deepGOLSA: Goal-directed planning with subgoal reduction models human brain activity

Huzi Cheng[1] and Joshua W. Brown[1]

[1]Department of Psychological and Brain Sciences, Indiana University Bloomington

March 18, 2024

Abstract

Goal-directed planning presents a challenge for classical Reinforcement Learning (RL) algorithms due to the vastness of combinatorial state and goal spaces. Humans and animals adapt to complex environments especially with diverse, non-stationary objectives, often employing intermediate goals for long-horizon tasks. Here we propose a novel method for effectively deriving subgoals from arbitrary and distant original goals, called the deep Goal Oriented Learning and Selection of Action, or deepGOLSA model. Using a loop-removal technique, the method distills high-quality subgoals from a replay buffer, all without the need of prior environmental knowledge. This generalizable and scalable solution applies across different domains. Simulations show that the model can be integrated into existing RL frameworks like Deep Q Networks and Soft Actor-Critic models. DeepGOLSA accelerates performance in both discrete and continuous tasks, such as grid world navigation and robotic arm manipulation, relative to existing RL models. Moreover, the subgoal reduction mechanism, even without iterative training, outperforms its integrated deep RL counterparts when solving a navigation task.

The goal reduction mechanism also models human problem-solving. Comparing the model's performance and activation with human behavior and fMRI data in a treasure hunting task, we found matching representational patterns between specific deepGOLSA model components and corresponding human brain areas, particularly the vmPFC and basal ganglia. The results suggest a new computational framework for examining goal-directed behaviors in humans.

1 Introduction

Humans and animals must develop capabilities to pursue time-varying goals in continuously changing environments. The pressure for survival prohibits slow, linear adaptation to different goals, i.e., learning value functions from scratch for each new objective. A quick and versatile paradigm is necessary for such goal-directed learning scenarios. However, traditional Reinforcement Learning (RL) algorithms are not specifically designed for this, encountering challenges in the goal-directed learning context. They

are highly optimized for scenarios with relatively fixed goals, e.g., winning in Go, or reducing building energy consumption [49]. In these situations, iterative methods like the Bellman equation are effective for approximating value/advantage/negative cost functions across various states, maintaining stability over time. Nonetheless, if the goal changes during training, classical RL becomes highly inefficient due to: 1) The significant increase in state space caused by the introduction of the goal set; 2) The inability to reuse experiences across different goals. To elaborate, if we denote the state space in a task as $s \in \mathcal{S}$, and introduce a set of goals $g \in \mathcal{G}$, the new state space becomes the Cartesian product $\mathcal{S} \times \mathcal{G}$. Since $\mathcal{G} = \mathcal{S}$ in many cases, e.g., in navigation where both state and goal are locations, the expanded state space becomes enormous. Humans and animals leverage past knowledge from the environment to solve tasks [50], even when the current goal differs. They apply environmental understanding to plan for new goals, using subgoals derived from previous experiences. This knowledge transfer and adaptation process is not inherent in Bellman equation-based RL algorithms. For instance, in goal-directed Q learning, the value function, $Q(s)$, evolves into $Q(s, g)$, without assuming shared structure when g changes [32].

On the other hand, traditional heuristic algorithms like Dijkstra's shortest path algorithm [12] and the A* pathfinding algorithm [20] excel in these problems. These algorithms leverage goal-independent knowledge to construct real-time maps during navigation. They use selective intermediate goal states to simplify the problem, thus reducing pathfinding time. By breaking down a distant goal into nearer ones, these algorithms utilize local knowledge, independent of goal changes, to solve the problem. This universality suggests a potential model for human problem-solving. Recently, a cognitive neural model, GOLSA (Goal-Oriented Learning and Selection of Action) by [17], demonstrated its potential in mimicking human behavior in goal-directed settings [54]. Like the A* algorithm, GOLSA employs a goal reduction mechanism to identify locally solvable intermediate goals leading to the ultimate objective. Importantly, GOLSA's biologically plausible construction suggests its feasibility in human brains, with fMRI results mapping its mechanisms to various brain regions [54].

Despite its promise, GOLSA's scalability remains a concern. Since GOLSA retrieves subgoals by virtually traversing all possible trajectories in parallel, this process necessitates a completely orthogonal representation space to perform a Markovian equilibrium-like calculation. This requirement naturally fits the one-hot neural representation and Hebbian network architecture to model the environmental knowledge. But one-hot representations scale linearly with problem dimensionality, limiting their applicability to high-dimensional real-world tasks, such as visual stream inputs. It also contravenes the distributed nature of the brain [21, 23, 4].

Modern deep RL algorithms, however, excel at managing complex inputs using deep neural networks. Given this and the advantages of goal reduction mechanisms in GOLSA and other heuristics, the critical question becomes: Can we develop a neural model capable of both handling high-dimensional inputs and generating valid goal states in navigation for goal-directed learning? Recent RL community attempts [52] include model-based RL approaches combining forward and backward planning [30], graph construction from sensory inputs for traditional pathfinding [55, 16], and subgoal generation through models to expedite learning [37, 41, 9, 22]. These studies, like GOLSA, underscore subgoals' importance in goal-directed settings. However, some rely on heuristic-based shortest path methods [16], with manually constructed sub-

goal quality, or bootstrap processes, like value functions learned through the Bellman equation, to generate beneficial subgoals [9].

We explore whether a solution can avoid traditional heuristics and with minimal reliance on global value functions. This paper proposes a goal reduction neural mechanism, demonstrating an efficient training method for a neural network to generate high-quality subgoals for goal-directed learning, independent of prior global environment knowledge. This mechanism bypasses iterative updates in standard Bellman equation-based learning, yet remains compatible with existing RL frameworks, enhancing their efficiency. We validate this in discrete spatial navigation and continuous robot arm manipulation tasks. Notably, when combined with a local policy not requiring iterative training, the framework functions akin to the GOLSA model [54]. We demonstrate that such combinations surpass traditional accelerated RL algorithms. Finally, we apply the trained neural model to the same tasks used by [54], showcasing its potential in simulating human goal-oriented behavior and explaining the underlying neural mechanism.

2 Results

In the following experiments, we demonstrate that by harnessing the power of episodic memory and a computational strategy, agents can be trained with exceptional goal reduction capabilities. These agents outperform classical RL algorithms in solving various tasks. Furthermore, using neuroscience tools like Representation Similarity Analysis (RSA), we show that the model's behavior in goal-directed tasks is akin to human's neural activity, by illustrating the representational similarity between the model and the brain.

2.1 Episodic Memories Without Loops Generate Better Subgoals

Consider a simple environment consisting of states $s_i \in \mathcal{S}$. These s_i could be images (e.g., in vision-based navigation), texts (e.g., text-based RPG games), or abstract states (e.g., career path planning). The agent may explore this environment with a goal $g \in \mathcal{G}$, set by either itself or the external world. The interaction between an agent and the environment can be modeled as $(s_t, g, a_t, r_t, s_{t+1})$, where a_t is the action taken by the agent, r_t is the reward received from the environment after executing the action, and s_{t+1} is the state of the environment following the action. In goal-directed learning, this variable g may change substantially, making conventional RL algorithms inadequate due to the enormous size of $\mathcal{S} \times \mathcal{G}$. (As previously mentioned, in many goal-directed learning scenarios, the state and goal occupy the same space, i.e., $\mathcal{G} = \mathcal{S}$. Henceforth, we may use $\mathcal{S} \times \mathcal{S}$ interchangeably to denote the space spanned by the state and the goal, except for the last experiment, where $\mathcal{S} \neq \mathcal{G}$) Additionally, in many scenarios, the reward is sparse, and the agent only receives a positive reward or avoids punishment when it reaches the goal state, i.e., $s_i = g$. This, combined with the complexity of the state space, further complicates goal-directed learning.

We propose that, given a specific g, if an agent can effectively reduce a goal g into a subgoal s_g that is "closer" to its current state s_t, it may alleviate the task's

difficulty. Furthermore this can be iterated to find subgoals that are arbitrarily close to the current state s_t.

The most straightforward solution to this problem is to train a function Φ that generates s_g, which we refer to as a *goal-reducer*:

$$\Phi(s_t, g) : \mathcal{S} \times \mathcal{G} \to \mathcal{G} \tag{1}$$

Training such a *goal-reducer*, however, can be challenging. There are two intuitive strategies. First, one can sample s_g uniformly from $\mathcal{S}$, as done by [9]. This approach is referred to as *Random sampling* (Figure 1 left).

An alternative is to sample s_g from past experiences, a strategy known as *Trajectory sampling*. Assuming a sequence of interactions $(s_t, g, a_t, r_t, s_{t+1})$ for $t = 1, 2, \cdots, T$ is stored in memory, s_g can be sampled from the s_t in this sequence (Figure 1 middle). A common technique in goal-directed RL, Hindsight Experience Replay (HER), proposed by [2], conceptually resembles this approach, as it encourages the agent to learn associations between current states and some of its future states in the same trajectory. Other studies, such as [39], also find trajectory-based sampling effective in goal-directed learning, though expert experience is not required in our case [39]. Furthermore, neuroscience research indicates that episodic memories, i.e., sequences of experienced states, are vital for learning [18].

