Leveraging Hyperbolic Embeddings for Coarse-to-Fine Robot Design

Heng Dong[1]*, Junyu Zhang[2]*, Chongjie Zhang[3]

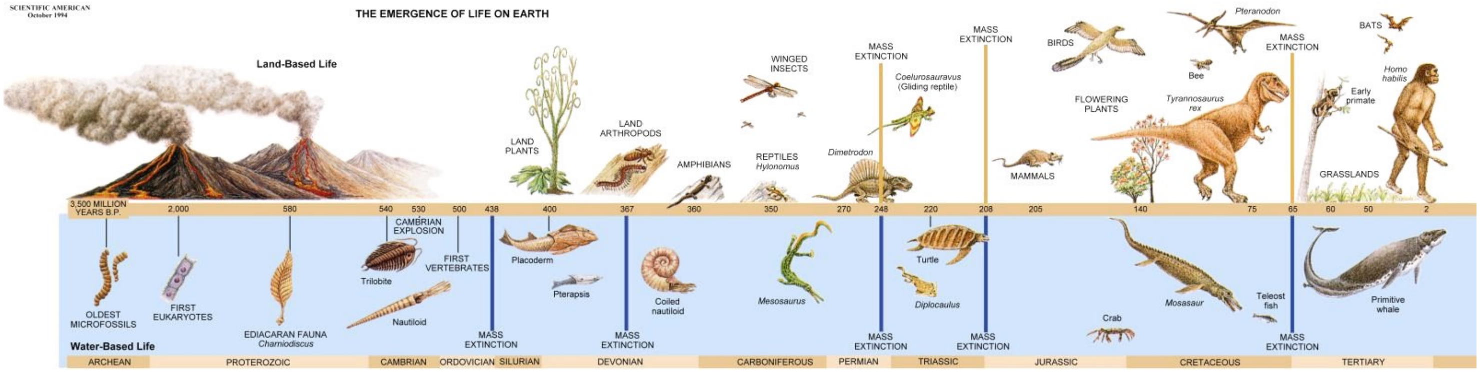
[1]Tsinghua University, [2]Huazhong University of Science and Technology, [3]Washington University in St. Louis

Machine Intelligence Group

清華大學 交叉信息研究院
Tsinghua University Institute for Interdisciplinary Information Sciences

Natural Evolution

- Creatures adapt to new environments to solve daily tasks better through natural evolution

