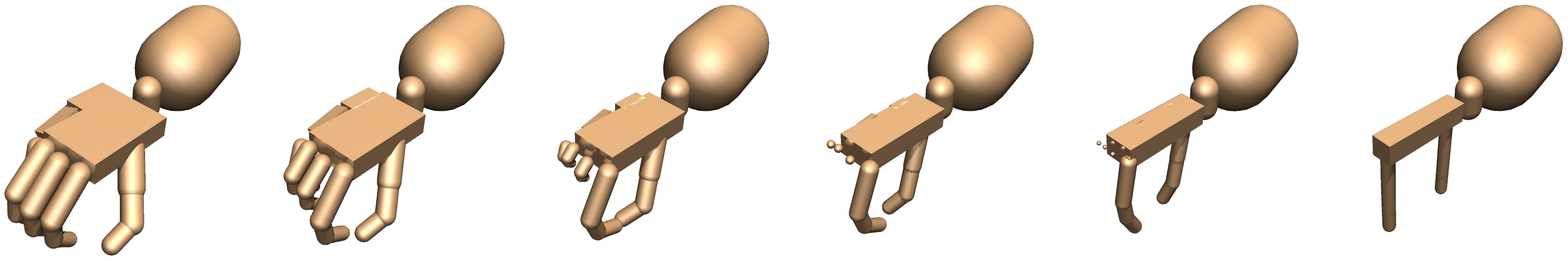


Train π

until sufficient
performance

Progressively train policy on a sequence of intermediate robots

Proposed Paradigm: Continuous Robot Evolution

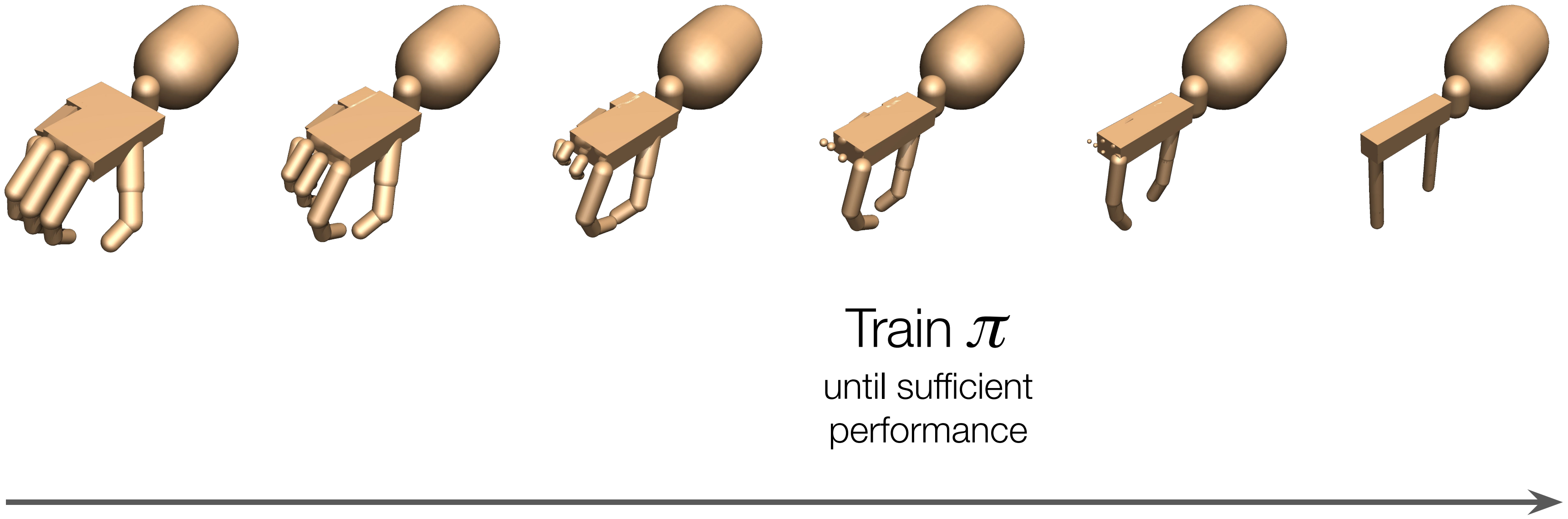


Train π

until sufficient
performance

Progressively train policy on a sequence of intermediate robots

Proposed Paradigm: Continuous Robot Evolution

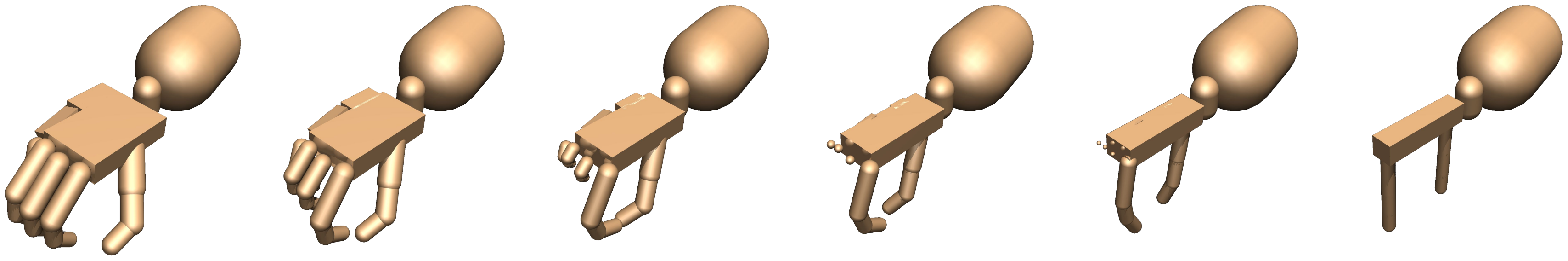


Train π

until sufficient performance

Progressively train policy on a sequence of intermediate robots

Proposed Paradigm: Continuous Robot Evolution

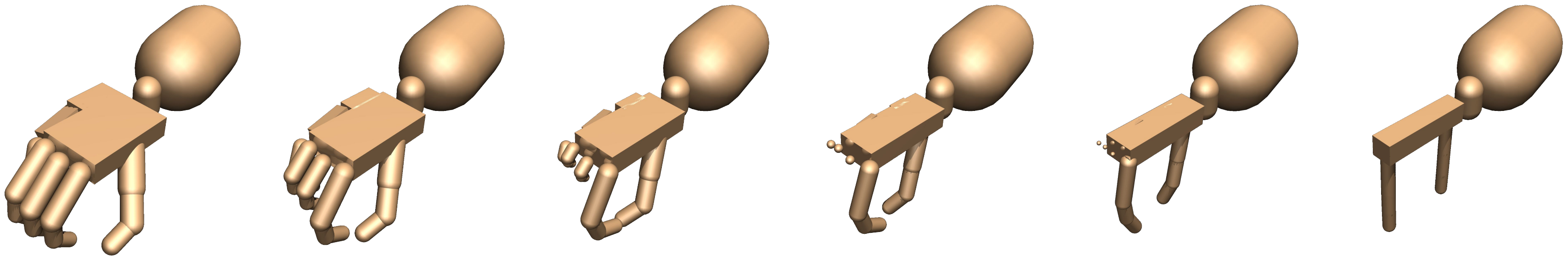


Train π

until sufficient performance

Progressively train policy on a sequence of intermediate robots

Proposed Paradigm: Continuous Robot Evolution

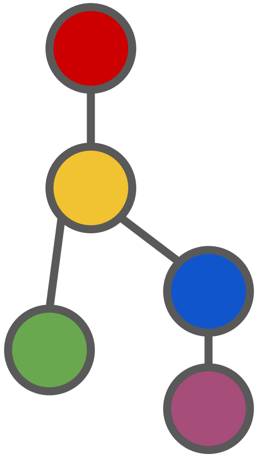


π_{target}

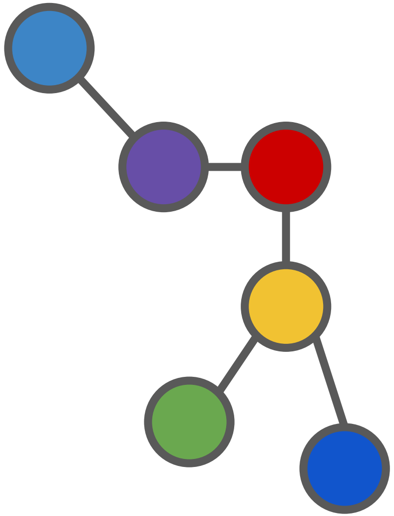
Progressively train policy on a sequence of intermediate robots

Continuous Robot Evolution**: How?**

Continuous Robot Evolution **Step 1: Morphology Matching**

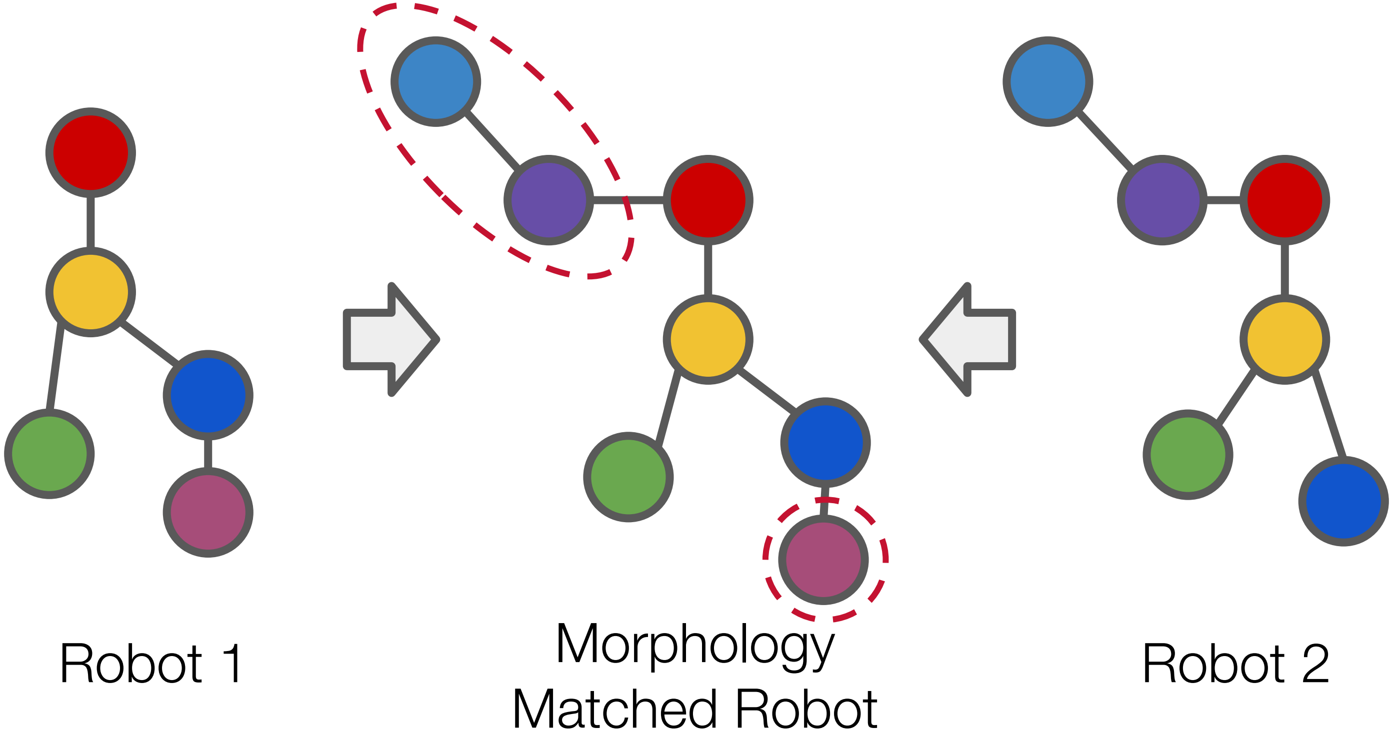


Robot 1

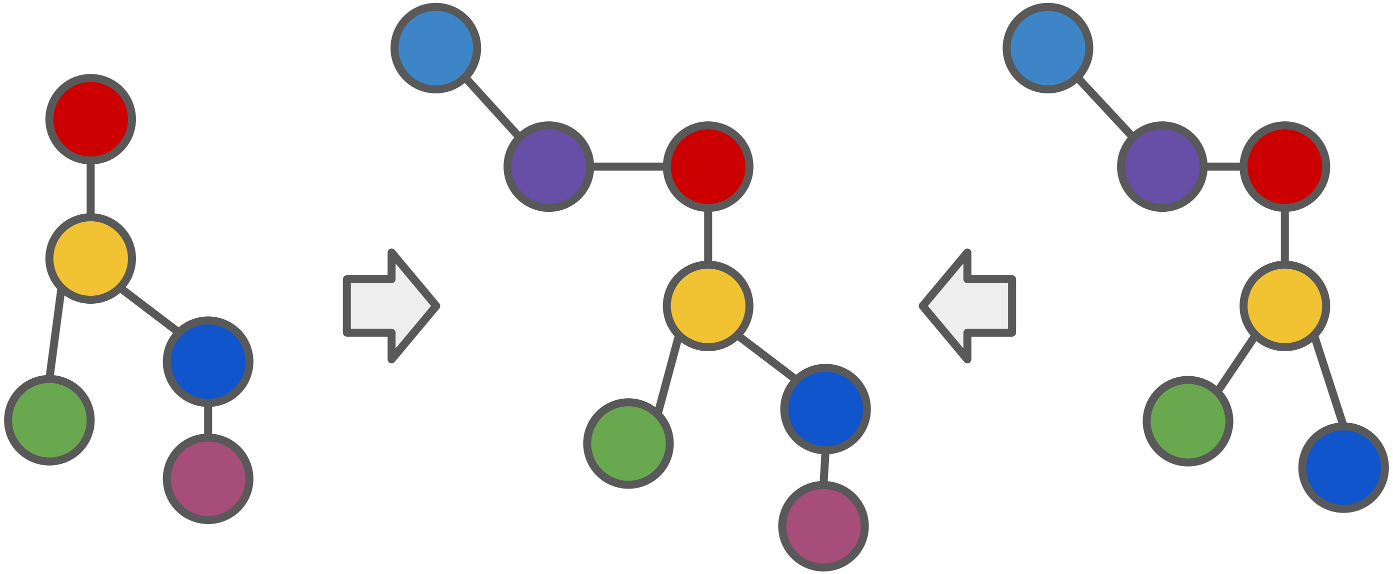


Robot 2

Continuous Robot Evolution **Step 1: Morphology Matching**

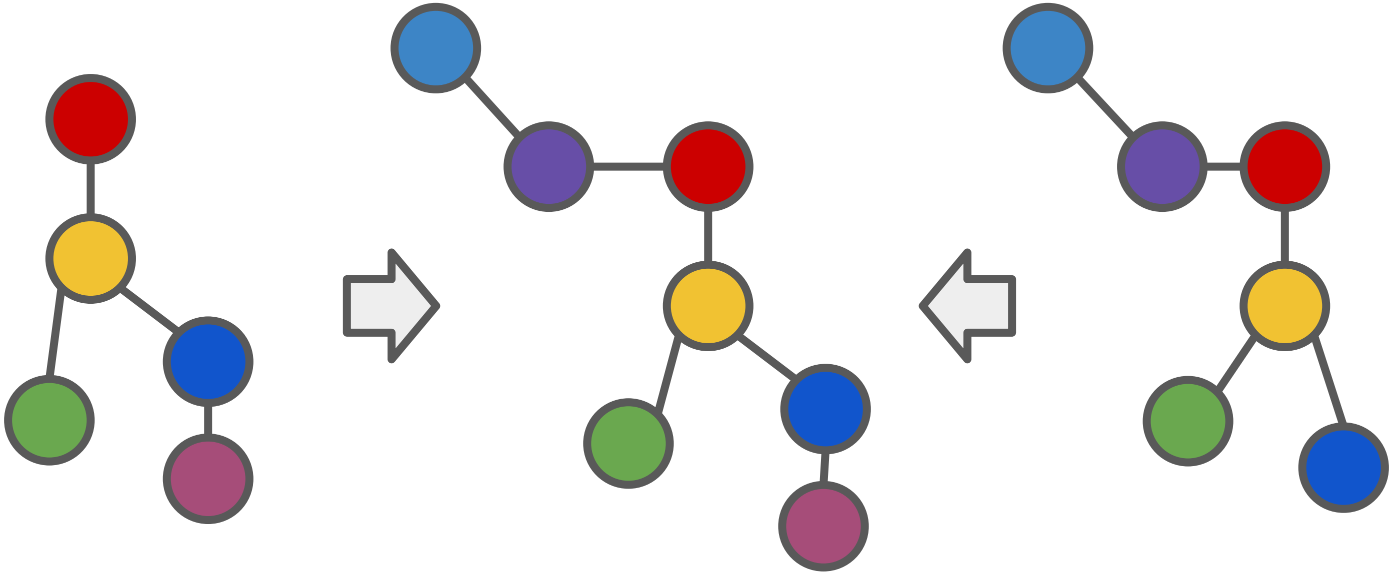


Continuous Robot Evolution **Step 1: Morphology Matching**



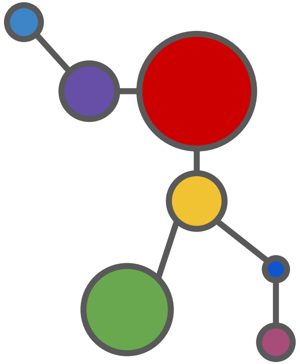
After this step, state and action space of the two robots are unified!

