Frames of Reference for Neural Pathfinding Navigation

Morgan Bryant
Departments of Psychology and Computer Science
Stanford University, Stanford, CA, 94305 USA

Abstract

Two cognitive visuospatial methods for representing object locations are Egocentric reference frames, which place objects relative to a special central entity, and Allocentric reference frames, which place objects on a stationary world. Humans have been shown to selectively use each reference frame, but it is unclear *why* humans use certain frames in certain situations. This work examines Deep-Q Reinforcement Learning Artificial Neural Networks that solve path-finding navigational puzzles using the two reference frames as models of human neural populations attempting the same puzzles. We provide a computational explanation for why humans generally use egocentric frames for navigational challenges, which is centered around cognitive skill transfer learning. Our results support the plausibility of connectionist cognitive planning as well as indicate the advantages that curriculum learning provides for teaching neural networks to plan effectively.

Introduction

This work is motivated by two natural observations: how children use locational reference frames in their cognitive spatial maps and the apparent difficulty neural networks have at developing long-range plans. Spatial mapping and planning, while not intrinsically related, are both necessary fundamental components of *navigation*, a human cognitive skill that requires both understanding of a spatial environment and using this understanding to interact with the environment. We study the hypothesis that neural networks model human cognition and therefore can learn to navigate comparably to humans. Specifically, in this work we look at two implications of this hypothesis: neural networks must be able to learn to plan, and human reference frame observations can have a neural-computational explanation. We make some first steps supporting this hypothesis by demonstrating that neural networks both can be taught to make rudimentary plans and when learning reference frames exhibit explainable learning dynamics that are similar to those observed in humans.

Egocentric and Allocentric Reference Frames

When an intelligent entity interacts with external objects, it must somehow maintain a cognitive spatial map of spatial relationships. Two kinds widely-recognized cognitive visuospatial frameworks that humans use in their cognitive maps are egocentric and allocentric frames of reference (Bremner, 1978; Klatzky, 1998). Both of these reference frames build useful models of relative locations of objects by relating everything to useful, foundational elements chosen from the environment. Egocentrism and allocentrism are systematically defined by Klatzky (1998), but the following definitions should suffice for our purposes. In brief, we say that an agent uses an egocentric framework when it puts the foundational origin location wherever a selected unique *ego* object is located. In this representation, the ego remains at the origin regardless of whether or not the ego object moves. The allocentric framework can have an ambiguous definition, but for our environments with mostly static objects, we say an agent uses an allocentric framework when it puts the foundational origin location on the unmoving ground. In this allocentric frame, every object on the fixed coordinate system moves in the agent's representation *only* when the object actually moves in the world, in an egalitarian sense. That is, if an agent X adopts an egocentric representation, when X moves, then any object Y will appear to move in the opposite direction of X's motion. If however X adopts an allocentric frame, when X moves, X's representation of X's own location moves according to the world coordinate frame,

and any other object Y will remain static in X's representation. Each of these representational frames are effective at putting relational information into a concrete, position-assignable map that can be accessed and used for other purposes.

Infant Frame Acquisition and Usage in Search

Bremner (1978), Piaget and Cook (1954) and others have observed that at nine months of age, infants are capable of searching for hidden objects and learning the locations of the objects, but are unable to realize when the object has moved. In a classic study by Piaget called $A\bar{B}$ or A-not-B, infants observed a researcher hide a toy under hiding spot A repeatedly, and they would learn that if they searched for the object at location A, they could find the toy. However, when the child observed the researcher hide the toy at location B, they would reliably search for the toy at location A. Piaget suggests that this error occurs because infants have not yet formed a coherent spatial framework. He further proposed that infants use an egocentric representation to encode the toy's location, such that they learn to point to an abstract location relative to their own bodies that has been successful (i.e., at A) instead of correctly identifying the toy's location in the world at B and pointing to that spot. Bremner's research has extended this study and confirmed that infants exhibit an egocentric understanding when searching in this and other related challenges, but also, they can also be compelled to use allocentric methods, given enough the right kinds of cues. Bremner raises the question of why infants default to using an incorrect egocentric framework to solve hidden object tasks, not an allocentric one, and by what mechanism this happens. Furthermore, Hardwick et al. (1976) have shown that children by age eight are capable of allocentric activity but also erroneously default to using egocentric models in experiments testing their ability to build and use a cognitive map of a classroom.

In this study, we propose a computational basis for why egocentrism is both more representationally powerful and more plausible as the dominant 'phenotype' among reference frames. Under a connectionist assumption, we demonstrate that neural networks exhibit the same bias towards egocentrism as these infants.

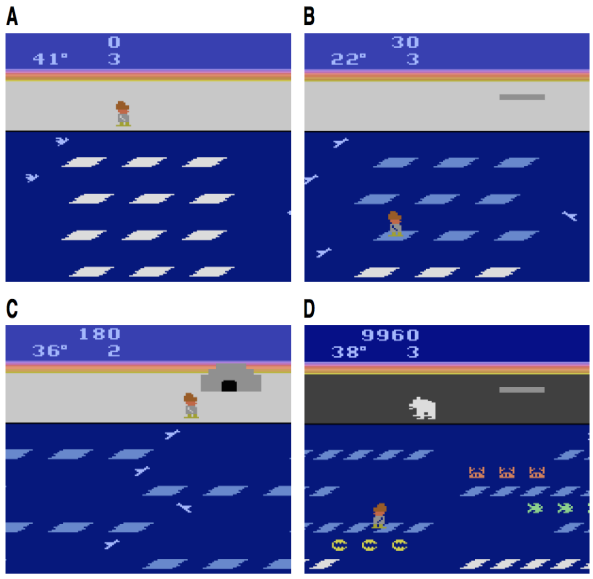
Navigation Requires Planning

In addition to building a cognitive map of the world, humans need to be able to interact with objects in the world. Specifically, as humans move around, they need to be able to guide themselves around objects that may be moving or interacting with each other. This fundamental skill – to plan a path – is a essential capability that cognitive agents must have (Lake et al., 2016). Furthermore, humans can learn to navigate very quickly and very well. The encompassing concept of navigation is, however, not particularly indicated as a skill that neural networks are good at, for which numerous reasons have been cited including inabilities to reason and do search (Lecun, 2015). In order for neural networks to be a compelling model of how human cognition works, it must be demonstrated that they can indeed form sensible plans.