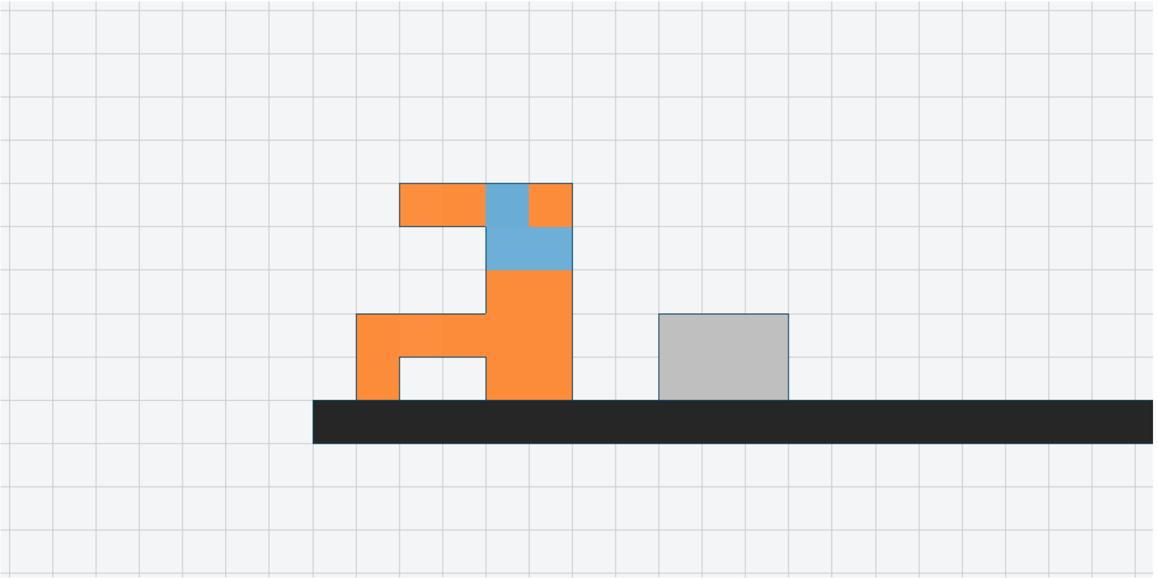


- Can we mimic this evolution process so that robots can solve new tasks better by changing their morphologies?

https://universe-review.ca/F10-multicell01.htm

Multi-Cellular Robot Design

- Evolution Gym



An example robot designed by our method

2D multi-cellular systems
design space: $10^{17}\sim10^{34}$

- Vertical Actuator
- Horizontal Actuator
- Rigid Cell
- Soft Cell
- Empty Cell

Bhatia, Jagdeep, et al. "Evolution gym: A large-scale benchmark for evolving soft robots." *Advances in Neural Information Processing Systems* 34 (2021): 2201-2214.

Major Difficulties of Robot Design

- Robot design problem can be formulated as a bi-level optimization problem

performance evaluation

control policy for design D

a robot design

$$D^* = \arg\max_D J(\pi_D, D)$$

$$\text{subject to} \quad \pi_D = \arg\max_\pi J(\pi, D)$$

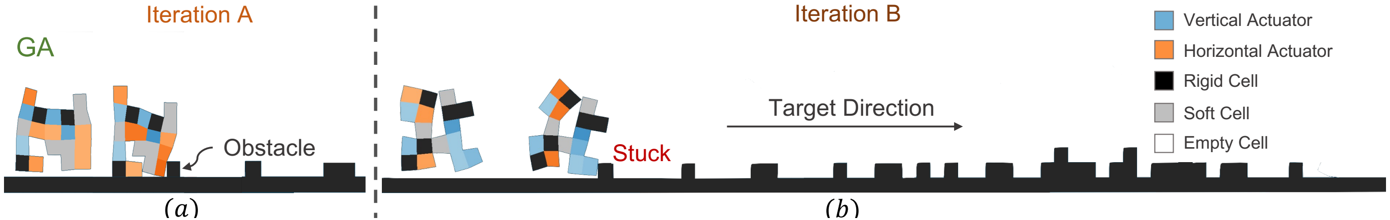
- **Outer level**: Search in the design space.
  - Immensely large design space
    - EvoGym: $10^{17} \sim 10^{34}$

- **Inner level**: Evaluate each candidate design
  - Computationally expensive to find its optimal controller
  - Inaccurate evaluation (due to the lack of optimal control policy)
    - It is hard to tell which one is better if the robots are similar at the beginning of training.

