# Parental Guidance: Evolutionary Distillation for Non-Prehensile Mobile Manipulation

**Octi Zhang**[†], **Quanquan Peng**[‡], **Rosario Scalise**[†], **Byron Boots**[†]
[†]Paul G Allen School, University of Washington
[‡]Shanghai Jiao Tong University

**Abstract:**

Developing robotic agents that can perform well in diverse environments while showing a variety of behaviors is a key challenge in AI and robotics. Traditional reinforcement learning (RL) methods often create agents that specialize in narrow tasks, limiting their adaptability and diversity. To overcome this, we propose a preliminary, evolution-inspired framework that includes a reproduction module, similar to natural species reproduction, balancing diversity and specialization. By integrating RL, imitation learning (IL), and a coevolutionary agent-terrain curriculum, our system evolves agents continuously through complex tasks. This approach promotes adaptability, inheritance of useful traits, and continual learning. Agents not only refine inherited skills but also surpass their predecessors. Our initial experiments show that this method improves exploration efficiency and supports open-ended learning, offering a scalable solution where sparse reward coupled with diverse terrain environments induces a multi-task setting.

## 1 Introduction

Developing robotic agents that can generalize across diverse environments while continually evolving their behaviors is a core challenge in AI and robotics. The difficulties lie in solving increasingly complex tasks and ensuring agents can continue learning without converging on narrow, specialized solutions. Quality Diversity (QD) [1, 2] methods effectively foster diversity but often rely on trial and error, where the path to a final solution can be convoluted, leading to inefficiencies and uncertainty.

Our approach draws inspiration from nature's inheritance process, where offspring not only receive but also build upon the knowledge of their predecessors. Similarly, our agents inherit distilled behaviors from previous generations, allowing them to adapt and continue learning efficiently, eventually surpassing their predecessors. This natural knowledge transfer reduces randomness, guiding exploration toward more meaningful learning without manual intervention like reward shaping or task descriptors.

What sets our method apart is that it offers a straightforward, evolution-inspired way to consolidate and progress, avoiding the need for manually defined styles or gradient editing [3, 4] to prevent forgetting. The agent's ability to retain and refine skills is driven by a blend of IL and RL, naturally passing down essential behaviors while implicitly discarding inferior ones.

We make the following contributions:

1. **Distributed Evolution Framework**: We propose a framework that distributes the evolution process across multiple compute instances, efficiently scheduling and analyzing evolution.
2. **Integration of Learning Paradigms**: We integrate RL to find locally optimal policies, automatic curriculum to continually challenge the learners, and IL to pass down specialists' optimal solutions via distillation within the aforementioned evolutionary framework.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

3. **Preliminary Evaluation of Method Efficacy**: We evaluate our framework using behavior cloning (an IL method) and RL as baselines. We compare a number of integration strategies, and we provide guidance on how to best transition between IL and RL to maximize the benefits of our framework.

## 2 Related Works

**Quality Diversity and Evolutionary Approaches** QD algorithms, combined with evolutionary methods [1, 5, 6], promote behavioral diversity by exploring a range of solutions optimized for different niches. These approaches have shown success in control tasks [7] by expanding skill sets within a parent's nearby space, improving exploration. However, single-lineage inheritance can struggle when tasks require joint capabilities discovered in distant regions. In nature, species accelerate exploration by learning from each other, and passing knowledge across generations. Mammals, known for their intelligence and complex social behavior, all reproduce sexually, an intriguing biological parallel to how we approach knowledge sharing in algorithms. Through mechanisms like knowledge distillation or reproduction, agents can share behaviors, develop generalizable skills, and reduce dependence on long inheritance cycles.

**RL, IL in Context of Continual Learning** RL [8, 9] has been shown to excel at specialized tasks but struggles to generalize beyond predefined hand-crafted rewards. IL effectively replicates expert behaviors but often lacks adaptability and improvement. The continual learning community recognizes this divergence and has begun to explore bridging RL and IL to mirror the natural process of parent-child learning [10], where a child inherits knowledge from both parents before becoming independent and excelling in its own unique way. A successful outcome in this context implies that the student policy's growth stems from both parental skills and its own continuous exploration. Our prelimiary experimentation is designed to validate this intuition. While previous works have focused on distillation techniques [11, 12] that ensure knowledge transfer, the transition from IL to RL remains underexplored in the context of coninual learning. We explore the IL-to-RL transition in a natural evolution framework, discussing its role with respect to the wholistic evolutionary framework.

## 3 Methodology

### 3.1 Distributed Evolution with a Central Orchestrator

At the core of our framework is a central scheduler that manages an evolutionary tree, represented as a Directed Acyclic Graph (DAG). The scheduler selects the next species to train based on the current state, governed by an evolutionary Markov Decision Process (MDP). Each job runs on a dockerized compute instance of Nvidia Isaac Lab [13], which requests tasks from the scheduler. The scheduler handles the outer-level evolutionary progression and data management, while the compute instances focus on executing the inner-level training.