Can Neural Networks Do Planning?

Robust planning and navigational pathfinding has been identified as a problem that is unsolved, generally difficult, and prerequisites any AI-complete system (Weston et al., 2015). Lake et al. (2016) present a particularly strong and concrete criticism against the possibility neural navigation. They look at the DQN, a deep reinforcement learning model presented by Mnih et al. (2015) that is challenged to learn various Atari video games. One of the complex control problems, a game called *Frostbite*, was particularly challenging for DQN. In short, *Frostbite* requires the player to jump treacherously across floating ice tiles to collect enough snow bricks to build an igloo; success requires planning to acquire the bricks. See Figure 1 for more on *Frostbite*. Although the network was able to reach human-level performance on 29 out of the 49 games, it was wholly unsuccessful at succeeding at *Frostbite* "and other games that required temporally extended planning strategies" (Lake et al., 2016). They blame the fact that the network is a "powerful pattern recognizer" in a model-free reinforcement setting, which diverts resources away from planning, and does not incorporate explicit model building, which they claim is a crucial aspect of intelligent systems.

In this work, we took acknowledge this criticism as a present unknown in deep learning work: can neural nets learn to plan? We second the DQN criticisms are especially reasonable because the way the DQN was trained was very unlike the way humans train. Lake et al. highlight that the *Frostbite* DQN was trained



Images of Atari game *Frostbite*, borrowed from Lake et al. (2016), a difficult challenge for neural networks. Lake's caption: "Screenshots of Frostbite, a 1983 video game designed for the Atari game console. A) The start of a level in Frostbite. The agent must construct an igloo by hopping between ice floes and avoiding obstacles such as birds. The floes are in constant motion (either left or right), making multi-step planning essential to success. B) The agent receives pieces of the igloo (top right) by jumping on the active ice floes (white), which then deactivates them (blue). C) At the end of a level, the agent must safely reach the completed igloo. D) Later levels include additional rewards (fish) and deadly obstacles (crabs, clams, and bears)."

Figure 1

from scratch and given reinforcement rewards only on completing the entire task, and when it failed on that, it was then trained to complete milestone sub-goals such as collecting one brick because "without these sub-goals, the DQN would have to take random actions until it accidentally builds an igloo," (Lake et al., 2016). They suggest that this learning strategy is very different from the way people do it. We also agree that it is cognitively unrealistic that humans approach this game as a single monolithic task. We suggest that it is practical to introduce a curriculum of prior sub-goals for teaching planning tasks to eventually solve the whole task (to be discussed in later sections). This aligns with another claim by Lake et al., that prior knowledge is essential for a human player's success. However, they insist that this prior knowledge must be encoded and combined in an explicit model. They claim that connectionist neural architectures are ineffective at making such models and therefore are unsuitable as cognitive models. We disagree that neural networks cannot make plans because of their inability to develop explicit models, and our preliminary work suggests that explicit model-building is not absolutely necessary for a neural network to exhibit planning. That is, we suggest that learn networks can learn behavior that resembles model-learning without being directed to do so.

PathFinder

PsychoPath (`www.kongregate.com/games/k2xl/psychopath`) is an online video game available for free through a web browser. The main objective is to try to navigate through a 2D maze to reach goal on a collection of designed mazes. The world also contains movable blocks, which the player may push by walking into. See Figure 2 for example challenges.

PathFinder is an adaptation we have developed that is inspired by the mechanics of *PsychoPath*. In PathFinder, as in *PsychoPath*, the objective is to navigate the virtual player starting from a given initial location through a mutable maze to the goal in as few steps as possible. The player can only be moved one space at a time in a cardinal direction (in this paper, referred to as UP, DOWN, RIGHT, and LEFT). The player cannot step on indicated wall squares or outside of the world, but they can push an unanchored block if they step from behind it and that the space that the movable block would slide onto is empty. The game is a full-information, single-player, non-zero-sum game without any temporal component; attempted invalid moves do not alter the game state in any way. For consistency's sake, we call unique initializations *challenges*, from which a variety of *game states* may be reachable based on actions taken, and a collection of challenges is a *task*. PathFinder was chosen because it provides an effective collection of problems that

