Comparing Human and AI Behavior in 3D Navigation Environments

Jeffrey Liu

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-111
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-111.html

May 14, 2021

Acknowledgement

I would first like to thank Professor Darrell for agreeing to be my advisor and for giving me the opportunity to be involved in the wonderful research community at Berkeley. I would also like to thank Professors Deepak Pathak, Pulkit Agrawal, and Alison Gopnik for their extended guidance and feedback on this project.

I would especially like to thank graduate students Jasmine Collins, David Chan, and Eliza Kosoy for their mentorship throughout this difficult period. I would also like to thank Adrian Liu for his valuable contributions.

Last, but certainly not least, I would like to thank all my peers and friends for making my time at Berkeley a truly unforgettable one.

Comparing Human and AI Behavior in 3D Navigation Environments

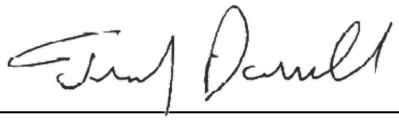
by Jeffrey Liu

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:



Professor Trevor Darrell
Research Advisor

5/14/21

(Date)

* * * * * * *

Professor Alison Gopnik
Second Reader

5/14/21

(Date)

Comparing Human and AI Behavior in 3D Navigation Environments

by

Jeffrey Liu

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell, Chair
Professor Alison Gopnik

Spring 2021

Comparing Human and AI Behavior in 3D Navigation Environments

Abstract

Comparing Human and AI Behavior in 3D Navigation Environments

by

Jeffrey Liu

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

While modern artificial intelligence agents have achieved superhuman performance in specific tasks, training artificial agents that can efficiently explore and generalize to new tasks remains an open problem. Recent work has turned to humans as a source of inspiration to tackle this problem. Humans have shown an ability to explore the world in such a way that translates to generalizable skills in later life, a trait that modern artificial intelligence has failed to replicate. Investigating the specific ways in which humans and artificial agents deviate in behavior may thus lend insight into ways that algorithms can be improved. In this work we present an online platform to design and carry out experiments comparing human and agent behavior in 3D navigation tasks. We also present a comparison of behaviors between humans and agents in both procedurally-designed and human-designed mazes, highlighting ways in which current algorithms are both similar and distinct from humans.

To my family and friends for their continuous support

Contents

List of Figures

List of Tables

Acknowledgments

I would first like to thank Professor Darrell for agreeing to be my advisor and for giving me the opportunity to be involved in the wonderful research community at Berkeley. I would also like to thank Professors Deepak Pathak, Pulkit Agrawal, and Alison Gopnik for their extended guidance and feedback on this project.

I would especially like to thank graduate students Jasmine Collins, David Chan, and Eliza Kosoy for their mentorship throughout this difficult period. I would like to specifically thank Jasmine Collins for helping with the data analysis, David Chan for his work on the maze generation algorithm and the data collection platform, and undergraduate student Adrian Liu for his work on the Unity maze environment and the data collection platform. This project would not have been possible without all of your valuable contributions.

Last, but certainly not least, I would like to thank all my peers and friends for making my time at Berkeley a truly unforgettable one.

Chapter 1

Introduction

Recently, modern techniques in reinforcement learning (RL) and artificial intelligence (AI) have been able to achieve human-like, and in some cases super-human, performance on tasks such as Atari games [8]. Such advances often rely on a paradigm of training an artificial agent from scratch in order to solve a specific task. One question that arises in this context is one of exploration. How should these agents interact with their environment and gather information in an intelligent manner? Various techniques have been proposed to guide an agent's exploration during training [9, 10], however many state of the art algorithms today still rely on simple exploration strategies such as ϵ-greedy. Another question that arises is one of generalization. While current agents are extremely good at the task they are trained to accomplish, they often struggle in new or unseen environments. These questions have long thought to be related, with the underlying principle being that agents that learn sophisticated exploration strategies will be able to generalize better in new environments.

Recent attempts to tackle these problems have turned to humans as a source of inspiration. From a young age, humans have demonstrated an ability to interact with the world in a systematically curious fashion [15, 16]. Such exploration has been shown to lead to overarching generalizations that are transferable to a variety of different contexts and tasks [18]. As both of these areas are ones in which modern AI struggle in, recent work has tried to leverage concepts in human behavior such as intrinsic motivation [12] and curiosity [17], resulting in promising results [11, 2].

Despite these advances, it is not entirely clear whether formulations based on humans actually do a good job of capturing human-like behavior. Recent work by Kosoy et al. [6] has suggested the use of 3D navigation environments such as DeepMind Lab [1] as a means to compare human and agent behavior in simulated exploration tasks. We present extensions in this direction through 2 main contributions.

1. An online platform for crowd-sourcing maze designs for exploration and running experiments on human subjects within these designs for comparison.

2. A comparison of the exploration behaviors of humans and agents both with and without curiosity across procedurally-designed and human-designed mazes.

Chapter 2

Related Work

2.1 Human Agent Comparisons

Many previous works have included the performance of artificial agents on benchmarks such as Atari games [8] or navigation tasks [7] relative to humans. However, such comparisons are often very surface level on only measure high-level performance differences, i.e differences in total reward or success rate. Such works do not typically include analysis on specific differences in behavior.

Recently, however, there has been more focus placed on investigating these more specific differences. Comparison of human and agent performance in a video game setting found that human performance drastically dropped when familiar object priors were masked, suggesting learning such reusable priors could be critical for more generalizable and efficient AI algorithms [3]. Such insights could potentially result in new and interesting research directions, and our work aims to provide further contributions in this area.

2.2 Deepmind Lab and Unity environments

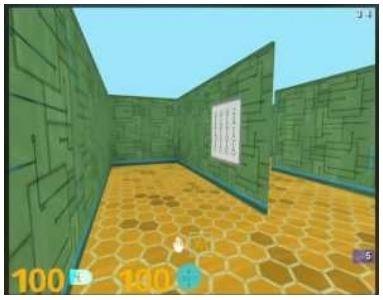
DeepMind Lab 2.1a is a learning environment based on the Quake game engine that provides complex 3D navigation tasks that can be used for both agents and humans. DeepMind Lab provides interfaces for both agents and humans to explore complex 3D mazes from a first person point of view, allowing for an interesting way to compare human and agent behavior. Previous work [6] has argued that such an environment makes for a more ecologically valid and appropriate setting for human-agent comparisons thanks to three main factors.

1. Rich visuals from a first-person point of view that more closely mirror human experience, as opposed to 2D Atari games or grid-world like settings.

