
Deep RL Needs Deep Behavior Analysis: Exploring Implicit Planning by Model-Free Agents in Open-Ended Environments

Riley Simmons-Edler^{1,2*} Ryan P. Badman^{1,2*} Felix Baastad Berg³ Raymond Chua⁴

John J. Vastola^{1,2} Joshua Lungert⁵ William Qian^{6,2} Kanaka Rajan^{1,2†}

¹Department of Neurobiology, Harvard Medical School

²Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University

³Department of Mathematics, NTNU

⁴School of Computer Science, McGill University & Mila

⁵Department of Computer Science, University of Toronto

⁶Biophysics Graduate Program, Harvard University

{riley_simmons-edler, ryan_badman, kanaka_rajan}@hms.harvard.edu

Abstract

Understanding the behavior of deep reinforcement learning (DRL) agents—particularly as task and agent sophistication increase—requires more than simple comparison of reward curves, yet standard methods for behavioral analysis remain underdeveloped in DRL. We apply tools from neuroscience and ethology to study DRL agents in a novel, complex, partially observable environment, ForageWorld, designed to capture key aspects of real-world animal foraging—including sparse, depleting resource patches, predator threats, and spatially extended arenas. We use this environment as a platform for applying joint behavioral and neural analysis to agents, revealing detailed, quantitatively grounded insights into agent strategies, memory, and planning. Contrary to common assumptions, we find that model-free RNN-based DRL agents can exhibit structured, planning-like behavior purely through emergent dynamics—without requiring explicit memory modules or world models. Our results show that studying DRL agents like animals—analyzing them with neuroethology-inspired tools that reveal structure in both behavior and neural dynamics—uncovers rich structure in their learning dynamics that would otherwise remain invisible. We distill these tools into a general analysis framework linking core behavioral and representational features to diagnostic methods, which can be reused for a wide range of tasks and agents. As agents grow more complex and autonomous, bridging neuroscience, cognitive science, and AI will be essential—not just for understanding their behavior, but for ensuring safe alignment and maximizing desirable behaviors that are hard to measure via reward. We show how this can be done by drawing on lessons from how biological intelligence is studied.

*Equal contributions

†Corresponding author

1 Introduction

As reinforcement learning (RL) researchers pursue more complex, naturalistic, and open-ended tasks [1], there is a growing need for tools to understand both the detailed behaviors and the neural dynamics of deep RL agents. This challenge mirrors long-standing problems in neuroscience, where researchers have developed extensive methods for joint behavioral and neural analysis in animals to understand biological intelligence. While machine learning (ML) has adopted neural representation analysis tools from neuroscience, mature behavioral analysis methods from neuroscience, cognitive science, and ethology [2, 3] remain underused in DRL. Open-ended RL tasks typically involve partial observability, memory, planning, and spatial reasoning—yet we still lack a systematic understanding of the strategies agents use to solve them [4–9]. Further, despite thousands of benchmarks—from video games [10] to 2D/3D gridworlds [11, 12]—there are no standard methods for analyzing or comparing agent behavior across tasks. RL evaluation still focuses primarily on reward curves and aggregate performance, which, while useful, offer limited insight into *how* agents solve structured tasks. The lack of behavioral analysis is a key bottleneck—limiting insight into why agents succeed or fail. Without understanding behavior, it’s difficult to diagnose algorithmic performance. Applied RL also stands to benefit from removing this bottleneck: more complex tasks require both better agent planning and deeper diagnostics to verify expected behavior [13–15].

Notably, this is not just a machine learning problem—interpreting behavior in partially observable, reward-driven environments is also a core challenge in neuroscience and cognitive science. Inspired by how animals are studied in neuroscience and ethology, we propose a behavior-first approach to analyzing DRL agents. This approach enables richer insight into both learned behavior and internal representations, drawing on joint behavioral-neural analysis tools that have proven essential for understanding biological intelligence. We ground this proposal in **ForageWorld**, a novel task suite in which agents forage and survive in procedurally generated, large-scale, partially observable environments with temporally extended dynamics. These pressures reflect the ecological tradeoffs animals encounter during natural foraging [16, 17], and align with calls to push foraging tasks in neuroAI beyond the traditional bandit paradigms [18–23]. ForageWorld builds on Craftax [12], combining complex naturalistic environment features with efficient training. We extend the GPU-accelerated Craftax with features such as patchy resources and structured threats to better capture key elements of animal foraging. These additions place strong demands on memory, spatial reasoning, and long-term planning, presenting a non-trivial challenge for modern DRL agents. In addition, foraging serves as a useful metaphor for core robotics tasks such as navigation, cleaning, and search and rescue [24], and underlies many popular RL benchmarks, including Minecraft and NetHack [25].

To support behavior-first evaluation in DRL, we introduce an analysis framework (Table 1) that maps key behavioral and neural targets to diagnostic methods. This framework is showcased by the results in this work and is reusable across neuroAI, interpretability, and broader RL applications.

Our key contributions are:

- (1) We develop **ForageWorld**, a naturalistic RL task suite with ecological structure—such as depleting resources, predators, partial observability, and open-ended survival goals—to study planning and memory in model-free agents.
- (2) We adapt tools from neuroscience and ethology—including path analysis, generalized linear models (GLMs), and recurrent state decoding—for **joint behavioral and neural analysis**, enabling biologically grounded interpretability of deep RL agents.
- (3) Using these tools, we show that model-free RNN agents exhibit structured, planning-like behavior through emergent dynamics, despite lacking explicit world models.
- (4) We release a reproducible pipeline and open-source codebase to support further research on internal representations and behavior.

We demonstrate these contributions in ForageWorld, analyzing trained agents across **five motif classes: exploration, decision-making, learning dynamics, internal representations, and generalization**.

The remainder of the paper reviews related work (Section 2), introduces our task and analyses (Section 3), and presents results (Section 4).

Table 1: *Understanding DRL agent behavior in open-ended tasks requires context-sensitive analysis frameworks, much like those used in animal neuroscience.* This table summarizes our reusable toolkit for probing memory, planning, exploration, and internal representations in DRL agents.

Analysis Target	Key Question	Methods Used
Goal Inference	What goals is the agent pursuing, and how are they shaped by experience?	Decision GLMs, in-context learning study, behavior phase segmentation
Memory Span	How far back in time can the agent remember?	RNN state decoding, memory module ablations
Planning Horizon	How far ahead does the agent plan?	Future decoding, auxiliary loss analysis for predictive objectives
Spatial Structure	How does the agent move through the environment?	Position occupancy entropy, tracking revisitation, path analysis
Network Capacity	Is the model too large or too small for the task difficulty?	Network size/pruning sweeps, comparing performance curves
Representations	What information is encoded in the agent’s internal states?	Decoding, GLMs, encoding profiles of task variables

2 Background: Open-Ended RL as a Bridge to NeuroAI

2.1 Open-Ended RL and the Need for Behavioral Analysis

Generalization in RL refers to the ability to transfer knowledge to new environments or tasks, and is critical for both practical applications and theoretical understanding in AI [26]. However, many RL algorithms struggle to generalize beyond their training tasks [27, 28]. This bottleneck has renewed interest in open-ended RL tasks that require rich exploration, subgoal discovery, and long-horizon reasoning [1, 29]. Open-ended RL tasks typically (i) scale with the learner, requiring skill composition in large state spaces, and (ii) incorporate naturalistic structure to better reflect real-world environments [29]. Many recent benchmarks embody this open-ended perspective—emphasizing compositional skills, dynamic goals, procedural generation, and expansive state spaces[1, 11–13, 30–39].

2.2 Joint Behavioral and Neural Analyses for Artificial Agents

To date, open-ended RL benchmarks have not drawn sustained interest from neuroscience or cognitive science, leading to a lack of deeper analysis frameworks in this research area. One reason is that few open-ended tasks reflect how animals explore, plan, and decide [3, 40]. Thus, there is a need for more naturalistic benchmarks—and for analysis pipelines that study agents with the same rigor applied to animal behavior and neural computation. Recent neuroscience and cognitive science work has primarily applied brain-inspired tools to ML systems for comparing internal representations [41–50]. This lack of behavioral emphasis mirrors limitations in how internal dynamics are typically studied—often in isolation from behavior, or without the tools needed to reveal structure. To date, most of this work has focused on representational similarity, not on joint analysis of behavior and underlying neural computation. A few recent exceptions—mostly outside RL—have begun to explore interpretability with a greater focus on behavior [47, 51, 52, 13, 53]. Researchers have also made calls for more interpretability work in the RL domain as well [54–56].