However, the aforementioned strategy does not guarantee the sampled s_g to be effective: an agent with limited environmental knowledge may simply engage in a random walk within the state transition graph, rendering the states experienced in the episodic memory ineffective for connecting the trajectory's start and end points. To address this, we introduce a third strategy, termed *Loop-removal sampling* (Figure 1 right), which may mitigate this issue. The underlying rationale is that when the agent has minimal knowledge of the environment, the trajectories it creates will likely be random and involve numerous "loops." A "loop" occurs when the agent revisits the same state at least twice within a trajectory. The *Loop-removal sampling* posits that by eliminating these "loops" from the episodic memory, the s_g sampled from the remaining trajectories will be more advantageous, as ineffective experiences are excluded, potentially resulting in a trajectory that more closely resembles a linear path in the best-case scenario.

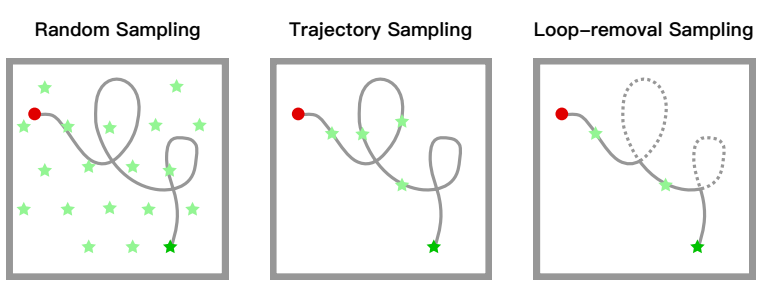


Figure 1: Three types of subgoal sampling strategies.

While *Loop-removal sampling* is effective when $\mathcal{S}$ is discrete, i.e., state representations are either the same or different, it faces challenges in environments where $s_i \in \mathcal{S}$ can be infinitely close, such as state embeddings learned via neural networks or continuous states like coordinates in 3D space navigation. This issue is also encountered in

earlier concepts like [25]. To overcome this, we adopt an idea from persistent homology [14], defining a filtration process to determine the existence of "loops." This process is depicted in Figure 2A: In a continuous space, trajectories are sparse, as distances between states are always greater than zero, making exact state overlaps unlikely. However, by assigning a filtration radius to each state and incrementally increasing it, the connectivity of the episodic memory trajectory changes. At a certain point, the algorithm detects a "loop" (proper case in Figure 2A), where *Loop-removal sampling* is most effective. If the filtration radius continues to increase, the trajectory becomes fully interconnected, leading to *Loop-removal sampling* failure. We demonstrate that with an appropriate filtration parameter, a random walk path in 3D space can be effectively simplified (Figure 2B) with a radius set to 0.8. The optimal radius is determined through a grid search between 0 and the longest distance between any pair of points in the space, with the objective of maximizing the rate at which task performance improves with learning. This search process is also used in later experiments (Figure 4) to maximize learning efficiency in terms of *Optimality* and *Equidex* (described in detail below).

This approach may seem akin to the *Search on the Replay Buffer* technique developed by [15], but there is a key distinction: *Search on the Replay Buffer* depends on a value function to estimate distances among states, whereas *Loop-removal sampling* operates independently of any prior environmental knowledge. Instead, *Loop-removal sampling* relies on a minimal assumption applicable to all state spaces: loops indicate redundancy.

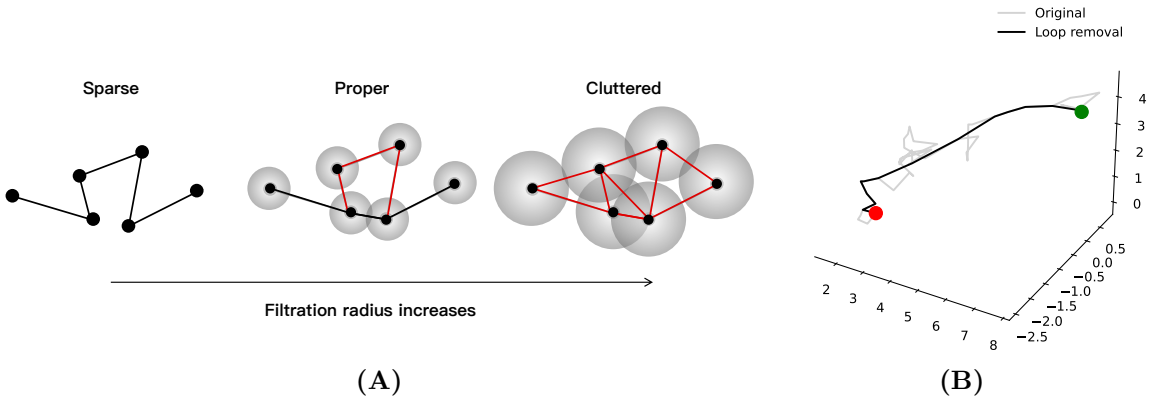


Figure 2: *Loop-removal sampling* in continuous state space. 2A: Illustration of the filtration process in *Loop-removal sampling*. This graph shows the progressive increase in the filtration radius, revealing loops formed within sparse state representations. Red lines indicate the trajectories that *Loop-removal sampling* targets for removal. 2B: Visualization of *Loop-removal sampling* applied to a random walk trajectory in 3D space.

To test our hypothesis that a *goal-reducer* trained with *Loop-removal sampling* generates superior subgoals compared to *Trajectory sampling* and *Random sampling*, we evaluate their performance across several datasets. This necessitates quantifying the quality of subgoals produced by *goal-reducer*. We introduce two metrics for this purpose: *Optimality* and *Equidex*. Firstly, we represent the effort required for an agent to reach goal g from state s as $||s, g||$, which is the distance across legal transitions of the

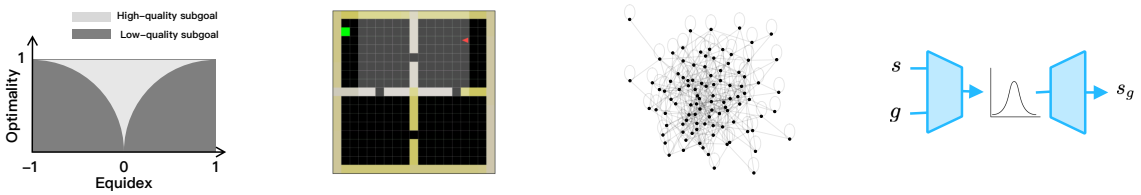
state graph. This concept mirrors the idea of shortest distances, though it may not be symmetrical, i.e., $||s, g|| \neq ||g, s||$, due to potentially irreversible state transitions. Also the state space graph of legal transitions may not be fully connected – because barriers may exist in the state space so that some transitions are not possible, s_g is generally not simply the linear midpoint between s and $g \in \mathcal{S}$. Building on this, *Optimality* is defined as:

$$Optimality(s, g, s_g) = \frac{||s, g||}{||s, s_g|| + ||s_g, g||} \tag{2}$$

When *Optimality* approaches 1, the total effort expended by an agent to reach g via s_g is near the optimal effort. However, a s_g with an *Optimality* close to 1 may not be informative if s_g is set to either g or s, where no new information is provided. To address this, *Equidex* is introduced:

$$Equidex(s, g, s_g) = \frac{||s_g, g|| - ||s, s_g||}{||s, s_g|| + ||s_g, g||} \tag{3}$$

When *Equidex* is close to 0, it indicates that the effort (or distance) from s to subgoal s_g is similar to the effort from s_g to the final goal g. This suggests that s_g is situated on the hyperplane formed by midpoints in the state space between s and g. Accordingly, as *Equidex* nears 1, the subgoal becomes closer to the current state, while a value nearing -1 suggests the subgoal is closer to the ultimate goal. In summary, the quality of a single subgoal can be characterized by both *Optimality* and *Equidex*. The ideal subgoal is one where *Optimality* equals 1 and *Equidex* equals 0 (refer to Figure 3A).



(A) **(B)** **(C)** **(D)**

Figure 3: 3A: A diagram showing the quality of subgoals depicted by *Optimality* and *Equidex*. 3B The gridworld environment. 3C: A randomly generated state transition graph. 3D: An adapted VAE-like encoder-decoder *goal-reducer*.

Using these indices, we assess the performance of the *goal-reducer* on two datasets. The first dataset is based on a four-room gridworld task, commonly used in multi-goal RL benchmark tests (Figure 3B). In this dataset, we simulate an agent exploring the environment without any prior knowledge of its structure. The second dataset derives from a constructed state graph, characterized by random connections among states and allowing self-connections (Figure 3C). We execute undirected random walks on this graph. For both datasets, we use *goal-reducer* architecture to learn the $s, g \rightarrow s_g$ association, employing an encoder-decoder network (Figure 3D). This network accepts concatenated representations of s and g as inputs, initially producing a latent probabilistic representation through the encoder Q. A prediction of s_g is then generated by the decoder P, utilizing the sampled latent representation. To circumvent sparse and

discrete encoding in both datasets, random embeddings are applied for each. Consequently, s, g, and s_g are represented as high-dimensional random vectors. For simplicity, all random embeddings remain fixed during training. The adopted loss function mirrors that of a classical VAE, comprising a subgoal reconstruction loss (MSE) and a weighted KL divergence penalty [27]:

$$L = -\alpha D_{KL}(Q(s_g|s,g)||P(z)) + \mathbb{E}_{Q(z|s,g)}[\log P(s_g|z)] \tag{4}$$

Training results (Figure 4A and 4B) indicate that in both datasets *Loop-removal sampling* surpasses *Trajectory sampling*, which in turn outperforms *Random sampling* in terms of training loss. The *Optimality* values follow a similar trend:

Loop-removal sampling > *Trajectory sampling* ≫ *Random sampling*

Given that all subgoals are uniformly sampled from the dataset, the optimality for all strategies approximates 0. Moreover, with frozen random embeddings representing states, the predicted s_g is identified by computing the Euclidean distance between the generated output and all embeddings, selecting the closest as the predicted state. To verify whether the prediction closely aligns with the intended states and diverges from others, we measure the separability of predictions (bottom row of Figure 4A, 4B). After normalizing these distances against the average distance between all state embeddings, we visualize the average distance to the closest embedding (red lines) and to all other embeddings (blue lines). The results demonstrate that *Loop-removal sampling* achieves superior separability compared to other strategies.

