DIRT: The Distributed Intelligent Replicator Toolkit

Chengrui Wang[1,2], Chase Van Amburg[1], Chloe Huangyuan Su[1], Joseph Bejjani[1], Yasin Mazloumi[1], Naeem Khoshnevis[1], Sham Kakade[1], Kiante Brantley[1], Aaron Walsman[1]

[1]Kempner Institute, Harvard University, USA
[2]Fudan University, China
aaronwalsman@fas.harvard.edu

Abstract

We introduce **DIRT**, the **D**istributed **I**ntelligent **R**eplicator **T**oolkit, a new evolutionary simulator designed to support GPU-accelerated massive populations and a large physical scale for artificial life research. DIRT is designed to simulate mobile agents in a natural environment that must consume resources in order to survive and reproduce. It uses highly configurable grid world dynamics that include multiple sensor types, complex terrain, water features, and a climate system. Agents in DIRT have traits that control their individual abilities. DIRT is interoperable with a wide variety of policy models and population algorithms. DIRT is built on JAX and as a result, it can support populations of over 10,000 agents on a single GPU. We also provide a rich set of measurement tools and a 3D viewer which allows fine-grained inspection and tracking of individual agents.

Submission type: **Late Breaking Abstract**

Data/Code available at: `http://github.com/aaronwalsman/dirt`

Introduction

Artificial Life seeks to understand the emergence of complexity and open-ended evolution in artificial organisms, yet progress has often been constrained by the computational cost of simulating large, interacting populations with sufficiently rich environments. Thus, we present **DIRT**, a scalable ecological simulator that supports large-scale evolutionary studies on modern accelerators. DIRT is motivated by the long and productive history of grid-world abstractions in ALife and RL, ranging from classical cellular automata and digital evolution to contemporary GPU-native simulators, while pushing scalability, configurability, and inspectability for population-level studies.

Ecological substrate. Grid-based abstractions have a long history in ALife stretching back to cellular automata (Von Neumann et al., 1966; Gardner, 1970; Wolfram and Gad-el Hak, 2003; Adamatzky and Martínez, 2016). DIRT builds on this rich body of work, while adding modern features such as elevation, hydrology and climate dynamics. Environmental processes are modeled as coupled fields, including discrete water flow with evaporation and precipitation, temperature and moisture cycles influenced by day/night and seasonal rhythms, and resource dynamics spanning light, biomass, energy, and water. Multimodal environmental signals such as visual color maps, odors and diffusion-based chemical cues, audio propagation, thermal measurements, and wind-driven transport, are represented as tensor fields updated via GPU-parallel operators, and then exposed to agents through parameterized sensors. This design builds on ecological simulation approaches in artificial life and multi-agent reinforcement learning (Lu et al., 2024; Sun et al., 2025), but provides a configurable toolkit for a variety of experimental setups.

Agents, traits, and actions. Agents in DIRT are situated in a 2D grid world and must acquire resources to survive and reproduce. This is consistent with the long tradition of digital organisms in ALife (Adami and Brown, 1994; Channon and Damper, 1998; Soros and Stanley, 2014). Each agent is parameterized by a policy that maps the agent's observations to actions and a set of heritable *traits* that specify the agent's capabilities and costs, echoing earlier work on morphology in evolving virtual creatures (Sims, 1994). Traits determine movement speed, attack radius and strength, metabolic efficiency and others, supporting ecological trade-offs observed in both natural and artificial systems.

DIRT defines a compact but expressive set of discrete action categories including *Move*, *Attack*, *Eat*, *Expell*, *Call*, *Mark Scent*, and *Reproduce*, which are flexibly modulated by traits and environmental state. This design reflects conventions from earlier agent-based models of flocking and signaling (Reynolds, 1987) and incorporates communicative and reproductive actions inspired by open-ended simulation frameworks such as JaxLife (Lu et al., 2024) and Amorphous Fortress (Charity et al., 2023). Importantly, all actions incur explicit energy and biomass costs, creating resource-based constraints that enforce trade-offs (e.g., faster movement or stronger attacks consume more energy), in line with ecological and evolutionary principles (MacArthur and Wilson, 2001; Rosenzweig, 1995). In addition, the constants that specify the exact nature of these trade-offs can be spec-