Despite this challenge, a key lesson from neuroscience is that neural activity alone is often insufficient to explain computation [3, 57–60]. For example, recent findings show that common similarity metrics can yield inconsistent conclusions [48, 61]. Several recent works argue that neuroAI should prioritize aligning behavior and internal dynamics in naturalistic settings, rather than relying on coarse similarity metrics in toy tasks [62, 60]. We thus adopt a behavior-first approach to studying DRL agents in the naturalistic, open-ended task ForageWorld, characterizing their strategies and internal representations using neuroscience-inspired tools—as is done in biological systems.

3 Methods

This section summarizes the environment, model architectures, training procedures, and analysis framework we used in this work. Full code for ForageWorld and our analysis is available at <https://github.com/RileySE/Craftax-Foraging/tree/foraging>, with additional details in the appendix.

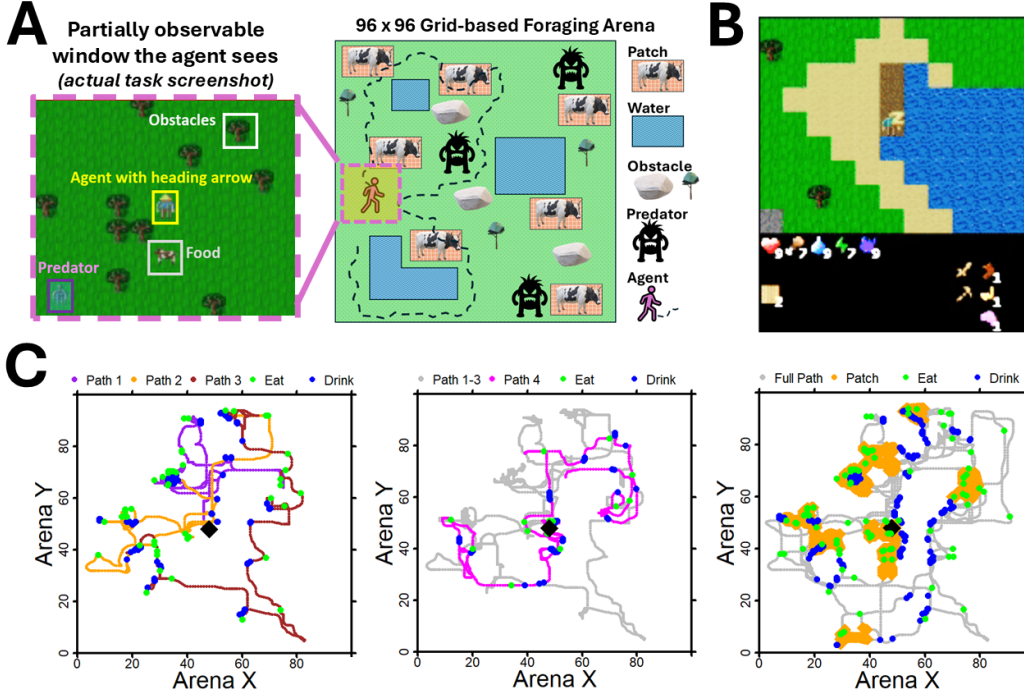


Figure 1: *Model-free agents exhibit structured exploration and revisitation behavior in an open-ended, partially observable environment.* (A) A 9×11 grid view (left, local observation window) shows the agent’s local observation, positioned within the full 96×96 environment (right). Cows (food) diffuse from spawn points and deplete when consumed; lakes (drink) are fixed and unlimited; predators intermittently pursue agents in view. (B) Agent observations include the visual window, inventory, and internal states (e.g., health, hunger, fatigue). This agent is sleeping—typically in corners to reduce predator contact. (C) Trajectories from a single episode: three early sequential exploratory paths (left), initial revisitation (middle), and full path with revisited patches in orange (right). The paths depicted belong to the same episode, and are directly consecutive from Path 1 through 4 in the behavioral time series logs for the episode. Path 1 starts at $t=0$ of the episode.

3.1 ForageWorld: Naturalistic Foraging with Internal States

We build on the Craftax benchmark [12], modifying it to support more naturalistic behavior under partial observability. ForageWorld introduces ecological constraints and uncertainty to better reflect real-world cognitive demands and enable deeper analysis. In particular, the need to localize food and avoid predators increases the memory demands relative to the original Craftax task. Cows have an arena-wide count limit, spawn at fixed locations, and diffuse over time—leading to temporary, localized food depletion and requiring the agent to leave and revisit patches periodically (Figure 1).

Drinking from procedurally placed water sources maintains hydration. Predators intermittently appear near food patches and pursue agents when within line of sight (Figure 1). Agents manage hunger, thirst, fatigue, and health. Fatigue recovers only through sleep, starvation and extreme thirst reduce health, and death occurs if the health variable hits zero. Sleep can only be performed when energy is below 50% of maximum. It immobilizes the agent while gradually restoring energy and, if applicable, health (Figure 1). Arenas are procedurally generated with randomized layouts, ensuring per-episode novelty. Unlike Craftax, our reward function is focused on agent survival: to receive reward, the agent must maintain its physiological resources (health, food, drink, energy) above half their maximum values. This objective correlates strongly with survival time and encourages the agent to maintain a buffer in resource levels—without promoting overconsumption or maxing out all variables. Specifically, the reward function is:

$$\mathcal{R}(s_t) = 0.1 \times (1 + \text{sign}(\text{health}_t - 5) + \text{sign}(\text{food}_t - 5) + \text{sign}(\text{drink}_t - 5) + \text{sign}(\text{energy}_t - 5)) \quad (1)$$

As defined in Equation 1, the agent’s reward is a function of physiological variables: it receives increasing positive reward for each key survival resource maintained above a threshold—health, food, drink, and energy.

For analysis, we log neural states, actions, and environment variables for each timestep (Table 2). This setup also supports held-out evaluation, enabling generalization assessment. Logged variables span both behavioral and neural axes from our analysis framework (Table 1).

3.2 RL Architectures and Model Variants

Our primary architecture was PPO with a GRU-based recurrent core (PPO-RNN), detailed in subsection A.6. In some experiments, we added an auxiliary objective to encourage interpretable spatial representations. In these experiments, the model also outputs its predicted position (x_t, y_t) relative to the first-timestep origin (path integration), via a small fully connected head that shares the RNN hidden state h_t with the policy and value heads. We train this auxiliary objective jointly with the PPO loss, using an L2 regression loss:

$$L_{\text{aux}} = \mathbb{E}_t[\|\hat{p}_t - p_t\|_2^2] \quad (2)$$

This term is weighted by a factor $\mathcal{W}_{\text{aux}}$ during optimization. This objective encourages RNN states to encode location, enabling spatial decoding (e.g., allocentric structure, planning horizon) and downstream analysis. Examining internal dynamics allows us to link behavior to emergent representations—paralleling methods in animal navigation studies.

To introduce biologically-inspired sparsity, we prune the weights of the RNN core using JaxPruner during training [63]. We targeted 90% sparsity across model layers, approximating biological brain sparsity, which ranges from 70% to 94% across species [64, 65]. We used magnitude-based pruning. Pruned agents performed comparably to unpruned ones and, in some cases, showed improved decoding of spatial variables from hidden states (Figure 16)—consistent with findings that sparse connectivity improves modularity and interpretability [66–68].

3.3 Decoding Allocentric Position and Planning Horizons

To evaluate memory and planning, we test whether hidden states encode past and future spatial positions. If location is internally represented, it may enable revisitation and planning. We first record hidden states h_t and ground-truth positions (x_t, y_t) throughout each episode. Then, at each timestep t , we train a decoder to predict allocentric displacement $Y_{t+\Delta t} = (\Delta x, \Delta y)$ between the current hidden state h_t and the agent’s position Δt steps in the past or future, using only $h_t \in \mathbb{R}^{512}$. This method tests whether the RNN encodes allocentric spatial information at various planning and memory horizons—analogueous to representations in animal navigation systems. Additionally, we train separate decoders for allocentric (relative to origin) and egocentric (relative to body orientation) coordinates to assess which coordinate frame the agent uses internally. We use ridge regression as the decoding model to preserve interpretability. Decoders were trained on the first 75% of each episode and evaluated on the final 25% to assess generalization. Accuracy is reported as RMSE, benchmarked against an average displacement-based baseline. All behavioral analyses were conducted in held-out test arenas, with weights frozen throughout, to ensure that behavior reflects generalization rather than ongoing learning. See subsection C.2 and subsection C.3 for full training details.

4 Results

4.1 Structured Exploration and Revisitation in Novel Environments

We begin by examining how agents explore novel environments and transition to revisitation strategies. Foraging has long served as a rich behavioral paradigm for studying goal-directed exploration in animal neuroscience [69, 70]. In rodents, foraging in novel, partially observable environments unfolds in distinct behavioral phases: mice initially engage in broad exploration, followed by direct and efficient revisitation of previously discovered locations once the environment is mapped [71].