Two robots designed by GA

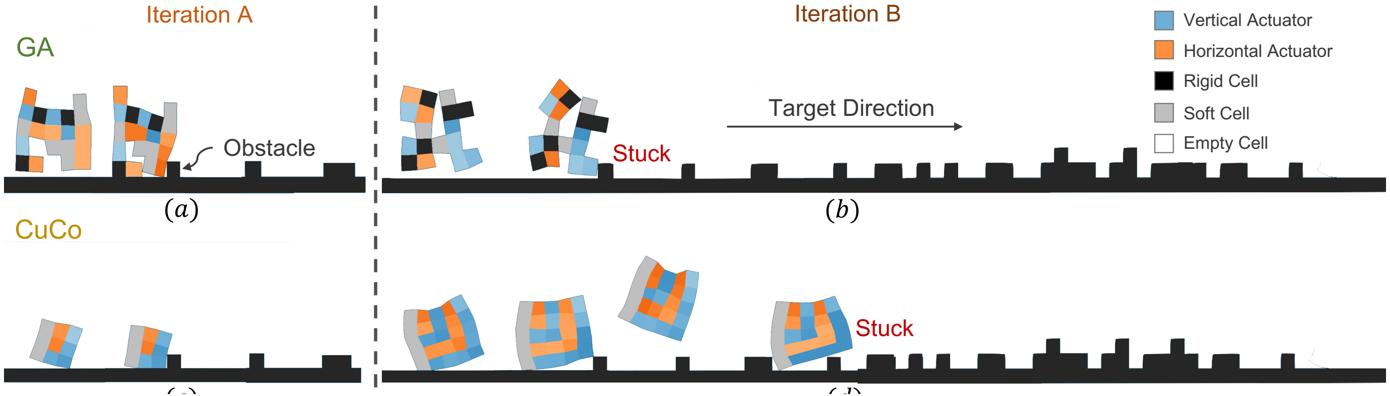
Intuition

- Previous work GA directly searches in the vast design space
  - fails to learn effective structures to cross the obstacles



Intuition

- CuCo adopts a predefined curriculum from smaller robots to larger robots
  - the smaller robot typically faces more challenges when solving the original tasks, e.g., the same obstacles could be more difficult for it.
  - cannot offer useful guidance for the remaining stage in the curriculum.



Curriculum-based Co-design of Morphology and Control of Voxel-based Soft Robots. ICLR 2023.