Continuous Robot Evolution **Step 1: Morphology Matching**

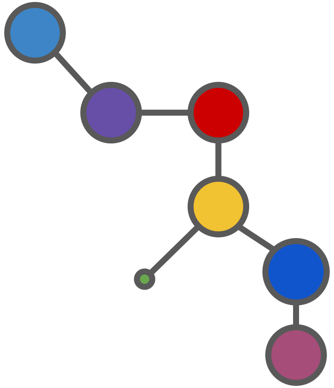


Now the only difference is transition dynamics

Continuous Robot Evolution **Step 2: Kinematic Interpolation**

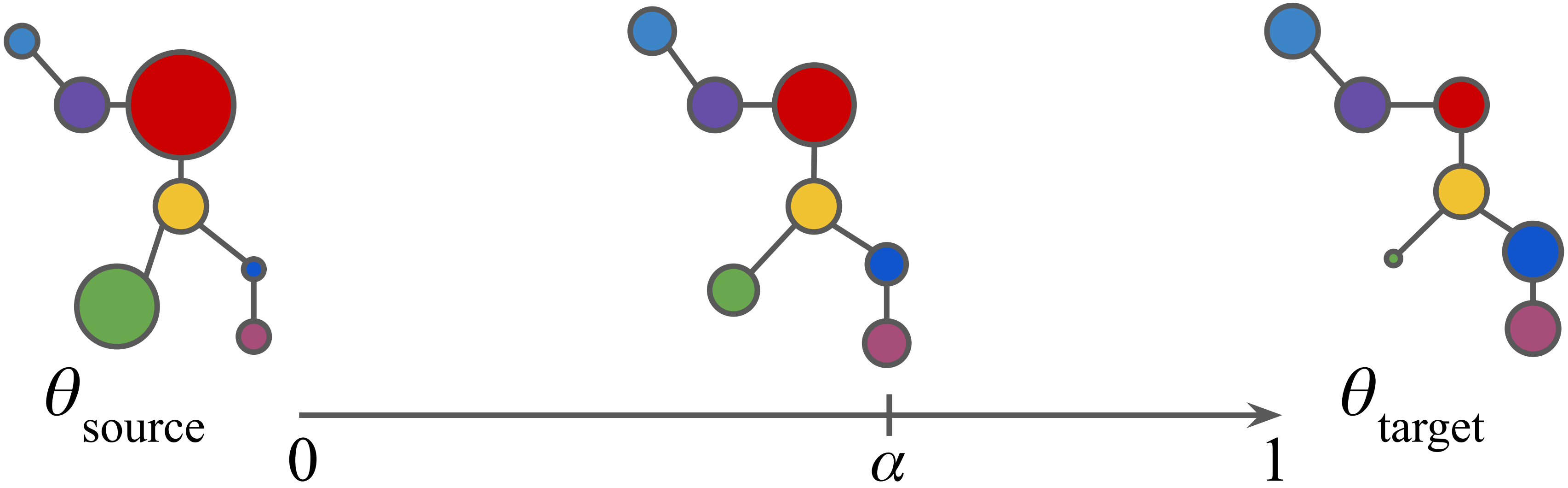


θ_{source}



θ_{target}

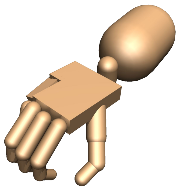
Continuous Robot Evolution **Step 2: Kinematic Interpolation**

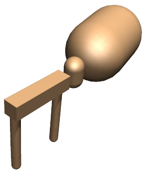


Hardware parameter of the robot at evolution progress α

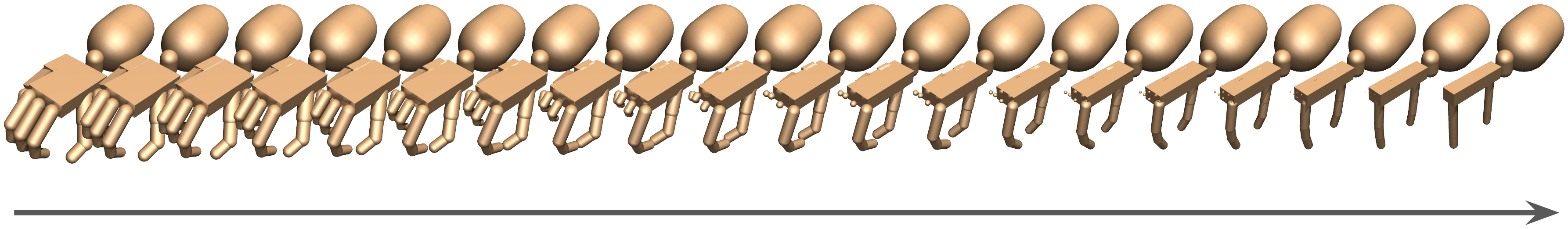
$$\theta(\alpha) = (1-\alpha)\cdot\theta_{\text{source}} + \alpha\cdot\theta_{\text{target}}$$

Policy Transfer on Continuously Evolving Robots

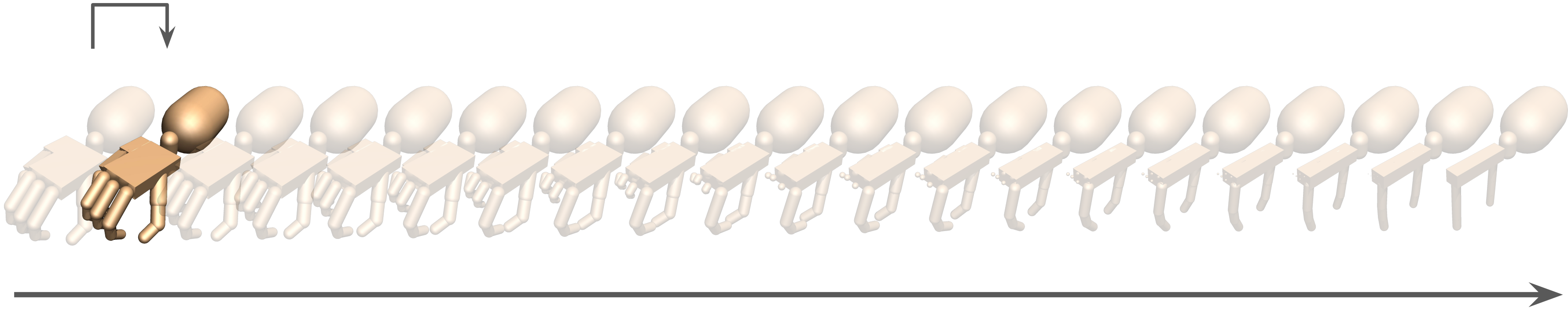




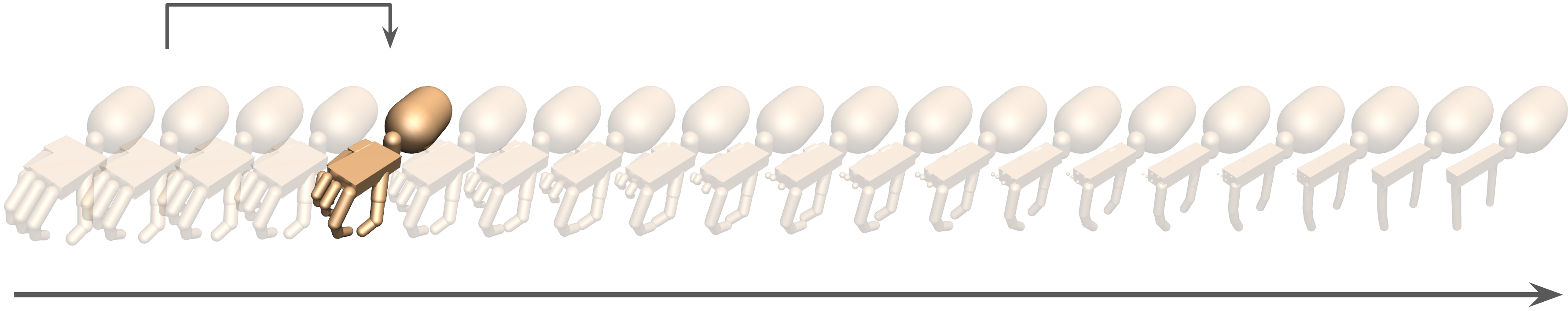
Policy Transfer on Continuously Evolving Robots



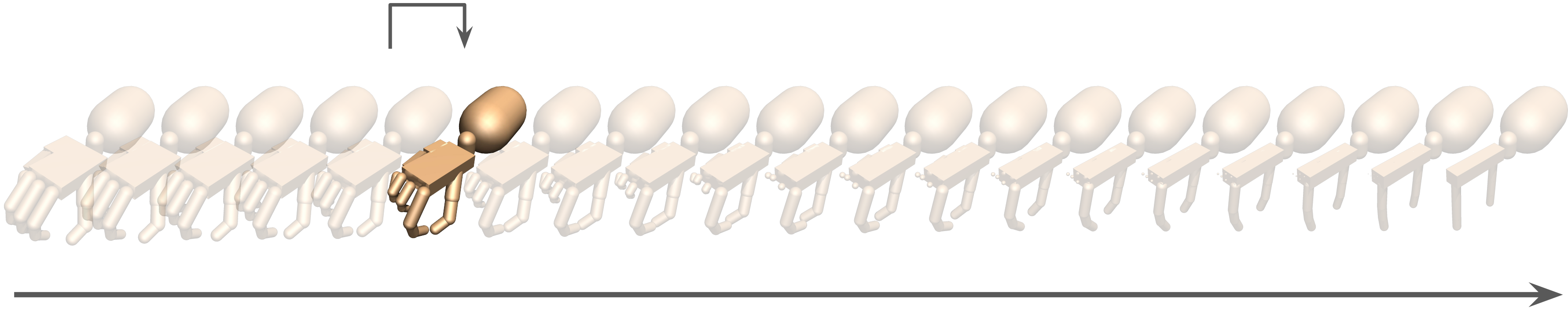
Policy Transfer on Continuously Evolving Robots: **Naive**



Policy Transfer on Continuously Evolving Robots: **Naive**

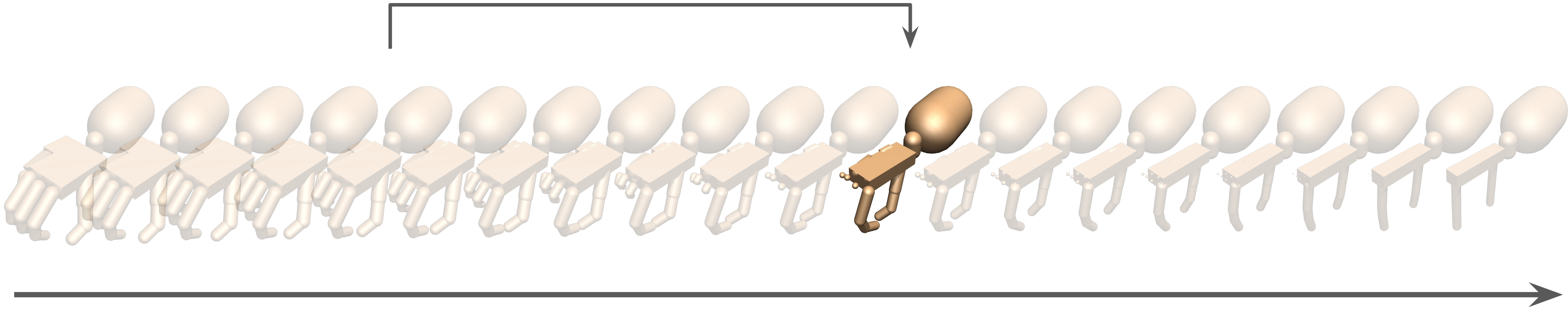


Policy Transfer on Continuously Evolving Robots: **Naive**



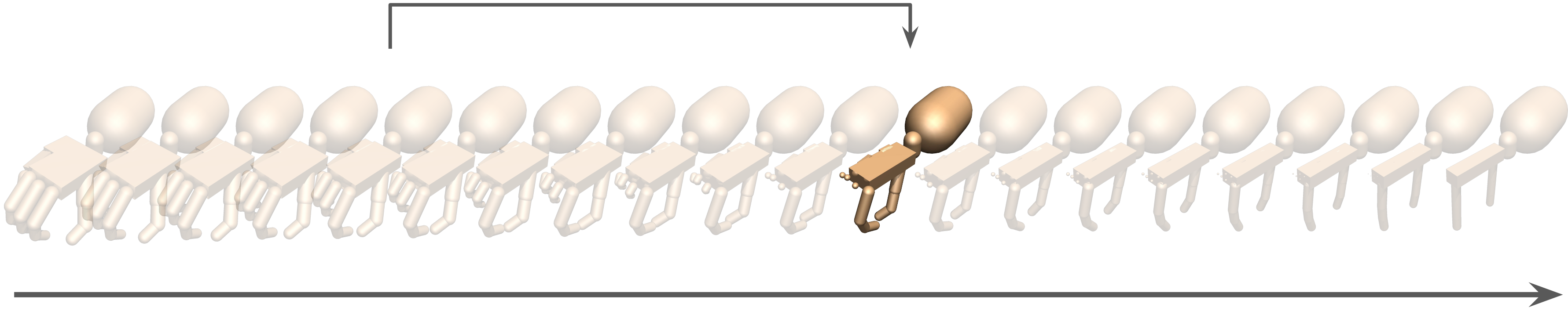
Small evolution progress?

Policy Transfer on Continuously Evolving Robots: **Naive**



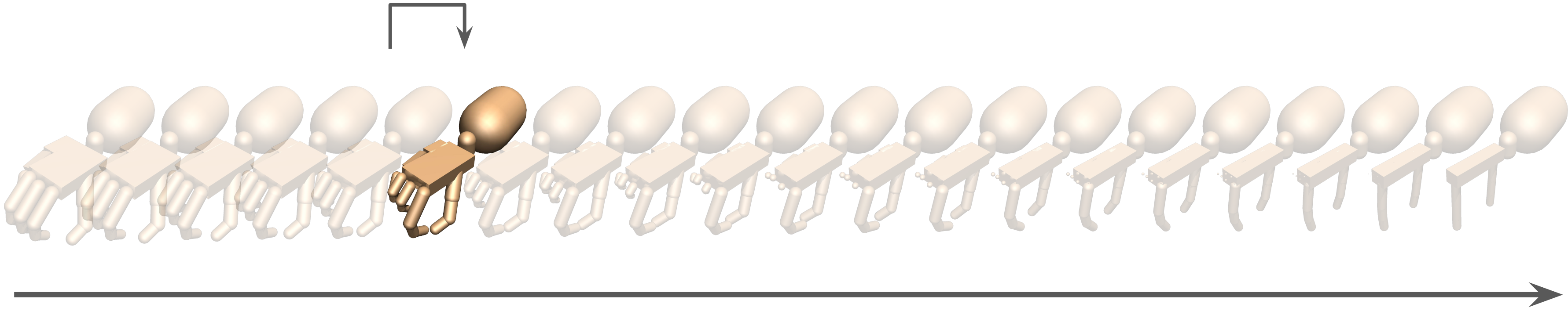
Large evolution progress?

Policy Transfer on Continuously Evolving Robots: **Naive**



What is the best evolution progression step size?

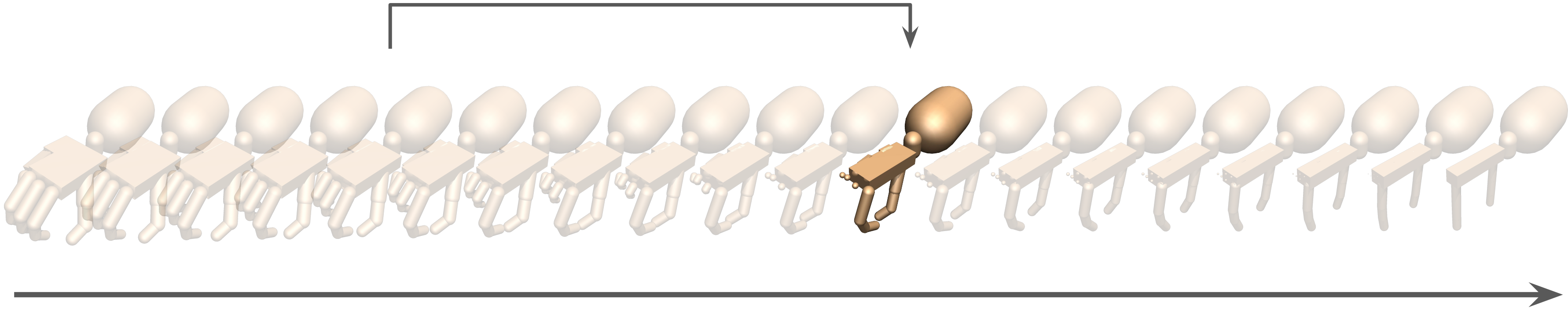
Policy Transfer on Continuously Evolving Robots: **Naive**



What is the best evolution progression step size?