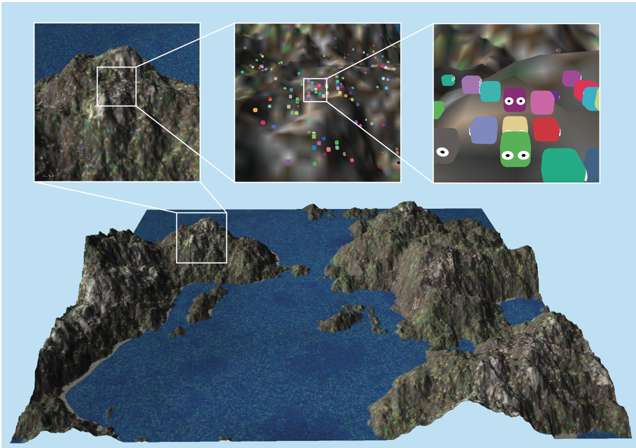

Figure 1: Simulator Overview: This map contains 1024×1024 grid cells and 10000 individual agents, which are too small to see when fully zoomed out. The inset areas progressively zoom in on one patch of terrain to reveal detail.

ified as system-wide parameters.

Interoperable population interface. A key design principle of DIRT is the explicit separation between environmental dynamics and *population algorithms*. Rather than prescribing heredity rules, learning strategies, or policy representations, DIRT reports demographic events such as births, deaths, and parentage, and delegates the instantiation of new policies and traits to user-provided algorithms. This flexible interface builds on the tradition of open-ended evolutionary systems such as Tierra (Ray, 1992), Avida (Ofria and Wilke, 2004), and Echo (Holland, 1992), while addressing their limitations by decoupling environmental simulation from evolutionary operators.

Conceptually, this modularity parallels the standardized APIs that have accelerated reinforcement learning research, such as Gym (Brockman et al., 2016), PettingZoo (Terry et al., 2021), Gymnasium (Towers et al., 2024), and Gymnax (Lange, 2022), which expose consistent agent–environment interfaces while leaving policy design unrestricted. By not embedding heredity or adaptation mechanisms into the simulator core, DIRT ensures compatibility with diverse research agendas, from open-ended evolution to large-scale multi-agent learning, while maintaining reproducibility and comparability across population-level experiments.

Acceleration and scale. DIRT is implemented entirely in JAX, with all major environment updates, including map field evolution, diffusion-based propagations, hydrology, and sensor extraction, compiled as batched tensor operations for GPUs and TPUs. This minimizes host–device communication overhead, especially if user-defined policies are colocated on the same GPU, which is a known bottleneck in large-scale simulation frameworks (Suarez et al., 2023). This design aligns with recent trends toward accelerator-native environments such as PGX (Koyamada et al., 2023),

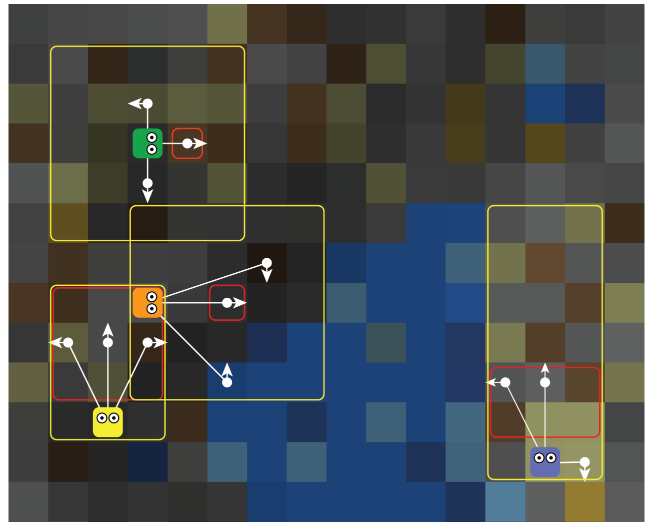

Figure 2: An example of several agent observation and action spaces. In this example, each agent has three motion primitives, shown here as white arrows. Note that these primitives perform different motions for each agent, as they each have different traits. Each agent has a single attack primitive in this example represented by the red box. The agent's visual range is represented by the yellow box. The pixelated background represents the color map that agents see, where blue pixels are water, grey is rock and shades of yellow and brown represent different distributions of resources.

JaxMARL (Rutherford et al., 2023), GigaStep (Lechner et al., 2023), and Craftax (Matthews et al., 2024).