To further examine the capability and stability of the trained *goal-reducer*, we assess its proficiency in recursive subgoal reduction. In this process, we input the predicted subgoal embedding into the *goal-reducer*, treating it as g to generate a subsequent subgoal s'_g. By iterating this procedure for several steps, we postulate that the *goal-reducer*, particularly when trained with *Loop-removal sampling*, will produce subgoals that are both effective and sufficiently close to the current states to facilitate easier navigation. This aspect will be elaborated upon later. The experimental results reveal that, after three iterations of recursive goal reduction (t=3, illustrated at the bottom of Figure 5A, 5B), the *goal-reducer* trained with *Loop-removal sampling* achieves the most favorable optimality distribution.

In terms of *Equidex*, as illustrated in Figure 6A and 6B, recursive goal reductions enhance the *goal-reducer*'s ability to predict subgoals that are nearer to the current states. This improvement is evidenced by the shift in the equidex distribution from t=0 to t=2. Consistent with our hypothesis, among all the strategies, *Loop-removal sampling* demonstrates the most favorable equidex distribution, particularly when compared to the background random distribution.

Together, the results in Figure 5A, 5B, 6A and 6B demonstrate the superiority of *Loop-removal sampling* as a training strategy for *goal-reducer* to produce proper subgoals.

A natural question arises after demonstrating that the *goal-reducer* with *Loop-removal sampling* can generate appropriate subgoals: Can this *goal-reducer* practically aid navigation in environments with constantly shifting goals? To explore this, we test the *goal-reducer* in two scenarios: the four-room gridworld navigation task (Figure 7B) and the robot arm reach task (Figure 7C). For the gridworld task, where the input space is inherently discrete, unlike the robot arm reach task, we increase the challenge

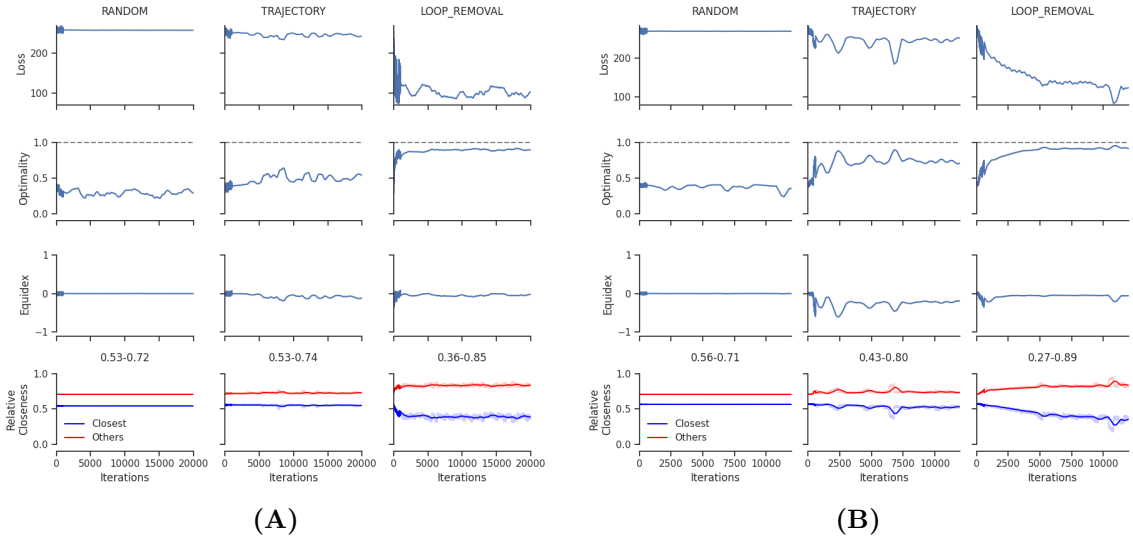


Figure 4: 4A: *goal-reducer* training results of the four-room gridworld task with different strategies. Top row: Training loss. Second row: Optimality. Shaded region represents standard deviation. Third row: Equidex. Shaded region represents standard deviation. Bottom row: Separability of predictions generated by *goal-reducer* between the closest state embedding (red) and the average of all other embeddings (blue). Shaded region represents standard deviation. The titles describe their numerical range after training. 4B: *goal-reducer* training results of the random graph with different strategies. Rows are the same as 4A

by using rendered images as direct inputs for the algorithm. Additionally, we create a partially observable environment by allowing the agent to see only its neighboring areas (Figure 7B). To eliminate the possibility of the agent relying on visual cues (the green square in Figure 7B) to locate the goal, we mask the goal indicator in the input images. This forces the agent to deduce its proximity to the goal using its internal state representation, shaped by the environment's layout and color. In each episode within this environment, the goal state is uniformly sampled from all possible locations, and the agent's starting position is randomized similarly. This approach ensures that the agent cannot master the task by merely memorizing a fixed goal location.

A widely adopted approach in modern RL for tackling such problems is to employ the Deep Q-learning (DQL) algorithm. This algorithm typically consists of two main components: a sensory encoder and a Q-network, while sometimes they're together referred as a Q-net. The sensory encoder's role is to transform sensory signals, i.e., images, into abstract state representations s_t and g. The Q-network, usually a deep forward network, takes these representations as inputs and produces distributions of Q-values for all possible actions, as the action space in this task is discrete. The action is then selected by sampling from the action probability distribution, $\pi(a_t|s_t, g)$, which is derived from the Q-value distribution using a normalization function like softmax. At the heart of DQL is the Bellman Equation, employed to train the Q-network in conjunction with the sensory encoder:

$$Q_{\phi_{k+1}} = \arg\min_{\phi} \frac{1}{2} \mathbb{E}_{(s_t,g,a_t,s_{t+1})\sim D} \left[Q_t^* - Q_\phi \left(s_t, g, a_t \right) \right]^2 \tag{5}$$

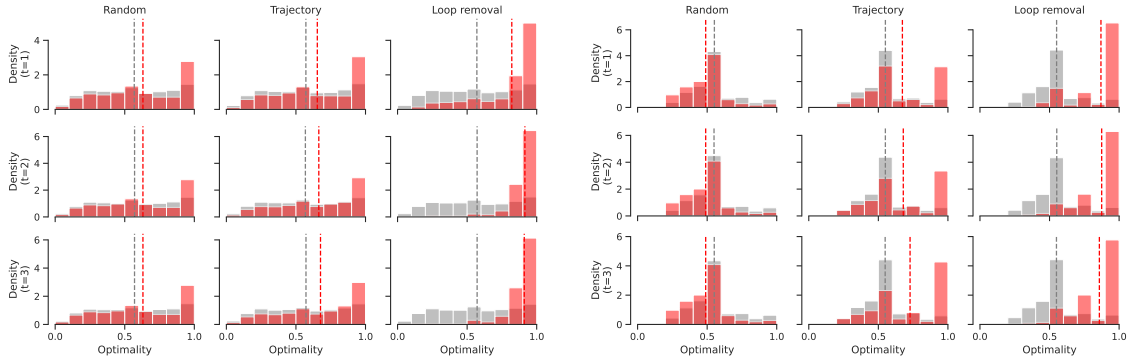


Figure 5: 5A: The evolution of subgoal optimality generated by *goal-reducer* trained with different strategies. Red histograms represent the generated subgoal distribution, while the gray histograms represent the a random optimality distribution used for comparison. Red and gray dashed lines represent the corresponding mean value for two types of distributions. 5B: The same as 5B, but for random graphs.

The target Q value is calculated as follows:

$$Q_t^* = r\left(s_t, g, a_t\right) + \gamma \mathbb{E}_{a_{t+1} \sim \pi(.|s_t, g)} Q_{\phi_k}\left(s_{t+1}, g, a_{t+1}\right) \tag{6}$$

In this update process, the replay buffer plays a role similar to that of episodic memory in animal learning, supplying training samples via sequences of state transitions stored as $(s_t, g, a_t, r_t, s_{t+1})$.

Fortunately, this replay buffer can also be utilized to train the *goal-reducer*, as long as it undergoes the *Loop-removal sampling* process as discussed earlier. Thus, we train the *goal-reducer* using the loss function from Equation 4, in parallel with the training governed by Equation 5.