Using trajectory visualizations, behavioral segmentation, and spatial entropy metrics, we find that ForageWorld induces structured, phase-like foraging behaviors in trained DRL agents—analogueous to those observed in biological systems. Standard performance metrics confirm that agents are competent at the task, showing long survival times (directly proportional to return) after training (Figure 2). However, our behavioral analysis reveals that DRL agents equipped only with RNNs—i.e., without explicit world models—develop in-context adaptive learning and exploration strategies that evolve across episodes.

Strikingly, after initially mapping a novel arena, agents shift into revisitation behaviors that reflect structured, multi-objective decision-making—mirroring phase transitions in animal foraging [71] and information seeking [72–74]. Their early exploration trajectories qualitatively resemble known

patterns of insect search behavior (Figure 8). Agents generated outwardly spiraling, azimuthally rotating loops from their starting position that gradually expanded to cover most of the arena (Figure 1). Even during revisitation phases, exploration persisted—most successful agents had covered the majority of the arena by the end of each episode (Figure 6). These extended trajectories also supported predator avoidance, enabling agents to discover shortcut paths between safe regions and resource patches (Figure 7). Similar patterns have been observed in ants and bees during nest displacement or early flights from the hive (Figure 8).

4.2 Multi-Objective Foraging and Strategic Patch Use

Having characterized agents’ exploration and revisitation behaviors, we next examine how they select among remembered locations. After initial exploration, agents displayed a clear shift to revisitation—returning to remembered patches via direct routes. Similar to rodents navigating partially observable mazes [71], our agents exhibit a rapid transition from broad exploration to targeted revisitation (Figure 1). This transition mimics phase-structured foraging as seen in rodents, while exploration resembles insect search behavior (Figure 8). Agents also periodically return to their starting position—potentially a reorientation strategy in the absence of an explicit spatial map.

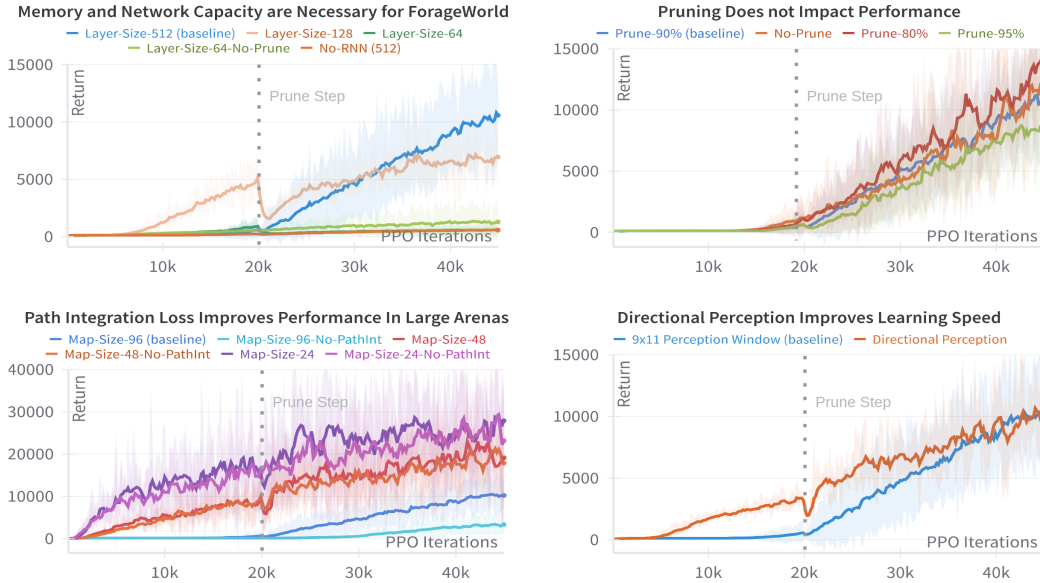


Figure 2: *Performance metrics and model ablations motivate deeper behavioral-neural analysis.* **(Top left)** ForageWorld requires memory and substantial network capacity to learn. Replacing the 512-unit RNN with a feedforward network significantly impairs performance, as does downsizing to a 64-unit network (550k parameters). A midsize 128-unit network also underperforms when pruned, suggesting that high capacity is needed to support sparsity. **(Top right)** Pruning does not degrade training performance but improves the spatial interpretability of internal representations (see Figure 16). **(Bottom left)** Removing the auxiliary path integration loss reduces performance in large arenas—but not small ones. Thus, the ability to predict current self-position appears critical for larger-scale navigation (see Figure 17 for representational effects). **(Bottom right)** Limiting perception to a forward-facing field of view improves early learning—perhaps due to reduced input complexity—but final performance conceals behavioral differences (see Figure 15). **(All plots)** Curves show the time-weighted EMA mean across 5 random seeds; shaded regions show one standard deviation.

Agents are not given an explicit list of patch locations. Instead, revisitation behavior emerges from recurrent state dynamics encoded during exploration and is reconstructed for analysis from logged trajectories. This result shows that model-free RNN agents can develop memory-guided, goal-directed behavior—such as selective revisitation—without explicit world models or symbolic memory structures. This memory dependence is further supported by a substantial performance gap between PPO agents with and without recurrence (Figure 2, Figure 11, Figure 10). Like animals that balance intake against physiological constraints (e.g., stomach volume, satiation), our agents are rewarded for maintaining low hunger, thirst, and fatigue—rather than for rapidly depleting resource patches. Under our baseline task parameters, expert agents ate at an average rate of 0.011 (approximately once

every 95 steps), drank at 0.034, and slept at 0.262. This emergent spacing enabled us to test whether agents made memory-guided, multi-objective decisions when selecting which patches to revisit. We analyzed patch revisitation choices and found that agents integrated spatial and task-relevant features in a multi-objective manner (Figure 3, Figure 7).

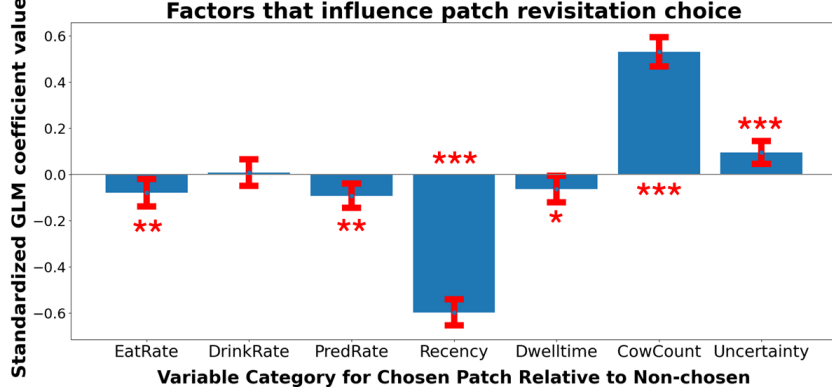


Figure 3: *Memory-guided, multi-objective revisitation strategies emerge in model-free agents without world models.* Generalized linear model (GLM) coefficients for patch history variables predicting an agent’s choice to revisit one patch over others. Choices are defined 50 timesteps before each patch eat event. From left to right: agents prefer patches with fewer prior eat actions (EatRate); show no preference for water proximity (DrinkRate); avoid patches with more predator encounters (PredRate); prefer patches visited more recently (Recency); show mild preference for longer dwell time (Dwelltime); prefer patches with more observed cows (CowCount); and prefer patches with higher prior position prediction error (Uncertainty). Significance levels: * < 0.05, ** < 0.01, *** < 0.001. Figure 13 has model output and VIF analysis. Error bars are 95% confidence intervals (CI).

4.3 Emergence of Behavioral Competencies Over Training

To examine how agent behavior evolves during training, we analyzed performance metrics and behavioral patterns across episodes. We find that ForageWorld supports staged learning: agents shift from undirected exploration to structured, goal-aligned strategies—transitions only partially visible in standard reward curves. These patterns suggest temporally organized skill acquisition. To test this, we tracked how different behaviors emerged and stabilized over the course of training. We used a panel of behavioral metrics to characterize the timeline of skill acquisition and identify which behaviors drove early versus late performance gains (Figure 4, Figure 9).

Early in training, PPO-RNN agents exhibited a “fishing” strategy—remaining stationary near the origin for extended periods. After approximately 20,000 training iterations, agents underwent an abrupt shift in behavior across multiple metrics. This transition included longer-distance travel during both early exploration and later revisitation, more strategic trade-offs between water and food collection, a sharp increase in tool-making frequency, and the emergence of gradual gains in predator defense. This shift marks the emergence of higher-level strategies in the training process.

