LegoTron: An Environment for Interactive Structural Understanding

Aaron Walsman[1], Muru Zhang[1], Adam Fishman[1], Karthik Desingh[1], Dieter Fox[1,2], Ali Farhadi[1,3]
[1]University of Washington, [2]NVIDIA, [3]Apple
{*awalsman, nanami17, afishman, kdesingh, fox, afarhadi*}@*cs.washington.edu*

github.com/aaronwalsman/ltron

Abstract

Visual reasoning about geometric structures with detailed spatial relationships is a fundamental component of human intelligence. As children, we learn how to reason about this structure not only from observation, but also by interacting with the world around us – by taking things apart and putting them back together again. We introduce a new learning environment designed to explore the interplay between interactive reasoning, scene understanding and construction by mining a previously untapped high-quality data source: fan-made Lego[1] creations that have been uploaded to the internet. To make use of this data we have built LegoTron, a fully interactive 3D environment that allows a learning agent to assemble, disassemble and manipulate these models. Our goal is to provide an interactive playground for agents to explore and manipulate complex scenes and recover their underlying structure.

1. Introduction

The physical world is made out of parts. Buildings are made out of roofs, rooms and walls, chairs are made out of seats, legs and arms, and cars have doors, wheels and windshields. The ability to reason about these parts in the physical world and the structural relationships between them are a key component of our ability to build tools, solve complex organizational problems and in general manipulate the world around us.

In order to explore the relationships between vision, interaction and structural reasoning, we introduce a new environment and Lego simulator, which we call LegoTron, designed specifically for interactive machine learning. LegoTron leverages the fan-made LDraw[6] parts library and LDCad[9] software which contain thousands of individual brick types, and annotations describing how they may be connected to each other. We also include 1491 fan-made reproductions of official Lego models from the LDraw Open Model Repository which provide a rich source of naturally occurring structural data.

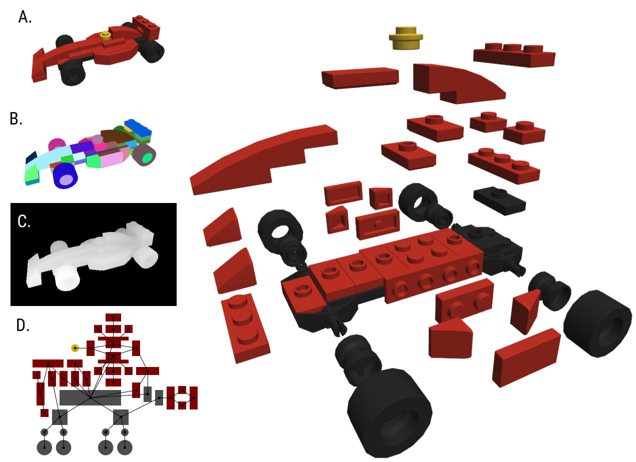

Figure 1. A car model from the Open Model Repository[6] in our environment. Our environment supports: color rendering (A), part masking (B), depth rendering (C), and connectivity structure with a graph-based representation (D).

We are not the first to use Lego bricks as a platform for AI [3, 4, 7, 8, 10, 14, 15, 17], but to our knowledge, LegoTron is the first interactive machine learning environment that provides access to the full library of Lego bricks and is compatible with a wide range of complex fan-made Lego models. Furthermore, we provide a comprehensive set of action and observation spaces using the OpenAI Gym [1] API to enable a wide range of tasks.

Interactive and embodied machine learning has been a popular research topic in recent years which has lead to a number of environments and challenges [2, 5, 11, 12, 13, 16]. While many of these environments offer visual and physical realism for indoor environments, LegoTron provides complex geometric scenes with granular labeling along with assembly and disassembly action primitives to support a variety of interactive reasoning tasks. In contrast to other embodied AI environments, every single object in LegoTron can be manipulated by a learning agent.

[1]Lego is an official trademark of the LEGO group, which has not endorsed this paper or dataset.

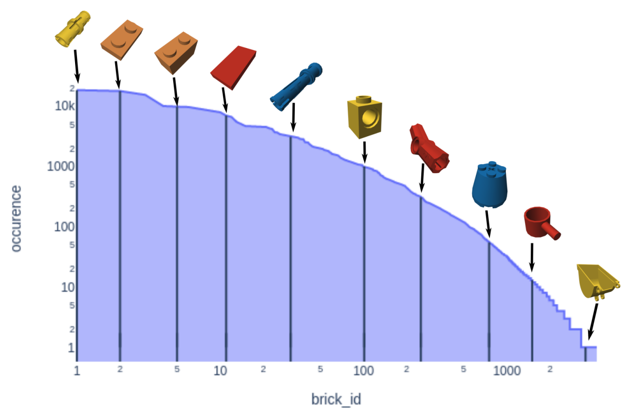

Figure 2. Distribution of brick frequency in the OMR data with examples of various common and rare bricks. The x-axis is the log-rank of each class sorted by frequency with the most common brick on the left and the least common on the right. The y-axis is the log-frequency of each class.