Depending on policy parameterizations, DIRT can sustain populations exceeding 10,000 agents on a single GPU, a scale that enables population-level experimental regimes including ecological niche formation, lineage dynamics, and large-scale selection pressures, which may be difficult to access in smaller systems. This scalability provides a practical foundation for exploring open-ended evolution and emergent population dynamics at scales that were previously unobtainable due to computational constraints.

Measurement and visualization. DIRT incorporates an integrated reporting framework designed to balance scalability with analytical depth. At each timestep, lightweight and customizable summaries are recorded, which can include global environmental fields, population-level statistics, and sampled agent states, while periodic checkpoints capture complete system state to permit exact reconstruction and selective replay. This structure enables both longitudinal analyses of population dynamics and fine-grained investigation of individual behaviors.

Dirt also includes an interactive 3D visualizer that can play back simulated sequences and render different environmental layers alongside agent overlays(Figure 1,2). This interface allows stepwise playback of simulations and targeted reruns from stored checkpoints, thereby supporting detailed inspection of emergent ecological and evolutionary processes.

References

Adamatzky, A. and Martínez, G. J. (2016). *Designing beauty: the art of cellular automata*, volume 20. Springer.

Adami, C. and Brown, C. T. (1994). Evolutionary learning in the 2d artificial life system. In *Artificial life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*, volume 4, page 377. MIT press.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

Channon, A. and Damper, R. (1998). Perpetuating evolutionary emergence.

Charity, M., Rajesh, D., Earle, S., and Togelius, J. (2023). Amorphous fortress: Observing emergent behavior in multi-agent fsms. *arXiv preprint arXiv:2306.13169*.

Gardner, M. (1970). Mathematical games. *Scientific american*, 222(6):132–140.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Koyamada, S., Okano, S., Nishimori, S., Murata, Y., Habara, K., Kita, H., and Ishii, S. (2023). Pgx: Hardware-accelerated parallel game simulators for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 45716–45743.

Lange, R. T. (2022). gymnax: A JAX-based reinforcement learning environment library.

Lechner, M., Yin, L., Seyde, T., Wang, T.-H., Xiao, W., Hasani, R., Rountree, J., and Rus, D. (2023). Gigastep - one billion steps per second multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*.

Lu, C., Beukman, M., Matthews, M., and Foerster, J. (2024). Jaxlife: An open-ended agentic simulator. In *ALIFE 2024: Proceedings of the 2024 Artificial Life Conference*. MIT Press.

MacArthur, R. H. and Wilson, E. O. (2001). *The theory of island biogeography*, volume 1. Princeton university press.

Matthews, M., Beukman, M., Ellis, B., Samvelyan, M., Jackson, M., Coward, S., and Foerster, J. (2024). Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *International Conference on Machine Learning (ICML)*.

Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial life*, 10(2):191–229.

Ray, T. S. (1992). Evolution, ecology and optimization of digital organisms. *Santa Fe*.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34.

Rosenzweig, M. L. (1995). Species diversity in space and time. *(No Title)*.

Rutherford, A., Ellis, B., Gallici, M., Cook, J., Lupu, A., Ingvarsson, G., Willi, T., Khan, A., de Witt, C. S., Souly, A., et al. (2023). Jaxmarl: Multi-agent rl environments in jax. *arXiv preprint arXiv:2311.10090*.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372.

Soros, L. and Stanley, K. (2014). Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In *Artificial Life Conference Proceedings*, pages 793–800. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . .

Suarez, J., Bloomin, D., Choe, K. W., Li, H. X., Sullivan, R., Kanna, N., Scott, D., Shuman, R., Bradley, H., Castricato, L., et al. (2023). Neural mmo 2.0: A massively multi-task addition to massively multi-agent learning. *Advances in Neural Information Processing Systems*, 36:50094–50104.

Sun, X., Mangan, M., Peng, J., and Yue, S. (2025). I2bot: an open-source tool for multi-modal and embodied simulation of insect navigation. *Journal of the Royal Society Interface*, 22(222):20240586.

Terry, J., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L. S., Dieffendahl, C., Horsch, C., Perez-Vicente, R., et al. (2021). Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al. (2024). Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

Von Neumann, J., Burks, A. W., et al. (1966). Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1):3–14.

Wolfram, S. and Gad-el Hak, M. (2003). A new kind of science. *Appl. Mech. Rev.*, 56(2):B18–B19.