Too small: waste RL iterations on too small robot changes

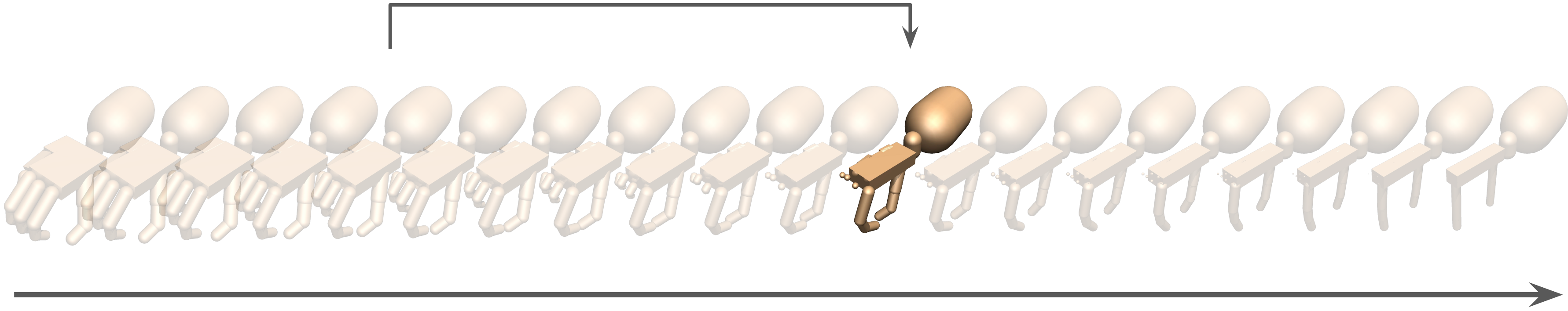
Policy Transfer on Continuously Evolving Robots: **Naive**



What is the best evolution progression step size?

Too small: waste RL iterations on too small robot changes
Too large: reward / success rate drop and hurt sample efficiency

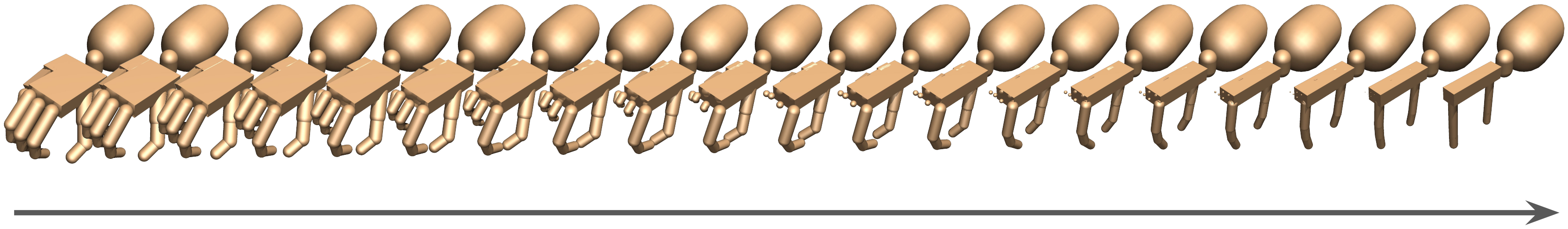
Policy Transfer on Continuously Evolving Robots: **Naive**



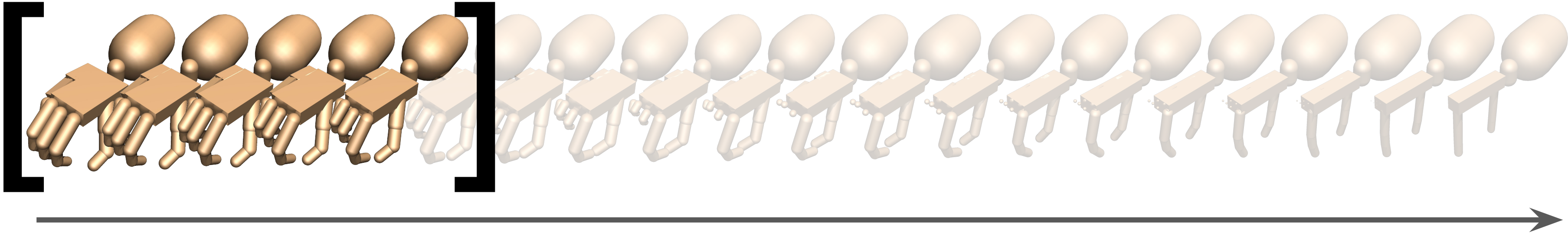
What is the best evolution progression step size?

The best evolution step size cannot be predicted beforehand

Proposed: Local Randomized Evolution Progression

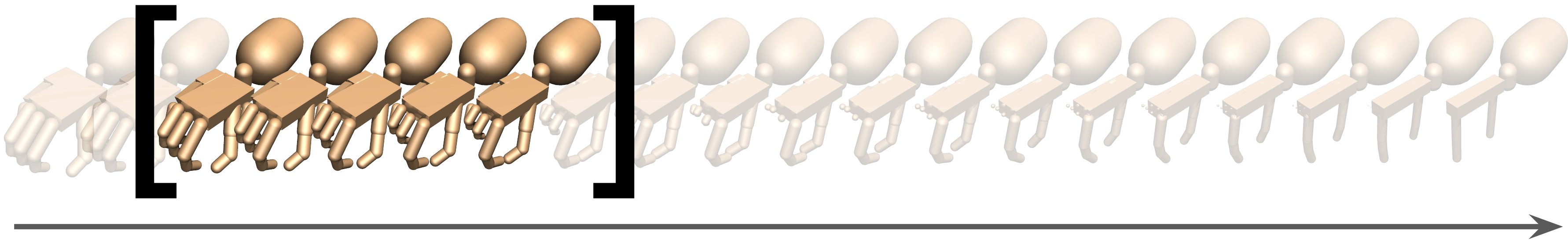


Proposed: Local Randomized Evolution Progression



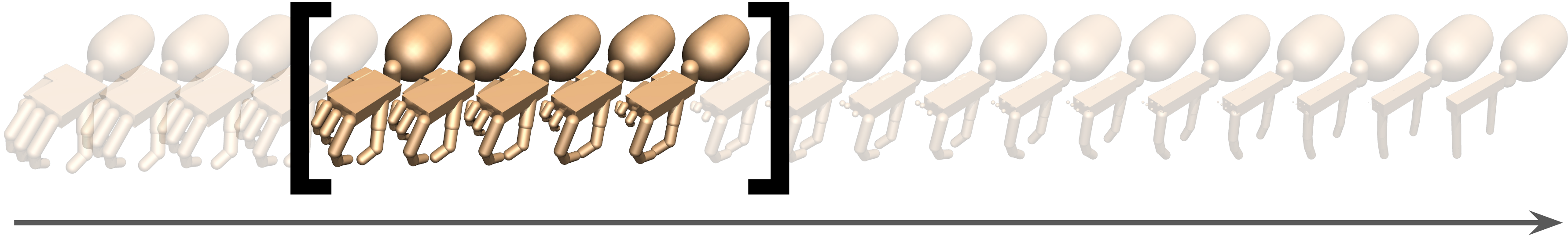
Sample from a moving window: both small and large evolution progress step sizes

Proposed: Local Randomized Evolution Progression



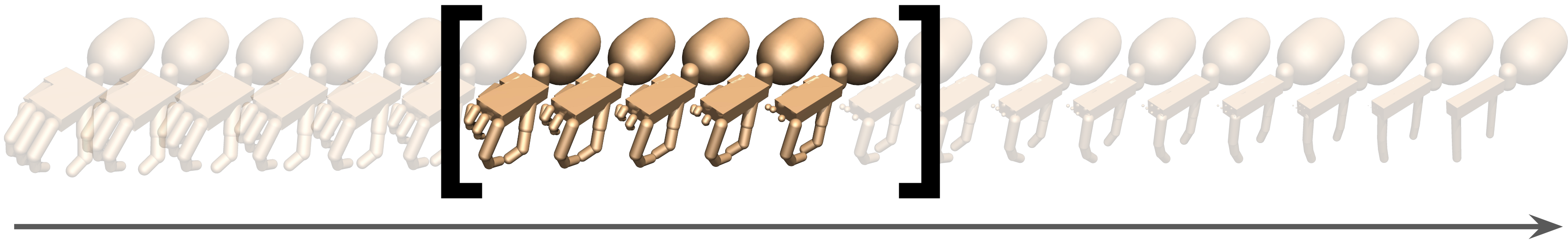
Sample from a moving window: both small and large evolution progress step sizes

Proposed: Local Randomized Evolution Progression



Sample from a moving window: both small and large evolution progress step sizes

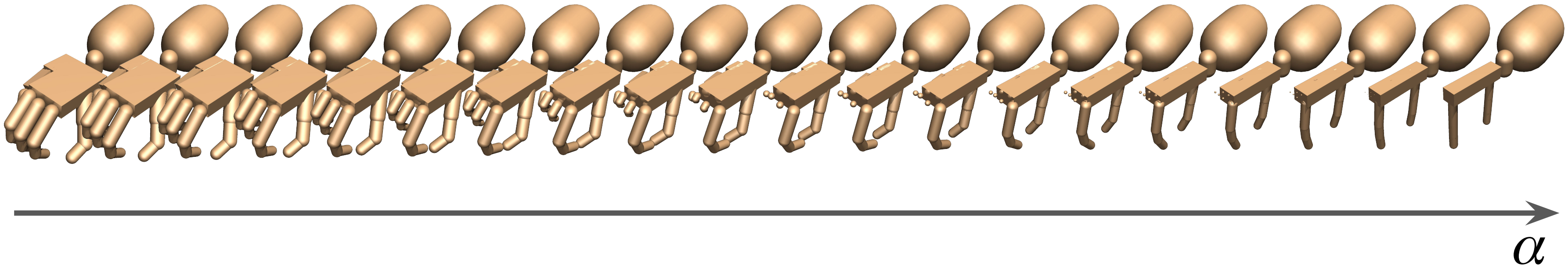
Proposed: Local Randomized Evolution Progression



Sample from a moving window: both small and large evolution progress step sizes

Small step size: maintain sufficient sample efficiency
Large step size: risk on large evolution to improve adaptation

Proposed: Evolution Reward Shaping



$$r'_t = r_t \cdot \exp(h \cdot \alpha)$$

Put more weight on reward received from robots with larger evolution progress α to improve adaptation towards target robot

Proposed: Evolution Reward Shaping

Theoretical results show the relationship between the evolution reward shaping and the optimization objective

Theorem 4.1. *Suppose the policy that optimizes the objective in Equation (6) with evolution reward shaping factor of* h *is the optimal policy* $\pi^*_{M_\varphi}$ *on robot* $M_\varphi = E(\varphi), \varphi \in [\alpha_k, \alpha_k + \xi]$, *i.e.*

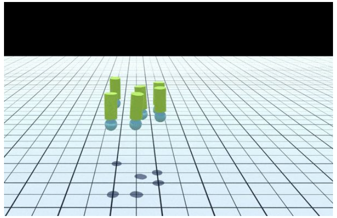
$$\arg\max_{\pi} \mathop{\mathbb{E}}_{\substack{\beta \sim U(\alpha_k, \alpha_k + \xi) \\ M_\beta = E(\beta)}} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim M_\beta(\cdot|s_t, a_t)}} \sum_t \gamma^t r_t \exp(h \cdot \beta)$$
$$= \pi^*_{M_\varphi} = \arg\max_{\pi} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim M_\varphi(\cdot|s_t, a_t)}} \sum_t \gamma^t r_t \tag{7}$$

Then when $\xi \to 0$, $\varphi = \alpha_k + \frac{1}{2}\xi + \frac{1}{4}h\xi^2 + o(\xi^2)$.

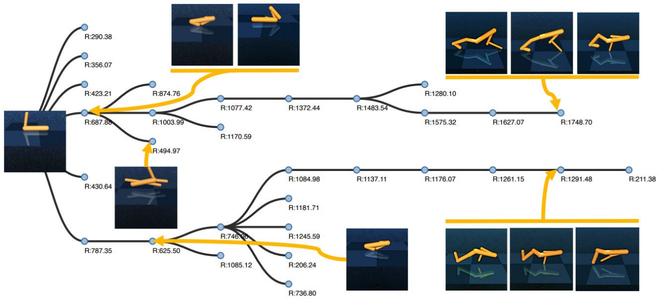
Related Works (1/2)

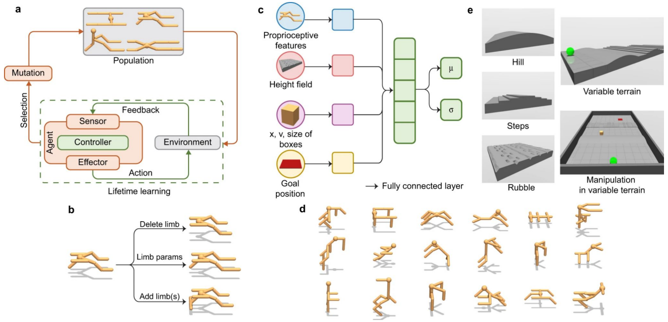
To Discover Robots that Generalize Better:

Evolved Virtual Creatures
K. Sims, SIGGRAPH 1994



Self-Assembling Agents
D. Pathak et al., NeurIPS, 2019





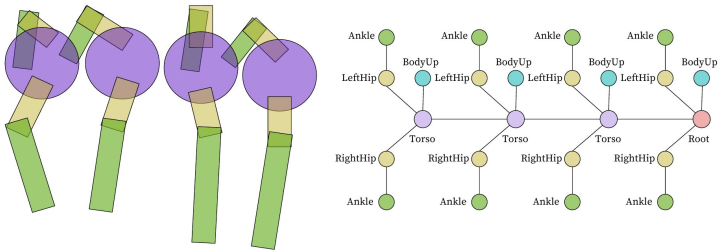
Neural Graph Evolution for Robot Design
T. Wang et al., ICLR 2019

Embodied Intelligence via Learning and Evolution
A. Gupta et al., Nature Communications 2021

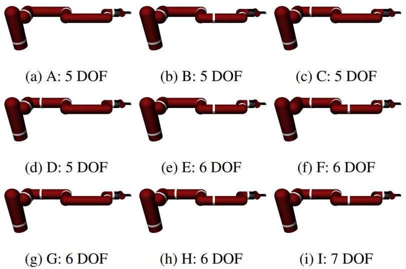
Ours: Transfer the Policy from a **Source** Robot to a **Predetermined Target** Robot

Related Works (2/2)

To Build Controllers that Generalize Across Robots:

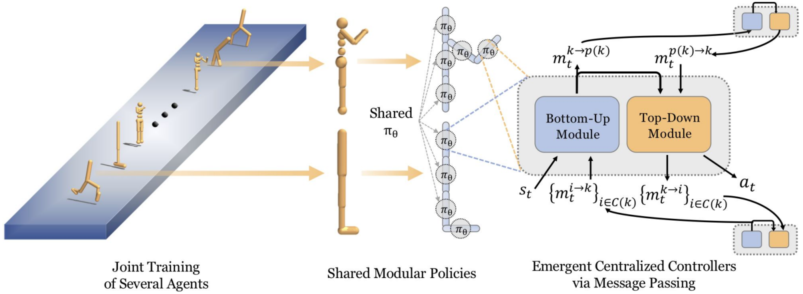


NerveNet: Structured Policy with GNN
T. Wang, ICLR 2018



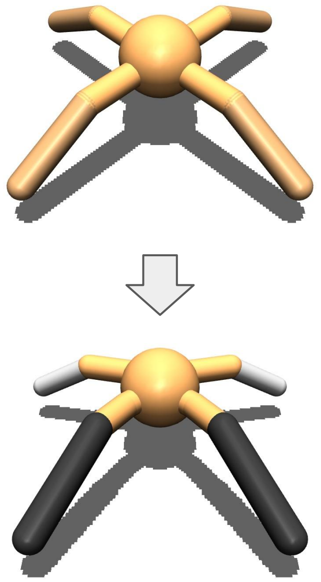
Hardware Conditioned Policies
T. Chen, NeurIPS 2018

Shared Modular Policies for Agent-Agnostic Control
W. Huang, ICML 2020

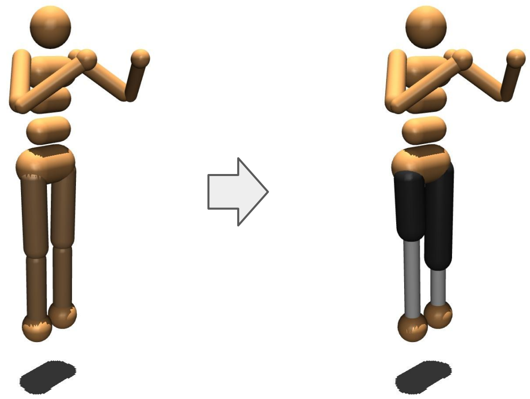


Ours: Assume Given Good Controller for Some Robot, Generate Controller for Some New Robot; **Does Not Need to Generalize** Across Robots