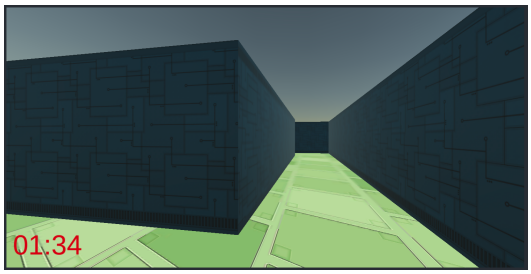
2. Controlled comparisons that offer more metrics for analysis and different avenues for research in terms of generalizaiton to new tasks

3. Customizability that allows for implementation of challenges for both humans and agents that can be used to test specific behaviors

In addition to DeepMind Lab, we also leverage Unity [5] 3D maze environments to carry out experiments on humans. Functionally the two environments are similar in that both contain rich 3D visuals and complex maze configurations, however the textures and physics engines are slightly different. Nevertheless, the two are similar enough to compare overarching behavioral trends on.



(a) DeepMind Lab



(b) Unity Maze

Figure 2.1: Screenshots of maze environments. We use DeepMind Lab for agent training and evaluation, while we use Unity for human evaluations

2.3 Intrinsic Curiosity

Recent work in the RL and AI fields has leaned on concepts in developmental psychology for ideas, with one example being the Intrinsic Curiosity Module (ICM), first presented by Pathak et al. [11]. Originally designed to help agents solve tasks in which external rewards are generally sparse, ICM draws upon ideas such as intrinsic motivation [12] and provides a self-supervised reward to encourage agents to explore their environment. Formally, consider an agent that receives observation x_t from the environment, takes an action a_t sampled from a policy $\pi(s_t; \theta_P)$ represented by a neural network with parameters θ_P, and transitions to a new state with observation x_{t+1}. ICM introduces two additional modules: an inverse dynamics module, and a forward dynamics module.

The inverse dynamics module consists of a neural network that is trained to encode the observation x_t into a feature vector $\phi(x_t)$, as well as a second sub-module that takes feature encodings $\phi(x_t), \phi(x_{t+1})$ as inputs and learns to predict the action a_t that is associated with the transition. The idea behind this module is to learn features that should be controllable by the agent, resulting in exploration rewards that should be invariant to noise in the environment. Such an exploration reward should then encourage the agent to explore truly novel states, rather than simply stay in states that provide a lot of noisy observations.

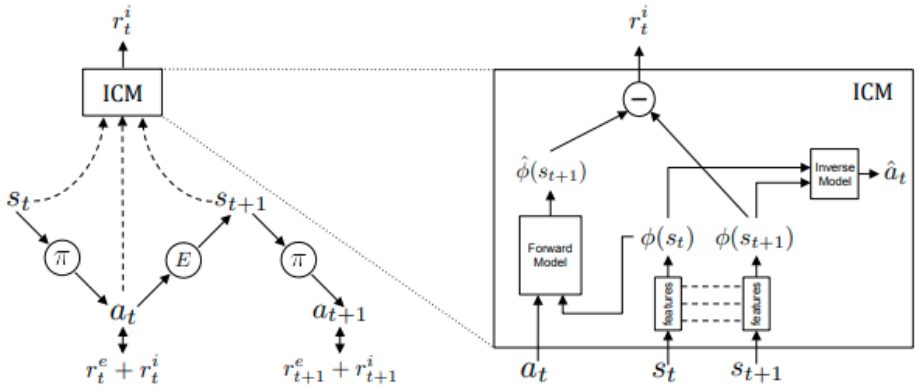


Figure 2.2: Overview of the ICM module, adapted from [11]

The forward dynamics module takes the feature representations learned by the inverse dynamics module and predicts the representation of the next state conditioned on the previous observation and action. The intrinsic curiosity exploration reward is defined as the error between the actual and predicted embeddings, i.e $||f(x_t, a_t) - \phi(x_{t+1})||_2^2$ where f is the learned dynamics module. The agent is then trained to maximize the sum $r_t = \alpha r_t^i + r_t^e$, where r_t^i is the curiosity reward previously described and r_t^e is the reward given by the environment. α is a coefficient that is used to scale the magnitude of the intrinsic reward. During training, the loss from the forward dynamics and inverse dynamics models are jointly optimized for along with the policy gradient loss.

Intuitively, ICM should incentivize agents to take actions that result in surprising outcomes for the agent, which should typically come in the form of stumbling across a new or novel state, mimicking human tendencies to explore and discover novel states. However, while the method may draw inspiration from human behavior, it is unclear if it is actually effective in replicating it, which we investigate further in later chapters.

Chapter 3

Human Data Collection Platform

The following chapter presents joint work with David Chan and Adrian Liu. Due to the COVID-19 pandemic, collecting human data for comparison presented a difficult task. To circumvent this, we designed and built a custom platform for human data collection that can be reused for future work along the same directions. The platform consists primarily of three components:

1. A maze creation tool for creating custom maze layouts
2. A maze rating tool for quality control
3. An online simulator built using Unity that collects data on human participants as they navigate through the maze

Our platform is publicly accessible at `explore.isx.ai`

3.1 Maze Creation

Currently, the dominant paradigm in training AI for navigation tasks is training on large numbers of procedural levels or designs [13]. However, are there systematic differences in human and procedural design that can result in downstream behavioral differences in agents? Do humans encode certain priors into their environment designs that are particularly exploitable by humans in comparison to artificial agents? Do these specific priors lead to better generalization in new environments? To answer these questions, we needed a way to gather a large number of human generated designs, which was the primary motivation behind the creation of our platform.

In order to gather maze designs, we created a tool that allowed for human participants to create custom maze designs through a simple web interface. Users are presented with a set of instructions detailing the basics of the interface as well as some general guidelines for maze design. Users can then create their own designs from a top-down perspective through a simple interface as shown in Figure 3.1a. Here, the dark spaces represent walls, white space

(a) Example maze design

(b) Rating example

Figure 3.1: Screenshots of maze design and rating tools

represents explorable areas, the yellow space represents the player's starting point, and the red space represents the end goal. The tool implements checks to ensure that designs are valid, e.g. a path exists from start to finish.

3.2 Maze Rating

While creating an online maze design tool made it easy to enlist large number of participants to help design mazes, it also presented a problem of quality control. We wanted to make sure that maze designs were sufficiently interesting and not trivial, e.g. consisting of just a short straight line. To address this problem, we implemented a design ratings tool into the platform as well. Users are presented with a set of instructions and are then presented with a set of mazes to rate on a scale out of 5 stars, as shown in 3.1b. To ensure that raters are rating in a somewhat reasonable manner, we include two of our own designs in each rating set. One is designed to be trivially simple, while the other is of reasonable complexity. If raters did not rate these control mazes above/below certain thresholds, we did not consider their ratings. When aggregated over multiple raters, these final ratings can then be used to filter out trivial or uninteresting designs.