### 3.2 Continual Learning via Behavioral Inheritance and Reinforcement Learning

Behavioral inheritance refers to the process where offspring agents inherit traits and behaviors from their parent species. This inheritance is facilitated through the DAgger [14], enabling agents to blend behaviors across multiple parent species. The process includes two key phases, weighted by a BC decay rate $\lambda \in (0, 1)$. During behavior distillation, offspring are trained on aggregated datasets derived from the experiences of both parents, allowing them to inherit and combine a diverse range of behaviors. As training progresses, RL progressively takes over to further refine and enhance their skills. We employed Proximal Policy Optimization (PPO) [15], though this approach is flexible and can accommodate other RL algorithmic variants. This phase enables agents to move beyond inherited behaviors, cultivating their unique capabilities while building on their parents' foundation. For further details on the learning process, see Algorithm. 1.

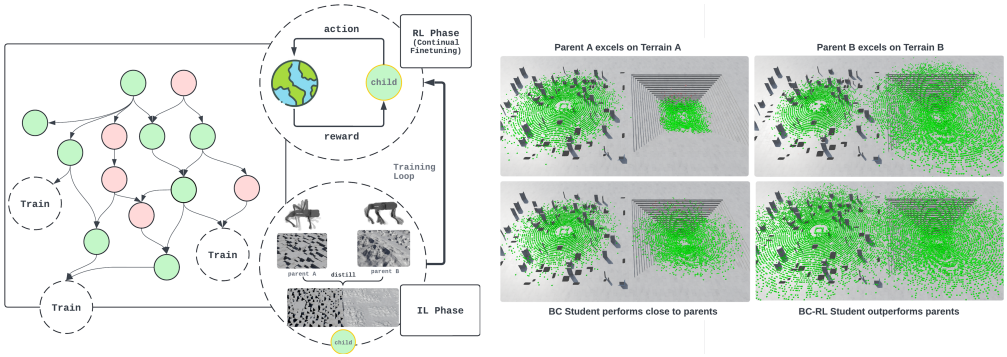


Figure 1: **Left**: Setup of Holistic Evolutionary Framework, an evolution scheduler that maintains a phylogenetic tree (a directed acyclic graph(DAG)). Each training cycle can be modularized and dispatched as a standalone process to an arbitrarily scalable number of compute nodes. Each node performs BC and RL and attaches the new child (the future parent) back to the phylogenetic tree. **Right**: Comparison of effect of BC-RL process. Each parent is a specialist in its own niche terrain and performs worse in other terrains. Each green dot represents treats successfully fetched by the agent. We let the agent run until the distance increment stops, indicating the agent skill has saturated. Distillation enables agents to do almost as well as the parent in both terrains, but BC-RL is able to exceed the performance of both parents on the union of tasks.

### 3.3 Training Setup

We set up training with a locomotion task, where the goal of the agent is to control its joints to fetch a treat. The treat in the environment is defined by its position $x_{\text{treat}}$, which is initialized near the agent's starting position $x_{\text{start}}$. The treat's position remains the same until the agent successfully fetches it. Once fetched, the treat spawns farther away. Specifically, the spawning radius of the treat is updated based on the agent's success at fetching the treat:

$$x_{\text{treat}}^{\text{episode}+1} = \begin{cases} x_{\text{treat}}^{\text{episode}} + \Delta x_{\text{treat}}, & \text{if } \|x_{\text{agent},t} - x_{\text{treat}}^{\text{episode}}\| \leq \epsilon, \\ x_{\text{treat}}^{\text{episode}} - \Delta x_{\text{treat}}, & \text{otherwise} \end{cases} \tag{1}$$

$\Delta x_{\text{treat}} > 0$ is the increment in the treat's position, $x_{\text{agent},t}$ is the agent's position at time step $t$, and $\epsilon$ is a small threshold indicating when the agent has successfully fetched the treat.

The overall objective of the student policy, $\pi_s$, maximizes the convex combination of $\mathcal{L}_{\text{BC}}$ and $\mathcal{L}_{\text{RL}}$ which seeks to balance both sub-objectives. The surrogate loss $\mathcal{L}_{\text{surrogate}}$ and $\mathcal{L}_{\text{value}}$ are defined as in PPO [15]:

$$\mathcal{L}_{\text{BC}} = \frac{1}{T} \sum_{t=0}^{T} \left( \mathcal{L}_{\text{BC}}^{\mu,t} + \mathcal{L}_{\text{BC}}^{\sigma,t} \right) \tag{2}$$

$$\mathcal{L}_{\text{RL}} = \mathcal{L}_{\text{surrogate}} - c \cdot KL[\pi_{s\_old}(\cdot|s_t), \pi_s(\cdot|s_t)] + \beta \cdot \mathcal{L}_{value} \tag{3}$$

$$\mathcal{L} = \lambda^i \cdot \mathcal{L}_{\text{BC}} + (1 - \lambda^i) \cdot \mathcal{L}_{\text{RL}} \tag{4}$$