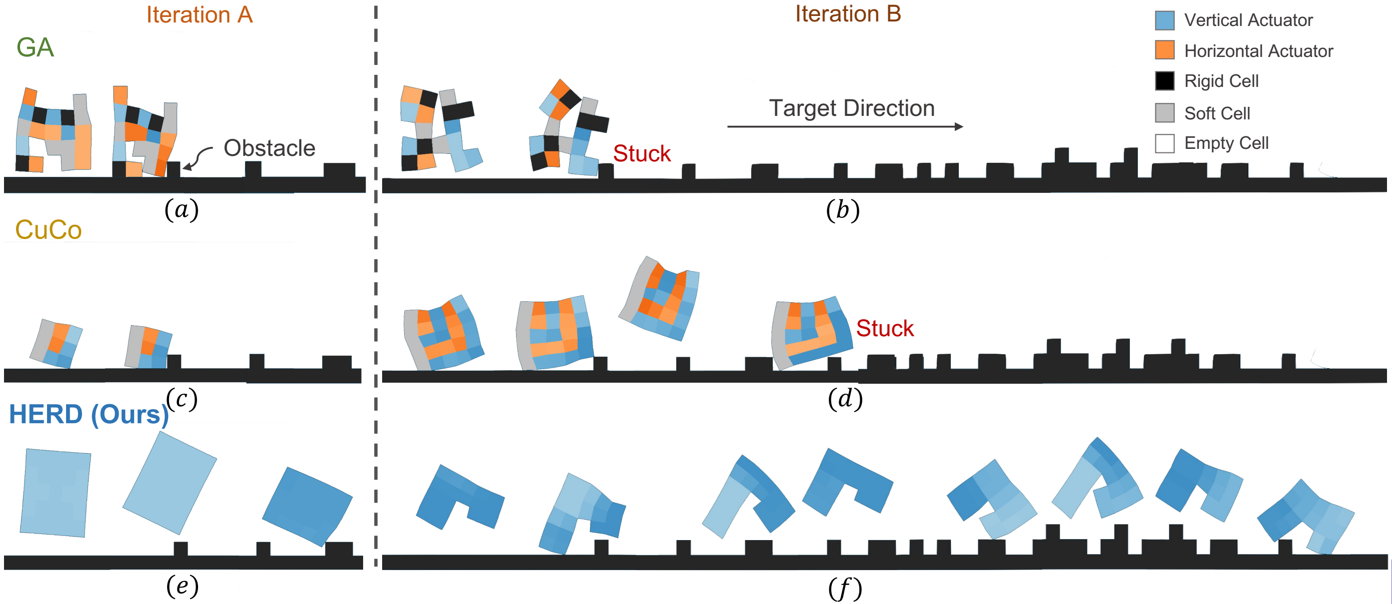
Our Idea

- Designing multi-cellular robots in a **coarse-to-fine** manner
  - first searching for coarse-grained robots with satisfactory performance
    - Smaller design space
  - subsequently refining them
  - an example of coarse-to-fine from painting

Coarse-grained → Fine-grained

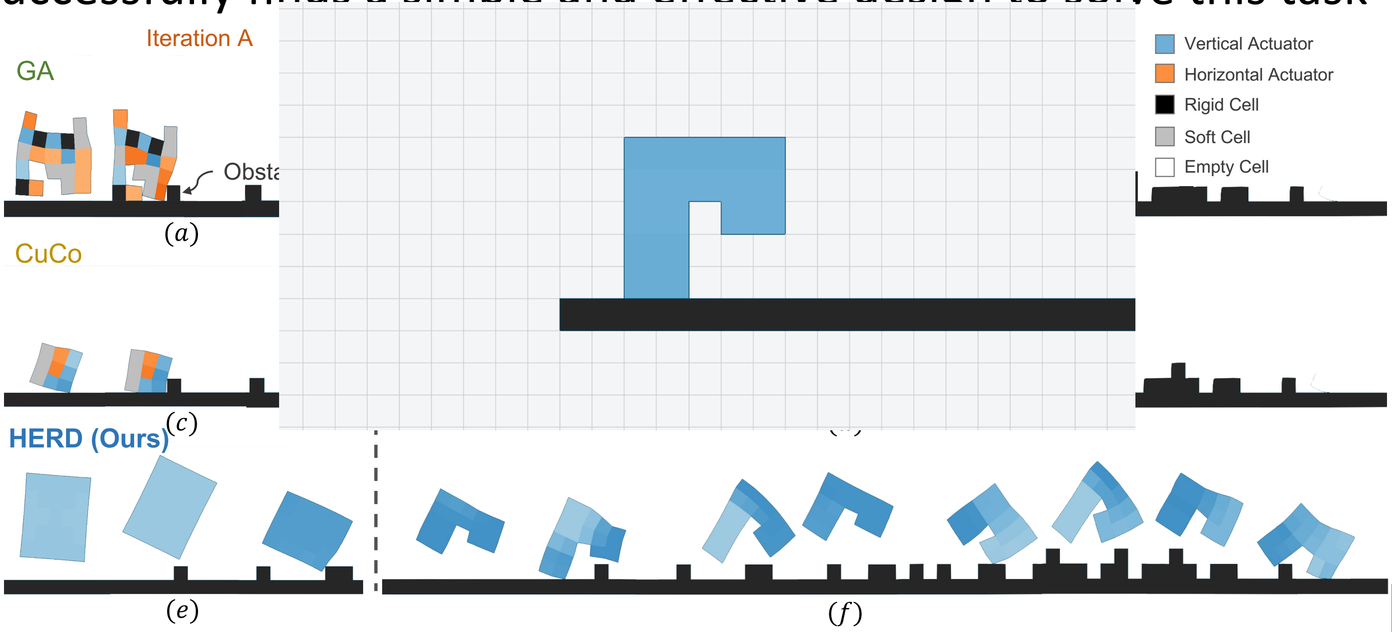
Intuition

- Our method designs robots in a coarse-to-fine manner
  - focus on promising regions with the helpful guidance of coarse-grained design
  - successfully finds a simple and effective design to solve this task