3.3 Maze Simulator

The final component of the platform consists of a maze simulator built on top of the Unity platform. Users can play through different maze layouts on a web browser using their keyboard to move around the maze. During gameplay, the platform collects data such as user position and velocity that can be analyzed later. Users can be asked to complete a customizable set of mazes, allowing for detailed and varied experimental design options.

Chapter 4

Comparison of Human and AI Behavior

4.1 Experimental Setup

Maze generation

Beyond comparing human and agent behavior, we were also interested in investigating differences in behavior in specific types of environments. Specifically, we were interested in seeing how humans/agents performed in procedurally/human-generated maze designs. Would humans find human mazes easier than procedural mazes? Would we see the same trends for artificial agents, or would they be reversed? To answer these questions, we collected 200 human maze designs using the platform presented above, sourcing participants from Amazon Mechanical Turk. We initially collected more than 200 designs before cutting the final number down by having Mechanical Turk workers rate the different maze designs and taking the top 200 highest rated ones. We also generated 200 maze designs in a procedural fashion using an algorithm based on a randomized version of Prim's algorithm that incrementally adds neighboring cells to the maze until a specific set threshold is reached. Since these maze produces a tree structure, we additionally introduce loops that are randomly added into the maze to allow for more complex designs. We enforce a total grid size of 15 x 15 for all maze layouts to ensure that no designs are much larger or more complex than the others. Code for the algorithm can be found at `https://gist.github.com/DavidMChan/3e839eb9caf8bde5f903dba50045da88`. Example maze designs can be found in Figure 4.1

Maze environments

For agent training and evaluation, we utilize the DeepMind Lab [1] environment, while for human experiments we utilize the Unity-based online simulator presented earlier. We keep things as consistent as possible between the two environments by utilizing the same action

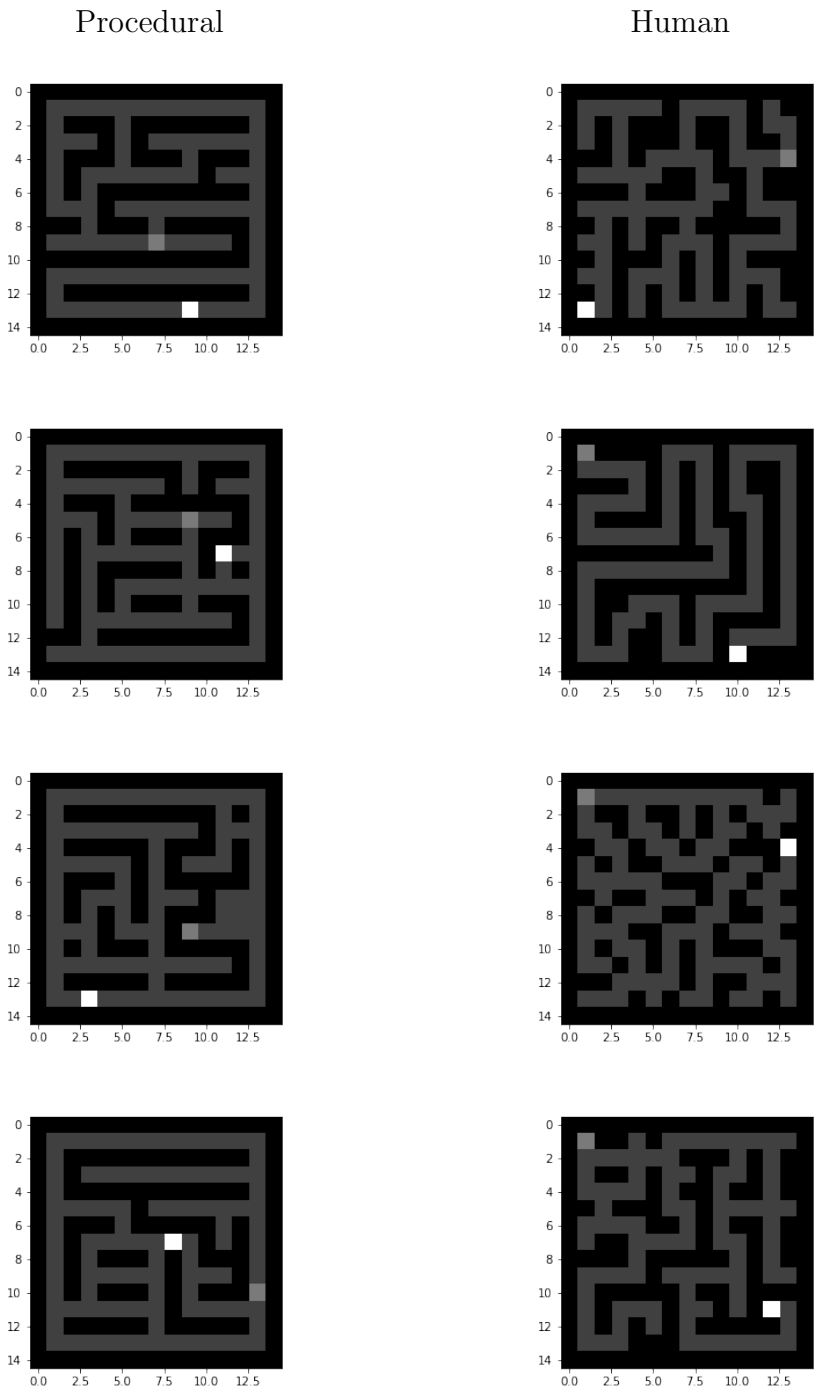


Figure 4.1: Examples of both procedurally-generated and human-generated mazes. The light gray squares represent traversable area, while the highlighted squares are the start and goal locations.

space (move forward, move backward, look left, and look right), similar textures (both environments feature a single wall and floor color throughout the entirety of the maze), and identical maze designs (the human and procedural designs collected earlier) in both environments. In both environments, there are no intermediate rewards - the only reward signal from the environment comes from the terminal goal.

Human data collection

To collect human trajectory data on the different maze designs, we again sourced workers from Amazon Mechanical Turk. Each participant was asked to complete 8 mazes, with 4 designs randomly drawn from the set of procedural mazes and 4 designs randomly drawn from the set of human-designed mazes. Participants were not paid unless all 8 mazes were attempted. Participants had 5 minutes to move around and attempt to find the goal for each maze. We collected data on player position and velocity as they played through each maze to compare to agent behavior. In total, we collected 810 trajectories from 101 participants.