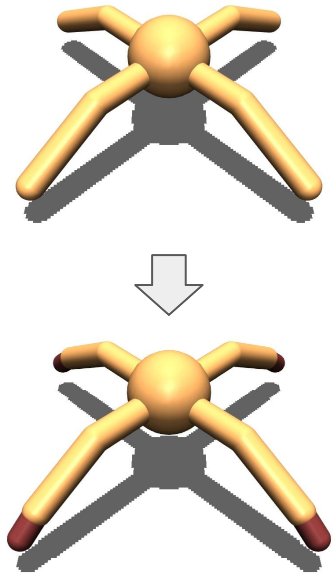
Experiments: MuJoCo Gym



Ant-length-mass

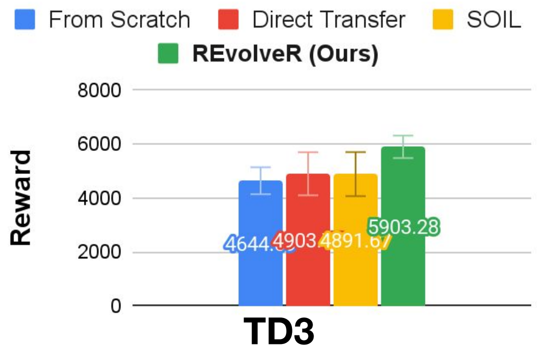


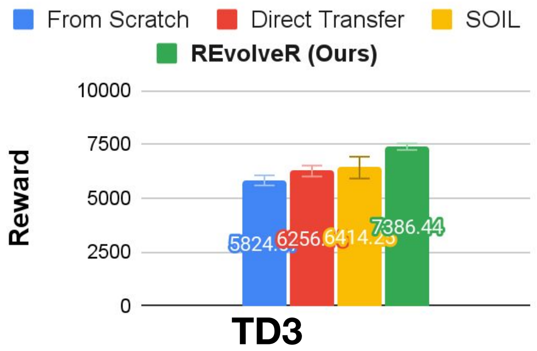
Humanoid-length-mass

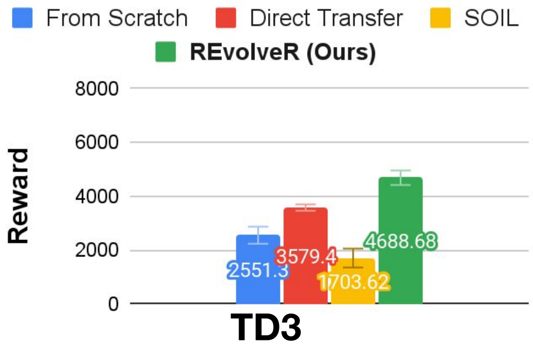


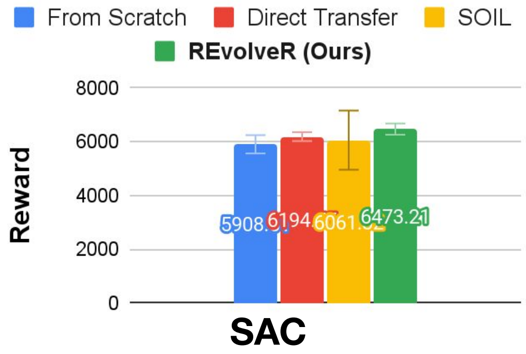
Ant-leg-emerge

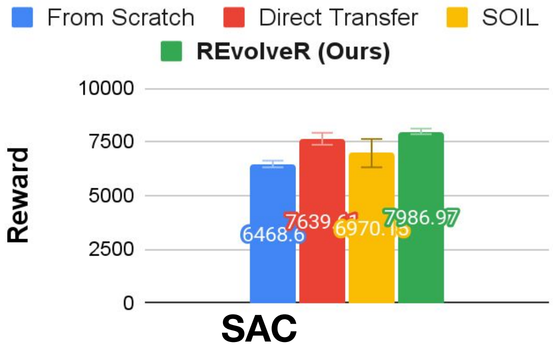
Experiments: MuJoCo Gym

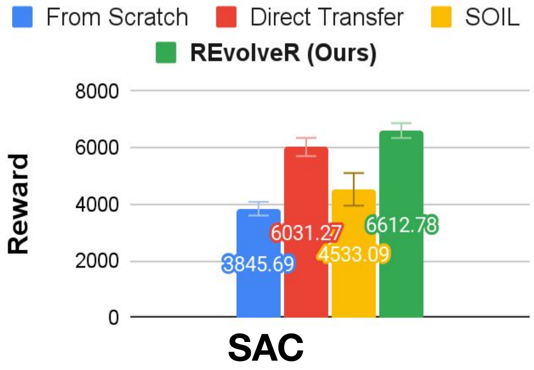










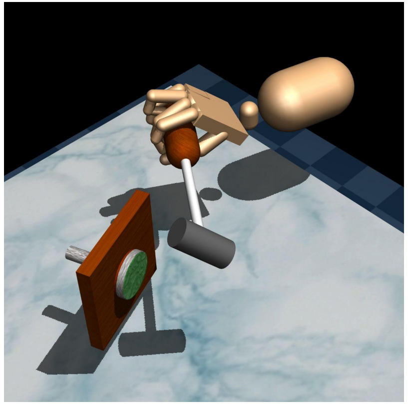


Ant-length-mass

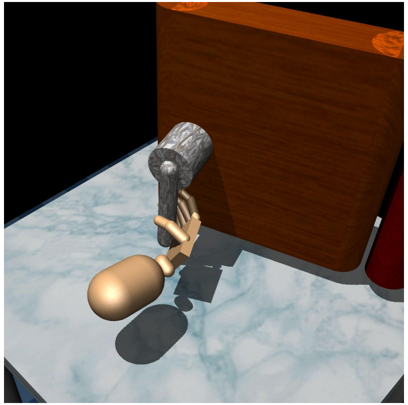
Humanoid-length-mass

Ant-leg-emerge

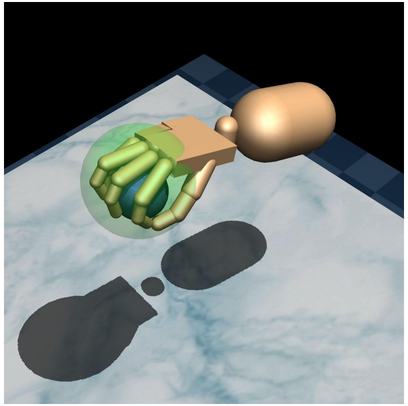
Experiments: Hand Manipulation Suite



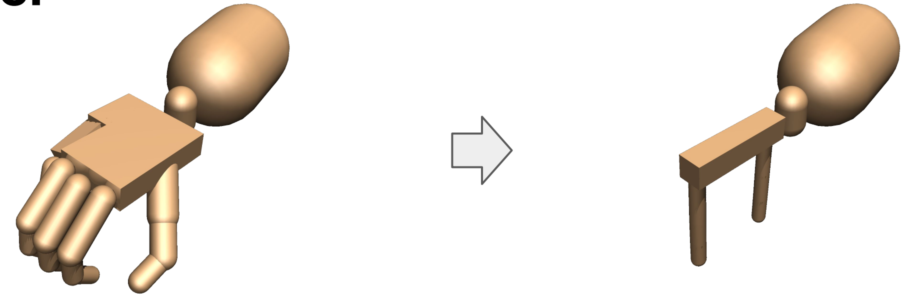
Hammer



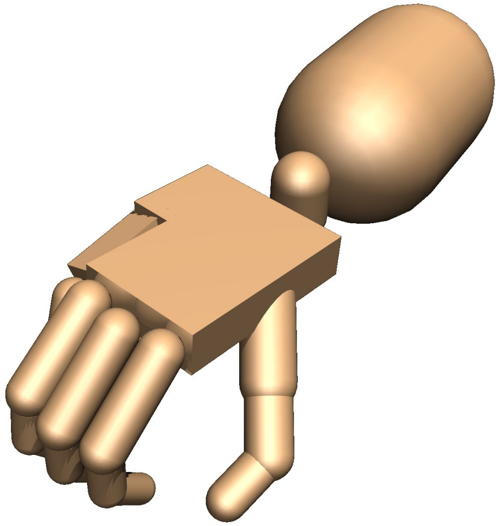
Door



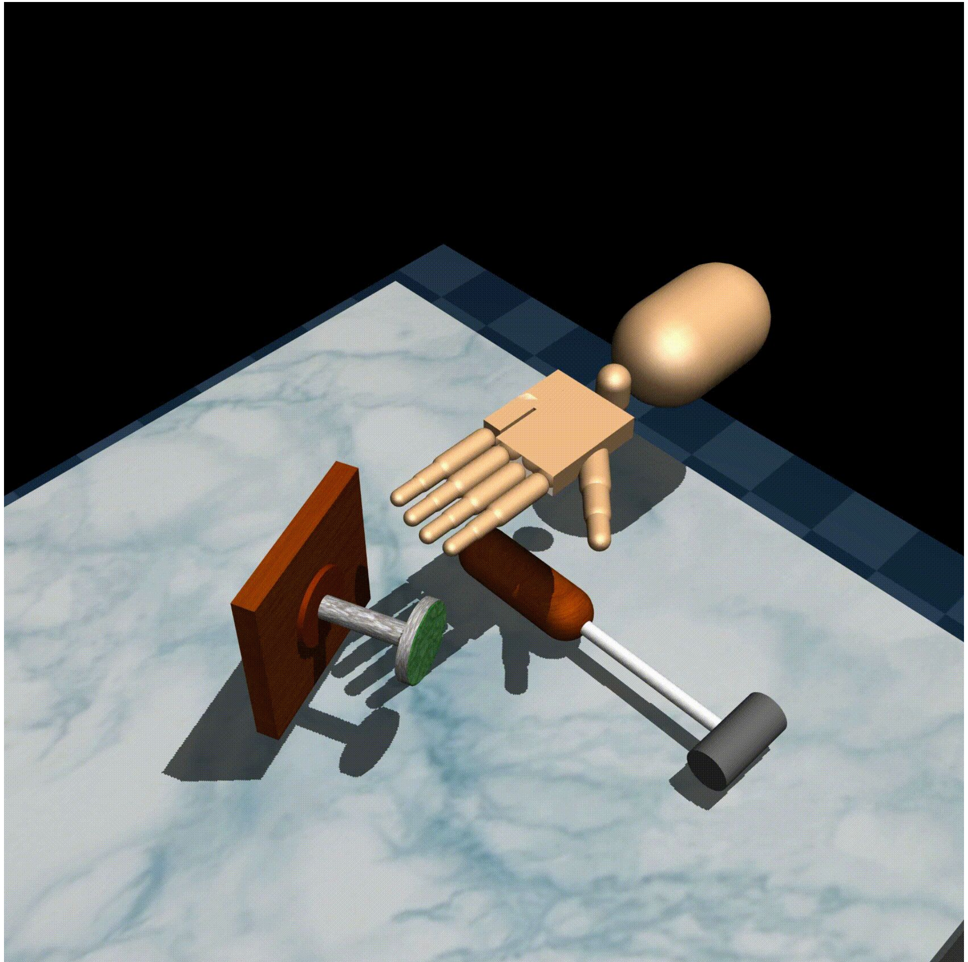
Relocate



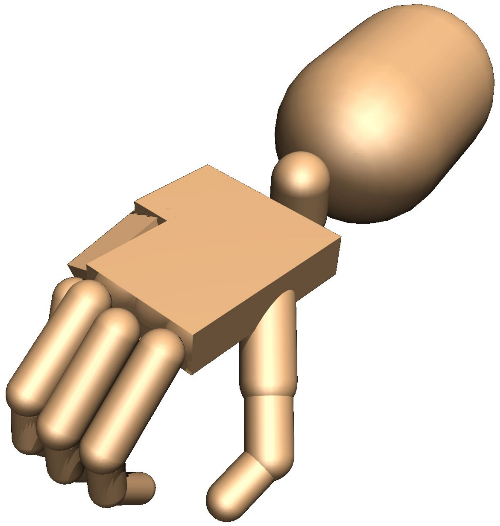
Experiments: Hand Manipulation Suite, **Hammer Task**



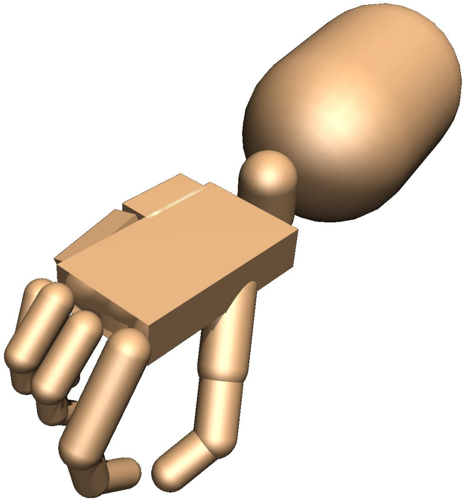
Robot at evolution progress $\alpha = 0.0$



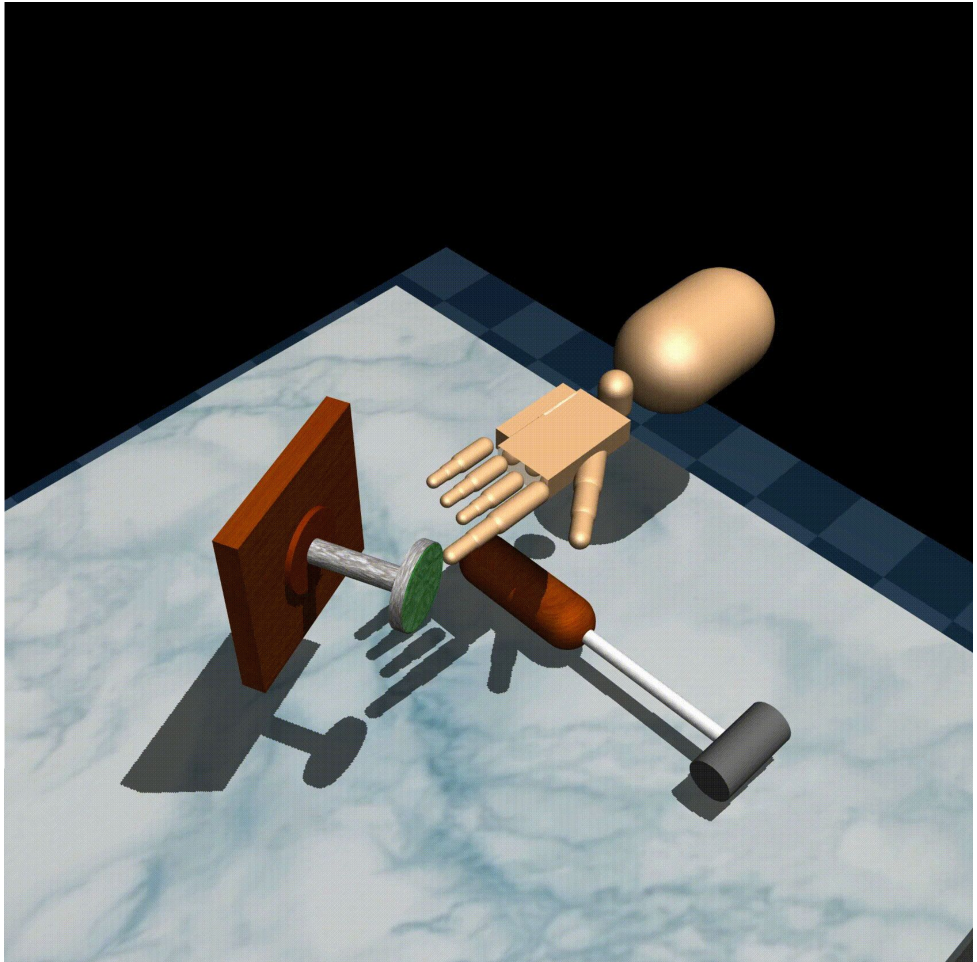
Experiments: Hand Manipulation Suite, **Hammer Task**



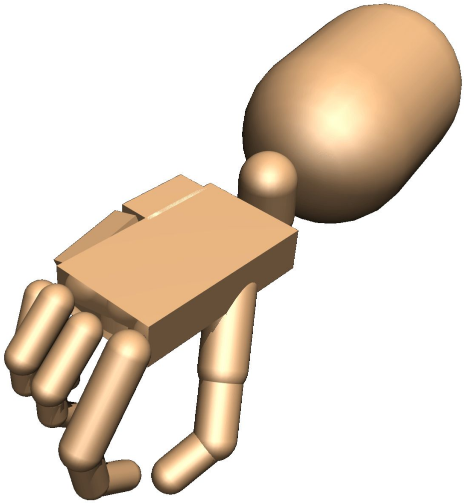
Experiments: Hand Manipulation Suite, **Hammer Task**



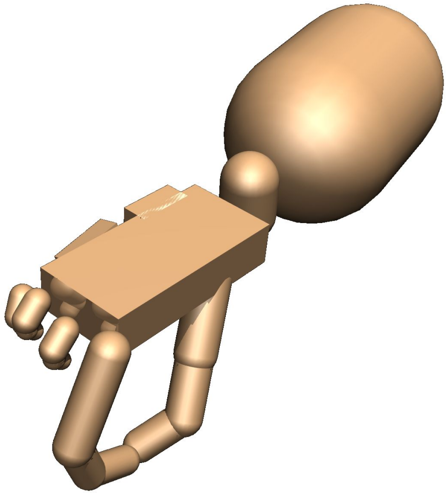
Robot at evolution progress $\alpha = 0.2$