Given the effectiveness of *Loop-removal sampling*, we hypothesize that the *goal-reducer* will converge more rapidly than the Q-network. Conversely, we anticipate that the Q-network will learn s, g pairs in an imbalanced manner, e.g., pairs closer to each other may be easier to learn. Combining these insights suggests a strategy for using a well-trained *goal-reducer* to enhance DQL. Since the optimal action for any location (subgoal) on the best path to a given goal should also be optimal for that ultimate goal, we can "align" the action distribution generated with g, i.e., $\pi(a_t|s_t, g)$, with the distribution generated using s_g, i.e., $\pi(a_t|s_t, s_g)$, via KL divergence. This approach is similar to the high-level policy training used by [9]. However, since it's not guaranteed that $\pi(a_t|s_t, g)$ is of lower quality than $\pi(a_t|s_t, s_g)$, a mechanism for alignment is needed. We opt to use the entropy $H[\pi(a_t|s_t, g)]$ as a proxy for confidence in quality assessment. Thus, an additional loss function for the Q-network is defined as:

$$\sum_{s_t, g} w_{s_t, g} \cdot D_{KL}\left[\pi\big(a_t|s_t, g\big) \middle\| \pi\big(a_t|s_t, \Phi(s, g)\big)\right] \tag{7}$$

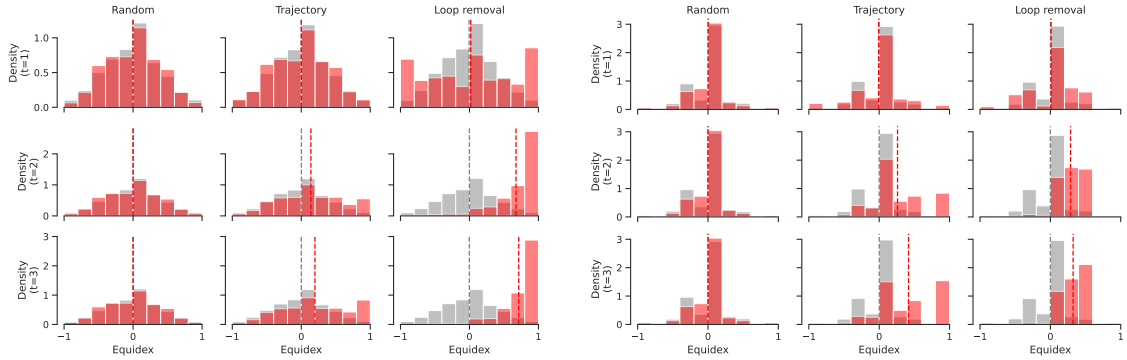


Figure 6: 6A: The evolution of subgoal equidex generated by *goal-reducer* trained with different strategies. All notions are the same with Figure 5A. 6B: The same as 6A, but for random graphs.

where

$$w_{s_t,g} = \begin{cases} 1, & \text{if } H[\pi(a_t|s_t,g)] \geq H[\pi(a_t|s_t,\Phi(s,g))] \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Through Equation 7, the *goal-reducer* is seamlessly integrated into the RL algorithm. For the discrete action spaces, where we usually used a deep Q network to estimate the action probability distribution, Equation 7 can be added to the Bellman value loss so they optimize the Q network together. In the continuous action space setting, a policy $\pi(a_t|s_t,g)$, e.g., an actor in Deep Deterministic Policy Gradient models or Soft Actor-Critic models, is usually an independent neural network and is optimized indirectly by inserting its output into a value function and then maximizing it: $max_{\pi_\theta} V(s_t, \pi(a_t|s_t,g))$. In this case, Equation 7 can be also calculated along with the value maximization loss and optimized simultaneously.

In the robot arm reach task, the overall methodology is similar, but with the key difference that the action space is continuous. To accommodate this, as discussed above, we slightly modify the aforementioned framework. Instead of DQL, we employ the Soft Actor-Critic (SAC) algorithm [19], augmented with Hindsight Experience Replay (HER), a standard technique used to improve RL algorithm's performance in goal-directed learning [2]. The reward mechanism remains consistent with the gridworld task, with the agent receiving a -1 reward for each step until it reaches the designated goal location.

In our experiments, we classify the SAC and DQL as DRL (Deep Reinforcement Learning) types, while the versions enhanced with the *goal-reducer* are categorized as GR+DRL types. The results, as depicted in Figure 7D, reveal that when combined with the *goal-reducer*, the performance of the respective DRL algorithms significantly improves in both the gridworld navigation task and the robot arm reach task.

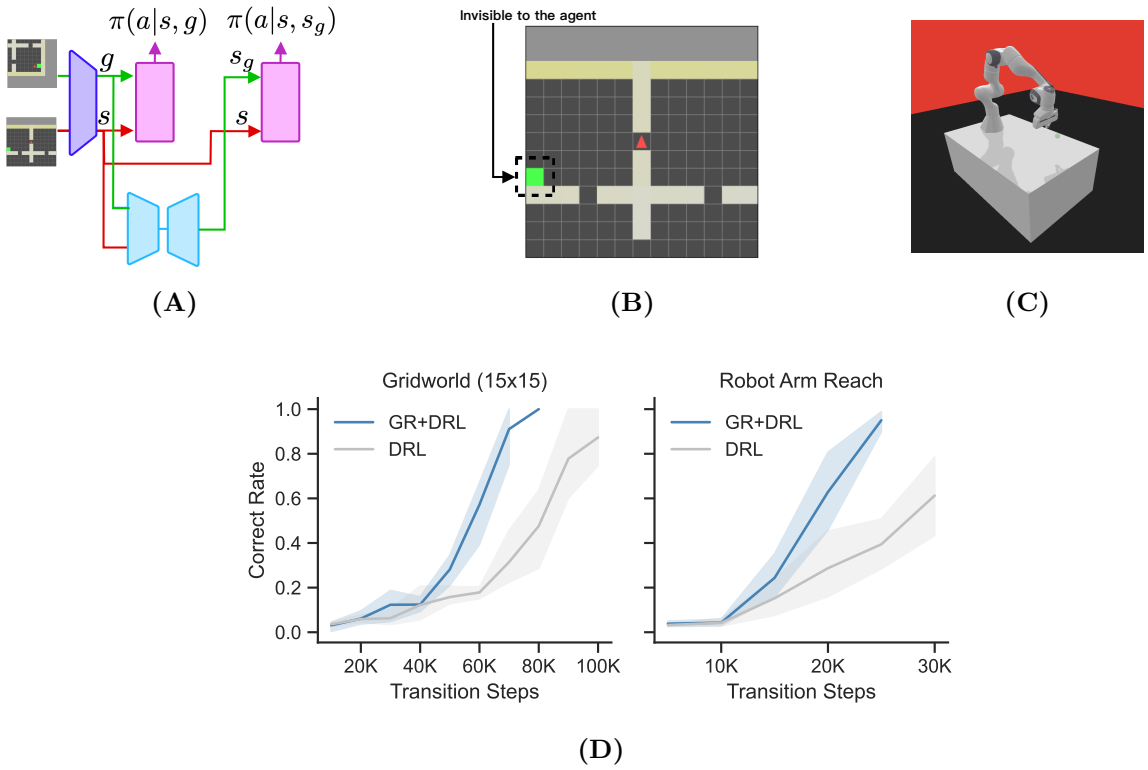


Figure 7: 7A: The mixed architecture of the *goal-reducer* combined with Deep Q-learning modules. 7B: An example input image of the partially-observable four-room gridworld task. Notice that for demonstration the goal state in presented as a green square, while in the run time it is masked out and thus invisible to the agent. 7C: An example initial configuration of the robot arm and target in the reach task. 7D: The performance comparison between DRL and *goal-reducer*-enhanced DRL in the gridworld and the robot arm reach task. Left: Gridworld task; Right: Robot arm reach task. Blue lines represent the correct rate curves of *goal-reducer*-enhanced DRL, while the gray lines represent the DRL correct rate curves. Shaded regions represent the corresponding standard error over multiple runs.

2.2 Standalone *goal-reducer* planning surpasses RL

Reflecting on the pivotal observations from Figure 5A and 6A, which indicate that the *goal-reducer* can progressively simplify the problem (as shown by the evolution of equidex) while preserving its effectiveness (evidenced by the evolution of optimality), we are prompted to consider whether a *goal-reducer* devoid of an RL component could independently facilitate navigation. Typically, the action execution process in RL depends, either explicitly or implicitly, on a value or advantage function during both training and inference phases. These value/advantage functions, which are crucial for assessing the significance of various actions, are often estimated using the Bellman equation. However, the convergence rate of the Bellman equation can be notoriously slow and unstable.

Consequently, we explore whether a standalone *goal-reducer* might exhibit performance comparable to its DRL counterparts. In this scenario, no global policy $\pi(a|s,g)$

is developed. Instead, we pair the *goal-reducer*, trained using the *Loop-removal sampling* strategy, with a local policy. This policy processes adjacent s and g to generate appropriate actions. Fortunately, such local policies are simpler to train and do not necessitate the use of the Bellman equation. When the goal g is substantially distant from the state s, the action execution process is decomposed into two consecutive phases: goal reduction and action generation. In the goal reduction phase, a trained *goal-reducer* iteratively applies $s_g = \Phi(s, g)$ and updates $g = s_g$, until a subgoal s_g is produced. This subgoal should be such that the uncertainty of the local policy, indicated by the entropy of the action distribution $H[\pi(a|s, s_g)]$, falls below a certain threshold.