Agent details

Since our goal was to compare exploratory behaviors between humans and AI, we trained our artificial agents on a training set of maze layouts consisting of a subset of the designs we collected from before, and evaluated them on the remaining layouts which they had previously never seen prior to evaluation-time. This way, agents could learn a general exploration policy that could be reasonably compared to that of humans on layouts in which neither had any prior exposure to.

Architecture

We train agents using the Proximal Policy Optimization (PPO) algorithm [14]. In our experiments, we focus on two architectures that we term *Nav PPO* and *Nav Curiosity*.

Nav PPO serves as a baseline and utilizes the same overarching architecture described in [7], consisting of a convolutional encoder and a two-layer stacked LSTM on top. The reward from the previous timestep is fed into the first recurrent layer, while the velocity and previous action are fed into the second recurrent layer. Drawing from the results presented in [7], we also implement an auxiliary depth prediction loss where depth is predicted from the top LSTM layer. The problem is formulated as a classification loss where the depth at each position in a 4x16 depth map is discretised into 8 different bands. The motivation for such a loss is to help build up feature representations that encode more useful information about the 3D space, leading to faster and more effective learning. The overall optimization problem that is being solved during learning is then

$$\min_{\theta_P} \left[\mathbb{E}_{\pi(x_t;\theta_P)}[\sum_t r_t] + \beta L_D \right]$$

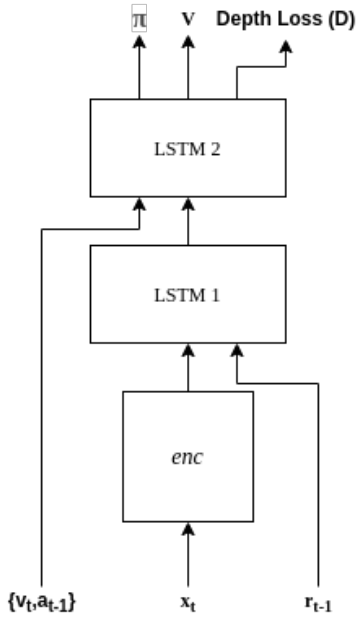


Figure 4.2: Overview of network architecture for both Nav PPO and Nav Curiosity, adapted from [7]

where L_D is the auxiliary depth prediction loss and $0 \leq \beta \leq 1$ is a scalar that weights the importance of the depth prediction loss. We refer to this configuration as *Nav PPO* moving forward.

Nav Curiosity utilizes the same underlying architecture as Nav PPO, but with the added components of the aforementioned Intrinsic Curiosity Module added on top. Note that for our experiments, features were not shared between the policy and the inverse/forward models. In this formulation, the overall optimization problem then becomes

$$\min_{\theta_P,\theta_I,\theta_F} \Big[\mathbb{E}_{\pi(x_t;\theta_P)}[\sum_t r_t] + \beta_D L_D + \beta_I L_I + \beta_F L_F\Big]$$

where θ_I, θ_F are the parameters of the inverse and forward models, L_I, L_F are the losses of the inverse and forward models, and $\beta_I, \beta_F > 0$ are scalars that weight each loss. The values for all the training hyperparameters selected for our experiments can be found in Table 4.1

The convolutional embedding networks in both the policy network and ICM were identical and based on standard architectures used in Atari experiments [8]. The exact specifications can be found at `https://github.com/openai/large-scale-curiosity` as we build our code on top of this open-source implementation. The output of these embedding networks had size 512 in both cases. We used LSTM layers of size 256 and 512 for the first and second layers respectively. The depth prediction module consisted of 64 single-layer MLP's that were each trained to predict the depth category of a single pixel in the depth map. Since the environments contain rewards that can be very sparse, we utilize rollouts of 900 steps and

	Nav PPO	Nav Curiosity
Learning rate	.0001	.0001
PPO entropy coefficient	.01	.01
Extrinsic reward coefficient	1	1
Intrinsic reward coefficient	.00001	.00001
Depth loss coefficient β_D	3.33	3.33
ICM forward loss coefficient β_F	-	1
ICM inverse loss coefficient β_I	-	1

Table 4.1: Training hyperparameters

perform 8 optimization epochs per rollout to more quickly latch onto these sparse rewards. We also run 32 environments in parallel in order to improve training stability and ensure that our policy did not quickly overfit to a few specific layouts encountered during training.

Training

Agents were trained in DeepMind Lab, using a customized environment that randomly loaded a maze layout during each episode. For both human and procedural mazes, we sample from a set of 100 mazes in total, drawing from the same set of mazes that we collected earlier. By training agents on multiple maze layouts, we hope to learn a generalized exploration policy rather than simply memorizing specific layouts. The agent receives 84x84 RGBD images from the environment, however only RGB images $\mathbf{x}_t \in \mathbb{R}^{84x84x3}$ are fed into the policy while the depth pixels are used purely to supervise the depth-prediction loss. The agent received agent-relative lateral and rotational velocity $\mathbf{v}_t \in \mathbb{R}^6$ from the environment as well. The agent received a reward of +10 for reaching the terminal goal, and 0 otherwise. By default DeepMind Lab episodes run for a certain number of timesteps in which agents can reach the goal multiple times during the episode, however to better mirror the setup that humans were presented with we modified this so that reaching the goal terminates the episode. Episodes were capped at 2700 timesteps, which corresponds to 3 minutes of real-world time. We trained both the Nav PPO and Nav Curiosity algorithms on procedurally-generated mazes and human-generated mazes separately, resulting in four different agent configurations that we consider for analysis. For each configuration, we averaged results across eight seeds that were able to achieve $> 90\%$ success rate on the training maze environment when trained to convergence. The final training performance of each agent setup is presented in Table **??**

Evaluation

We evaluated each of the four different agent configurations on the remaining human and procedural maze layouts (100 each). These maze designs were not seen by any agents during training, and thus the data collected here represents the zero-shot exploratory behavior of

Setup	Final Training Reward (out of 10)
Nav PPO Procedural	9.60 ±.13
Nav Curiosity Procedural	9.85 ±.08
Nav PPO Human	9.60 ±.17
Nav Curiosity Human	9.56 ±.11

Table 4.2: Final training performance of the different setups, averaged across all seeds. We see that during training, all algorithms achieve similarly good performance in both types of environments, indicating that a good general exploration policy that does not overfit to specific maze layouts is being learned.

the agents. Each agent was run for 20 evaluation trajectories of a maximum length of 1000 steps (roughly one minute of real-world time) on each of the 100 layouts. Similarly to the humans, we recorded data on position and velocity that allowed us to analyze specific paths and behaviors of the agents in each maze design.