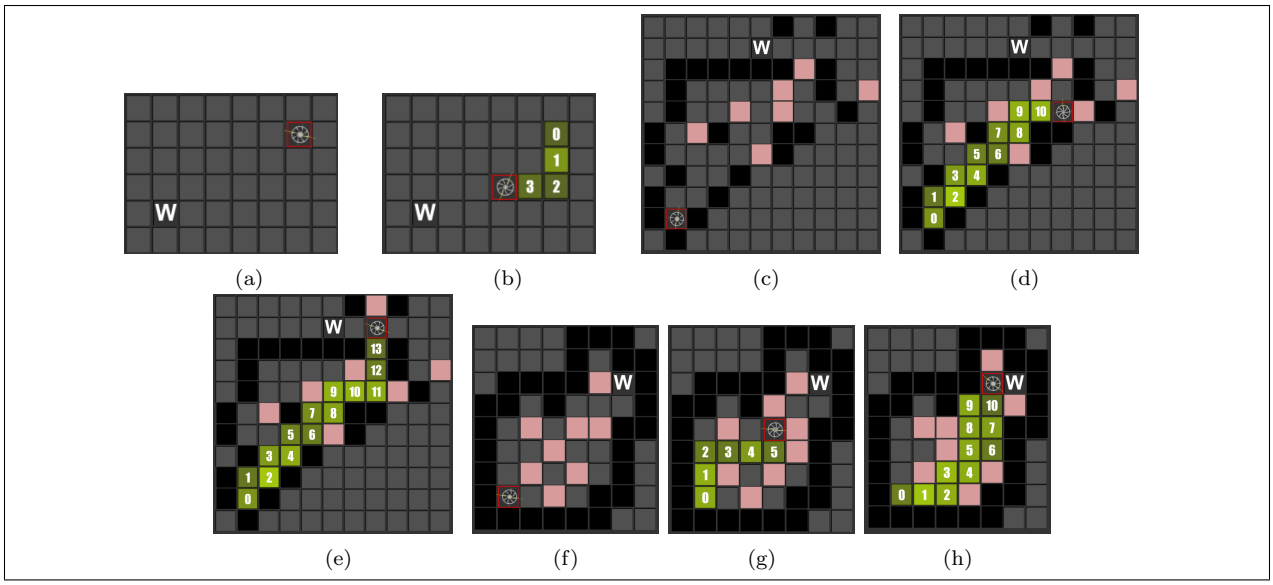


Figure 2: Some *PsychoPath* game states. The player is located at the circular icon, W represents the goal, black squares are immobile wall blocks, pink (or light gray if image is in black and white) squares are blocks that the player can push, dark gray squares are vacant tiles, and the numbered squares trace the player's moves. In some challenges, it may be difficult to determine what path to attempt to clear.
(a) Easy challenge: initial state.
(b) Easy challenge: state after two moves down and two moves left on keyboard.
(c) Movable block challenge: initial state.
(d) Movable block challenge: after several moves. On the latest move RIGHT, a block was pushed RIGHT.
(e) Movable block challenge: the player's moves and block pushes have cleared a path to the goal.
(f) Difficult challenge initial state. It is perhaps unclear what paths are best or valid.
(g) Difficult challenge: oops - in this round, the player's choices have blocked any possible path to the goal!
(g) Difficult game: RIGHT, RIGHT were better initial choices.

can test the differences between allocentrism and egocentrism. Additionally, this game can test a neural network's ability to exhibit short-range planning. The relative difficulty of simple challenges is intuitive, and literature recommends mazes as tests for visuospatial navigation experiments (Grön et al., 2000).

Frames of Reference and one-away versus two-away challenges

When tasks are simple, such as when the agent and the goal are initialized directly next to each other, then learning to reach the goal amounts to identifying the single direction to move in, and the allocentric and egocentric frames are equally informative and equally capable. For example, the task *all-1-away* (see Figure 4) causes no representational difference between the two possible frames being used: each initial challenge is monolithic and functionally unique, requiring simply the single correct action per initial state. However, when the tasks become more complex, the distinct frames of reference take on different representational behavior. In task *all-2-away* (see Figure 5), the initialized state is placed within two steps of the centered agent. In this scenario, when the agent takes its first action, the same initial state is representationally invariant between egocentric and allocentric frames. After the first action, if the action was in the right direction, then in the resulting state the agent will be beside the target state. However, at this point, the two frames have different appearances, even between identical initial states and initial actions. If the frame is allocentric, then the view of the agent and goal is in an unprecedented, unique arrangement, and this newfound state must be learned from scratch. If the frame is egocentric, the representation has adjusted so that the goal appears to have moved *towards* the agent. This resulting egocentric view now resembles exactly one of the states from the *all-1-away* task, which prior skill can hypothetically inform how to choose the current action correctly. Figure 3 visualizes the functional differences between allocentric and egocentric frames in PathFinder.

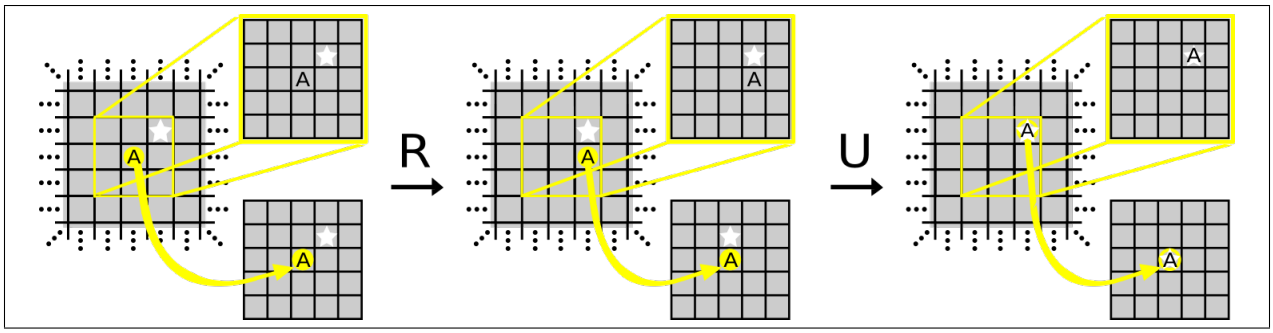


Figure 3: Different views or frames of reference on the same states and same actions RIGHT and UP. The upper images are allocentric views, and the lower images are egocentric views. Yellow guides indicate what the frame of reference keeps constant between moves and surfaces as the representation of the underlying game state.