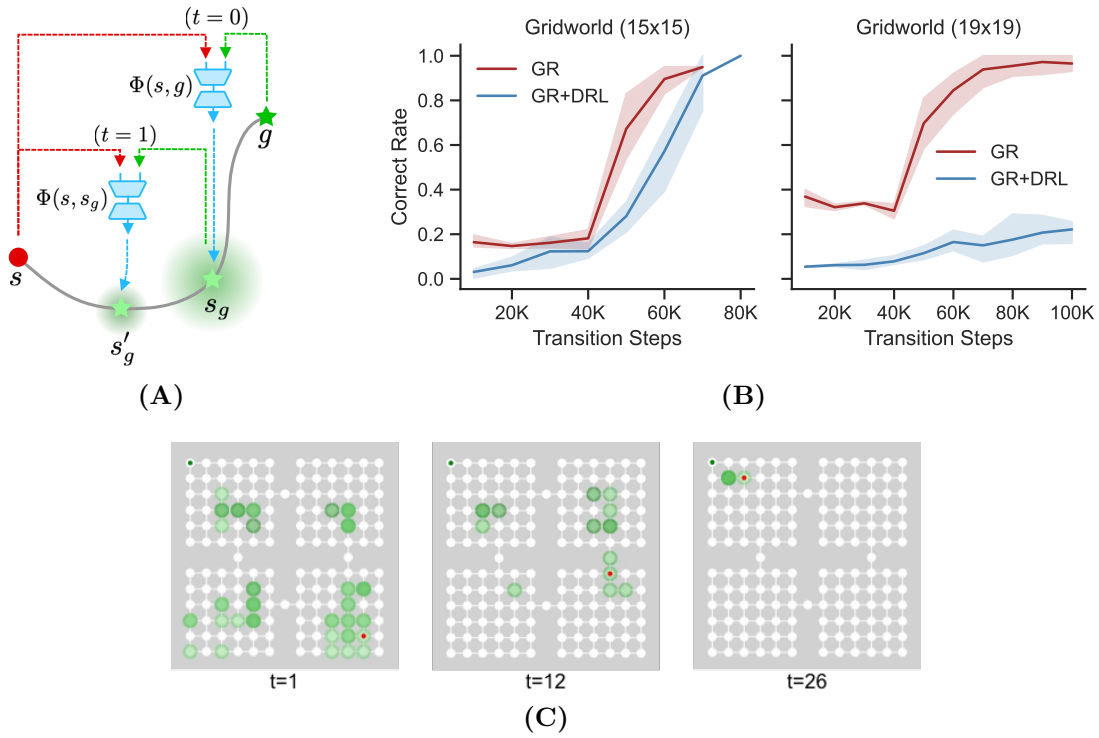


Figure 8: 8A: The standalone planning process using *goal-reducer* and a local policy. 8B: The performance comparison between standalone *goal-reducer* and *goal-reducer*-enhanced DRL algorithms in the 15x15 (left) and 19x19 (right) gridworld navigation task. 8C: A demonstration of navigation progress using standalone *goal-reducer* without RL in a 15x15 gridworld. Red dots indicate the agent's location, while the green dots in the upper left corner denote the ultimate goal. Larger scattering shadowed green circles represent subgoals generated by *goal-reducer*. Since *goal-reducer* involves a sampling process, a single location may be assigned as a subgoal multiple times. The depth of the green color indicates the relative frequency of a location being sampled as a subgoal.

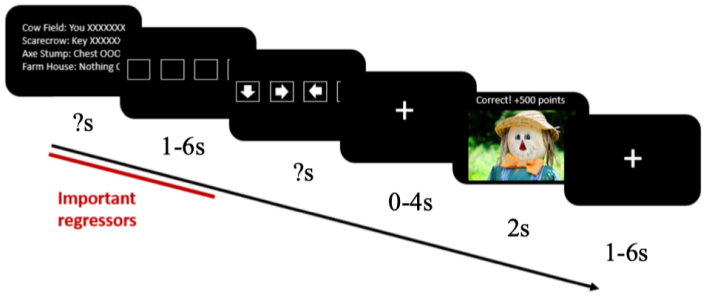
The entropy threshold in this context serves to balance the exploration-pondering trade-off, akin to that commonly employed in certain DRL algorithms [46, 19]. We choose the gridworld task as a testbed for this framework due to its relative simplicity, facilitating easier analysis and visualization of the entire internal planning process.

To evaluate the impact of scalability, we utilize mazes of two different sizes: a 15x15 maze and a larger 19x19 maze. We compare the performance of the standalone *goal-reducer* in planning against the *goal-reducer*-enhanced DRL algorithms, omitting the pure DRL solutions as they've been shown to be inferior to the latter in previous experiments (Figure 7D). All parameters remain consistent with previous experiments. The findings, presented in Figure 8B, suggest that a standalone *goal-reducer* can effectively solve the navigation task. Moreover, it demonstrates greater efficiency compared to *goal-reducer*-enhanced DRL algorithms. To better understand the internal working mechanism, Figure 8C illustrates the standalone *goal-reducer* navigation process. As time proceeds, the generated subgoals progressively converge in space, consistently providing a subset that directs the agent's actions towards the ultimate goal.

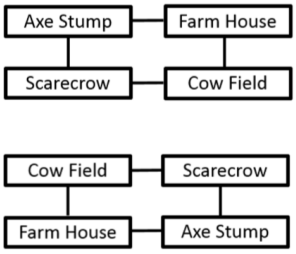
2.3 The *goal-reducer* in the brain

In the above experiments, we have demonstrated the superiority of the *goal-reducer* in tasks involving both discrete and continuous action spaces. Next, we question whether this algorithm, i.e., a goal reduction mechanism paired with a straightforward local policy, can model human behavior in problem solving. As previously mentioned, it is unlikely that humans solve problems solely through the slow value-based iteration mechanism in an ever-changing environment. Instead, evidence suggests that the brain, particularly the prefrontal cortex, employs rules for controlling/inhibiting behavior and monitoring performance [36, 1, 13]. When recast in the control theory framework, such rules appear as evidence of humans using internal goals to achieve desired states. This framework also aligns with the algorithm proposed in this study, where an agent executes actions by simplifying or reducing ultimate goals into smaller, more achievable goals through a simple policy. Therefore, we compared the model's behavior with that of human subjects in a Treasure Hunt task [54], which necessitates flexible goal representation changes.

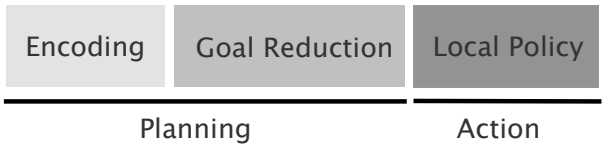
All research involving human subjects was approved by Indiana University's IRB. Subjects provided informed consent, and the procedures adhered to applicable guidelines and the Declaration of Helsinki. The data were re-used from [54], which also provides the full methods which are summarized briefly here. In the task, human subjects were placed in one of four possible starting states on one of two maps (Figure 9B) and were required to reach states designated as a "chest" (the ultimate goal). However, possession of a "key" is necessary when reaching the "chest" to obtain a reward. The locations of the "key" and "chest" are presented to the subject at the start of each episode (Figure 9A) and change randomly across episodes. The model, following the architecture depicted in Figure 8A, was also trained on the same task. Unlike previous models [54], for scalability and universality, we directly represented all information in text for the model (Figure 9D), similar to the sentences displayed on the screen to the human subjects. These texts were encoded by the OpenAI embedding model "text-embedding-ada-002" and then passed to the *goal-reducer* and the local policy π. Another difference in the task settings between the model and human subjects is that the reward unit for reaching "key" and "chest" in human subjects is set to 500 points, while for the model we normalize them to 1. So the model would receive a total reward of 2 when it successfully finishes an episode, either taking one or two steps. After training on the task, we analyzed the model's data and compared it with human subjects' brain activity measured via fMRI, using representational similarity analysis



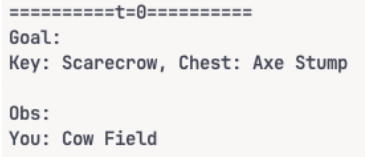
(A)



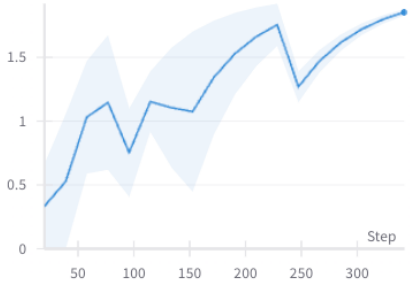
(B)



(C)



(D)



(E)

Figure 9: 9A: The timeline of a Treasure Hunt task, showing a series of key and chest location information at the beginning of each episode. After sufficient planning time, action options are presented and the subject chooses the action to execute. The outcome is then revealed to the subject. 9C: Two maps used in the Treasure Hunt task. 9D: An example of text-based input to the model in the Treasure Hunt task. 9E: The reward curve of the *goal-reducer* paired with a local policy in the Treasure Hunt task during training. The maximum reward is set to 2.

(RSA) [28]. RSA considers all conditions that occurred in the task and compares the activity similarity of the model/brain between each pair of conditions via Pearson correlation, thereby forming a symmetric representational dissimilarity matrix (RDM). The entries in the RDM range from 0 to 2 (1 minus the possible correlation ranging from -1 to 1), where a lower value indicates higher similarity.