4.2 Results

In total, we compare results from 810 trajectories for humans and $16,000$ trajectories for each agent configuration. We report results from humans and from agents across a couple of metrics that we define and justify below.

Metrics

1. **Success Rate**: Percentage of trajectories in which the subject reaches the goal. While not a perfect measure of exploration as subjects can explore many parts of the maze without finding the goal, this metric does give some indication of how well subjects are exploring the mazes overall.

2. **Steps taken**: We discretize each trajectory into different cells and measure the number of unique cells visited during the trajectory. This provides a measure of pure movement, as a higher step count indicates more raw area being covered.

3. **Percent explored**: Using the same discretized trajectory from before, we take the steps taken metric and divide it by the total number of cells in the maze on a per-maze basis. This is intended to normalize results across mazes which have more or less available area compared to others, giving a measure of area being explored relative to the total possible area in the maze.

4. **Percent revisited**: Using the same discretized trajectory from before, we measure the number of cells that were re-visited during the trajectory and divide it by the

total number of cells visited. A low steps taken number and high percent re-explored would indicate inefficient exploration behavior for example, as this would mean that the subject continuously stays in the same areas without exploring new parts of the maze.

5. **Normalized steps taken**: The metrics introduced above are sensitive to intrinsic qualities of the maze such as how many cells are in the maze or how far the goal is away from the start. To measure how efficiently subjects are exploring the maze in a way that normalizes for differences in complexity or difficulty, we divide the steps taken metric by the optimal path lengths in each maze. This metric then compares maze coverage in comparison to an optimal policy that knows the shortest path beforehand.

Analysis

Procedural vs human mazes

Our results across both humans and agents show that there are indeed specific qualities being encoded into the human maze designs we collected that make them markedly distinct from the procedural ones we generated. One of the key differences comes in the form of distance from the start to the goal. Figure 4.6 shows that the human maze designs have much longer optimal path distances on average, while also exhibiting a wider variance in path lengths. In contrast, the longest procedural mazes only end up being slightly longer than the average human maze length. Our other results provide further evidence to support this conclusion. Agents that were trained in human mazes, for example, achieved significantly better performance in new human layouts when compared to agents trained in procedural mazes. In contrast, the discrepancy in performance in new procedural layouts was much smaller, suggesting that the human designs had more distinctive characteristics that agents were able to learn to specifically exploit during training when compared to the procedural designs.

Despite this discrepancy in maze length, our results suggest that both humans and agents find the procedural designs more difficult to explore in an efficient manner. For all agent configurations and humans, the normalized steps taken in the procedural mazes is significantly higher than in the human mazes, suggesting more deviation from the optimal path.

Interestingly, despite the apparently increased complexity of the procedural mazes, the training performance in Table 4.2 for both Nav PPO and Nav Curiosity is roughly equal in both procedural and human maze environments, suggesting that both algorithms are able to learn to explore and solve complicated layouts to a proficient degree by the time they converge.

Results in Figure 4.5 show that despite the increased maze complexity, humans were actually more successful in finding the goals in procedural mazes. This could largely be explained with the path length discrepancy discussed earlier, as humans also took over less

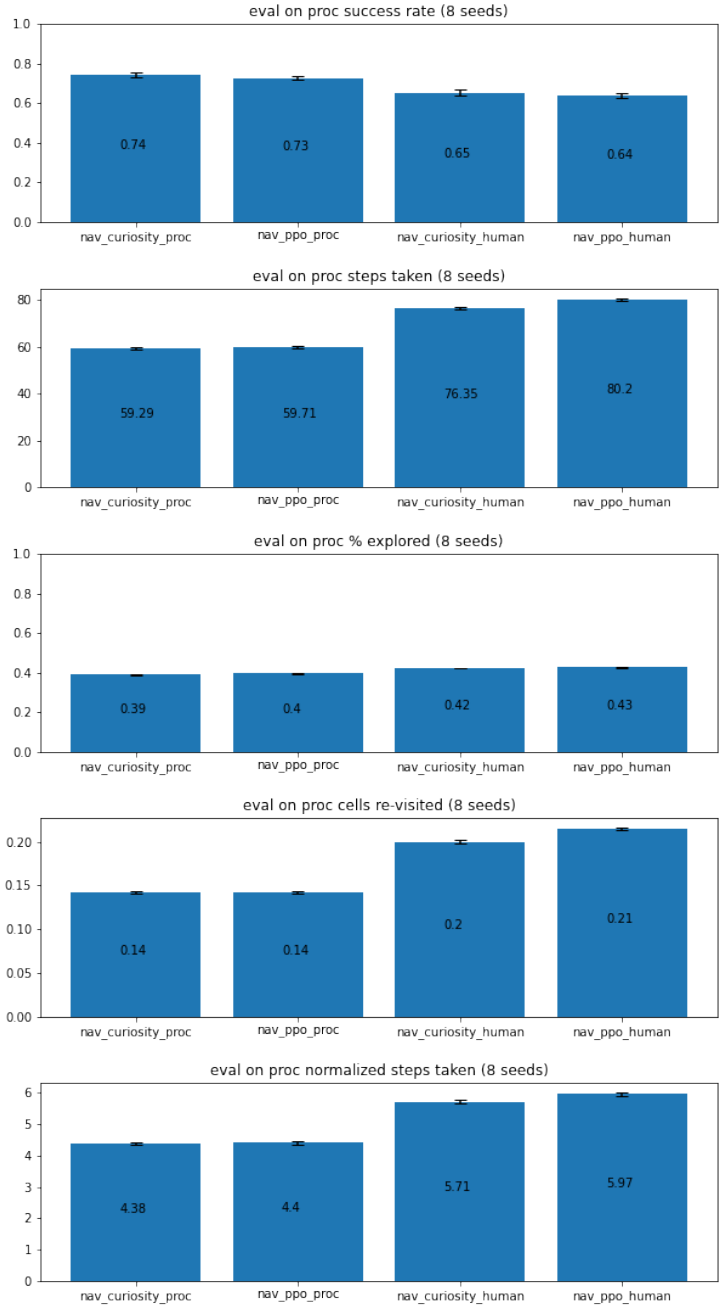


Figure 4.3: Results of agent evaluations on procedurally generated mazes reported across the metrics we defined previously. The labels on the x-axis refer to the architecture used and the set of mazes that were used during training.