The *R-U-RU* task

To compare the effects of egocentric and allocentric representations, we tested the ability for neural networks to learn a simple proof-of-concept PathFinder task, which we call the *R-U-RU* task (Figure 4). The *R-U-RU* task consists of three challenges, each in a grid world five tiles wide and five tiles tall. For each, the agent is initialized in the center. In challenge *R*, the goal state is immediately to the right of the agent. In challenge *U*, the goal is one tile above the agent. In challenge *RU*, the goal is up and to the right of the agent. This challenge is designed to be both stripped to the barest parts while still resembling the $A\bar{B}$ challenge given to the infants, who remain seated while the target varies location.

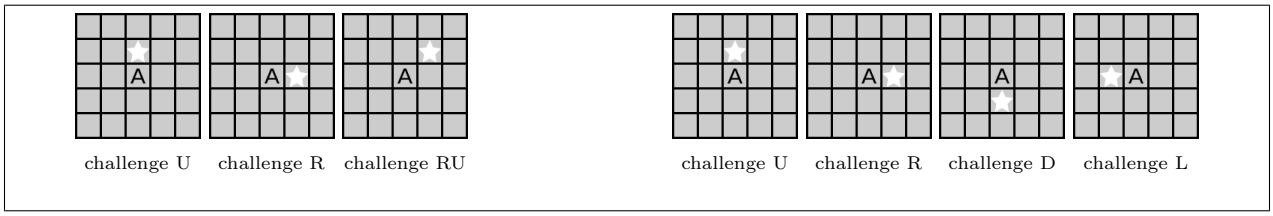


Figure 4: Two tasks. Left: the *R-U-RU* challenge represents the simplest case that demonstrates an ego-allo difference. Right: the *all-1-away* task with the Agent centered has all situations in which the goal is within one step.

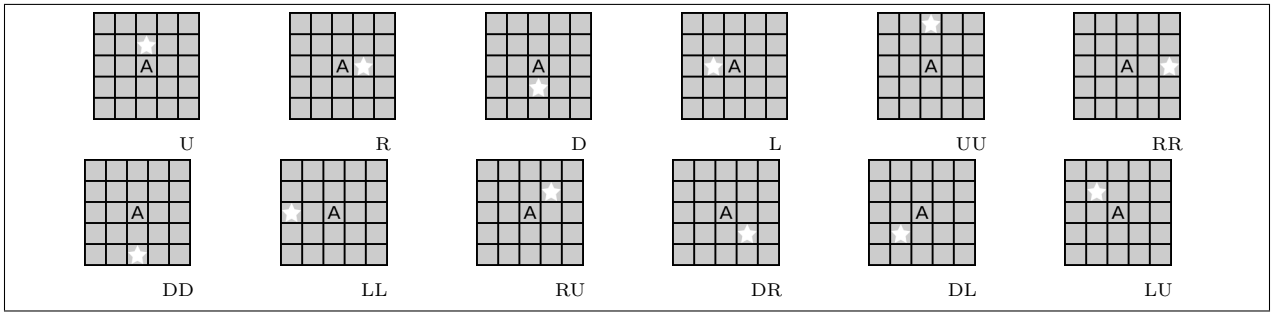


Figure 5: Task *all-2-away*: with the Agent centered, this task has all situations in which the goal is within two steps.

Experiments and Results

Data representation and environment

We encoded our world states as four binary 5x5 matrices for the agent, the goal, immobile block locations, and mobile block locations, where a '1' indicated that such an object was present at the location. In the *R-U-RU* task, all states were reachable in two steps, so the network was given two opportunities to move

and attempt to reach the goal. A cardinal action would result in a deterministic game state change in the relationships between objects and the agent, but the different frames of reference have their own ways to present the new state to the agent. These states were generated programmatically.

Reinforcement learning setting, hyperparameters, and network topology

For interacting with the environment, we used a Deep Q-Network (DQN) reinforcement learning paradigm. Over the course of training, the reinforcement agent maintained and learns 'Q-values' unique to state-action pairs, among which higher-valued actions in a state indicated preferred actions for that state. In standard so-called Q-learning, a table of values for each state-action pair is maintained. However, this state-action value storage method is impractical when the number of states and actions grows or when the agent must generalize to unseen data, since the table method does not degrade gracefully. Instead, deep Q-learning attempts to alleviate these issues by approximating Q-values using a neural network to estimate values as output given an image as input. For more information on Q-learning and deep Q-learning, please see Gosavi (2009); Mnih et al. (2015, 2013). In our setting, the reinforcement agent received a game state represented as four binary matrices and was tasked with determining the best action to take in order to receive a delayed reward; the action selected is the one whose estimated state-action Q-value was greatest in that state. The reinforcement agent would acquire its Q-value estimates by feeding the represented state to a neural network, which would output values for each possible action given the state.

The reinforcement agents and trained networks were randomly seeded for all comparative experiments. The agent was permitted up to two actions, in which it attempted to achieve the goal reward (of value 1), and all other actions would result in no reward. The action that was taken was always the one with the highest value as determined by the network. In *R-U-RU*, there are no invalid actions to be handled. For all the possible actions, the error signal was given only to the action that was actually taken. The agent was trained on all three challenges per training epoch, in a single batch of size 32, which sampled the three states with replacement. For each challenge, there was a random chance of taking an action chosen uniformly randomly; the probability of this chance was 0.8 in the presented trials. The agent was tested intermittently between training trials on each challenge in the task, once each without sampling; these results are reported. The Bellman equation gamma discount was set to 0.9, and network updates were applied immediately without a buffer.