In our study, each RDM for a brain voxel or model component is a 64×64 matrix, where $64 = 4 \times 8 \times 2$. Here, the number 4 represents the four possible current subject/model locations on the map. The number 8 represents all possible episode configurations (comprising four two-step key-chest configurations when the subject/-

model is at the starting location, two one-step "key-chest" configurations when the subject/model is at the starting location, and two two-step key-chest configurations when the subject/model is at the middle location). The factor 2 accounts for the two phases of each movement (planning and action phase) as depicted in Figure 9C. For the human fMRI data, such RDMs are calculated for each voxel in the brain for each subject, with a neighborhood radius set to 10 mm. For the model, RDMs are first calculated for learned $\mathcal{S}$ and $\mathcal{G}$ representations and the intermediate representations in the corresponding neural networks. Moreover, we calculated RDMs for the *goal-reducer*'s hidden representations as well when the goal reduction is performed. The action layer, i.e., the output distribution of at each step is also calculated. Together, these form the following model RDMs: $\mathcal{S}$ RDM for state representations, `Intermediate` $\mathcal{S}$ RDM for state representations in hidden layers, $\mathcal{G}$ RDM for goal representations, `Intermediate` $\mathcal{G}$ RDM for goal representations in hidden layers, π RDM for action representation, and Φ RDM for *goal-reducer*'s hidden layers.

The comparison between the model RDMs and human fMRI RDMs aims to examine whether there exist positive correlations between some of the model's components and brain areas in different human subjects. Specifically, Pearson correlation between each voxel and a model RDM's upper triangular part, given their symmetrical nature, is calculated. This forms a series of fMRI RSA maps for each subject, showing whether certain brain regions are positively correlated with the aforementioned model components. Second-level analysis is then performed to extract the population effect. To do so, we first transform the individual correlations into z-scores using the Fisher transform. Then, all z-score maps are projected into a shared space, and a 1-sample T-test is performed for each voxel in the shared brain space to examine whether the corresponding correlation is significant across subjects. After that, cluster-based correction is performed to filter out statistically significant regions. With the above analysis pipeline, we found that different brain regions have a high match with distinct components in the deepGOLSA model. For a complete list of all clusters, see Table 1. Here, we highlight several regions that appear strongly related to the model.

As discussed above, one of the key ingredients in the deepGOLSA model is the *goal-reducer* $\Phi(s, g)$, which is applied to the goal space $g \in \mathcal{G}$. This reducer generates newer, simpler goals, that is, $\Phi(s, g) \to g'$, which can be resolved with less effort. We therefore examined regions correlated with goal representations $g \in \mathcal{G}$ as well as regions associated with intermediate layer neurons in $\Phi(s, g)$.

The RSA results in Figure 10A show that the activity in the ventromedial prefrontal cortex (vmPFC) corresponds to the neurons representing $g \in \mathcal{G}$ in the deepGOLSA model. Following this, we evaluated the activity matching the internal neurons of the *goal-reducer* $\Phi(s, g)$. As depicted in Figure 10B, there are several regions whose activities correlate with Φ, including parts of the vmPFC (indicated by the right circle in Figure 10B) and the left accumbens. This finding is particularly intriguing for two reasons. First, numerous studies have established that the vmPFC is related to value and goal representations [26, 44]. Second, the close spatial relationship between regions matching $\mathcal{G}$ and Φ (Figure 10B) reflects the deepGOLSA model's organization (Figure 10D), wherein Φ and $\mathcal{G}$ are interconnected through the generation and input of different goals. This recurrent connectivity suggests that the widespread recurrent connections in the vmPFC [35] may fulfill the role of goal reduction or perform reverse future simulation [5], thereby simplifying the task. The finding is consistent with a

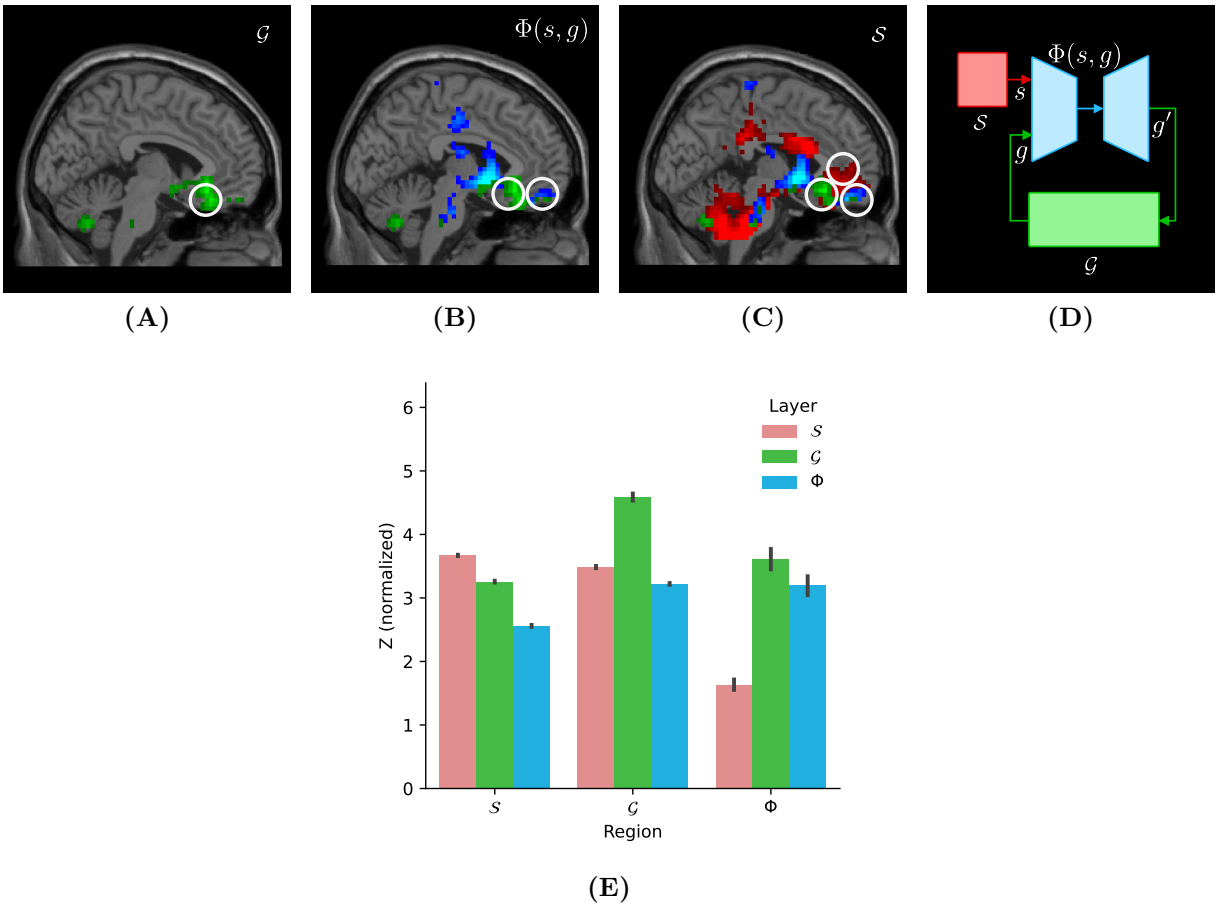
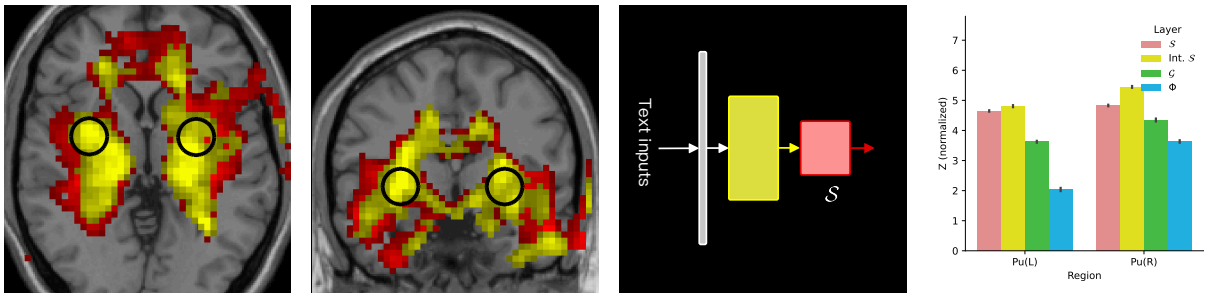


Figure 10: 10A: Population Z maps (x=-4) for goal representation layer $\mathcal{G}$ (green). 10B: Population Z maps (x=-4) for intermediate layers in the *goal-reducer* (blue), $\Phi(s, g)$, stacked with $\mathcal{G}$ (green). 10C: Population Z maps (x=-4) for state representations $\mathcal{S}$ (red). 10D: The interplay between $\Phi(s, g)$, $\mathcal{S}$ and $\mathcal{G}$, through the input and generation of different goals. 10E: The distribution of z values of different layers (color marked, consistent with the colors in 10C and 10D) for three regions specified using a radius of 5 millimeter and three centers, $\mathcal{S} : (-4, 43, -5)$,$\mathcal{G} : (-4, 25, -12)$ and $\Phi(s, g) : (8, 49, -21)$.

posterior-to-anterior hierarchy of goal planning in the prefrontal cortex [7], and our results may suggest the function of such a hierarchy - namely that anterior regions provide goal reduction functionality for more posterior regions.