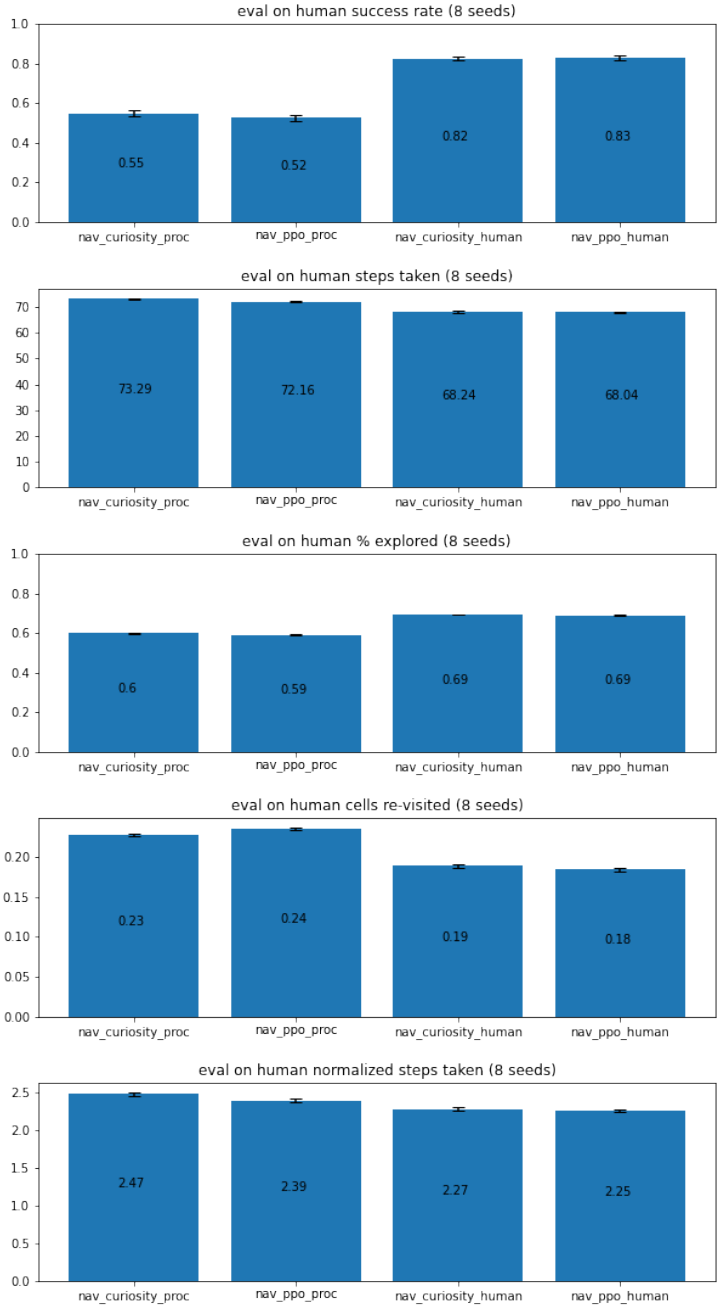


Figure 4.4: Results of agent evaluations on human generated mazes across the metrics we defined previously. The labels on the x-axis refer to the architecture used and the set of mazes that were used during training.

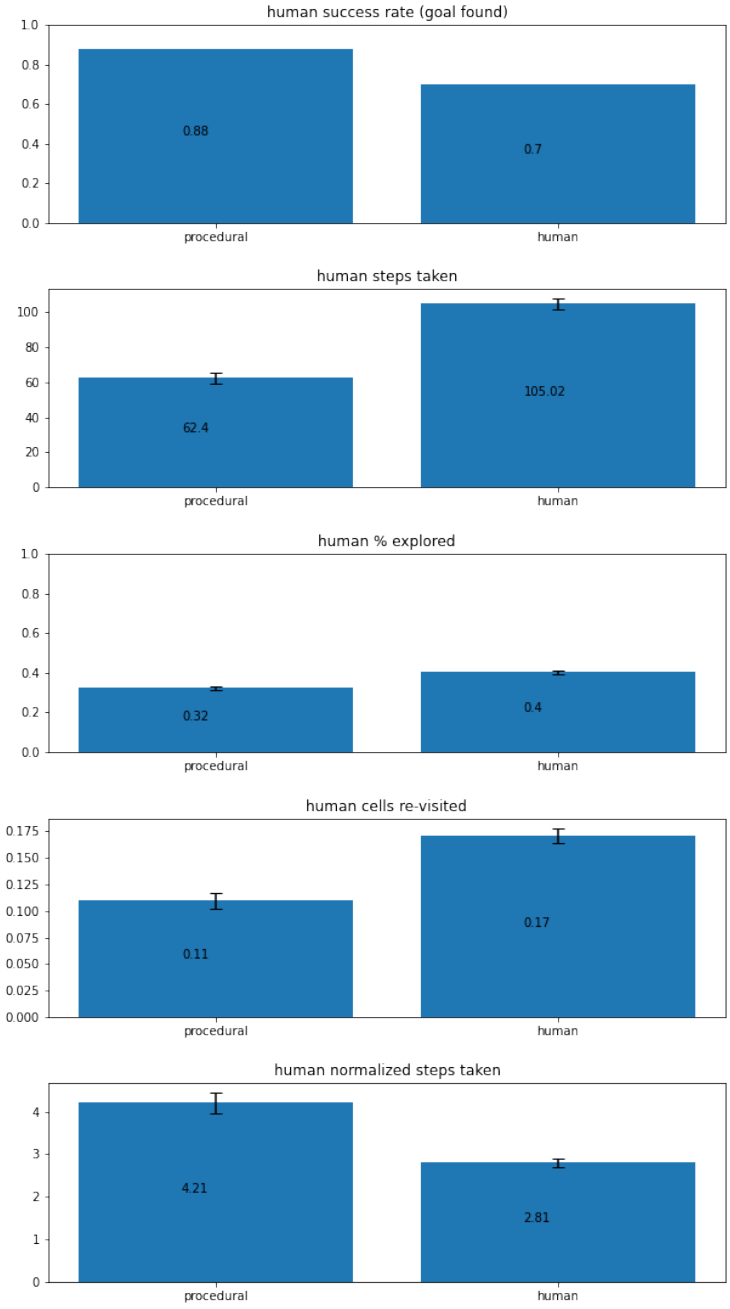


Figure 4.5: Results of human evaluations on both procedural and human mazes. The labels on the x-axis refer to the set of mazes that were evaluated.