The neural networks presented here were feedforward networks. They took as input the binary matrix of size 5x5x4, which was passed through a single fully connected hidden layer with 64 units and finally fed to output nodes representing Q-value estimates for the distinct actions available to the agent; in *R-U-RU*, the output had four nodes representing actions moving the agent in four cardinal directions. After each layer, the linear output was passed through RELU activations; the first hidden layer applied dropout with probability 0.5 after its RELU, and the output was passed through a tanh squashing function. The network was trained using the Adam optimizer (epsilon 1e-6) variant of gradient descent with learning rate 1e-4. Losses for the gradient were calculated using a huber loss (with maximum gradient 3e-5) applied to the difference between the selected action's corresponding output Q-value and the reinforcement's gamma times the output Q-value plus the reward. Network weights were initialized to random small nonnegative values for all the layers except the last, which took small mean-zero values, each scaled proportional to their layer sizes. Twenty seeded sample networks were trained for 2000 epochs each under both the egocentric and allocentric frameworks on *R-U-RU*.

We chose to use fully connected networks as opposed to convolutional networks due primarily to the simplicity of the challenges. On harder challenges, we expect that convolutional networks and deeper networks would be more appropriate.

Curriculum

During preliminary explorations, we observed that networks that could not reliably learn to solve two-step-away challenges. Specifically, when a network was given a task with any two-step-away challenges, it was usually unsuccessful; networks given only one-step-away challenges were almost always successful. If, however, we simply let the networks learn the one-step-away tasks before presenting any two-step-away tasks, they were reliably successful *if the network was using an egocentric frame*. Networks using the allocentric frame were still generally unreliably successful.

We give thorough explanation for this phenomenon in a later section, but the simple explanation is that egocentric frames can learn to reuse skills that they have previously learned *if they have already learned it.* If however the network is tasked with learning both one-step and two-step challenges concurrently, it was less effective at learning both learn a prior skill and use the prior skill at the same time. If the frame is allocentric, there is no opportunity to transfer skills from simpler challenges to more difficult challenges in *R-U-RU.*

The relevant point here is that we introduced curricular aspects into our experiments because, in part, they had a significant impact on the success of learning.

Statistical Results

Figure 6 and Figure 7 present analyses of the effects of curricula and frames, respectively, for our experiments on the *R-U-RU* task.

Our experiments tested the effects of various curricula on PathFinder success (see Figure 6). We have plotted the training results of 20 equally-seeded samples on the effectiveness to learn the *R-U-RU* task within 2000 epochs on three curricula and two reference frames. Figure 7 shows the results from our tests for the effectiveness of the egocentric and allocentric frames of reference. Using a binomial test over 20 equal-seeded networks, we demonstrate that egocentric frames are at least as good as allocentric ones at solving the *R-U-RU* task. When currcula are used, then egocentric frames are clearly better at solving the task ($p < 0.001$). When curricula are not used, however, it is much less clear which of the two frameworks is more successful. This supports the hypothesis that egocentric frameworks indeed make use of prior knowledge of simple challenges when solving difficult challenges, but only when the prior skill is actually fully learned. If not, then it is sensible that either framework would treat the three challenges as arbitrary challenges to concurrently learn and would perform comparably. In the cases that used a curriculum, the allocentric frame more reliably learned the two easy tasks and more reliably did not learn the full task within 2000 epoch compared to the egocentric frame, which less reliably learned all three staring tasks.

Analysis of results

Networks trained to first solve R and U and only later be tasked with the RU exhibited qualitatively better and faster convergence. Specifically, we observed that in egocentric frameworks, exposing the network to only simple challenges for some time before introducing tougher ones significantly improved convergence, and exposing the tough challenges gradually was slightly more effective than suddenly. These results support the claim that networks that learn simpler tasks first can learn to reuse them effectively in more difficult tasks, but only if they fully learn the simpler skills prior. In the allocentric cases, we observe an opposite phenomenon: due to the inability to reuse its skills, the network learns best when it is exposed to all the challenges equally soon. However, while the allocentric model learns better on average with no curriculum than with one, the network is also less reliable and is the only place among our tests we saw networks that could not reliably learn the two simple tasks. It is also important to note that the allocentric success rate was consistently rising near the end of the trials with curricula. Future work should run tests for longer and verify end learning. This aligns with current research findings that allocentric knowledge is slowly incorporated into hippocampal storage (Holdstock et al., 2000).

State recognition analysis: why egocentrism performs better than allocentrism

We give the following explanation for the observation that egocentrism performs better than allocentrism on the *R-U-RU* task. In egocentric case, if the agent was given time to learn easy challenges R and U, then when it was presented with RU, it simply had to learn to place itself beside the block, which it knew how to solve. This is because once it was there, the egocentric representation allowed it to view the new scenario in terms of a solved scenario. So, when presented with a two-step-away state, the network only needed to learn a single move. The allocentric model did not have this capability: the new scenario had no shared representation to take advantage of. The extraordinary improvement the network received indicates that such a curriculum was useful and that such a skill reuse was done. See Figure 8 for a diagram of skill reuse.

This skill transfer is an example of a rudimentary plan that the network learned to do: simply, the network was guided to learn to first to solve a simple task and then to learn to put itself into that situation again. For future work, extensions of this can test a neural network's ability to choose to navigate to paths it knows hows to solve, much like a human driver choosing to navigate to a main road which is familiar instead of traveling along a potentially shorter but less reusable path.

State space analysis: why egocentrism converges faster than allocentrism

Besides the egocentric frame provides a more effective representation than allocentrics frame, even in the test without a curriculum, egocentric-framed networks learn faster than allocentric ones (see Figure 6). Another reason for this phenomenon has to do with the number of states each network must learn to handle. Egocentric representations contains fewer states than allocentric ones due to the reusability of representations for multiple challenges. See Figure 9 as an example. Due to this, egocentric networks have fewer states they have to learn and so converge more quickly.