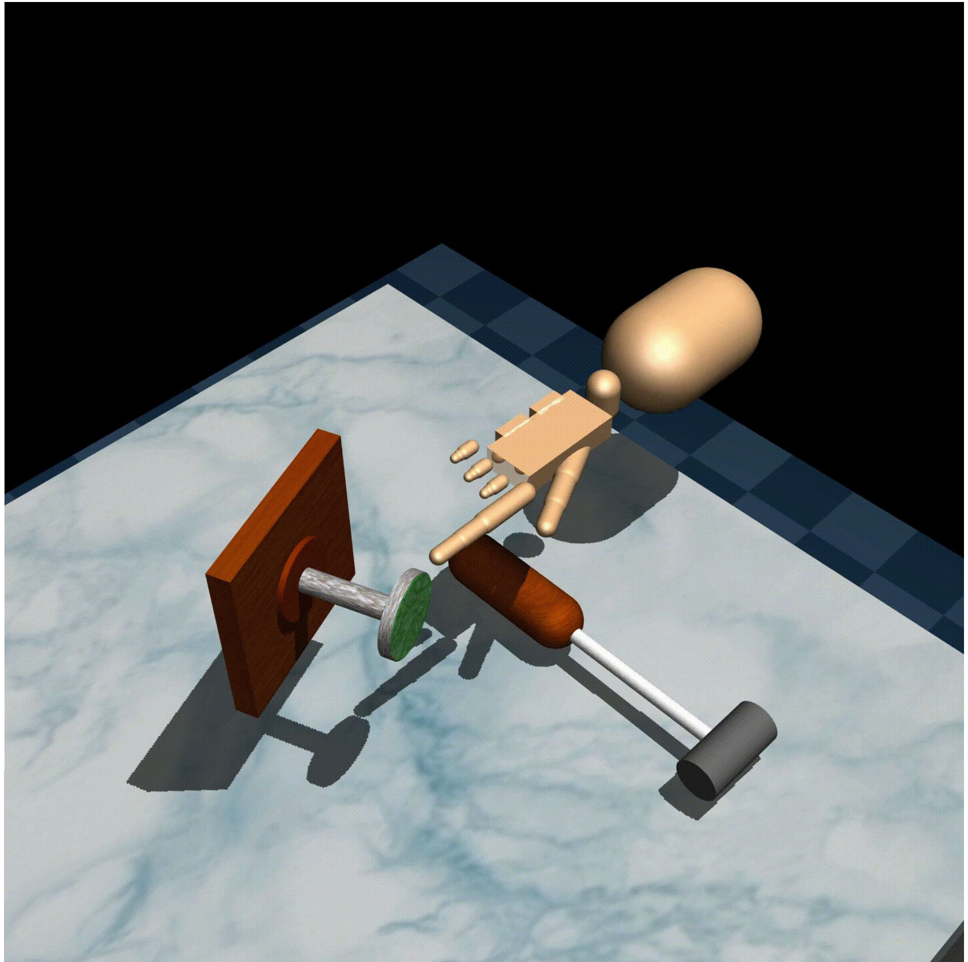
Experiments: Hand Manipulation Suite, **Hammer Task**



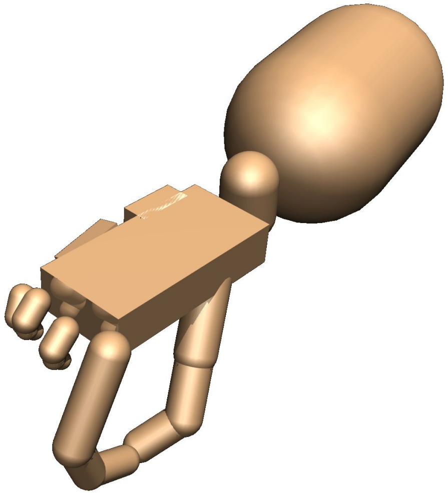
Experiments: Hand Manipulation Suite, **Hammer Task**



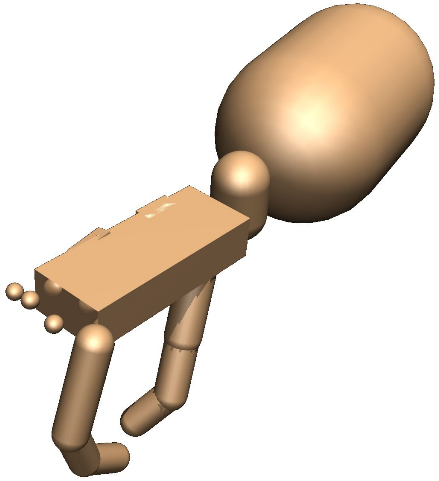
Robot at evolution progress $\alpha = 0.4$



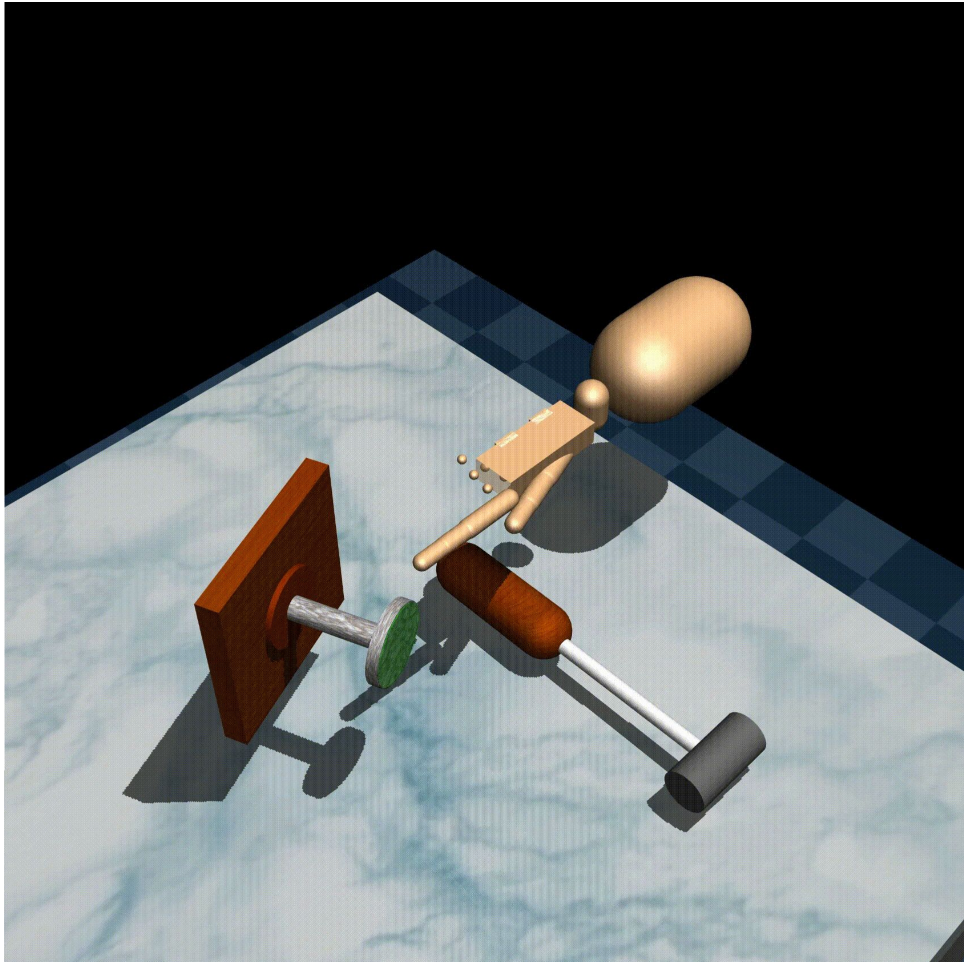
Experiments: Hand Manipulation Suite, **Hammer Task**



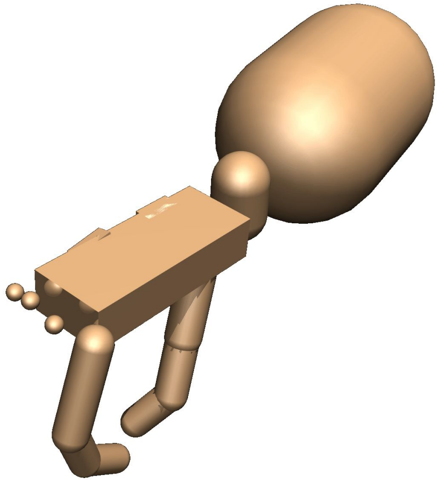
Experiments: Hand Manipulation Suite, **Hammer Task**



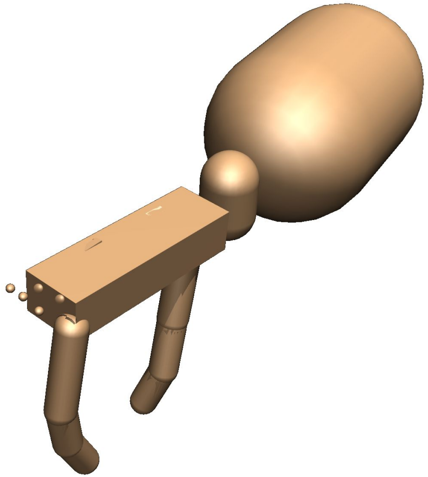
Robot at evolution progress $\alpha = 0.6$



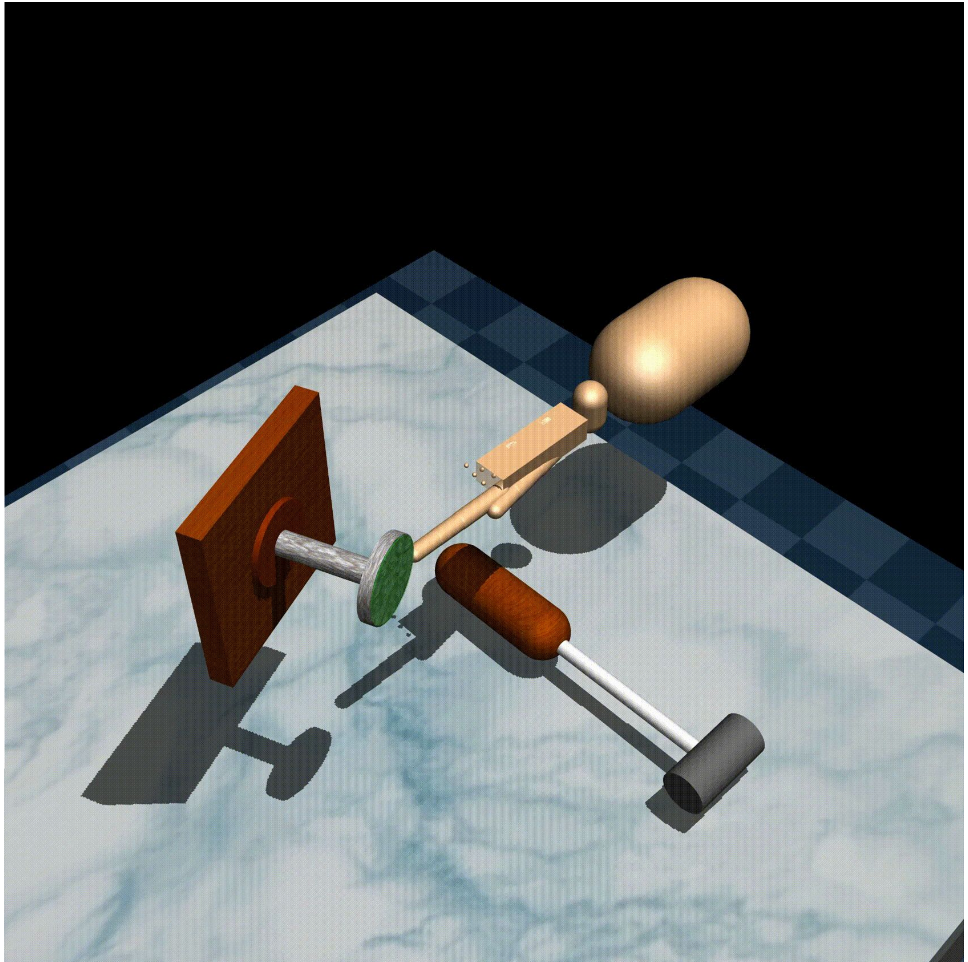
Experiments: Hand Manipulation Suite, **Hammer Task**



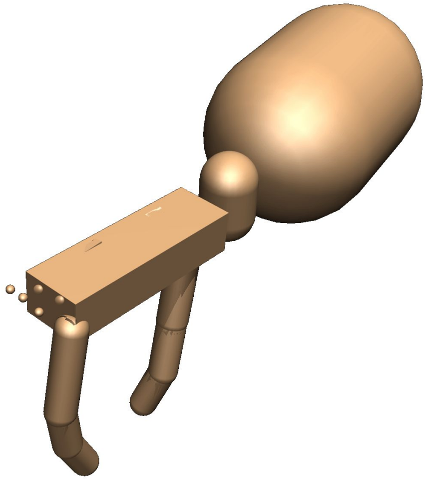
Experiments: Hand Manipulation Suite, **Hammer Task**



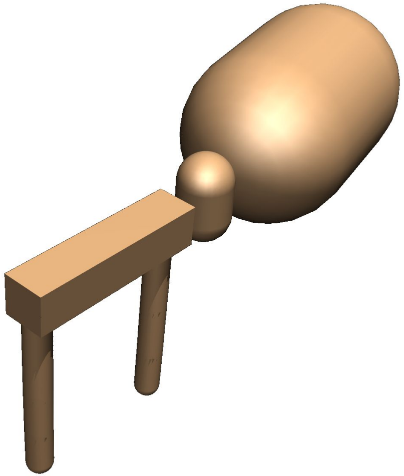
Robot at evolution progress $\alpha = 0.8$



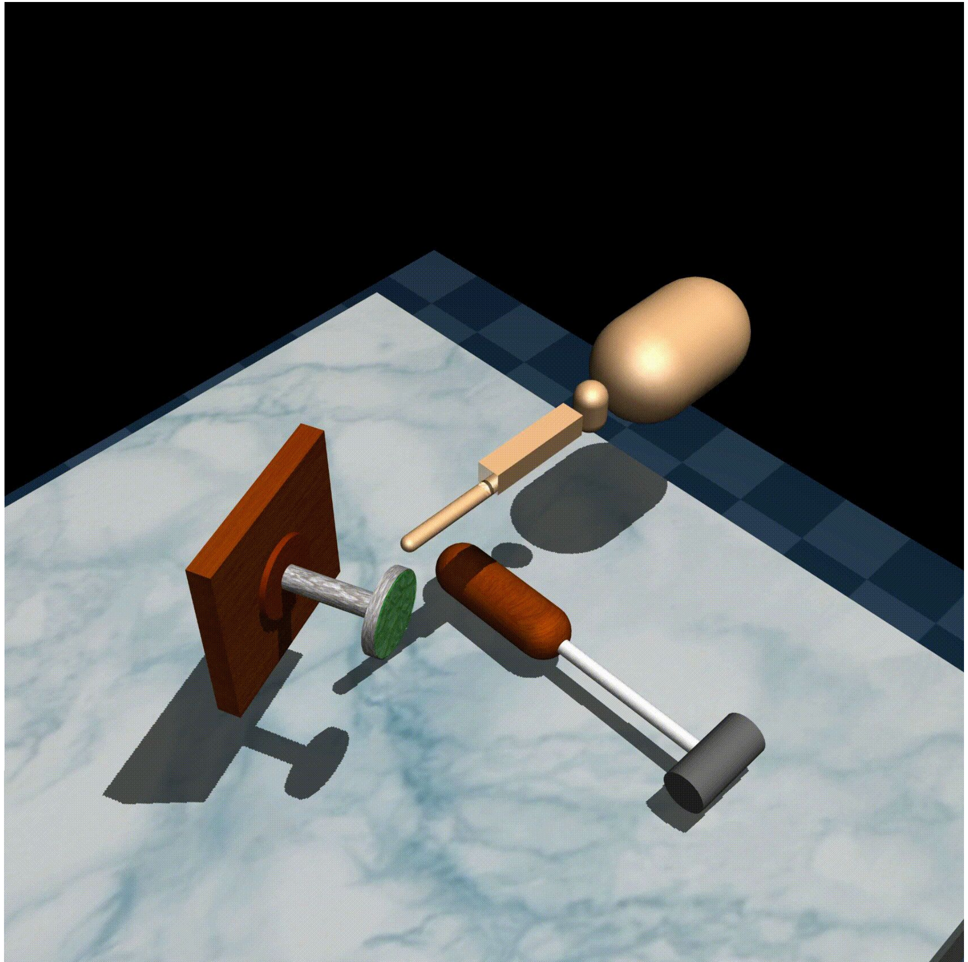
Experiments: Hand Manipulation Suite, **Hammer Task**



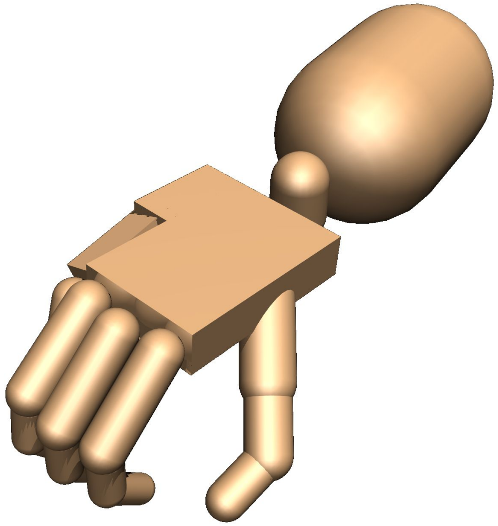
Experiments: Hand Manipulation Suite, **Hammer Task**