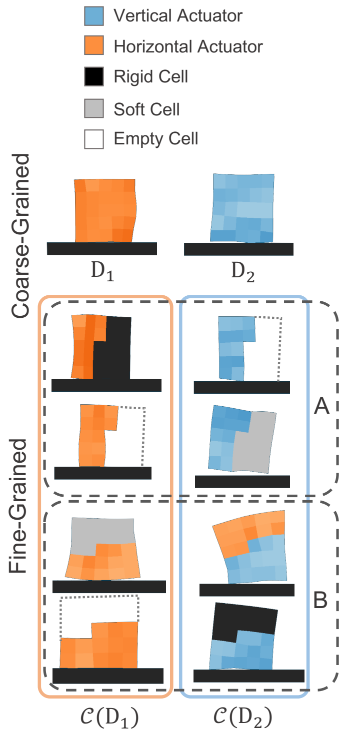
Intuition

- Our method designs robots in a coarse-to-fine manner
  - focus on promising regions with the helpful guidance of coarse-grained design
  - successfully finds a simple and effective design to solve this task

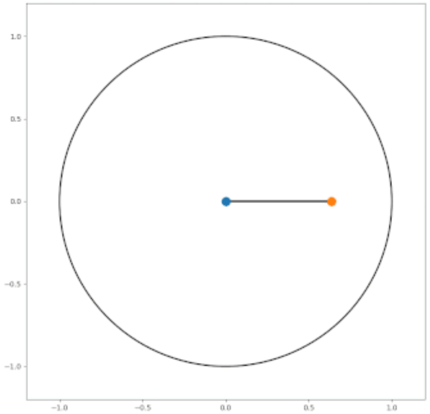


Method

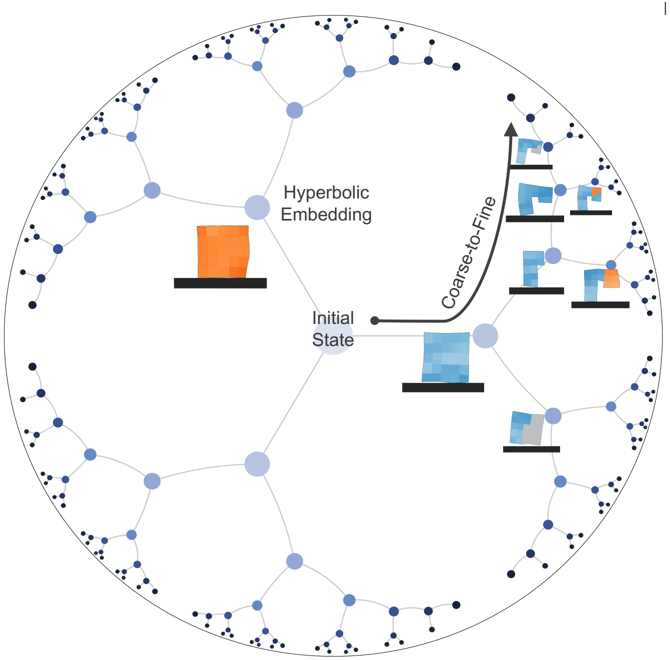
- The space of robot designs can be organized as a **hierarchy**



(a)