This reasoning would also indicate a possible reason that human infants use egocentric frames more readily but children and adults slowly lose this bias. If networks are indeed good models for human spatial mapping, then egocentric representations would be faster to learn in humans as well, despite the presence of allocentric frames even in an infant's toolbox. The fact that networks and infants indeed share characteristics around frame choice indicates networks can be a decent model of human spatial mapping; indeed, some initial research is indicating as much (Ekstrom et al., 2014; Samsonovich and McNaughton, 1997).

Discussion, Predictions, Future Work

We have demonstrated that networks hold as suitable models of infant spatial mapping observations. The network and the infant both exhibit a preference for egocentric representations early in development, which we suggest is because egocentric frames are more rapidly acquired and more useful at simple compositional levels of abstraction. Our analyses provide a reasonable explanation for why all humans, even infants, have the both frames at their disposal, but it takes many years into childhood before a human is able to consistently avoid defaulting to egocentric frames erroneously. We suggest that infants have both frameworks that are learned with the same amount of information after birth; since egocentric frames can take advantage of simple compositional patterns faster better than allocentric ones, we claim there is little biological motivation to do otherwise. In infants, the egocentric frameworks are first frames to develop into a sufficiently coherent map, which we conclude based on our connectionist hypothesis and the results in this work. Then, when egocentric and allocentric frames differ, a competition between frames is indicated. In a competition between the two models, infants tend to default to egocentric models because they are generally more developed. Competitive constructs have been suggested by Burgess (2006) and others. We leave it to future work to build an artificial network that has both frames at its disposal and gets to choose which frame to use in a given situation.

We have demonstrated that neural networks can exhibit rudimentary planning, supporting the greater feasibility of neural planning. In particular, we showed that appropriate frames help networks plan in a model-free fashion: the networks did not need models to transfer the use of prior skill knowledge. Notably, a reasonable curriculum proved to be useful and possibly critical for neural planning. Future work can examine more rigorous curricula on harder challenges.

Other extensions of this work includes an examination of actions. In this work, the agent could slide in the four cardinal directions, which is unrealistic, since humans are *directed* creatures; for example, when a human needs to go backwards, humans tend to rotate around and walk forward instead of walk backwards.

Open questions. It currently still remains to what extend neural networks can plan and learn to plan. It is unclear how curricula help and for which problems it is unhelpful, useful, or crucial. Model-free planning was indicated by this work, but it remains open how much neural networks can make models implicit.

Much thanks to Andrew Lampinen and Jay McClelland for providing insight and guidance for this project.

References

Bremner, J. G. (1978). Egocentric versus allocentric spatial coding in nine-month-old infants: Factors influencing the choice of code. *Developmental Psychology*, 14(4):346.

Burgess, N. (2006). Spatial memory: how egocentric and allocentric combine. *Trends in Cognitive Sciences*, 10(12):551 – 557.

Ekstrom, A. D., Arnold, A. E., and Iaria, G. (2014). A critical review of the allocentric spatial representation and its neural underpinnings: toward a network-based perspective. *Frontiers in human neuroscience*, 8:803.

Gosavi, A. (2009). Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, 21(2):178–192.

Grön, G., Wunderlich, A. P., Spitzer, M., Tomczak, R., and Riepe, M. W. (2000). Brain activation during human navigation: gender-different neural networks as substrate of performance. *Nature neuroscience*, 3(4):404–408.

Hardwick, D. A., McIntyre, C. W., and Pick Jr, H. L. (1976). The content and manipulation of cognitive maps in children and adults. *Monographs of the society for research in child development*, pages 1–55.

Holdstock, J., Mayes, A., Cezayirli, E., Isaac, C., Aggleton, J., and Roberts, N. (2000). A comparison of egocentric and allocentric spatial memory in a patient with selective hippocampal damage. *Neuropsychologia*, 38(4):410 – 425.

Klatzky, R. L. (1998). Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. In *Spatial cognition*, pages 1–17. Springer.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building machines that learn and think like people. *arXiv preprint arXiv:1604.00289*.

Lecun, Y. (2015). What's wrong with deep learning? The Conference on Computer Vision and Pattern Recognition.