Robot at evolution progress $\alpha = 1.0$

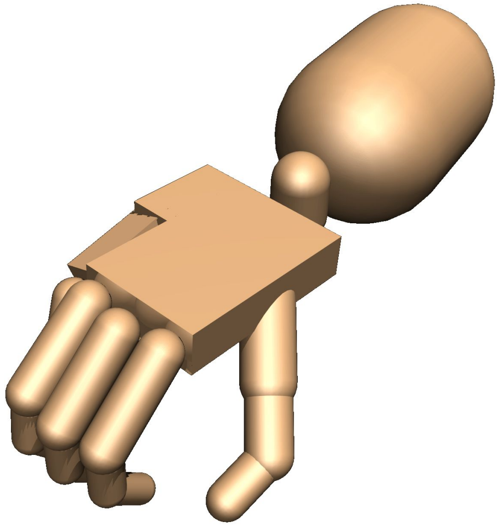


Experiments: Hand Manipulation Suite, **Door Task**

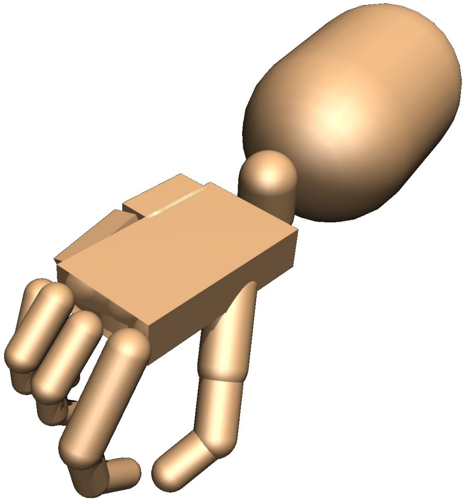


Robot at evolution progress $\alpha = 0.0$

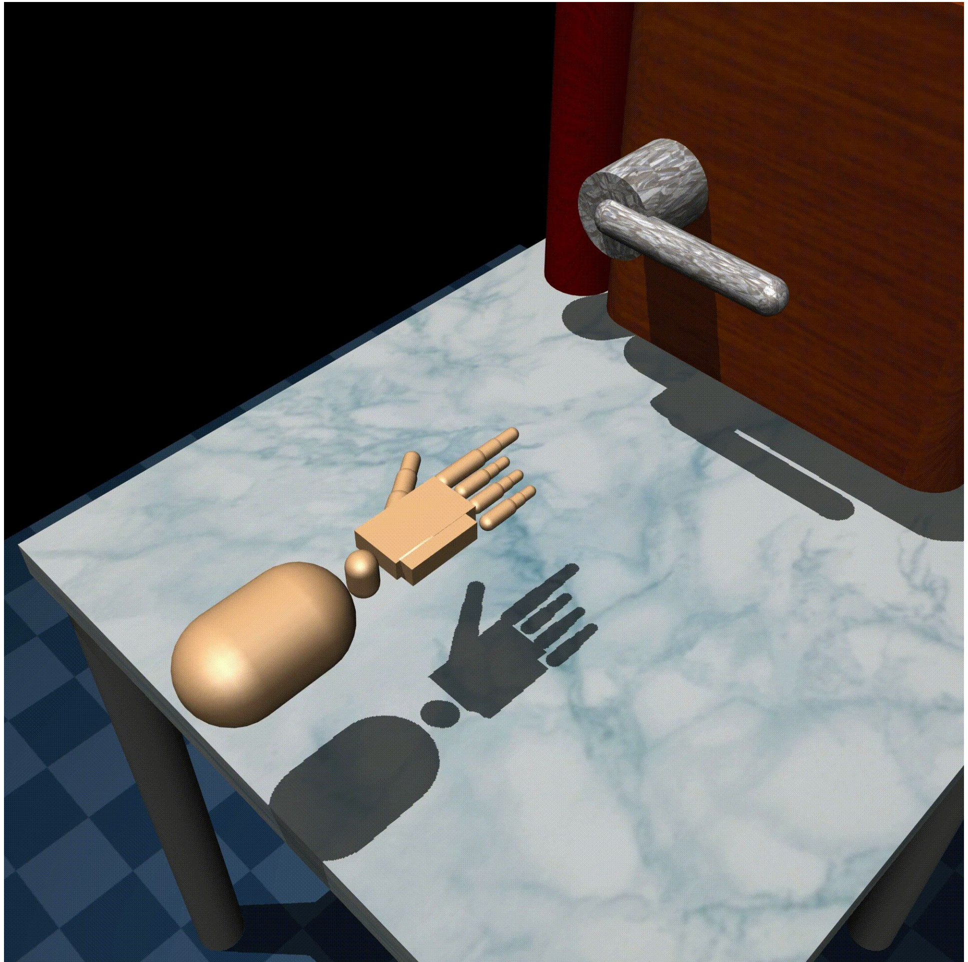
Experiments: Hand Manipulation Suite, **Door Task**



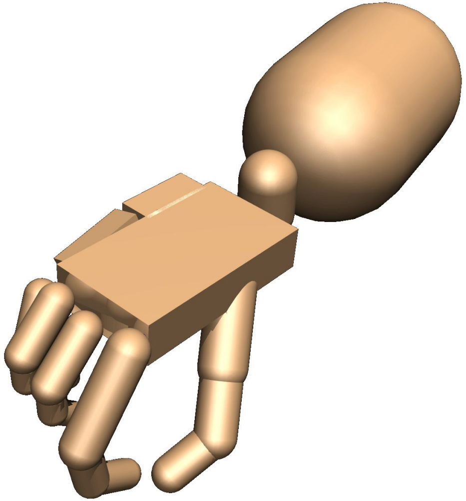
Experiments: Hand Manipulation Suite, **Door Task**



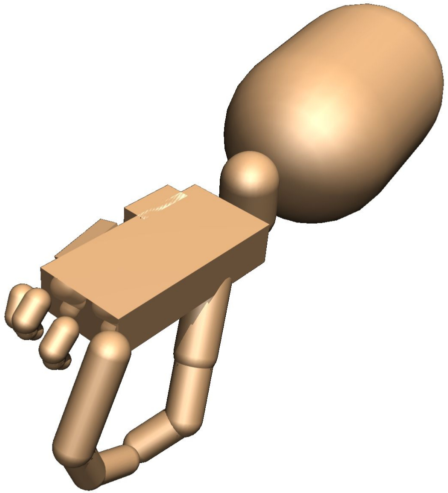
Robot at evolution progress $\alpha = 0.2$



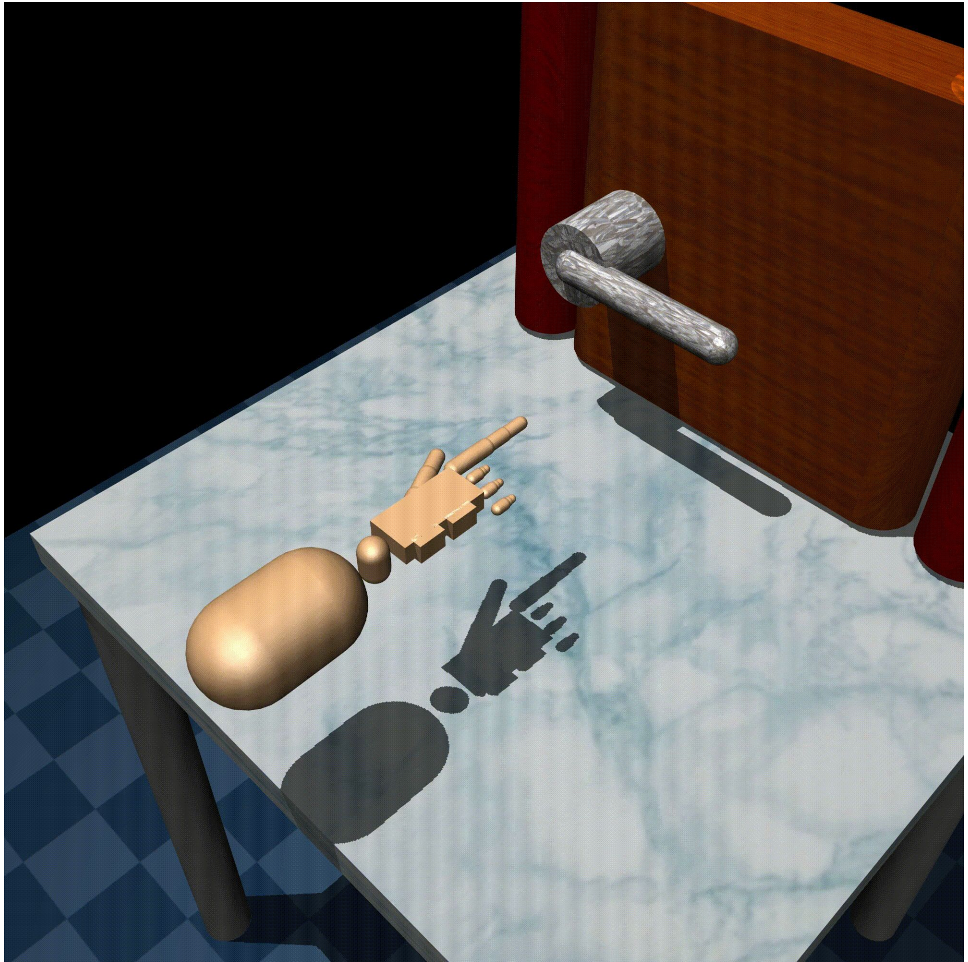
Experiments: Hand Manipulation Suite, **Door Task**



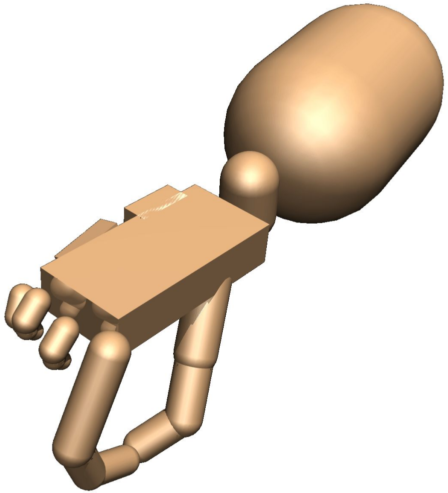
Experiments: Hand Manipulation Suite, **Door Task**



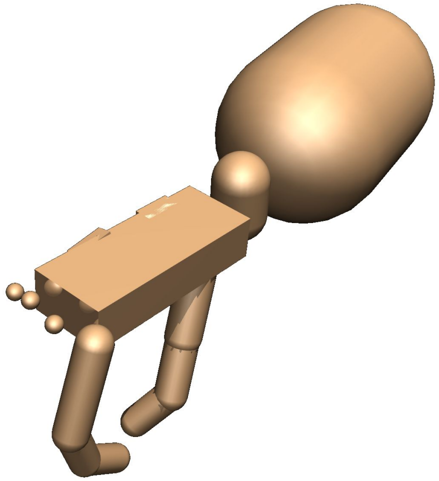
Robot at evolution progress $\alpha = 0.4$



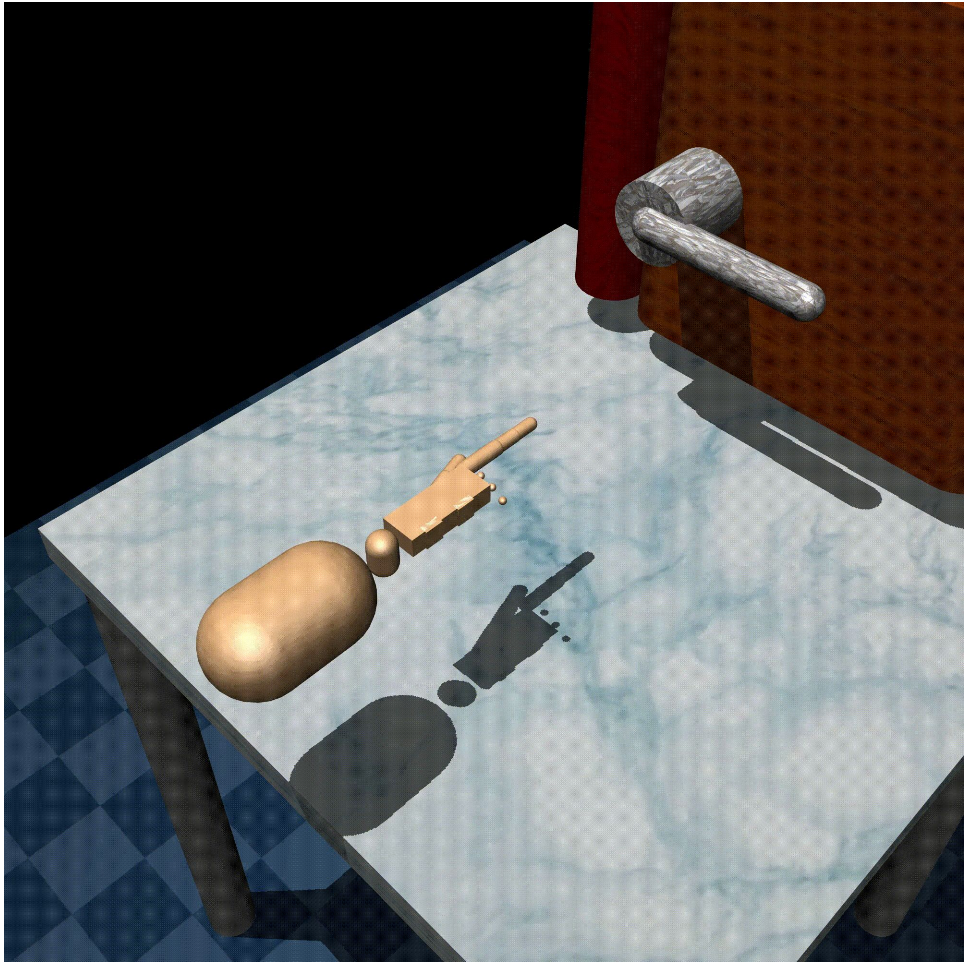
Experiments: Hand Manipulation Suite, **Door Task**



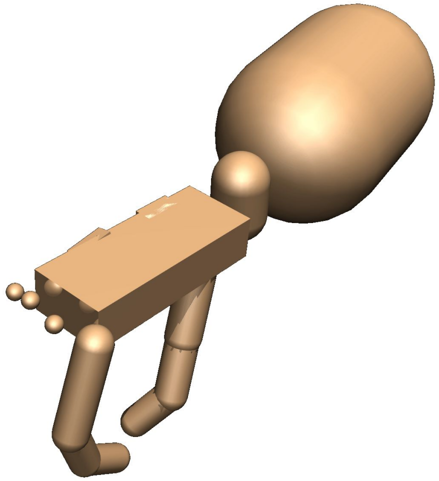
Experiments: Hand Manipulation Suite, **Door Task**



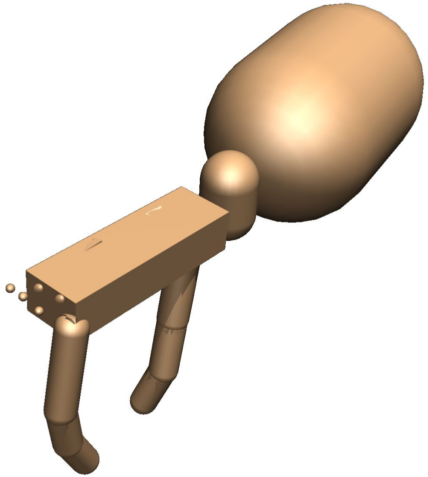
Robot at evolution progress $\alpha = 0.6$



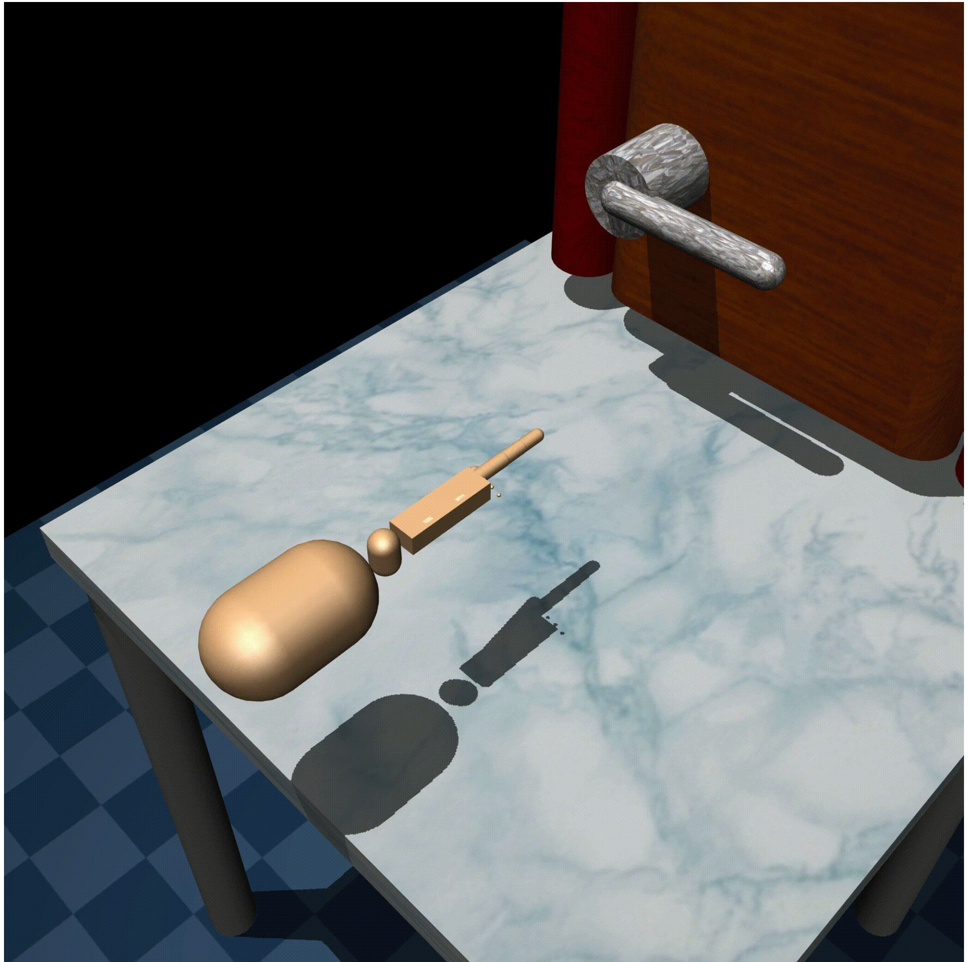
Experiments: Hand Manipulation Suite, **Door Task**