To evaluate the contribution of specific components of the environment and model, we compared our PPO-RNN baseline to several variants with targeted objectives or capacities removed. One key ablation altered the agent’s field of view (FOV) to include only the forward grid rows, rather than the default Craftax FOV of a 9×11 rectangle centered on the agent. This change increased the need for information-seeking behavior and more closely resembled animal vision, which is predominantly front-facing. We found that front-FOV agents explored farther earlier in training and often engaged in more localized, dense search patterns (Figure 14, Figure 15). We also trained a different DRL objective, an off-policy RL architecture called parallelized Q-network (PQN) [75] (Appendix E). PQN-LSTM models performed comparably to the PPO-GRU baseline Figure 29, and with overall similar learning histories and pathing (Figure 31, Figure 30), but with the noticeable behavioral differences of much lower rates of predator killing (i.e. reducing predator HP to 0) (Figure 31). Thus, PPO-GRU seems to converge to a predator fighting strategy, while PQN-LSTM converges to predator evasion.

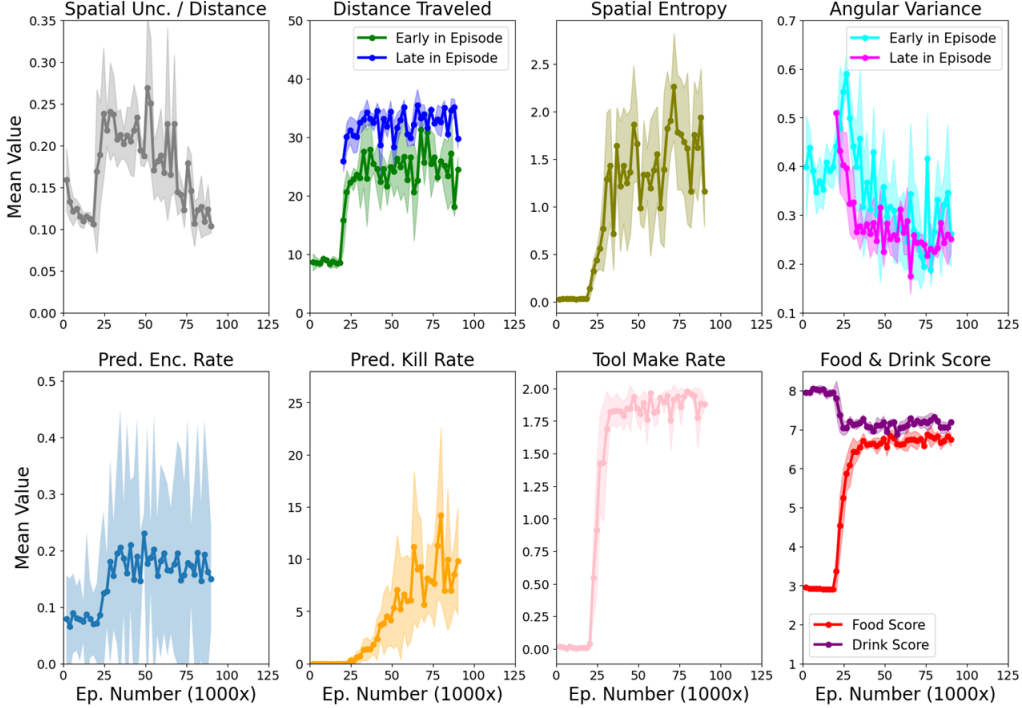


Figure 4: *Distinct behavioral competencies emerge over training, with early gains in exploration followed by refinement of survival strategies.* Training dynamics show staged acquisition of task-aligned skills. Early learning emphasizes spatial exploration and arena coverage, while later training refines predator response, tool use, and pathing patterns. Metrics include: spatial uncertainty (normalized by distance from origin); distance from origin (early vs. late, meaning before vs. after the first 1500 timesteps in an episode); state occupancy entropy of agent position; angular orientation variance across 250 timestep intervals; predator field-of-view exposure; tool-making rate (1 = one tool crafted, 2 = both); and food/water satiation levels. Error bars are 95% CI. These plots can be compared to the generally linear performance gains seen in survival times across training in Figure 2.

4.4 Recurrent State Representations Support Memory and Planning

We next ask whether agents encode task-relevant structure in their internal representations. Using linear decoders trained on recurrent states, we find that agents represent allocentric position across time—revealing implicit memory and planning capacity not apparent from performance metrics alone. Thus, our framework can (1) decode the memory and planning timescales represented in agent RNN states, and (2) identify the internal encoding structure that supports foraging behavior.

A growing body of work in RL has examined whether model-free RNN agents can perform implicit planning without the explicit world models typically used for planning tasks [76–78]. This question also arises in neuroscience, where insects like ants and bees show planning-like behavior despite lacking mammalian hippocampal-entorhinal map systems [79–81]. Our analyses show that DRL agents revisit patches based on multiple factors such as proximity, predator history, and position prediction error (Figure 3, Figure 7). An agent’s position relative to the origin can also be decoded from its current RNN state—up to 50–100 timesteps into the past and future—supporting its capacity for long-horizon memory and planning (Figure 5). Furthermore, approximately 100/512 neurons were found to be position-sensitive at the single neuron level using generalized linear model (GLM) analysis. The neural activity response to position change (GLM coefficient magnitude) increased with radial distance from the agent’s origin as well, suggesting a distance accumulation circuit that modulates origin-revisitation behavior (Appendix D).

Decoding accuracy improved over the course of training, consistent with the agent acquiring more stable and structured internal representations. In contrast, the agent’s position relative to itself (egocentric orientation) could not be reliably decoded. This preference for origin-relative encoding may help explain why agents frequently revisit the arena origin during later phases (Figure 1).

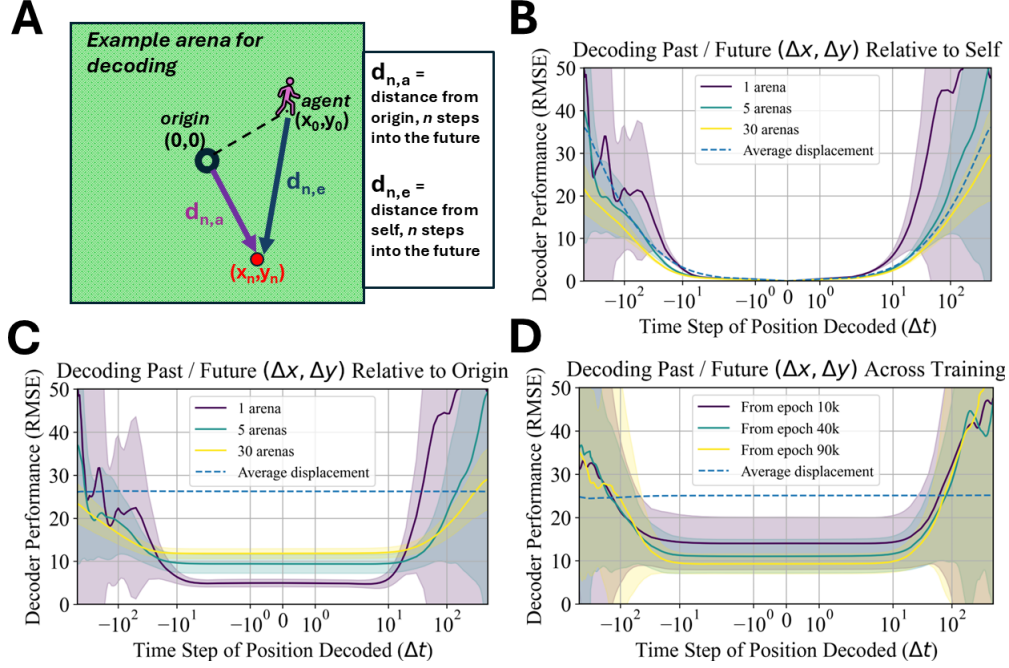


Figure 5: *RNN states encode allocentric position and temporal structure, revealing emergent memory and planning capacity in model-free agents.* (A) A single decoding model was trained per agent to test whether spatial information was encoded in allocentric (relative to origin) or egocentric (relative to agent) coordinates. (B) Late-training decoding performance for egocentric distance remained at chance level across models. (C) Allocentric position could be decoded above chance up to approximately 50–100 timesteps into the past and future depending on the run. Training arena count per decoder was varied to show that models did not rely on arena-specific cues. (D) Allocentric decoding improved over training. Error bars are larger in (D) because decoding was limited to timesteps 1000–6000 per arena, due to shorter survival in early-training (10k epoch) agents. All plots use average displacement per timestep as a chance baseline. Error bars reflect 95% CI.