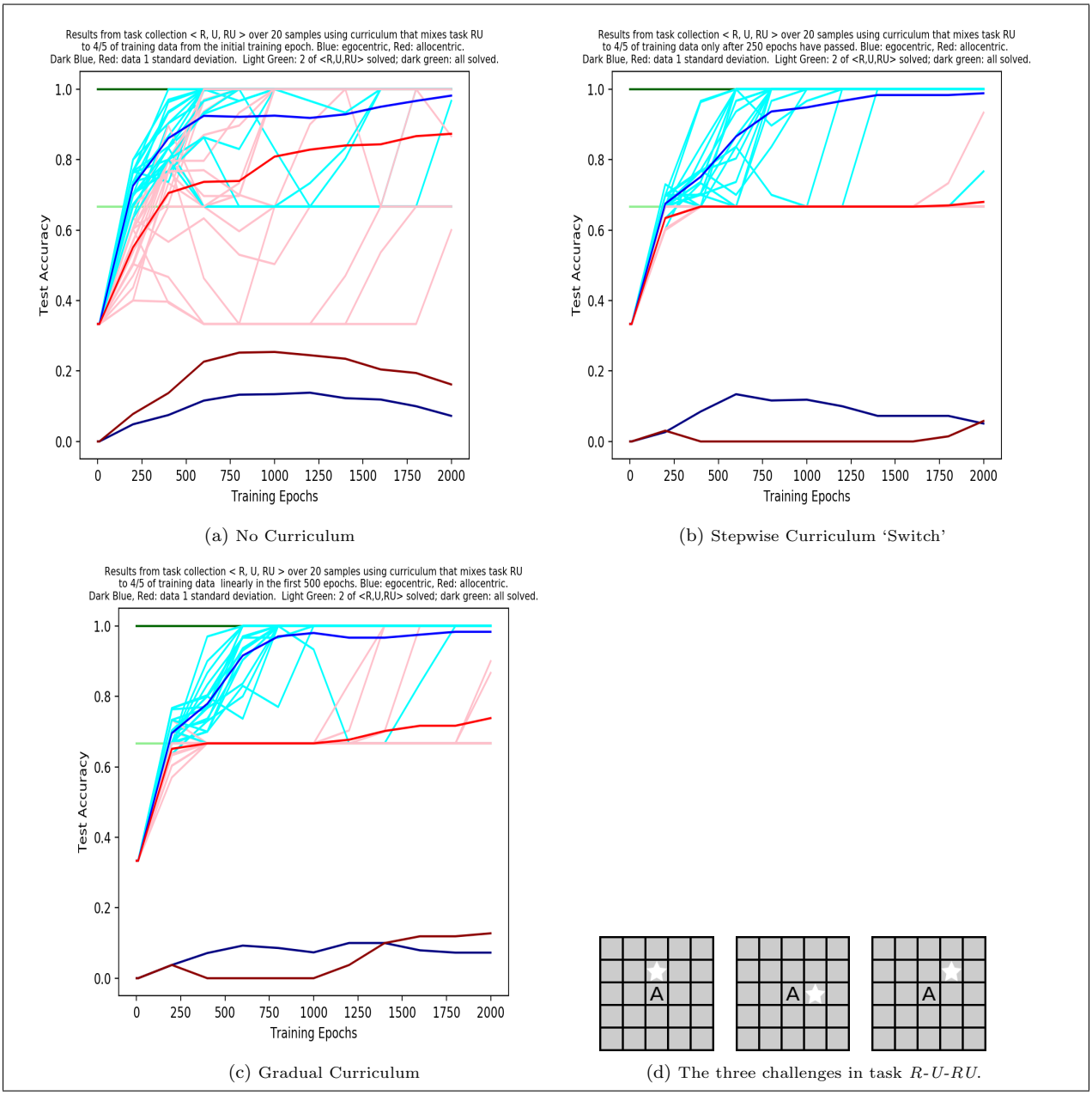
Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Piaget, J. and Cook, M. T. (1954). The construction of reality in the child.

Samsonovich, A. and McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, 17(15):5900–5920.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.



(a) No Curriculum

(b) Stepwise Curriculum 'Switch'

(c) Gradual Curriculum

(d) The three challenges in task *R-U-RU*.

Figure 6: Comparison of test accuracy per epoch with varied curricula. (a): during training, the agent was presented challenge RU 80% of the time and either U or R 20% of the time, on all epochs. (b): during the first 250 epochs, the agent was presented only with U or R challenges. After epoch 250, the network received 20% R or U and 80% RU. (c): in the first epoch, the network was only given R and U challenges, but over the course of the first 500 epochs, the network was gradually given more and more RU challenges until after epoch 500, when the network was given 20% R or U and 80% RU. These networks were tested every 200 epochs.