2. LegoTron Environment

Scene Structure: The atomic units of our simulator are individual Lego bricks. Each brick consists of a single rigid shape with some number of annotated connection points that describe where and how it may be attached to other bricks. There are several different connection types that may exist on individual bricks. The most common by far are the prototypical studs that cover the top surface of many bricks and the corresponding cavities that exist on the underside. We use the LDCad [9] snap point description metadata to describe these connection points.

In order to make Lego models accessible to a variety of learning agents, we provide a suite of observation and action spaces that can be combined to train different agents with different modalities.

Observation: LegoTron supports color, depth and segmentation rendering for visual observation spaces. We use a custom OpenGL renderer to allow both interactive and headless rendering with EGL. We also support an observation space that returns a symbolic scene representation consisting of a list of bricks, along with the color and 3D transform of each, as well as a list of connection points that are attached to each other.

We provide two camera motion action spaces that allow a learning agent to manipulate the virtual camera that views the 3D scene. The first is an orbital camera that remains aimed at a fixed center point. In this setting an agent can rotate the camera about the center point and translate closer or farther from it. The second is a free floating camera that can translate and rotate in any direction relative to the center of the camera.

Action: We provide assembly and disassembly action spaces that allow the agent to add, move and remove bricks from the scene. To add new bricks to a scene, an agent must specify its part ID, color and placement, which may be specified by either providing a position and orientation of the new brick relative to the camera, or by specifying connection points to attach to.

Moving existing bricks is similar. An agent can specify a brick to move using either a 2D pixel location or an ID based on the symbolic representation of the scene. The new location of the brick can be specified using either connection points or raw position and rotation values. Additionally we support discrete motion actions that translate or rotate bricks by a fixed step size.

Bricks may also be removed from the scene entirely by specifying either their symbolic ID or a 2D pixel location.

Data: We bundle 1491 fan-made reproductions of official Lego sets into LegoTron in order to provide a source of natural data for learning agents. These use a total of 4,224 brick types which follow a long-tail distribution shown in Figure 2. The models vary in size from 5 to 5197 bricks, with the largest being a massive reproduction of the Millennium Falcon space ship from Star Wars. The mean set size is 336.8 bricks, and the median is 131.5. Several examples are shown in Figure 3.

Tasks: Our goal in building LegoTron is to provide a resource that supports interactive construction, generative modelling and scene understanding problems. However, at the moment we do not provide explicit tasks or reward structures along these lines and instead provide the simulator and data together as an open-ended learning environment. We are however eager to develop a standard set of challenges to explore this data comprehensively.

Figure 3. Examples of models from the Open Model Repository[6]

.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. 1

[2] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3164–3174, 2020. 1

[3] Alexandre Devert, Nicolas Bredeche, and Marc Schoenauer. Blindbuilder: A new encoding to evolve lego-like structures. In *European Conference on Genetic Programming*, pages 61–72. Springer, 2006. 1

[4] Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. Duplotrack: A reatime system for authoring and guiding duplo model assembly. In *Proceedings of the 25th annual ACM symposium adjunct on User interface software and technology*, New York, NY, USA, 2012. ACM. 1

[5] David Hall, Ben Talbot, Suman Raj Bista, Haoyang Zhang, Rohan Smith, Feras Dayoub, and Niko Sünderhauf. The robotic vision scene understanding challenge. *arXiv preprint arXiv:2009.05246*, 2020. 1

[6] James Jessiman. LDraw. `http://www.ldraw.org`. 1, 2

[7] J. Kim. Survey on automated lego assembly construction. 2015. 1

[8] Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: Optimizing lego designs. *ACM Transactions on Graphics (TOG)*, 34(6):1–12, 2015. 1

[9] Roland Melkert. LDCad. `http://www.melkert.net/LDCad`. 1, 2

[10] Maxim Peysakhov and William C Regli. Using assembly representations to enable evolutionary design of lego structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM*, 17(2):155, 2003. 1

[11] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 1

[12] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, et al. igibson, a simulation environment for interactive tasks in large realisticscenes. *arXiv preprint arXiv:2012.02924*, 2020. 1

[13] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 1

[14] Romain Testuz, Yuliy Schwartzburg, and M. Pauly. Automatic generation of constructable brick sculptures. In *Eurographics*, 2013. 1

[15] Rylee Thompson, Elahe Ghalebi, Terrance DeVries, and Graham W Taylor. Building lego using deep generative models of graphs. *arXiv preprint arXiv:2012.11543*, 2020. 1

[16] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multi-on: Benchmarking semantic map memory using multi-object navigation. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1

[17] Jie Zhou, Xuejin Chen, and Y Xu. Automatic generation of vivid lego architectural sculptures. In *Computer Graphics Forum*, volume 38, pages 31–42. Wiley Online Library, 2019. 1

Appendix: Data Sourcing

The data files included in LegoTron represent the creative works of hundreds of individuals who have generously made their work available for free online. All data and external software used in LegoTron has been ethically sourced from online repositories with Creative Commons or other open licensing terms. Wherever possible we have made efforts to contact the authors and community in order to make them aware of our usage of this work. For more information see github.com/aaronwalsman/ltron/LICENSE and github.com/aaronwalsman/ltron/CONTRIBUTORS.