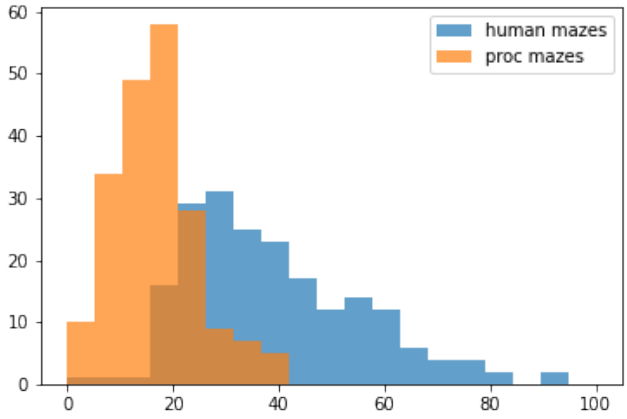


Figure 4.6: Comparison of optimal path length distributions in procedural and human mazes. One potential explanation for this discrepancy is that in the maze generation algorithm, start and goal locations are randomly sampled. However, humans have preconceived notions of where starts and goals should be located (i.e at the end of hallways and at opposite corners) which leads to longer maze path lengths overall.

than half as many steps in the procedural mazes, showing that the simple fact that the goals were closer to the start was enough to offset this structural complexity.

Figure 4.7 shows some interesting qualitative differences in the downstream behaviors that arise from being trained in the different types of mazes. For instance, the agents trained in procedural mazes seem to be more inclined to branch off and visit offshooting paths, while agents trained in human mazes seem much more content to continue on the main path that they were originally on. Such a behavior could be a result the agent learning to more efficiently explore the complex grid-like pattern that is often found in the procedural mazes, compared to the more long-winding paths that are typical of the human mazes, as evidence in Figure 4.1. The other maze layout shows agents trained in human mazes to be much more adept at following long paths compared to the agents trained in procedural mazes, which again is likely an artifact of the discrepancy in maze designs.

Agent generalization across maze types

Our results show that across all metrics, agents explore more effectively and efficiently when exposed to new maze layouts that are generated in the same way as the mazes that the agent was trained in, reaching the goal more often and revisiting less of the maze. Since we have established that the procedural mazes are significantly different from the human mazes, this is not a surprising result, as we would expect agents to have more difficulty adapting across distributions. When looking at the differences in agent performance when generalizing across

different maze types, however, some interesting observations arise.

When looking at how agents behave different when exposed to a different type of maze structure, agents exhibit a few consistent behaviors. For one, the raw step count goes up, which holds true even for agents that go from the longer human mazes to the shorter procedural ones. In addition, they begin to revisit more cells in the maze, both when compared to their behavior in their native maze structure as well as when compared to the behavior of the other agent configurations in their native maze structure. This suggests that certain systematic differences in the new maze structures are confusing the agent and causing it get stuck in specific areas more often.

Overall, our results show some evidence that agents trained in human mazes are able to generalize to the other distribution better. If we consider the performance of agents that were trained and evaluated in the same maze type as the benchmark, then training on human mazes consistently comes closer to benchmark performance in the procedural mazes when compared to the reverse direction. Additionally, agents trained in human mazes explore comparable amounts of the maze compared to the procedural maze benchmark, while agents trained in procedural mazes end up exploring much less of the maze when compared to the human maze benchmark.

Effects of curiosity

Our results show the ICM module having small but mostly insignificant impacts on performance for agents trained in either setting. Metrics such as success rate remain very similar across all settings, and while significant differences in steps taken can be observed when looking at agent generalization to different maze types, the trend is not consistent. While agents that were trained in human mazes seem to be taking less steps in procedural mazes when trained with curiosity, the direction is reversed when looking at agents that were trained in procedural mazes being evaluated in human mazes. As such, it is difficult to ascertain the systematic effects that curiosity is having on generalization performance.

One explanation for such a result could be that during training, as agents get better and better at solving the training maze the effect of the curiosity reward diminishes. This is because the agent learns to prioritize the primary external reward provided by the environment. As such, the ultimate policy that the agent learns ends up being dictated more so by the design of the environment that the agent is trained in, rather than the specific algorithm used. This could suggest that variations in environment design could be equally important, if not more important, than algorithmic modifications when it comes to attaining better generalization performance for RL agents.

Human agent comparison

Overall, agents and humans exhibited surprisingly similar behavior in a couple of key aspects. In procedural mazes especially, when looking at metrics besides success rate, humans performed very closely in line with the benchmark agent. Interestingly, humans were still