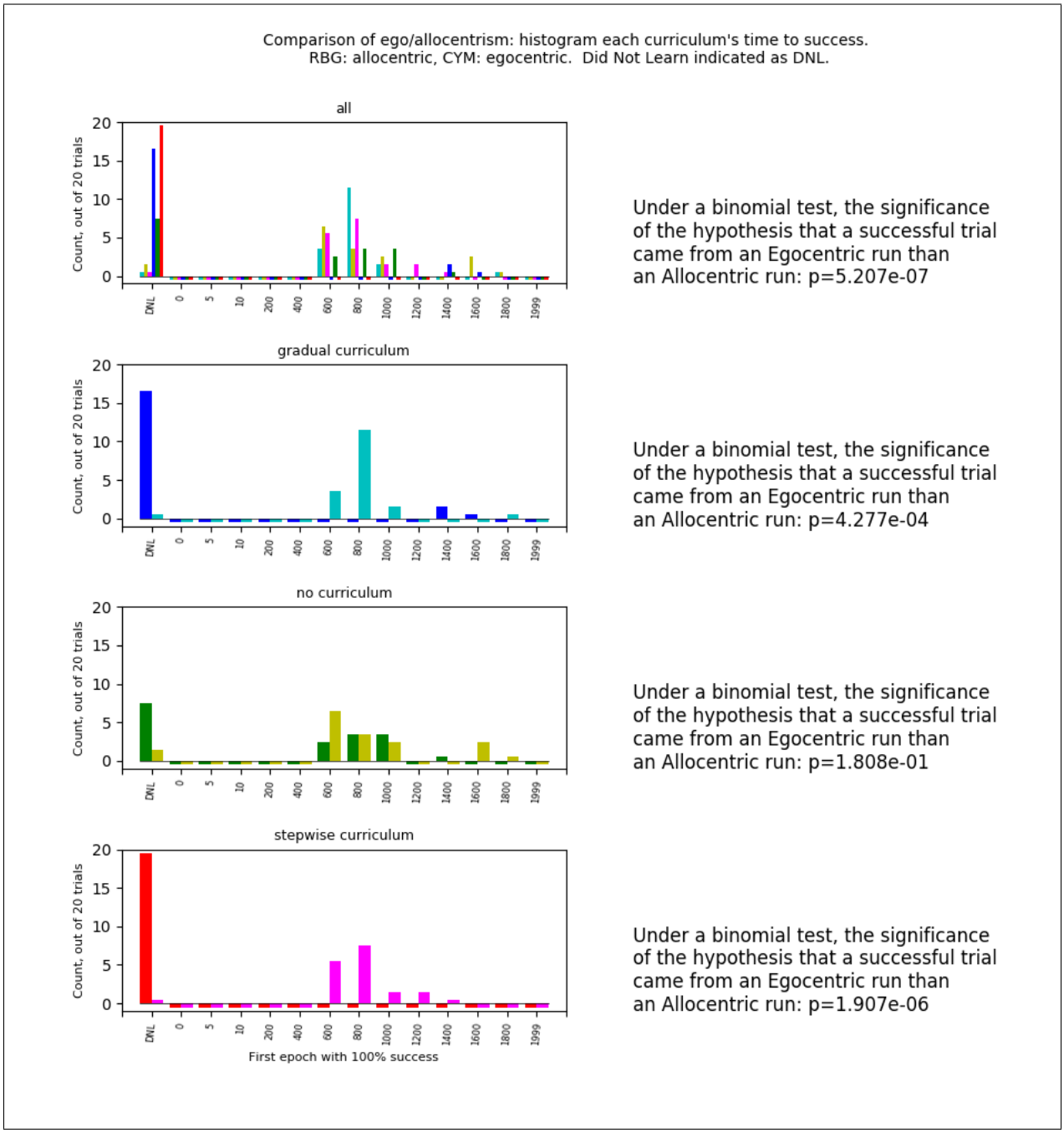


Figure 7: Comparison of the effectiveness of egocentrism vs allocentrism subject to different curricula.

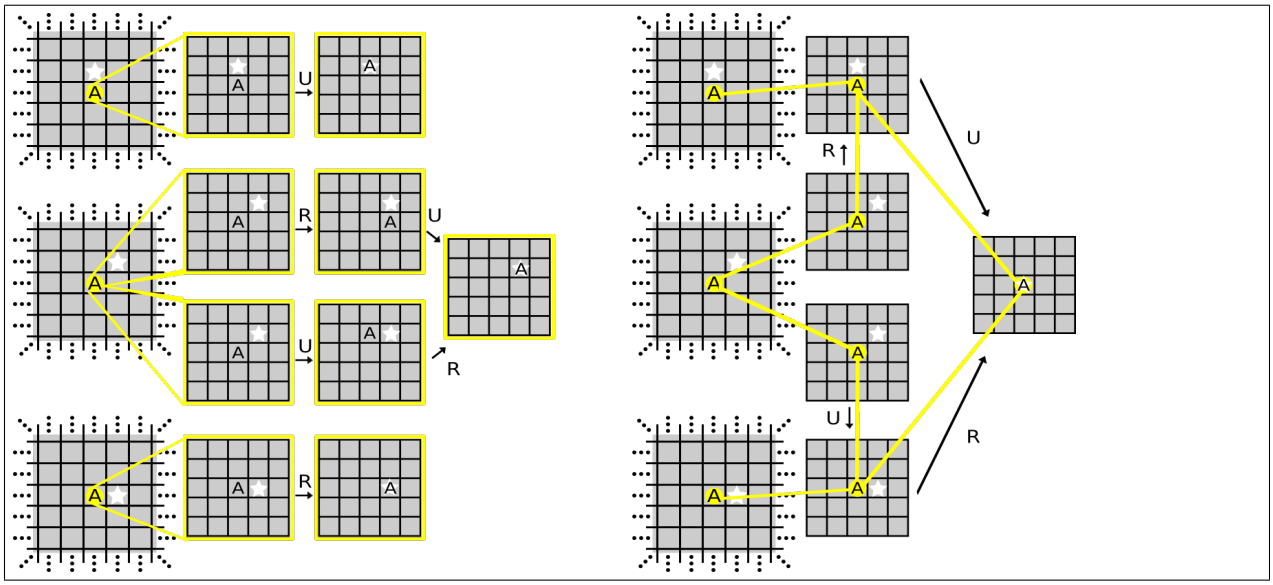


Figure 8: The successful paths for each starting state in *R-U-RU* in allocentric (left) and egocentric (right) frames. Note in the egocentric frame, the harder task decomposes into a simpler task.

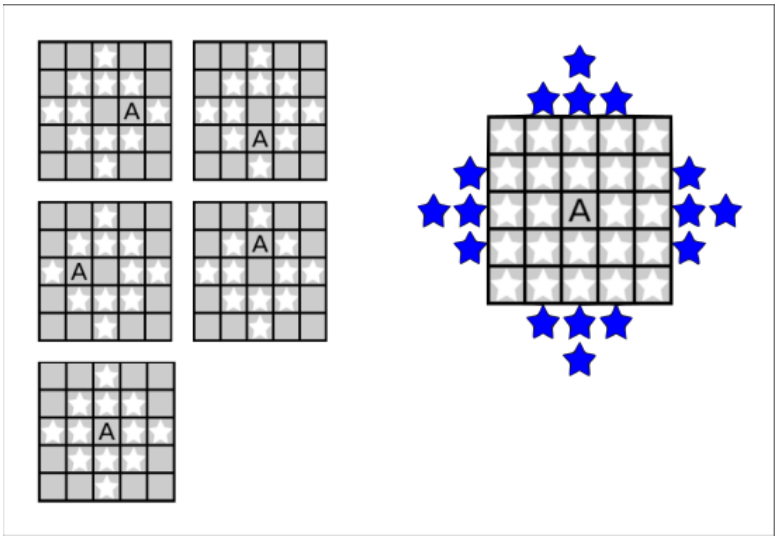


Figure 9: All the possible states in 2-away tasks. Stars cover all the locations that a goal might be placed in an allocentric (left) or egocentric (right) representation with the agent centered. In this scenario, there are 56 possible states, all of which the allocentric represents as distinct. On the contrary, the egocentric representation views only 40 possible views.