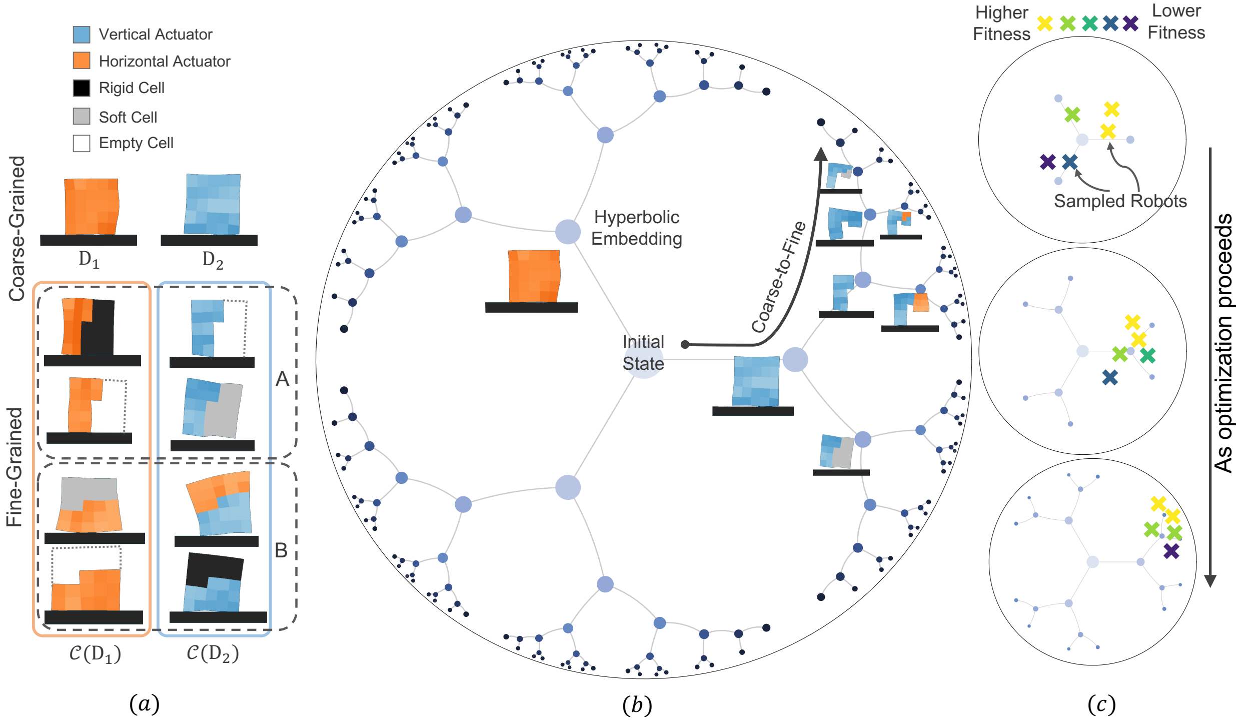
We propose to embed the hierarchy of robots into Poincaré Disk using Sarkar's construction for better optimization.



(b)

Method

- Sampling robots from the center of Poincaré Disk to the border is exactly the process of coarse-to-fine robot design.



Method

- **HERD:** Leverage **H**yperbolic **E**mbeddings for coarse-to-fine **R**obot **D**esign

Algorithm 1: HERD: Hyperbolic Embeddings for Coarse-to-Fine Robot Design

Input: robot design space $\mathbb{D}$, hierarchy size N, Poincaré ball $\mathbb{B}_c^d$, population size of CEM N_v

Embedding

$\{D_i, \mathcal{C}(D_i)\}_i^N \leftarrow$ build the robot hierarchy for each design $D_i \in \mathbb{D}$ using K-Means;

$\mathbb{S} = \{D_i, \boldsymbol{z}_i\}_i^N \leftarrow$ embed the robot hierarchy $\{D_i, \mathcal{C}(D_i)\}_i^N$ in Poincaré ball $\mathbb{B}_c^d$ by applying Sarkar's Construction in Algorithm 2 recursively;

Optimization

initialize control policy π, CEM mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$;

while *not reaching max iterations* **do**

  replay buffer $\mathcal{H} \leftarrow \emptyset$;

  for $i \in \{1, 2, \cdots, N_v\}$ **do**

    $\boldsymbol{v}_i \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}))$; `// sample an embedding from Euclidean space`

    $\bar{\boldsymbol{z}}_i = \exp_{\boldsymbol{0}}^c(\boldsymbol{v}_i)$; `// map to Poincaré ball, Equation (2)`