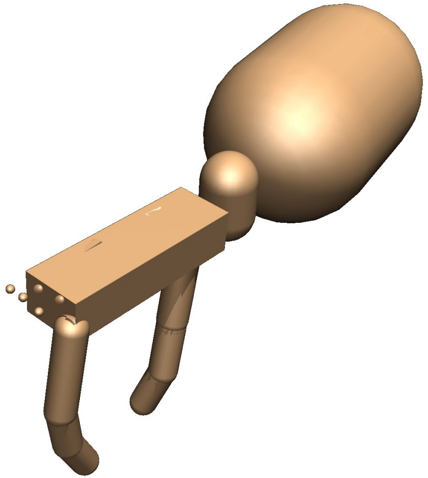
Experiments: Hand Manipulation Suite, **Door Task**



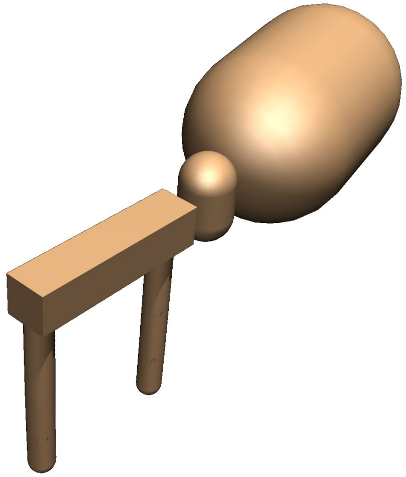
Robot at evolution progress $\alpha = 0.8$



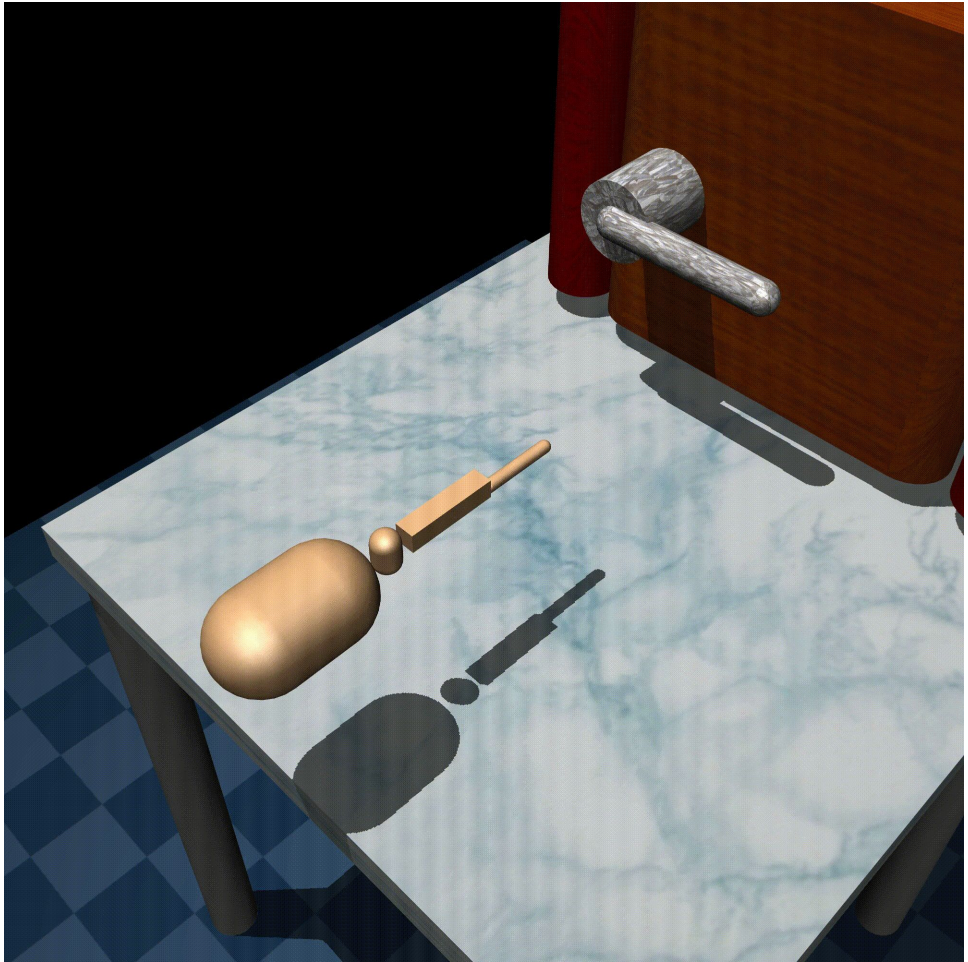
Experiments: Hand Manipulation Suite, **Door Task**



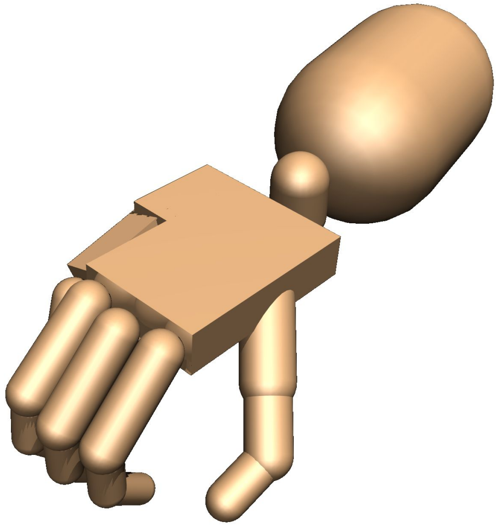
Experiments: Hand Manipulation Suite, **Door Task**



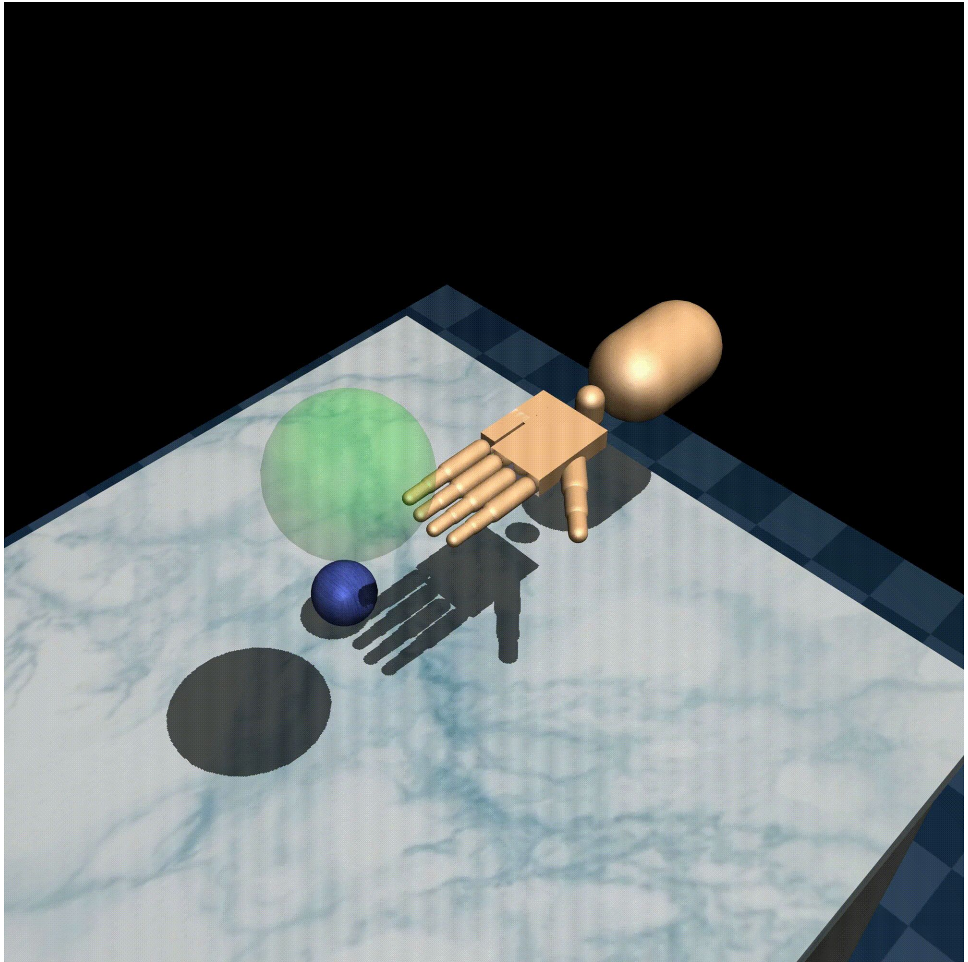
Robot at evolution progress $\alpha = 1.0$



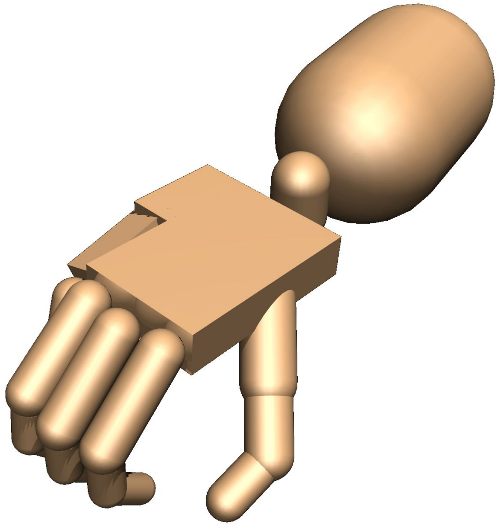
Experiments: Hand Manipulation Suite, **Relocate Task**



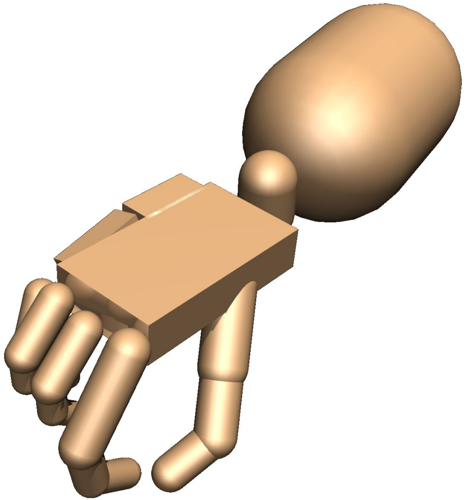
Robot at evolution progress $\alpha = 0.0$



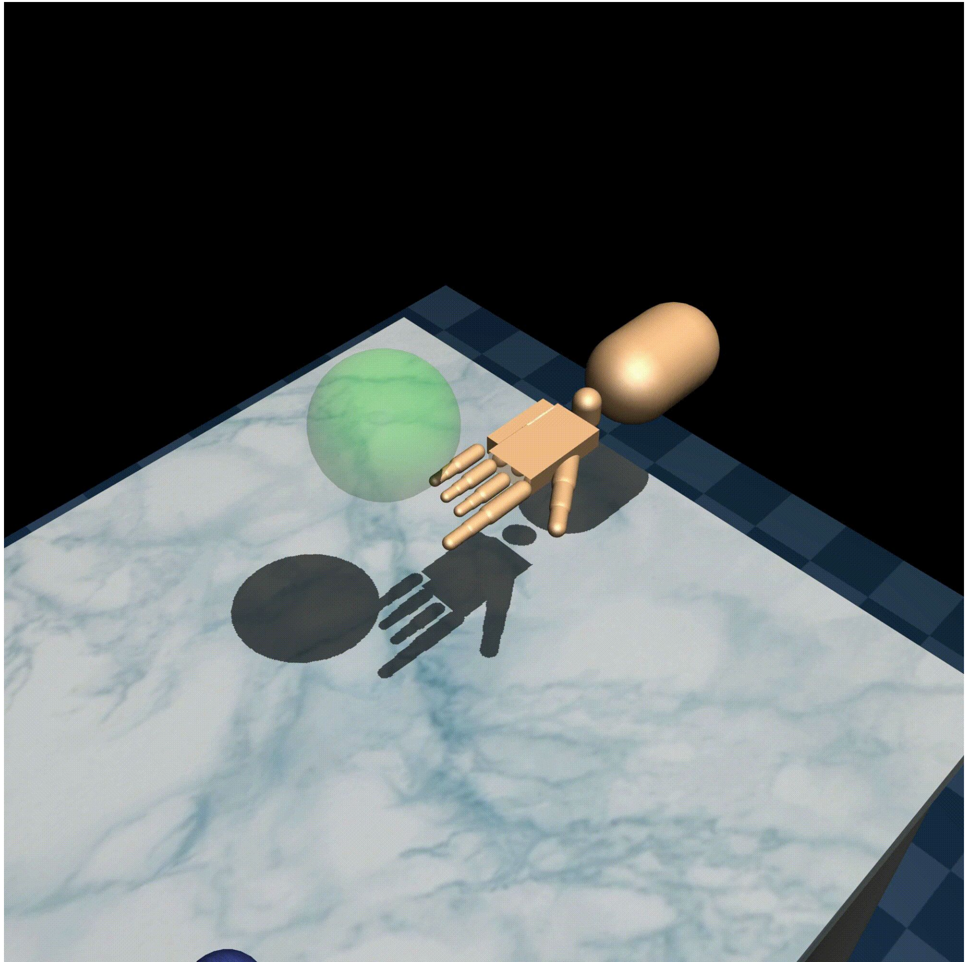
Experiments: Hand Manipulation Suite, **Relocate Task**



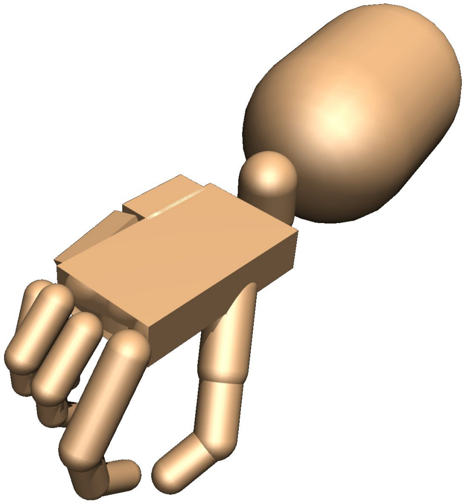
Experiments: Hand Manipulation Suite, **Relocate Task**



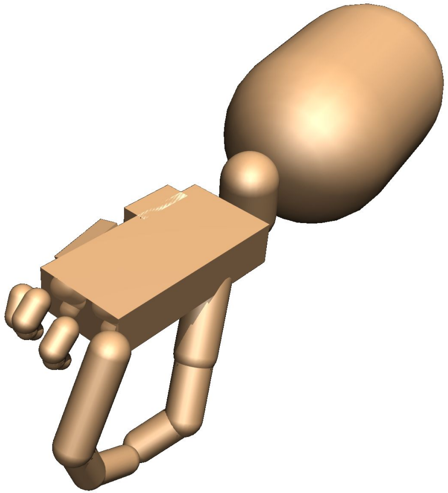
Robot at evolution progress $\alpha = 0.2$



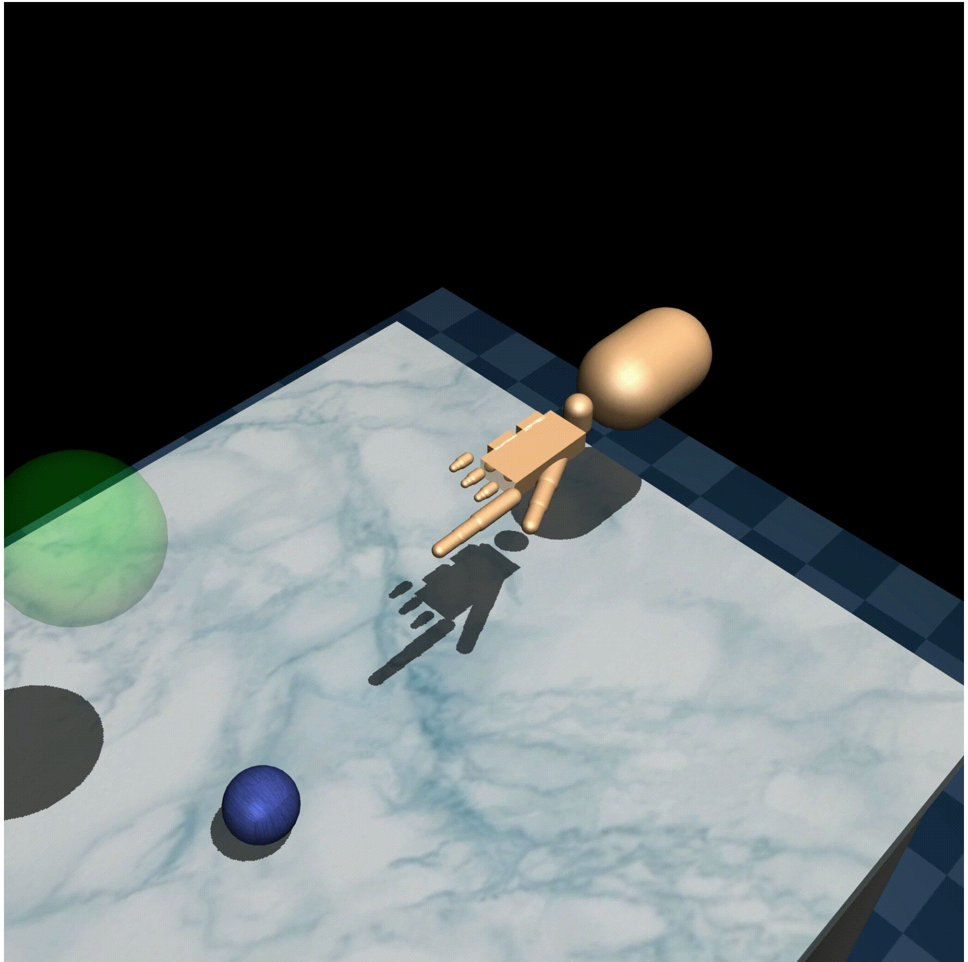
Experiments: Hand Manipulation Suite, **Relocate Task**