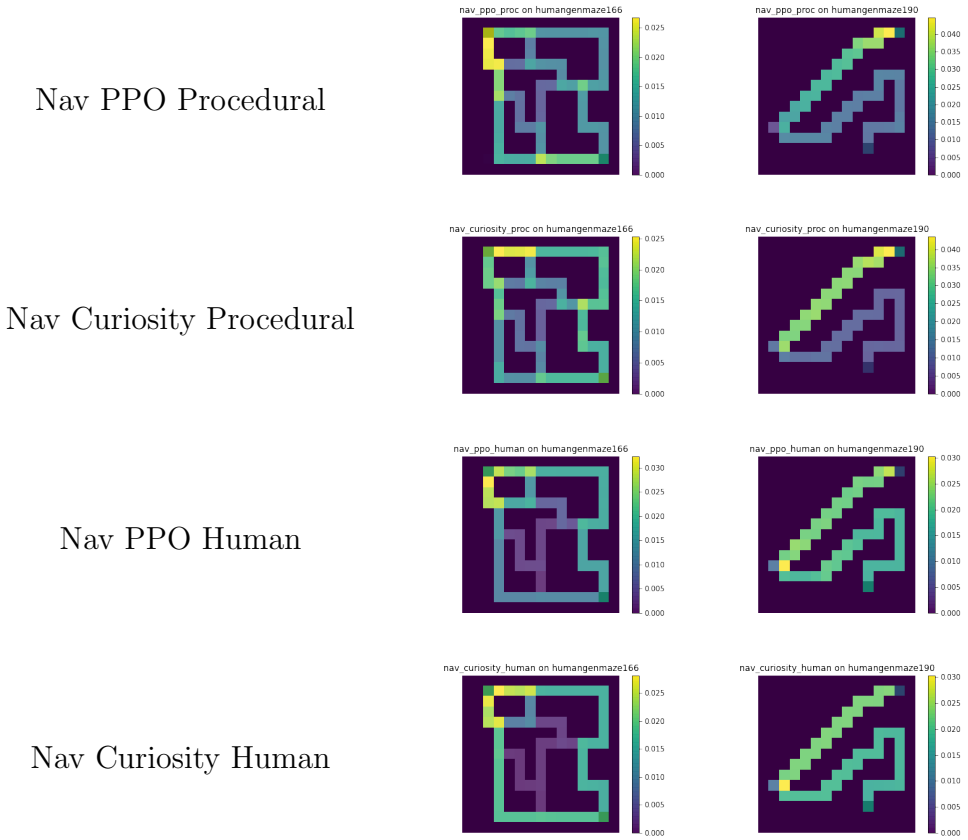


Figure 4.7: We investigate qualitative differences in behavior of different subjects using heatmaps of all trajectories in a couple of layouts. In the maze layout on the left, while agents trained on human layouts tend to stick to the outer portion of the maze, agents trained on procedural layouts explore the inner portions of the maze more consistently. In the maze layout on the right, we see that while agents trained on procedural mazes have difficulty following the path all the way down, agents trained on human mazes are able to do so more consistently.

able to find the goal much more consistently, despite the close similarity in the rest of the metrics.

Somewhat surprisingly, human behavior ended up being more similar to agents that were trained in procedural mazes rather than agents that were trained in human mazes. Both humans and the benchmark procedural agents demonstrated similar behavior not only in procedural mazes, but also in human mazes, Both were less successful, took more steps, started re-visiting more cells, and took a similar amount of normalized steps in the human mazes when compared to the procedural ones. Again, however, humans were much more successful at finding the goal in human mazes than the benchmark procedural agents.

The above results indicate that while certain priors are being encoded into the human designs, these are not priors that humans themselves are particularly good at leveraging. In fact, agents that were trained in human mazes ended up being more successful in human mazes than humans across all metrics, as they were able to find the goal more often while exploring more of the maze layout and taking less normalized steps, suggesting that the agents were able to more effectively exploit the specific traits of the human layouts better than humans were able to.

Chapter 5

Conclusion and Future Work

In this work, we presented two contributions in the space of human-AI comparisons. The first was an online platform for collecting human maze designs and exploration data that can be used for future experiments in this line of research. The second was a comprehensive comparison of the exploration behavior of humans and artificial agents in 3D navigation environments. We specifically explored the effects of the intrinsic curiosity module [11] on agent behavior in agent comparisons to determine if it accurately models human behavior. We also explored differences in behavior of both agents and humans across procedurally generated and human generated maze designs. Our results showed that while curiosity ends up having little effect on agent behavior, training agents in different types of maze environments ends up having drastic effects on exploration behavior in unseen environments from the same maze distribution as well as unseen environments from a different distribution. Our results also showed that while certain agent configurations behaved extremely similarly to humans across many of the metrics we examined, humans were more successful in finding goals within the mazes as well and more robust to differences in maze distribution.

One interesting line for future work could be the development of metrics that better describe some of the discrepancies we observed between human and agent behavior. The fact that humans were more successful at finding goals despite their similarity to agents across many of our metrics suggests that there are certain modes of behavior not being fully captured by the metrics we analyzed. The development of metrics to capture this behavior could lend further insight into what makes human exploration specifically more effective, which in turn could better inform future algorithms. Another line of future work lies in modifications to the algorithmic setups that we present. While the results we presented did not show ICM making a significant difference in agent behavior, other formulations of curiosity [13] or spatial models [4] could yield different results. Finally, future developments to the data collection platform, such as the ability to collect video logs of human trajectories, could aid in the development of new metrics and ultimately allow for more nuanced means of behavioral comparison.

Bibliography

[1] Charles Beattie et al. *DeepMind Lab*. 2016. arXiv: 1612.03801 [cs.AI].

[2] Yuri Burda et al. "Large-Scale Study of Curiosity-Driven Learning". In: *ICLR*. 2019.

[3] Rachit Dubey et al. "Investigating Human Priors for Playing Video Games". In: *ICML*. 2018. URL: http://proceedings.mlr.press/v80/dubey18a.html.

[4] Saurabh Gupta et al. "Cognitive Mapping and Planning for Visual Navigation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2017, pp. 7272–7281. DOI: 10.1109/CVPR.2017.769. URL: https://doi.org/10.1109/CVPR.2017.769.

[5] Arthur Juliani et al. *Unity: A General Platform for Intelligent Agents*. 2020. arXiv: 1809.02627 [cs.LG].

[6] Eliza Kosoy et al. "Exploring Exploration: Comparing Children with Agents in Unified Exploration Environments". In: *Proceedings of the 42th Annual Meeting of the Cognitive Science Society , CogSci*. Ed. by Stephanie Denison et al. cognitivesciencesociety.org, 2020. URL: https://cogsci.mindmodeling.org/2020/papers/0386/index.html.

[7] Piotr Mirowski et al. "Learning to Navigate in Complex Environments". In: *ICLR*. 2017.

[8] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[9] Ian Osband et al. "Deep Exploration via Bootstrapped DQN". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[10] Georg Ostrovski et al. "Count-Based Exploration with Neural Density Models". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. 2017, pp. 2721–2730.

[11] Deepak Pathak et al. "Curiosity-driven exploration by self-supervised prediction". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2778–2787.

[12] Richard M. Ryan and Edward L. Deci. "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions". In: *Contemporary Educational Psychology* 25.1 (2000), pp. 54–67. ISSN: 0361-476X. DOI: https://doi.org/10.1006/ceps.1999.1020. URL: https://www.sciencedirect.com/science/article/pii/S0361476X99910202.

[13] Nikolay Savinov et al. "Episodic Curiosity through Reachability". In: *7th International Conference on Learning Representations, ICLR 2019*. 2019.

[14] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347 (2017).

[15] Laura Schulz. "The origins of inquiry: Inductive inference and exploration in early childhood". In: *Trends in cognitive sciences* 16 (June 2012), pp. 382–9. DOI: 10.1016/j.tics.2012.06.004.

[16] Laura Schulz and Elizabeth Bonawitz. "Serious Fun: Preschoolers Engage in More Exploratory Play When Evidence Is Confounded". In: *Developmental psychology* 43 (Aug. 2007), pp. 1045–50. DOI: 10.1037/0012-1649.43.4.1045.

[17] Paul J. Silvia. "Curiosity and motivation". In: *The Oxford Handbook of Motivation* (2012).

[18] Z. Sim and F. Xu. "Learning Higher-Order Generalizations Through Free Play: Evidence From 2- and 3-Year-Old Children". In: *Developmental Psychology* 53 (2017), pp. 642–651.