    $D_i, \boldsymbol{z}_i \leftarrow \arg\min_{(D,\boldsymbol{z})\in\mathbb{S}} \mathcal{D}_c(\bar{\boldsymbol{z}}_i, \boldsymbol{z})$; `// find the nearest valid embedding and its corresponding design, Equation (1)`

    use π to control current robot design D_i and store trajectories to $\mathcal{H}$;

  update π with PPO using samples in $\mathcal{H}$;

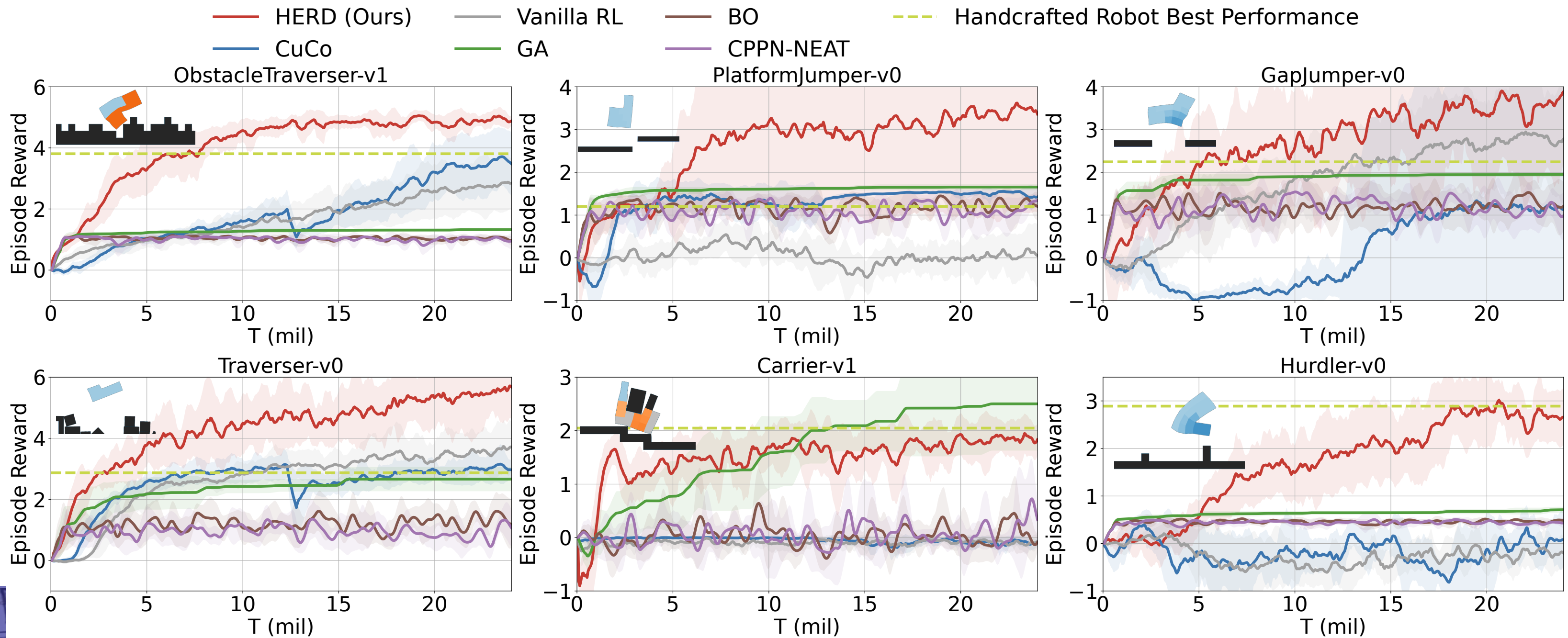
  update $\boldsymbol{\mu}$ by averaging the elite $\boldsymbol{v}_i$s based on the performance in $\mathcal{H}$, and linearly decrease $\boldsymbol{\sigma}$;