**Algorithm 1:** Method Overview

**Input:** Number of iterations $N$, Expert policies $\{\pi_E^t\}$, Student policies $\pi_S$

Initialize $\pi_S$, $\lambda$; ▷ $\lambda$ is bc weight decay rate;
**for** $i = 1$ **to** $N$ **do**
  **for** *step* $= 1$ **to** *num_steps_per_env* **do**
    Use policy $\pi_S$ to sample rollouts $\mathcal{D} := \{(obs, a_s, reward)\}$ ;
    **for** *each terrain* $t$ **do**
      Obtain expert $\pi_E^t$ actions mean $\mu_E^t$ and std $\sigma_E^t$ from $\pi_E^t$;
  **for** *each* $d$ *in* $\mathcal{D}$ **do**
    Compute total loss $\mathcal{L}$ and update $\pi_S$ based on Eqn. 4

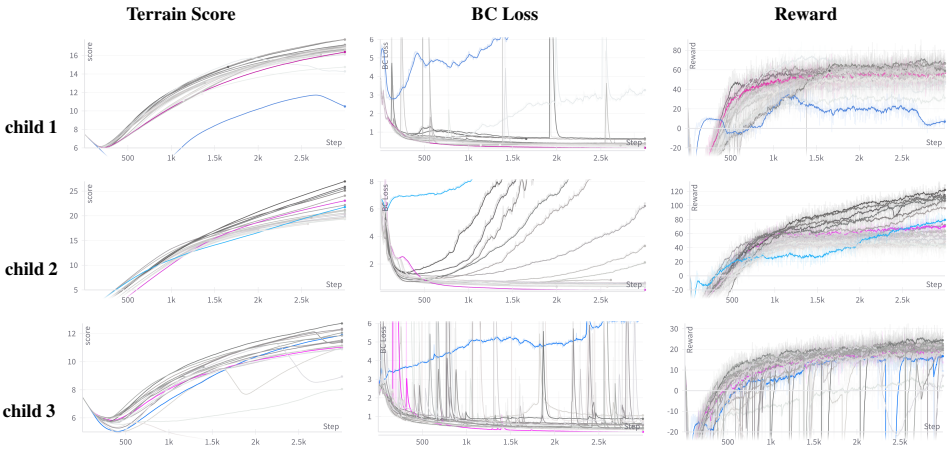


Figure 2: Comparison of 80 BC-RL transitions in which score, BC Loss, and reward plots across 3 children from 3 distinct families. The gradient from black to white ranks transitions from best to worst performance. The purple curve represents pure BC, and the blue curve represents pure RL.

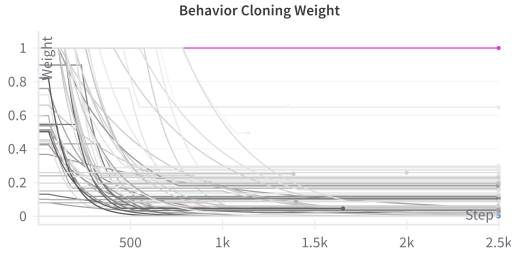


Figure 3: Comparison relationship between BC-to-RL transitions (same data from figure 2) and their performance. Again, the gradient from black to white ranks transitions from best to worst performance. The purple curve represents pure BC, and the blue curve represents pure RL.

## 4 Experimental Results

We tested 80 transitions from BC to RL by progressively reducing the BC weight. The most successful transitions showed continuous improvement, leveraging parent behaviors to enhance exploration and avoid local minima. Pure BC (fixed weight at 1) demonstrated a steady reduction in BC loss, while pure RL (weight at 0) saw BC loss increase during training. Transitions that shifted to RL early, within 200 steps, maintained a BC loss belower than pure RL yet performed better than pure BC or RL, indicating more efficient learning.

Transitions that stayed longer in the BC phase, however, remained closer to parent behaviors and failed to outperform them in later RL stages, emphasizing the need to transition to RL early enough to enable meaningful exploration and improvement.

## 5 Conclusion

We introduce an evolution-inspired optimization framework that uniquely merges the strengths of IL and RL into a reproduction module—a first of its kind within an open-ended agent system. The goal is to understand how transitioning from IL to RL helps agents consolidate knowledge and adapt to complex environments, where parents have specialized in different tasks. Our initial results show that this module plays a key role in efficiently merging skills from parent agents, helping offspring perform better than single-task optimization. This approach opens up opportunities for further research on how reproduction can enhance knowledge transfer and the development of more adaptable agents in the context of open-ended learning.

## References


[1] R. Wang, J. Lehman, J. Clune, and K. O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[2] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[3] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[5] J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[6] J. Lehman and K. Stanley. Evolving a diversity of creatures through novelty search and local competition. *Genetic and Evolutionary Computation Conference, GECCO'11*, pages 211–218, 07 2011. doi:10.1145/2001576.2001606.

[7] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk. Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training. *arXiv preprint arXiv:2305.12127*, 2023.

[8] V. Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[9] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[10] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020.

[11] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[12] G. Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[14] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.