4.5 Generalization and Modularity in Recurrent Dynamics

We next examine whether the internal representations learned by DRL agents generalize across environments and exhibit modular substructure. These properties are important not only for performance, but also for interpretability—revealing which units encode what, and under what conditions. Decoding performance was moderately higher in pruned (sparse) networks than in fully connected ones (Figure 16), confirming more efficient information encoding in sparse architectures—consistent with recent findings [66–68]. Removing the auxiliary objective for position prediction eliminated above-chance decoding performance (Figure 17), suggesting that the associated task performance loss (Figure 2) stems from the model’s inability to encode position information using the PPO objective alone. These results suggest that the auxiliary objective improved both agent performance and the interpretability of its internal representations—a dual benefit revealed only through neural analysis. Past and future position encoding relied on overlapping but functionally modular subpopulations of RNN units, which diverged more clearly at longer time horizons (Figure 18). Notably, a single decoding model generalizes across episodes—predicting past and future position from RNN states recorded in many different arenas for the same agent. This cross-episode generalization likely reflects the consistent spatial structure of the environment: although arena configurations vary, the 96×96 grid layout and global orientation are preserved, providing a stable allocentric reference frame.

In summary, model-free RNN agents trained in open-ended tasks can develop structured internal representations that support memory, planning, and spatial generalization—without relying on explicit world models. ForageWorld induces naturalistic behavioral and neural motifs in DRL agents, and our analysis framework enables these motifs to be detected and interpreted using behavior-first tools.

5 Discussion

We provide two main findings: (1) ForageWorld induces behavior–neural motifs relevant to planning and memory, and (2) our toolkit reveals these motifs in ways that simple performance metrics cannot.

We also help challenge the common view in neuroscience that sophisticated planning and memory require explicit world models, mammalian-like brains, or symbolic memory [80, 82], a topic that is of growing interest in computer science as well [83, 84]. Despite being trained with model-free objectives, RNN-based agents in our study exhibit revisitation, uncertainty-driven exploration, and allocentric encoding—behaviors typically associated with model-based agents. While recent work suggests that temporal credit assignment and environment interaction can give rise to planning-like behavior in RNNs [76–78], our study provides the first systematic evidence that such behavior can emerge in complex, naturalistic environments without world models. Thus, internal interpretability is shaped not just by training outcomes, but by design choices that constrain or promote structure.

We also find that architectural choices such as recurrence, pruning, and auxiliary losses significantly influence both performance and interpretability. Pruning improved decoding accuracy (Figure 16) while leaving performance and behavior substantively unchanged (Figure 2); the auxiliary position prediction objective enhanced predator-related behavior and spatial encoding structure (Figure 4, Figure 11, Figure 17) while also improving performance (Figure 2). This win-win fits with observations from neuroscience—biological agents do not simply maximize reward, but also engage in latent path learning and other subgoals [82]. Thus, there are benefits for RL researchers to embracing this complexity of learning objectives and bridging the gaps between artificial and natural intelligence.

These findings also speak to neuroscience research on small nervous systems. Insects, for example, navigate using structured internal cues [80, 85]. The fact that similar behaviors emerge in DRL agents with only a few hundred recurrent units suggests that such agents may offer useful models for evaluating the cognitive capacities of small nervous systems. Taken together with behavioral consistency across arenas, the ability of a single decoding model to generalize across episodes likely reflects a stable, compass-like encoding of space—mirroring strategies used by insects and other animals to maintain orientation using global cues such as magnetic fields or polarized light [86–88]. This encoding strategy may explain how DRL agents reliably plan navigation across environments with differing local layouts.

More broadly, we dispute the idea that DRL models are too complex for neuroscience-style analysis. The brain is at least as complex—and likely more so—than current DRL agents [45]. If neuroscience tools cannot explain DRL agents with full behavioral and neural access, their utility for understanding the brain is limited. Testing these tools in complex simulation environments also addresses practical challenges in naturalistic neuroscience, including low statistical power, high data costs, and limited experimental control [89]. By generating high-volume, ecologically grounded datasets with joint neural-behavior logging, we provide a platform for evaluating brain analysis techniques. Our diagnostic framework (Table 1) offers a guide for interpreting agent behavior and internal dynamics across diverse RL tasks. We hope this framework serves both as a roadmap for future neuroAI work and as a foundation for more interpretable RL evaluation.

Limitations & Broader Impacts The 96×96 grid size limit in Craftax prevents us from evaluating how our results generalize to even larger arenas. A second limitation is that we could not log the precise locations of all trees, rocks, and lakes, which limited our ability to study landmark-based navigation in detail. A third limitation is that incorporating a wider range of RL architectures with full behavioral-neural logging is currently non-trivial in this environment, which is why we focused on PPO- and PQN-based models with shared components. We believe this work is likely to have a positive societal impact by fostering closer collaboration between neuroscientists, cognitive scientists, ethologists, and RL researchers in the analysis of deep RL agents—advancing all fields through shared insight into the structure and function of learned behavior.

6 Acknowledgments

We thank Albert Lee and members of the Rajan Lab for the helpful comments and feedback during the course of the project. We also thank the research assistants Yang Wu and Yunxin Wu for their help during revisions. R.S., R.P.B. and K.R. were collectively funded by the NIH (RF1DA056403, U01NS136507), James S. McDonnell Foundation (220020466), Simons Foundation (Pilot Extension-00003332-02, McKnight Endowment Fund, CIFAR Azrieli Global Scholar Program, NSF (2046583), Harvard Medical School Neurobiology Lefler Small Grant Award, Harvard Medical School Dean’s Innovation Award, and Harvard Medical School Leonard and Isabelle Goldenson Fellowship. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. F.B.B. received support from Fulbright Program, Jansons Legat, and the Norwegian State Educational Loan Fund.

References

- [1] OEL Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.
- [2] John W Krakauer, Asif A Ghazanfar, Alex Gomez-Marin, Malcolm A MacIver, and David Poeppel. Neuroscience needs behavior: correcting a reductionist bias. *Neuron*, 93(3):480–490, 2017.
- [3] Yael Niv. The primacy of behavioral research for understanding the brain. *Behavioral Neuroscience*, 135(5):601, 2021.
- [4] Thomas Wolbers and Jan M Wiener. Challenges for identifying the neural mechanisms that support spatial navigation: the impact of spatial scale. *Frontiers in human neuroscience*, 8: 571, 2014.
- [5] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017.
- [6] Arne D Ekstrom, Derek J Huffman, and Michael Starrett. Interacting networks of brain regions underlie human spatial navigation: a review and novel synthesis of the literature. *Journal of neurophysiology*, 118(6):3328–3344, 2017.
- [7] Kai Zhu and Tao Zhang. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*, 26(5):674–691, 2021.
- [8] Eloy Parra-Barrero, Sandhiya Vijayabaskaran, Eddie Seabrook, Laurenz Wiskott, and Sen Cheng. A map of spatial navigation for neuroscience. *Neuroscience & Biobehavioral Reviews*, 152:105200, 2023.
- [9] Tamir Eliav, Shir R. Maimon, Ayelet Sarel, Shaked Palgi, Liora Las, and Nachum Ulanovsky. Fragmented replay of very large environments in the hippocampus of bats. *Cell*, 2025. doi: 10.1016/j.cell.2025.05.024.
- [10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [11] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.
- [12] Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Jackson, Samuel Coward, and Jakob Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning, 2024. URL <https://arxiv.org/abs/2402.16801>.