To substantiate this hypothesis, a pertinent question arises: Are there any regions near the Φ and $\mathcal{G}$ representations that correspond to the activity of $s \in \mathcal{S}$, given their close spatial association in the *goal-reducer* (Figure 10B)? Accordingly, we investigated the Z map of state representations and indeed discovered a region above Φ and $\mathcal{G}$ in the vmPFC that is correlated with $\mathcal{S}$ (as denoted by the top circle in Figure 10C). The average z-values in these regions show differential representational loading on the deepGOLSA layers (Figure 10E), although the ROI z-value comparison is a descriptive statistic only given that the regions were selected for strong loading on the various model layers [29]. Together, these regions suggest a central role of vmPFC in goal-

oriented learning: just like the *goal-reducer* Φ, the vmPFC may act as a context-aware reverse world model, continuously reduce the complexity of a given task to an easier subtask. Superficially, this reduction process have many names, such as goal ranking and retrieving [38], goal value-coding [34], or reward value comparison [47]. Lesion studies also show vmPFC is related to deficits in goal-directed choice in human subjects [42, 45].



(A) **(B)** **(C)** **(D)**

Figure 11: 11A and 11B: Population Z maps (z=1: 11A) and (y=-3: 11B) for the state representation layer $\mathcal{S}$ (red) and its intermediate layers (yellow). The regions are circled out black. 11C: A schematic depicting the generation of state representations. As all information in the treasure hunting task is encoded as text, we use OpenAI embeddings to convert this information into high-dimensional vectors (gray). Subsequently, a feedforward neural network processes the embedding vector to produce a relatively low-dimensional state representation (red), with the intermediate layer of this network indicated in yellow. 11D: Distribution of Z values across different layers (colors corresponding to those in 10C, 10D, 11A, and 11C) for the left (Pu(L)) and right (Pu(R)) putamen, defined within a 5-millimeter radius around two centers, $(-26, 1, 8)$ and $(25, 1, 1)$.

Aside from goal reduction, another crucial component in the deepGOLSA model is the state representation denoted as $\mathcal{S}$. State representation is involved in both policy execution and goal reduction, serving as a context signal to determine appropriate actions and simplified goals. We performed representational similarity analysis (RSA) on the state representation and its intermediate neural network layers. As Figure 11 illustrates, the RSA results reveal that the z-map patterns for the state representation layer and its intermediate layer substantially overlap, confirming our hypothesis that they inherently represent the same variable.

Moreover, in addition to the ventromedial prefrontal cortex (vmPFC) region described earlier, RSA indicates significantly elevated representational loading for these two layers in the bilateral putamen regions (Figure 11), while there is a lower level of loading for the goal representation layer and an even lesser extent for the goal reduction layer (Figures 11A, 11B and 11D). The putamen, a component of the larger basal ganglia system, is recognized for its involvement in habitual behaviors [53] and goal-directed actions [3, 24]. Within the deepGOLSA model, this suggests that such an area should engage in both local policy enactment (the habitual aspect) and goal reduction (goal-directed component). The congruence between the state representation and its intermediate layer with this role may illuminate why activation related to the goal and

its reduction is lower in this region (Figure 11D) since they are not as essential for the habitual component of the local policy.

3 Discussion

Goal conditioned learning is crucial to various complex behaviors and cannot be solved trivially by just expanding the classical state space to the Cartesian product of the state space and the goal space, as illustrated by the robot arm and the four-room maze navigation experiments above. On the other hand, the success of humans in vastly complex environments with various goals suggests the existence of methods to deal with goal conditioned learning. The deepGOLSA model addresses this by proposing how a neural model can solve such problems. The first observation we rely on is that "closer" goals implicitly encode solutions to "faraway" goals. Classical navigation algorithms like A* take this recursive approach and have many real-world use cases.

However, when it comes to the neural representation space, an effective tool to produce "closer" goals with "faraway" goals becomes non-trivial, and it is unclear how typical distance-to-goal heuristics used to direct a search (as in the A* algorithm) can be implemented in the general case of high dimensional state graphs. This is because one premise of such algorithms is the distance/cost/effort (or negative value) taken from any state to any goal can be measured and compared effectively. In classical algorithms this can be implemented via graph search mechanisms like BFS. While in the neural representation, without assuming superposition, no such metrics exist. We might suppose that another network can be trained to generate a distance or value function, but this merely removes the problem by one degree rather than solving it. The size of state space in goal conditioned learning makes the convergence of a value function unrealistically slow. Nevertheles, some recent studies on graph theory [6] and metric learning [48] imply that some functions may exploit the relative order relationships like the triangular inequality [48] between different states in the space. To a certain extent our approach embodies the same spirit: We make use of the fact that loops in trajectories of space imply sub-optimality or even abyss in distance/value estimates, which constitutes a weaker assumption and thus a more generally powerful approach than metric learning. Loop removal makes realistic sense specifically when the representation space is discrete and relatively restricted, where neural representation lives in a continuous high dimensional real space. We mitigate this issue using a similar idea adopted in VAE [27] and other generative models like VAE [11] and GAN[8], i.e., using noise to create robust overlapping representations of states. With such representations, a filtration technique can be applied to remove loops in a trajectory. Building on this approach, in RL problems, a *goal-reducer* Φ can efficiently sample subgoals from trajectories in the replay buffer. Thus it can not only accelerate classical RL algorithms, but also learn a task completely with only a local policy and without value function iterations (i.e. Bellman equation-free) at all. Given its computational superiority in solving tasks with changing goals and the aforementioned biological adaption to complex environments, we applied this algorithm to a human problem-solving task and looked for neural correlates of the deepGOLSA model in human brains. The analysis reveals that regions including the vmPFC and putamen are indeed correlated with some components in the model. Notably, the vmPFC shows a separated pattern that three regions matches three spatially interconnected compo-

nents, i.e., the state, goal and goal reduction layer in the model. In the past, vmPFC has been assigned various roles in goal directed behaviors from goal comparison to value retrieval [38, 34, 47]. These roles can be unified through the lens of goal reduction in the deepGOLSA model, as the goal reduction process implicitly performed the corresponding behavioral process. For example, multi-run goal reductions can be seen as a pondering goal comparison, while since the candidate subgoals are getting simpler and simpler to solve, their value are increasing in the process. Also, we found that regions like the bilateral putamen match the state representation layer and especially its intermediate layer, suggesting that the basal ganglia system may participate in the decision making process by continuously injecting current state representations as a context signal.

With all these findings, our study tries to build a model for a general goal reduction mechanism, both biologically plausible and computationally advantageous. Nevertheless limitations exist in various aspects. First, the neural approach we take is a closer step to biological realization but not completely, as the deepGOLSA model simulations use the backpropagation algorithm [43], known for its incompatibility with plausible neural learning rules [51]. This may not be a major issue though, as we and others have shown more biologically plausible methods that approximate full gradient descent in feedforward [31] and recurrent [10] networks. Second, in the loop removal process we adopted the filtration technique. In the language of neuroscience, this suggests a neural mechanism that can avoid closed-loop line attractors in associative memories. To our knowledge currently there are no empirical studies focusing on the existence of this mechanism. This may be a crucial direction to explore, especially in the study of hippocampus, as the hippocampus is famous for its role in learning through reverse replay, which in principle is exactly the subgoal reduction process in the deepGOLSA model. A possible experiment design would be to involve and detect if loops appear during reverse replay for tasks involving looped movement trajectories and if yes, how that may influence the efficacy of learning.

4 Conclusion

In this study, we proposed the deepGOLSA model, an alternative algorithm to the classical Bellman equation based RL algorithms to solve goal conditioned learning tasks, through efficient goal reductions. To test the advantage of deepGOLSA, we compared its performance against several baseline RL algorithms on one four-room navigation task with different sizes and a robot arm manipulation task. In all tests, RL algorithms accelerated with deepGOLSA consistently outperform their RL-only counterparts. Moreover the deepGOLSA model trained without the Bellman question and just a local policy solves these tasks even more quickly than those with both RL and goal reduction accelerations, in the four-room task. To examine the potential of this model to explain human problem solving process, we applied the model to a treasure hunting task and compared it with human subject's brain imaging data recorded while performing the same task. The RSA analysis reveals that regions including the vmPFC and putamen, which are considered to have played several distinct roles in goal directed behaviors, match the crucial goal reduction layer and its surrounding components in the deepGOLSA model. The model thus, aside from its computation potential, serve as a candidate theory to explain the neural mechanism of general goal directed problem

solving in humans, and in a manner that can be scaled up to arbitrarily large and complex problem spaces.

References

[1] William H Alexander and Joshua W Brown. Medial prefrontal cortex as an action-outcome predictor. *Nature neuroscience*, 14(10):1338–1344, 2011.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[3] Bernard W Balleine and John P O'doherty. Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action. *Neuropsychopharmacology*, 35(1):48–69, 2010.

[4] David L Barack and John W Krakauer. Two views on the cognitive brain. *Nature Reviews Neuroscience*, 22(6):359–371, 2021.

[5] Roland G Benoit, Daniel J Davies, and Michael C Anderson. Reducing future fears by suppressing the brain mechanisms underlying episodic simulation. *Proceedings of the National Academy of Sciences*, 113(52):E8492–E8501, 2016.

[6] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[7] Iva K Brunec and Ida Momennejad. Predictive representations in hippocampal and prefrontal hierarchies. *J. Neurosci.*, 42(2):299–312, January 2022.