$D^*, \boldsymbol{z}^* \leftarrow \arg\min_{(D,\boldsymbol{z})\in\mathbb{S}} \mathcal{D}_c(\exp_{\boldsymbol{0}}^c(\boldsymbol{\mu}), \boldsymbol{z})$; `// optimal robot design`

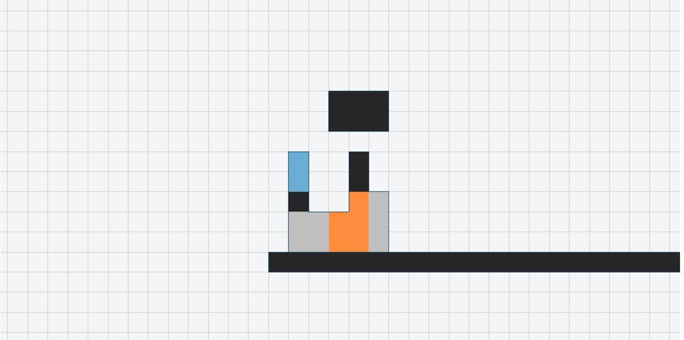
Output: optimal robot design D^*, control policy π

Results

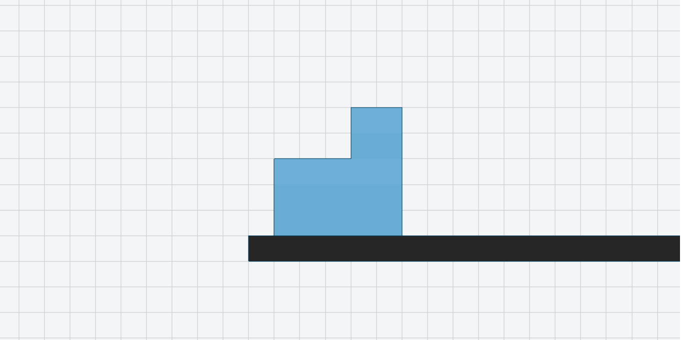
- Hard Tasks
  - Robot design can improve performance compared to handcrafted robots
  - HERD can effectively help design robots



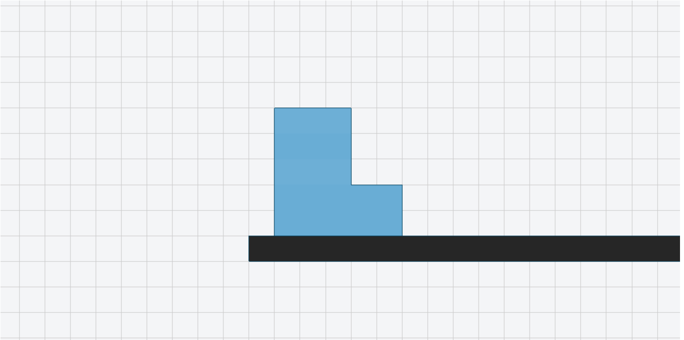
Replay



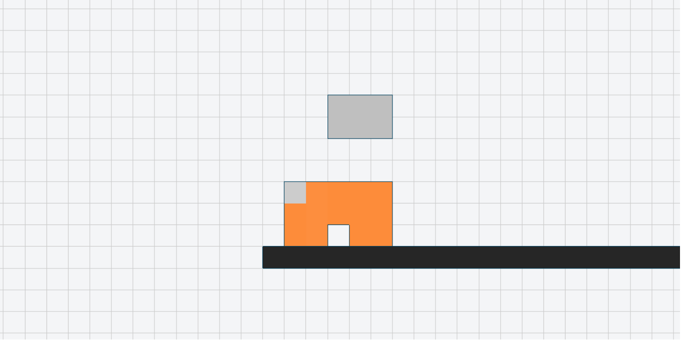
Carrier-v1



PlatformJumper-v0



GapJumper-v0



Carrier-v0

https://sites.google.com/view/hyperbolic-robot-design

Thanks for Your Listening