- [13] Stephen Chung, Ivan Anokhin, and David Krueger. Thinker: Learning to plan and act. *Advances in Neural Information Processing Systems*, 36:22896–22933, 2023.
- [14] Christopher R Fetsch. The importance of task design and behavioral control for understanding the neural basis of cognitive functions. *Current opinion in neurobiology*, 37:16–22, 2016.
- [15] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. *arXiv preprint arXiv:2111.01674*, 2021.
- [16] Nils Kolling, Marco Wittmann, and Matthew FS Rushworth. Multiple neural mechanisms of decision making and their competition under changing risk pressure. *Neuron*, 81(5):1190–1202, 2014.
- [17] Alexander T Lai, German Espinosa, Gabrielle E Wink, Christopher F Angeloni, Daniel A Dombeck, and Malcolm A MacIver. A robot-rodent interaction arena with adjustable spatial complexity for ethologically relevant behavioral studies. *Cell reports*, 43(2), 2024.
- [18] Sam Hall-McMaster and Fabrice Luyckx. Revisiting foraging approaches in neuroscience. *Cognitive, Affective, & Behavioral Neuroscience*, 19(2):225–230, 2019.
- [19] Juliano Morimoto. Foraging decisions as multi-armed bandit problems: Applying reinforcement learning algorithms to foraging data. *Journal of theoretical biology*, 467:48–56, 2019.
- [20] Charlotte N Boccara, Michele Nardin, Federico Stella, Joseph O’Neill, and Jozsef Csicsvari. The entorhinal cognitive map is attracted to goals. *Science*, 363(6434):1443–1447, 2019.
- [21] Ryan Paul Badman, Thomas Trenholm Hills, and Rei Akaishi. Multiscale computation and dynamic attention in biological and artificial intelligence. *Brain sciences*, 10(6):396, 2020.
- [22] Wan-Chen Jiang, Shengjin Xu, and Joshua T Dudman. Hippocampal representations of foraging trajectories depend upon spatial context. *Nature neuroscience*, 25(12):1693–1705, 2022.
- [23] Laura L Grima, Hannah Haberkern, Rishika Mohanta, Mai M Morimoto, Adithya E Rajagopalan, and Emma V Scholey. Foraging as an ethological framework for neuroscience. *Trends in Neurosciences*, 2025.
- [24] Alan F. T. Winfield. *Towards an Engineering Science of Robot Foraging*, pages 185–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00644-9. doi: 10.1007/978-3-642-00644-9_16. URL https://doi.org/10.1007/978-3-642-00644-9_16.
- [25] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.
- [26] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.
- [27] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.
- [28] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pages 1282–1289. PMLR, 2019.
- [29] Minqi Jiang, Tim Rocktäschel, and Edward Grefenstette. General intelligence requires rethinking exploration. *Royal Society Open Science*, 10(6):230539, 2023.
- [30] Hua Wei, Jingxiao Chen, Xiyang Ji, Hongyang Qin, Minwen Deng, Siqin Li, Liang Wang, Weinan Zhang, Yong Yu, Liu Linc, et al. Honor of Kings arena: an environment for generalization in competitive reinforcement learning. *Advances in Neural Information Processing Systems*, 35:11881–11892, 2022.

- [31] Emma Stensby Norstein, Kai Olav Ellefsen, and Kyrre Glette. Open-ended search for environments and adapted agents using map-elites. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 651–666. Springer, 2022.
- [32] Shaofei Cai, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13734–13744, 2023.
- [33] Mikayel Samvelyan, Akbir Khan, Michael Dennis, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Roberta Raileanu, and Tim Rocktäschel. Maestro: Open-ended environment design for multi-agent reinforcement learning. *arXiv preprint arXiv:2303.03376*, 2023.
- [34] Chi Zhang, Penglin Cai, Yuhui Fu, Haoqi Yuan, and Zongqing Lu. Creative agents: Empowering agents with imagination for creative tasks. *arXiv preprint arXiv:2312.02519*, 2023.
- [35] Tristan Tomilin, Meng Fang, Yudi Zhang, and Mykola Pechenizkiy. COOM: A game benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36:67794–67832, 2023.
- [36] Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Skill reinforcement learning and planning for open-world long-horizon tasks. *arXiv preprint arXiv:2303.16563*, 2023.
- [37] Karolis Jucys, George Adamopoulos, Mehrab Hamidi, Stephanie Milani, Mohammad Reza Samsami, Artem Zholus, Sonia Joseph, Blake Richards, Irina Rish, and Özgür Şimşek. Interpretability in action: Exploratory analysis of VPT, a Minecraft agent, 2024. URL <https://arxiv.org/abs/2407.12161>.
- [38] Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. Xland-minigrid: Scalable meta-reinforcement learning environments in jax. *Advances in Neural Information Processing Systems*, 37:43809–43835, 2024.
- [39] Guy Davidson, Graham Todd, Julian Togelius, Todd M Gureckis, and Brenden M Lake. Goals as reward-producing programs. *Nature Machine Intelligence*, 7(2):205–220, 2025.
- [40] Toby Wise, Kara Emery, and Angela Radulescu. Naturalistic reinforcement learning. *Trends in Cognitive Sciences*, 28(2):144–158, 2024.
- [41] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018.
- [42] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.
- [43] Martin Schrimpf, Jonas Kubilius, Michael J Lee, N Apurva Ratan Murty, Robert Ajemian, and James J DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 108(3):413–423, 2020.
- [44] Soma Nonaka, Kei Majima, Shuntaro C Aoki, and Yukiyasu Kamitani. Brain hierarchy score: Which deep neural networks are hierarchically brain-like? *IScience*, 24(9), 2021.
- [45] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021.
- [46] Mitchell Ostrow, Adam Eisen, Leo Kozachkov, and Ila Fiete. Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. *Advances in Neural Information Processing Systems*, 36:33824–33837, 2023.

- [47] Ida Momennejad, Hosein Hasanbeig, Felipe Vieira Frujeri, Hiteshi Sharma, Nebojsa Jojic, Hamid Palangi, Robert Ness, and Jonathan Larson. Evaluating cognitive maps and planning in large language models with CogEval. *Advances in Neural Information Processing Systems*, 36:69736–69751, 2023.
- [48] Ansh Soni, Sudhanshu Srivastava, Konrad Kording, and Meenakshi Khosla. Conclusions about neural network to brain alignment are profoundly impacted by the similarity measure. *bioRxiv*, pages 2024–08, 2024.
- [49] Han Wang, Erfan Miah, Martha White, Marlos C Machado, Zaheer Abbas, Raksha Kumaraswamy, Vincent Liu, and Adam White. Investigating the properties of neural network representations in reinforcement learning. *Artificial Intelligence*, 330:104100, 2024.
- [50] Wilka Carvalho, Momchil S Tomov, William de Cothi, Caswell Barry, and Samuel J Gershman. Predictive representations: Building blocks of intelligence. *Neural Computation*, 36(11): 2225–2298, 2024.
- [51] Demet Yesiltepe, P Fernández Velasco, Antoine Coutrot, A Ozbil Torun, Jan M Wiener, Christoph Holscher, Michael Hornberger, R Conroy Dalton, and Hugo J Spiers. Entropy and a sub-group of geometric measures of paths predict the navigability of an environment. *Cognition*, 236:105443, 2023.
- [52] Jaedong Hwang, Zhang-Wei Hong, Eric Chen, Akhilan Boopathy, Pulkit Agrawal, and Ila Fiete. Grid cell-inspired fragmentation and recall for efficient map building. *arXiv preprint arXiv:2307.05793*, 2023.
- [53] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.
- [54] Drew H Abney, Caitlin M Fausey, Catalina Suarez-Rivera, and Catherine S Tamis-LeMonda. Advancing a temporal science of behavior. *Trends in Cognitive Sciences*, 2025.
- [55] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International conference on machine learning*, pages 5045–5054. PMLR, 2018.
- [56] Zelei Cheng, Jiahao Yu, and Xinyu Xing. A survey on explainable deep reinforcement learning. *arXiv preprint arXiv:2502.06869*, 2025.
- [57] Jerry Hirsch. Nothing in neurobiology makes sense—except in the light of behaviour. *Contemporary Psychology*, 1986.
- [58] Eric Jonas and Konrad Paul Kording. Could a neuroscientist understand a microprocessor? *PLoS computational biology*, 13(1):e1005268, 2017.
- [59] Robert Geirhos, Roland S Zimmermann, Blair Bilodeau, Wieland Brendel, and Been Kim. Don’t trust your eyes: on the (un) reliability of feature visualizations. *arXiv preprint arXiv:2306.04719*, 2023.
- [60] Drew Linsley, Pinyuan Feng, and Thomas Serre. Better artificial intelligence does not mean better models of biology, 2025. URL <https://arxiv.org/abs/2504.16940>.
- [61] Tianyu Cui, Yogesh Kumar, Pekka Marttinen, and Samuel Kaski. Deconfounded representation similarity for comparison of neural networks. *Advances in Neural Information Processing Systems*, 35:19138–19151, 2022.
- [62] Rosa Cao and Daniel Yamins. Explanatory models in neuroscience, part 2: Functional intelligibility and the contravariance principle. *Cognitive Systems Research*, 85:101200, 2024.
- [63] Joo Hyung Lee, Wonpyo Park, Nicole Elyse Mitchell, Jonathan Pilault, Johan Samir Obando Ceron, Han-Byul Kim, Namhoon Lee, Elias Frantar, Yun Long, Amir Yazdanbakhsh, et al. Jaxpruner: A concise library for sparsity research. In *Conference on Parsimony and Learning*, pages 515–528. PMLR, 2024.