[8] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*, 2022.

[9] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. *ArXiv*, abs/2107.00541, 2021.

[10] Huzi Cheng and Joshua W Brown. Replay as a basis for backpropagation through time in the brain. February 2023.

[11] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[12] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[13] Maël Donoso, Anne GE Collins, and Etienne Koechlin. Foundations of human reasoning in the prefrontal cortex. *Science*, 344(6191):1481–1486, 2014.

[14] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[15] Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Neural Information Processing Systems*, 2019.

[17] Justin M. Fine, Noah Zarr, and Joshua W. Brown. Computational neural mechanisms of goal-directed planning and problem solving. *Computational Brain & Behavior*, 3:472 – 493, 2019.

[18] Samuel J Gershman and Nathaniel D Daw. Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual review of psychology*, 68:101–128, 2017.

[19] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[20] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[21] Geoffrey E Hinton. Distributed representations. 1984.

[22] Christopher Hoang, Sungryull Sohn, Jongwook Choi, Wilka Carvalho, and Honglak Lee. Successor feature landmarks for long-horizon goal-conditioned reinforcement learning. In *Neural Information Processing Systems*, 2021.

[23] Alumit Ishai, Leslie G Ungerleider, Alex Martin, Jennifer L Schouten, and James V Haxby. Distributed representation of objects in the human ventral visual pathway. *Proceedings of the National Academy of Sciences*, 96(16):9379–9384, 1999.

[24] Marjan Jahanshahi, Ignacio Obeso, John C Rothwell, and José A Obeso. A fronto–striato–subthalamic–pallidal network for goal-directed and habitual inhibition. *Nature Reviews Neuroscience*, 16(12):719–732, 2015.

[25] Zhao Jin, Jian Jin, and WeiYi Liu. Autonomous discovery of subgoals using acyclic state trajectories. In *Information Computing and Applications: First International Conference, ICICA 2010, Tangshan, China, October 15-18, 2010. Proceedings 1*, pages 49–56. Springer, 2010.

[26] Joseph W Kable and Paul W Glimcher. The neural correlates of subjective value during intertemporal choice. *Nature neuroscience*, 10(12):1625–1633, 2007.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, page 4, 2008.

[29] Nikolaus Kriegeskorte, W Kyle Simmons, Patrick S F Bellgowan, and Chris I Baker. Circular analysis in systems neuroscience: the dangers of double dipping. *Nat. Neurosci.*, 12(5):535–540, May 2009.

[30] Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. Bidirectional model-based policy optimization. *ArXiv*, abs/2007.01995, 2020.

[31] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.*, 7(1):13276, November 2016.

[32] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[33] Nikos Makris, Jill M Goldstein, David Kennedy, Steven M Hodge, Verne S Caviness, Stephen V Faraone, Ming T Tsuang, and Larry J Seidman. Decreased volume of left and total anterior insular lobule in schizophrenia. *Schizophrenia research*, 83(2-3):155–171, 2006.

[34] Daniel McNamee, Antonio Rangel, and John P O'doherty. Category-dependent and category-independent goal-value codes in human ventromedial prefrontal cortex. *Nature neuroscience*, 16(4):479–485, 2013.

[35] Earl K Miller. The prefontral cortex and cognitive control. *Nature reviews neuroscience*, 1(1):59–65, 2000.

[36] Earl K Miller and Jonathan D Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202, 2001.

[37] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *ArXiv*, abs/1909.05829, 2019.

[38] John P O'Doherty. Contributions of the ventromedial prefrontal cortex to goal-directed action selection. *Annals of the New York Academy of Sciences*, 1239(1):118–129, 2011.

[39] Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] William D Penny, Karl J Friston, John T Ashburner, Stefan J Kiebel, and Thomas E Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Elsevier, 2011.

[41] Karl Pertsch, Oleh Rybkin, Frederik Ebert, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Long-horizon visual planning with goal-conditioned hierarchical predictors. *ArXiv*, abs/2006.13205, 2020.

[42] Justin Reber, Justin S Feinstein, John P O'Doherty, Mimi Liljeholm, Ralph Adolphs, and Daniel Tranel. Selective impairment of goal-directed decision-making following lesions to the human ventromedial prefrontal cortex. *Brain*, 140(6):1743–1756, 2017.

[43] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[44] Matthew FS Rushworth, MaryAnn P Noonan, Erie D Boorman, Mark E Walton, and Timothy E Behrens. Frontal cortex and reward-guided learning and decision-making. *Neuron*, 70(6):1054–1069, 2011.

[45] Brett Schneider and Michael Koenigs. Human lesion studies of ventromedial prefrontal cortex. *Neuropsychologia*, 107:84–93, 2017.

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[47] Caleb E Strait, Tommy C Blanchard, and Benjamin Y Hayden. Reward value comparison via mutual inhibition in ventromedial prefrontal cortex. *Neuron*, 82(6):1357–1366, 2014.

[48] Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. *arXiv preprint arXiv:2304.01203*, 2023.

[49] Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *Proceedings of the 54th annual design automation conference 2017*, pages 1–6, 2017.

[50] James C R Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E J Behrens. The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263.e23, November 2020.

[51] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

[52] Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *ArXiv*, abs/2202.04478, 2022.

[53] Henry H Yin and Barbara J Knowlton. The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience*, 7(6):464–476, 2006.

[54] Noah Zarr and Joshua W. Brown. Foundations of human spatial problem solving. *Scientific Reports*, 13, 2023.

[55] Lunjun Zhang, Ge Yang, and Bradly C. Stadie. World model as a graph: Learning latent landmarks for planning. *ArXiv*, abs/2011.12491, 2020.

Component	X	Y	Z	$p_{\text{corrected}}$	cluster_{k_E}	Z	Region(s)
Intermediate State Layer	32	-9	21	0	15,323	6.51	Right Cerebral White Matter
Intermediate State Layer	29	-9	11	0	15,323	6.44	Right Putamen
Intermediate State Layer	15	-12	-2	0	15,323	6.20	Right Cerebral White Matter
Intermediate State Layer	39	43	38	0.02	82	3.79	Frontal PoleRight Cerebral Cortex
Intermediate State Layer	-50	-64	28	0.01	96	4.24	Lateral Occipital Cortex, superior divisionLeft Cerebral Cortex
Intermediate State Layer	-44	-67	21	0.01	96	4	Lateral Occipital Cortex, superior divisionLeft Cerebral Cortex
Intermediate State Layer	-50	-47	35	0.01	96	3.60	Supramarginal Gyrus, posterior divisionLeft Cerebral White Matter
Intermediate Goal Layer	-37	-12	-26	0	18,578	5.82	Parahippocampal Gyrus, anterior divisionLeft Cerebral Cortex
Intermediate Goal Layer	-26	-23	1	0	18,578	5.78	Left Cerebral White Matter
Intermediate Goal Layer	-26	-16	32	0	18,578	5.50	Left Cerebral White Matter
Goal Reducer Intermediate Layer	-30	-9	15	0	10,362	6.45	Left Cerebral White Matter
Goal Reducer Intermediate Layer	-23	-26	32	0	10,362	6.01	Left Cerebral White Matter
Goal Reducer Intermediate Layer	25	-40	21	0	10,362	5.73	Right Cerebral White Matter
Goal Reducer Intermediate Layer	-13	32	52	0	183	4.59	Superior Frontal GyrusLeft Cerebral Cortex
Goal Reducer Intermediate Layer	-16	43	45	0	183	4.49	Frontal PoleLeft Cerebral Cortex
Goal Reducer Intermediate Layer	-13	50	35	0	183	3.53	Frontal PoleLeft Cerebral Cortex
State Layer	29	-9	18	0	14,266	6.60	Right Cerebral White Matter
State Layer	25	-23	35	0	14,266	5.82	Right Cerebral White Matter
State Layer	18	-12	1	0	14,266	5.80	Right Cerebral White Matter
State Layer	-50	-50	38	0.05	70	3.79	Supramarginal Gyrus, posterior divisionLeft Cerebral Cortex
State Layer	-47	-54	28	0.05	70	3.73	Angular GyrusLeft Cerebral White Matter
State-Goal Conjunction Layer	29	-9	-13	0	1,148	4.79	Right Cerebral White Matter
State-Goal Conjunction Layer	-2	-26	-40	0	1,148	4.72	Brain-Stem
State-Goal Conjunction Layer	22	-19	32	0	1,148	4.49	Right Cerebral White Matter
State-Goal Conjunction Layer	-16	36	-2	0.01	104	3.91	Left Cerebral White Matter
State-Goal Conjunction Layer	-13	32	-9	0.01	104	3.76	Frontal Medial CortexLeft Cerebral White Matter
State-Goal Conjunction Layer	-9	22	-13	0.01	104	3.35	Subcallosal CortexLeft Cerebral White Matter
Goal Layer	-16	8	-9	0	16,195	6.16	Left Putamen
Goal Layer	29	-12	28	0	16,195	6.04	Right Cerebral White Matter
Goal Layer	25	-5	8	0	16,195	5.81	Right Putamen

Table 1: Significant similarity clusters derived from SPM5 [40] for RSA analysis. Anatomical region labels are derived from [33]. The $p_{\text{corrected}}$ and cluster size (cluster_{k_E}) values are both corrected using SPM5. Note that some layers may involve more than one region due to the probabilistic nature of atlases in [33].