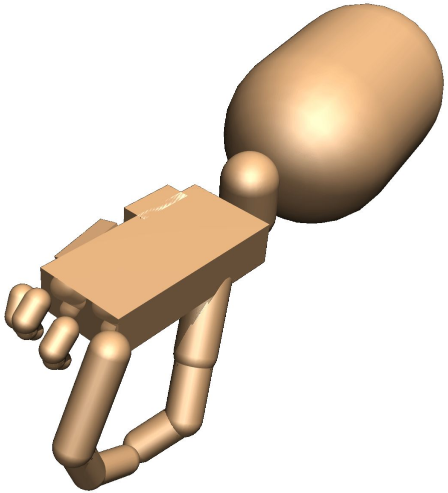
Experiments: Hand Manipulation Suite, **Relocate Task**



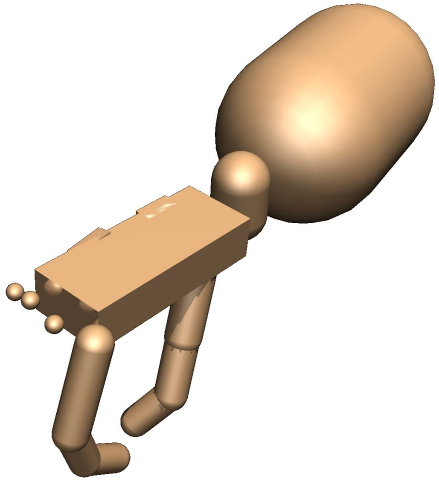
Robot at evolution progress $\alpha = 0.4$



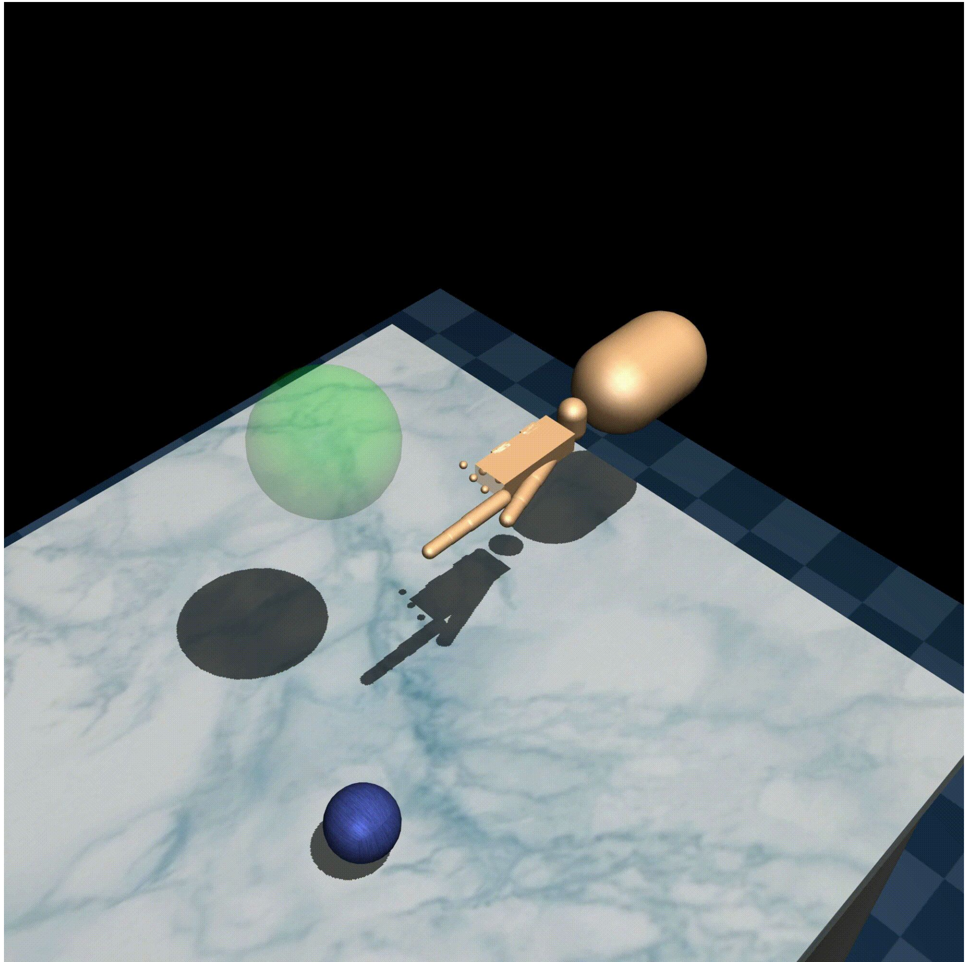
Experiments: Hand Manipulation Suite, **Relocate Task**



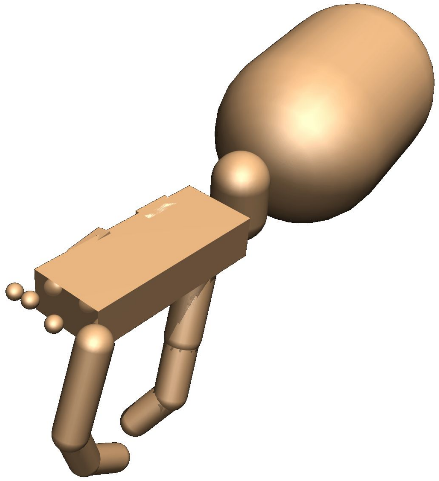
Experiments: Hand Manipulation Suite, **Relocate Task**



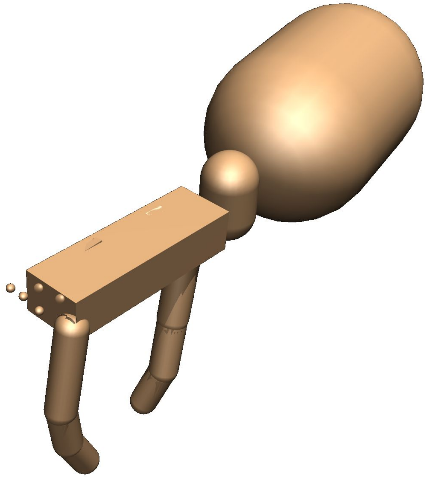
Robot at evolution progress $\alpha = 0.6$



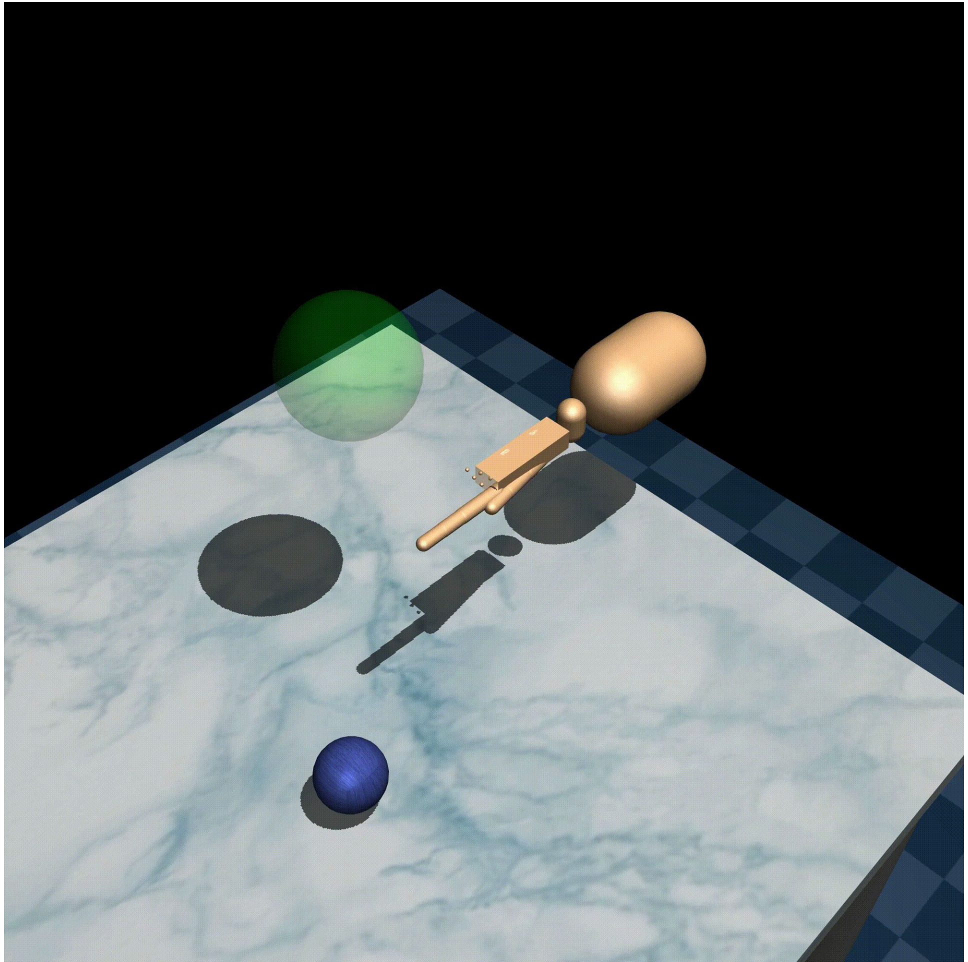
Experiments: Hand Manipulation Suite, **Relocate Task**



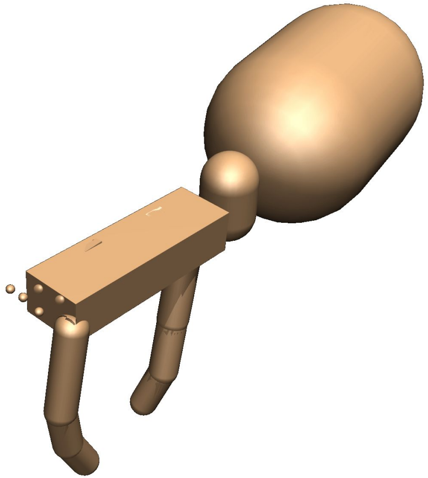
Experiments: Hand Manipulation Suite, **Relocate Task**



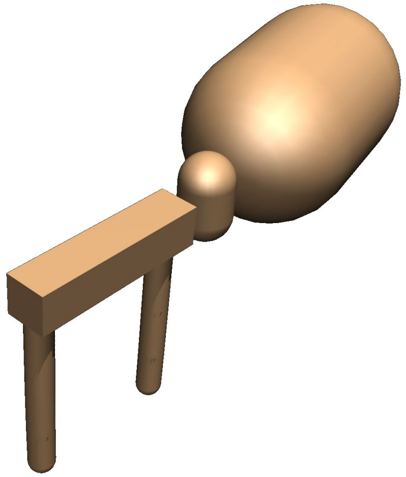
Robot at evolution progress $\alpha = 0.8$



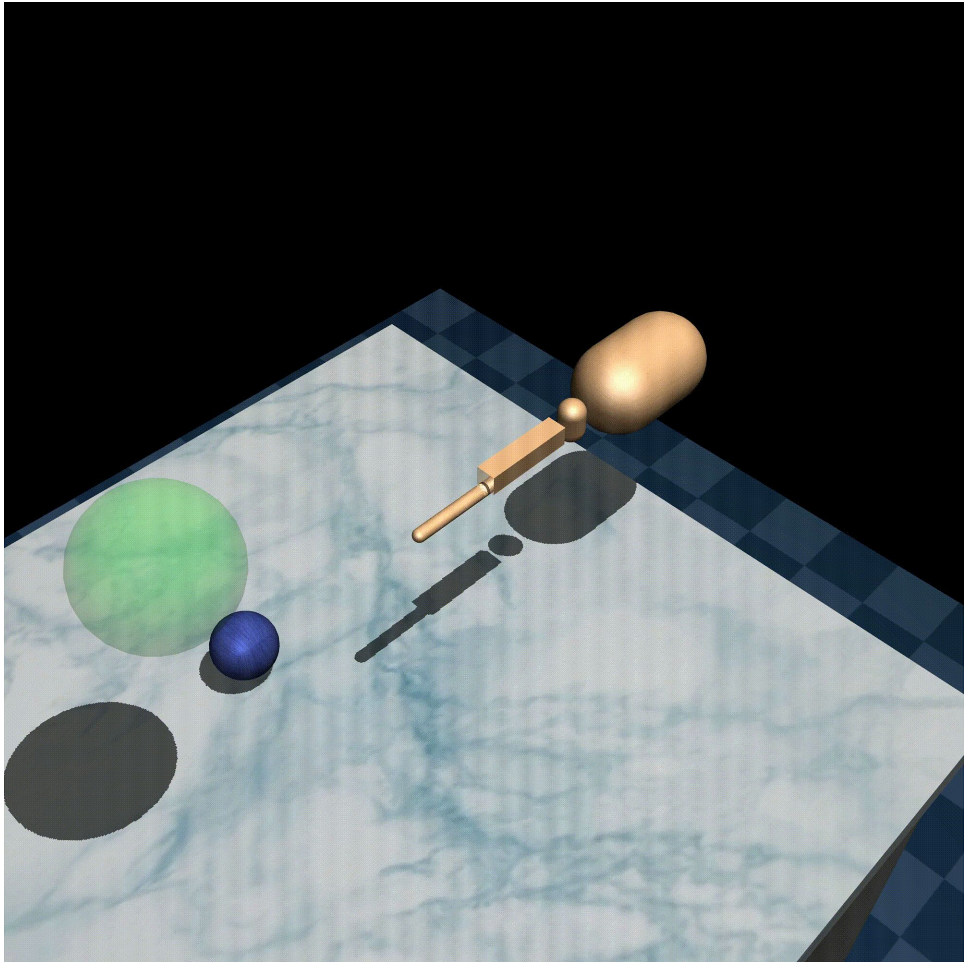
Experiments: Hand Manipulation Suite, **Relocate Task**



Experiments: Hand Manipulation Suite, **Relocate Task**



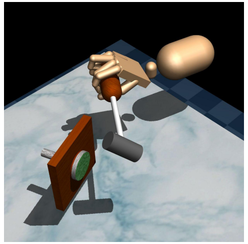
Robot at evolution progress $\alpha = 1.0$



Experiments: Hand Manipulation Suite

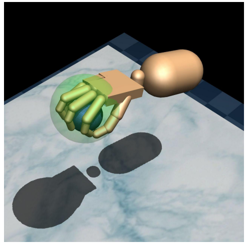
	Dense Reward	Sparse Reward
From Scratch	-	∞
Direct Finetune	7.6K	∞
DAPG	5.4K	∞
Ours	-	**2.6K**

Door Task



	Dense Reward	Sparse Reward
From Scratch	$>$100K	∞
Direct Finetune	43.5K	∞
DAPG	23.3K	∞
Ours	-	**18.1K**

Hammer Task



	Dense Reward	Sparse Reward
From Scratch	$>$100K	∞
Direct Finetune	$>$100K	∞
DAPG	17.1K	∞
Ours	-	**11.9K**

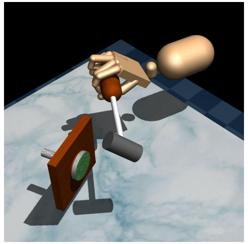
Relocate Task

Number of epochs needed to reach 90% success rate

Experiments: Hand Manipulation Suite

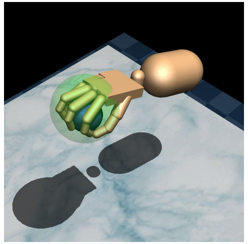
	Dense Reward	Sparse Reward
From Scratch	-	∞
Direct Finetune	7.6K	∞
DAPG	5.4K	∞
Ours	-	**2.6K**

Door Task



	Dense Reward	Sparse Reward
From Scratch	>100K	∞
Direct Finetune	43.5K	∞
DAPG	23.3K	∞
Ours	-	**18.1K**

Hammer Task



	Dense Reward	Sparse Reward
From Scratch	>100K	∞
Direct Finetune	>100K	∞
DAPG	17.1K	∞
Ours	-	**11.9K**

Relocate Task

Number of epochs needed to reach 90% success rate

REvolveR: Continuous Evolutionary Models for Robot-to-robot Policy Transfer

Thank you