- [64] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature communications*, 10(1):1017, 2019.
- [65] Xin-Jie Zhang, Jack Murdoch Moore, Gang Yan, and Xiang Li. Universal structural patterns in sparse recurrent neural networks. *Communications Physics*, 6(1):243, 2023.
- [66] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.
- [67] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, pages 14014–14024, 2019.
- [68] Neehal Tumma, Mathias Lechner, Noel Loo, Ramin Hasani, and Daniela Rus. Leveraging low-rank and sparse recurrent connectivity for robust closed-loop control. *arXiv preprint arXiv:2310.03915*, 2023.
- [69] Benjamin Y Hayden. Economic choice: the foraging perspective. *Current Opinion in Behavioral Sciences*, 24:1–6, 2018.
- [70] Thomas T Hills. Animal foraging and the evolution of goal-directed cognition. *Cognitive science*, 30(1):3–41, 2006.
- [71] Matthew Rosenberg, Tony Zhang, Pietro Perona, and Markus Meister. Mice in a labyrinth show rapid learning, sudden insight, and efficient exploration. *Elife*, 10:e66175, 2021.
- [72] P-Y Oudeyer, Jacqueline Gottlieb, and Manuel Lopes. Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. *Progress in brain research*, 229:257–284, 2016.
- [73] Jennifer J Bussell, Ryan P Badman, Christian D Márton, Ethan S Bromberg-Martin, LF Abbott, Kanaka Rajan, and Richard Axel. Representations of information value in mouse orbitofrontal cortex during information seeking. *bioRxiv*, pages 2023–10, 2023.
- [74] Ilya E Monosov. Curiosity: primate neural circuits for novelty and information seeking. *Nature Reviews Neuroscience*, 25(3):195–208, 2024.
- [75] Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *arXiv preprint arXiv:2407.04811*, 2024.
- [76] Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free RL can be a strong baseline for many POMDPs. *arXiv preprint arXiv:2110.05038*, 2021.
- [77] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.
- [78] Stephen Chung, Scott Niekum, and David Krueger. Predicting future actions of reinforcement learning agents. *arXiv preprint arXiv:2410.22459*, 2024.
- [79] Rüdiger Wehner and Mandyam V Srinivasan. Searching behaviour of desert ants, genus *cataglyphis* (formicidae, hymenoptera). *Journal of comparative physiology*, 142:315–338, 1981.
- [80] Lars Chittka. *The mind of a bee*. Princeton University Press, 2022.
- [81] Marilia Freire, Antonio Bollig, and Markus Knaden. Absence of visual cues motivates desert ants to build their own landmarks. *Current Biology*, 33(13):2802–2805, 2023.
- [82] Leo Clement, Sebastian Schwarz, and Antoine Wystrach. Latent learning without map-like representation of space in navigating ants. *bioRxiv*, pages 2024–08, 2024.

- [83] Thomas Bush, Stephen Chung, Usman Anwar, Adrià Garriga-Alonso, and David Krueger. Interpreting emergent planning in model-free reinforcement learning. *arXiv preprint arXiv:2504.01871*, 2025.
- [84] Antoine Dedieu, Joseph Ortiz, Xinghua Lou, Carter Wendelken, Wolfgang Lehrach, J Swaroop Guntupalli, Miguel Lazaro-Gredilla, and Kevin Patrick Murphy. Improving transformer world models for data-efficient RL. *arXiv preprint arXiv:2502.01591*, 2025.
- [85] Sercan Sayin, Einat Couzin-Fuchs, Inga Petelski, Yannick Günzel, Mohammad Salahshour, Chi-Yu Lee, Jacob M. Graving, Liang Li, Oliver Deussen, Gregory A. Sword, and Iain D. Couzin. The behavioral mechanisms governing collective motion in swarming locusts. *Science*, 387(6737):995–1000, 2025. doi: 10.1126/science.adq7832.
- [86] Rüdiger Wehner. Polarized-light navigation by insects. *Scientific American*, 235(1):106–115, 1976.
- [87] Roswitha Wiltschko. *Magnetic orientation in animals*, volume 33. Springer Science & Business Media, 2012.
- [88] Stanley Heinze. Polarized-light processing in insect brains: recent insights from the desert locust, the monarch butterfly, the cricket, and the fruit fly. *Polarized light and polarization vision in animal sciences*, pages 61–111, 2014.
- [89] Samuel A Nastase, Ariel Goldstein, and Uri Hasson. Keep it real: rethinking the primacy of experimental control in cognitive neuroscience. *NeuroImage*, 222:117254, 2020.
- [90] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [91] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [92] Joel Z Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*, 2019.
- [93] Minqi Jiang, Michael Dennis, Edward Grefenstette, and Tim Rocktäschel. minimax: Efficient baselines for autocurricula in jax. *arXiv preprint arXiv:2311.12716*, 2023.
- [94] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source LLMs. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6710–6717. IEEE, 2025.
- [95] Joshua A Cullen, Caroline L Poli, Robert J Fletcher Jr, and Denis Valle. Identifying latent behavioural states in animal movement with m4, a nonparametric bayesian method. *Methods in Ecology and Evolution*, 13(2):432–446, 2022.
- [96] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. ctree: Conditional inference trees. *The comprehensive R archive network*, 8:1–34, 2015.
- [97] Juliet L Osborne, Alan Smith, Suzanne J Clark, Don R Reynolds, Mandy C Barron, Ka S Lim, and Andy M Reynolds. The ontogeny of bumblebee flight trajectories: from naïve explorers to experienced foragers. *Plos one*, 8(11):e78681, 2013.
- [98] Edoardo Balzani, William Broderick, Guillaume Viejo, and Alex Williams. NEural MOdelS, a statistical modeling framework for neuroscience. <https://github.com/flatironinstitute/nemos>, 2025.
- [99] Charles V Vorhees and Michael T Williams. Assessing spatial learning and memory in rodents. *ILAR journal*, 55(2):310–332, 2014.

- [100] Edvard I Moser, Yasser Roudi, Menno P Witter, Clifford Kentros, Tobias Bonhoeffer, and May-Britt Moser. Grid cells and cortical representation. *Nature Reviews Neuroscience*, 15(7): 466–481, 2014.
- [101] Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.
- [102] Peter M Todd and Thomas T Hills. Foraging in mind. *Current Directions in Psychological Science*, 29(3):309–315, 2020.
- [103] Luca Ambrogioni and H Freyja Ólafsdóttir. Rethinking the hippocampal cognitive map as a meta-learning computational module. *Trends in Cognitive Sciences*, 27(8):702–712, 2023.
- [104] Payam Piray and Nathaniel Douglass Daw. Reconciling flexibility and efficiency: Medial entorhinal cortex represents a compositional cognitive map. *bioRxiv*, pages 2024–05, 2024.
- [105] Jozsef Csicsvari, Joseph O’Neill, Kevin Allen, and Timothy Senior. Place-selective firing contributes to the reverse-order reactivation of ca1 pyramidal cells during sharp waves in open-field exploration. *European Journal of Neuroscience*, 26(3):704–716, 2007.
- [106] H Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.
- [107] Eric L Denovellis, Anna K Gillespie, Michael E Coulter, Marielena Sosa, Jason E Chung, Uri T Eden, and Loren M Frank. Hippocampal replay of experience at real-world speeds. *Elife*, 10:e64505, 2021.
- [108] Payam Piray and Nathaniel D Daw. Linear reinforcement learning in planning, grid fields, and cognitive control. *Nature communications*, 12(1):4942, 2021.
- [109] Kiah Hardcastle, Surya Ganguli, and Lisa M Giocomo. Environmental boundaries as an error correction mechanism for grid cells. *Neuron*, 86(3):827–839, 2015.
- [110] Kiah Hardcastle, Niru Maheswaranathan, Surya Ganguli, and Lisa M Giocomo. A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. *Neuron*, 94(2): 375–387, 2017.
- [111] Marielena Sosa, Mark H Plitt, and Lisa M Giocomo. Hippocampal sequences span experience relative to rewards. *bioRxiv*, 2023.
- [112] James CR Whittington, Joseph Warren, and Timothy EJ Behrens. Relating transformers to models and neural representations of the hippocampal formation. *arXiv preprint arXiv:2112.04035*, 2021.
- [113] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705): 429–433, 2018.

7 Appendix

This Appendix provides additional details on the ForageWorld environment and agent training setup, supplementing the descriptions in subsection 3.1 and subsection 3.2 of the main text. It includes key environment parameters, procedural generation constraints, agent sensory input, and logging infrastructure used for downstream analysis. We also provide supplementary analyses and figures that expand on the main results, including trajectory examples, revisitation models, training dynamics, ablations, and neural decoding structure.

A ForageWorld: Task and Training Details

We designed ForageWorld to balance between task complexity, transparency to analysis, training efficiency (using JAX GPU acceleration), biological plausibility in navigational planning computations, and connection to joint RL and recurrent memory architectures that are of particular interest to neuroscientists. More complex deep RL tasks exist, but many such tasks (for example, Minecraft) are much more difficult to study and in particular more difficult to instrument for behavior analysis. In addition, the increased computational cost of experiments in such environments is prohibitive to some statistical analysis, and many computational neuroscientists in particular do not have access to compute resources sufficient to run experiments in such complex environments. Relative to common neuroscience tasks used to study foraging, e.g. n-armed bandit tasks which lack spatial or temporal components, ForageWorld pushes the complexity boundary on the neuro-side of neuroAI while still remaining tractable for neuroscience tools and neuroscientists to study. Further, ForageWorld as it exists is not trivial to learn—our agents require billions of timesteps of experience to train to competency on the task, and Craftax, the benchmark which we build upon to develop ForageWorld, has not been solved by any agent or architecture to date.

In the following section, we expand here on the ForageWorld environment and agent training setup described in subsection 3.1 and subsection 3.2 of the main text. This includes key environment parameters, procedural generation constraints, agent sensory input, and logging infrastructure used for downstream analysis.

A.1 Procedural Arena Generation

Each arena is procedurally generated at the start of an episode using a fixed-size 96×96 grid. Spawn points for cows (food) and lakes (drink) are initialized in random patches using Perlin noise, providing episodic variability alongside naturalistic structure. Obstacles are added to break line-of-sight and create navigational constraints. The agent begins at the center of the arena.

A.2 Episode Structure and Termination Conditions

Each episode runs for a maximum of 100,000 timesteps or ends earlier if the agent’s health drops to zero—which is the typical outcome, even for highly competent agents. Agents earn reward by maintaining internal homeostasis—balancing hunger, thirst, and fatigue—and avoiding predator encounters. The reward function is defined in subsection 3.1.

A.3 Sensory Input and State Representation

At each timestep, the agent receives a 9×11 grid-centered egocentric view of the environment (see Figure 1), along with an inventory vector containing current health, satiation levels (food and water), fatigue, and any collected items. This full observation is encoded by a feedforward neural net layer before being passed to the recurrent network.

A.4 Logging Infrastructure

To support joint behavioral and neural analysis, we log a wide range of signals at each timestep, including the full recurrent hidden state, internal reward components, movement trajectories, environmental state, and agent decisions. The complete logging schema is shown in Table 2. These logs enable replay, ablation, and representational analysis across multiple analytic pipelines.

A.5 Training Hyperparameters

Agents were trained using Proximal Policy Optimization (PPO) [90] with Generalized Advantage Estimation (GAE). Each agent trained for 3 billion timesteps, resulting in a variable number of episodes—typically in the hundreds of thousands—depending on competence and learning speed. Episodes were capped at 100,000 timesteps, though this cap was rarely reached in practice. The PPO rollout horizon was 64, with a minibatch size of 8192 and a learning rate of 0.00025. We used the Adam optimizer with default momentum parameters. Additional hyperparameters are provided in Table 3. Policy and value losses were weighted using standard PPO coefficients (see Table 3). For stability, gradient norms were clipped to 1.0.

Each training run used a single GPU (NVIDIA A100 or H100) and typically converged within 24–48 hours, depending on configuration. While memory usage varied by configuration, most experiments were feasible on GPUs with at least 24 GB of memory. Logging occurred every 2048 PPO iterations (roughly every 134 million timesteps), and the best-performing checkpoints were selected based on held-out arena performance.

Together, these settings define a procedurally rich, ecologically grounded RL training setup that supports the emergence of complex behaviors and structured internal representations. The design balances task realism, computational feasibility, and interpretability—enabling the behavioral and neural analyses presented in this work.

A.6 Model Architecture Details for PPO-RNN

Our primary architecture combined Proximal Policy Optimization (PPO) [90] with a recurrent neural network, using Generalized Advantage Estimation (GAE) for stability. PPO optimized a clipped surrogate objective that balanced advantage maximization with policy stability:

$$L_{\text{clip}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (3)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$ is the probability ratio between current and previous policies, and $\hat{A}_t$ is the GAE-computed advantage. The full PPO loss also includes value and entropy terms, following standard practice.

To handle partial observability in the task, the agent uses a gated recurrent unit (GRU) [91] to build memory over time. At each timestep t , the GRU took the current observation o_t and previous hidden state h_{t-1} , producing an updated hidden state h_t . Both the policy $\pi(a_t | o_{\leq t})$ and the value estimate $V(o_{\leq t})$ are computed from h_t via separate output heads. This architecture is shared between the actor and critic networks. We used a single-layer GRU with 512 hidden units. GRUs offer a gating mechanism that supports long-range memory retention while being more computationally efficient than LSTMs. This advantage makes them especially well suited for tasks involving memory and interpretability. We followed the implementation provided in the Craftax benchmark [12], which demonstrated that memory-equipped PPO agents perform substantially better on partially observed, long-horizon tasks. In total, our standard configuration contains approximately 6.5 million trainable parameters.

A.7 Future directions in ForageWorld

The Craftax platform also provides an opportunity for future work not yet explored here: incorporating curriculum learning into ForageWorld and related environments [92, 93]. The staged emergence of behavioral competencies we observed (e.g., in Figure 4) suggests that ForageWorld naturally supports curriculum-based training regimes and Craftax is compatible with JAX curriculum learning packages. We hope future work will explore curriculum learning [92, 93]. Future studies could leverage this structure—and the motif-based analysis framework—to probe how learning history shapes internal representations, and how staged training affects generalization, revisitation strategies, and neural dynamics. Along this direction, it would also be interesting to test LLM models’ foraging behavior, as navigation is an underexplored topic in LLM literature generally [94] (though note that transformers and LLMs would be harder to analyze than the simpler RNNs chosen here for their brain-like recurrent activity). However, Craftax is currently not coded for language-based labels of state variables so additional modification would have to be done to support LLM tests.

Lastly, note that Craftax and ForageWorld do not support multi-agent environments yet, but this would be an interesting modification to explore.

B Interpreting Agent Behavior: Supplementary Figures and Analysis

This section provides supplementary behavioral analyses and supporting visualizations. It expands on the main text results in section 4, including example trajectories, phase structure segmentation, and multi-objective revisitation strategies. We also present ablations targeting memory, auxiliary objectives, and predator combat to support interpretability and robustness claims.

B.1 Structured Exploration and Revisitation Trajectories

We present representative trajectories from trained agents that illustrate the structured exploration and patch revisitation behaviors described in section 4. Figure 6 shows trajectories from three episodes, with early exploratory loops followed by increasingly direct revisitation as the environment becomes familiar. These examples demonstrate that agents undergo a consistent shift from broad exploration to memory-guided foraging, even in novel arenas.

B.2 Behavioral Phase Segmentation via Unsupervised Clustering

To understand how agents modulate their movement patterns across time and task demands, we applied unsupervised clustering to the agent’s locomotion features. We used the `bayesmove` package to identify three latent movement states based on turning angle and step size over a 7-timestep moving window [95]. We next used conditional inference trees to identify which task variables best predicted transitions between these states [96].

As shown in Figure 7, the identified states correspond to interpretable movement modes: short-range (state 3), mid-range (state 1), and long-range (state 2) navigation. These were differentially associated with predator presence, eating behavior, and positional uncertainty. In particular, state 1 frequently coincided with predator events, suggesting an evasive or scanning behavior, while state 3 tended to occur near food and was enriched for eat actions. This analysis highlights the agent’s ability to adapt its movement patterns in response to internal and external signals, without requiring explicit mode-switching logic.

B.3 Comparison to Real Insect Search Patterns

To contextualize these emergent strategies, we compared agent trajectories to documented search behaviors in ants and bees. As shown in Figure 8, both species exhibit progressively expanding search loops when navigating novel or ambiguous environments. These loops are typically centered around a known location, such as a hive or nest. These patterns strongly resemble the outward-spiraling, azimuthally rotating loops exhibited by our agents during early exploration in novel ForageWorld arenas. Although the sensory and neural mechanisms differ across biological and artificial systems, the structural similarity in their trajectories suggests that simple control policies—when shaped by partial observability, uncertainty, and survival constraints—may converge on similar search dynamics.

B.4 GLM Outputs and Multicollinearity Checks

To support our analysis of patch revisitation strategies, we fitted generalized linear models (GLMs) to evaluate the influence of task variables on agent decision-making. Each GLM was trained to classify whether a patch was chosen at a revisitation point, using historical variables—such as recent reward rates, predator presence, and spatial uncertainty—as input features. Each model was trained on data from multiple agents and episodes, with agent identity included as a fixed effect to account for inter-agent variability.

Figure 13 shows the full GLM outputs underlying Figure 3, as well as variance inflation factor (VIF) scores used to assess multicollinearity among the regressors. As expected, uncertainty, predator encounter rate, and recency of visitation emerged as significant predictors, while variables such as dwell time and drink rate contributed less. VIF analysis confirmed that none of the regressors exhibited problematic multicollinearity. While revisitation behavior captures spatial strategy, we also tracked how decision confidence evolved over training.

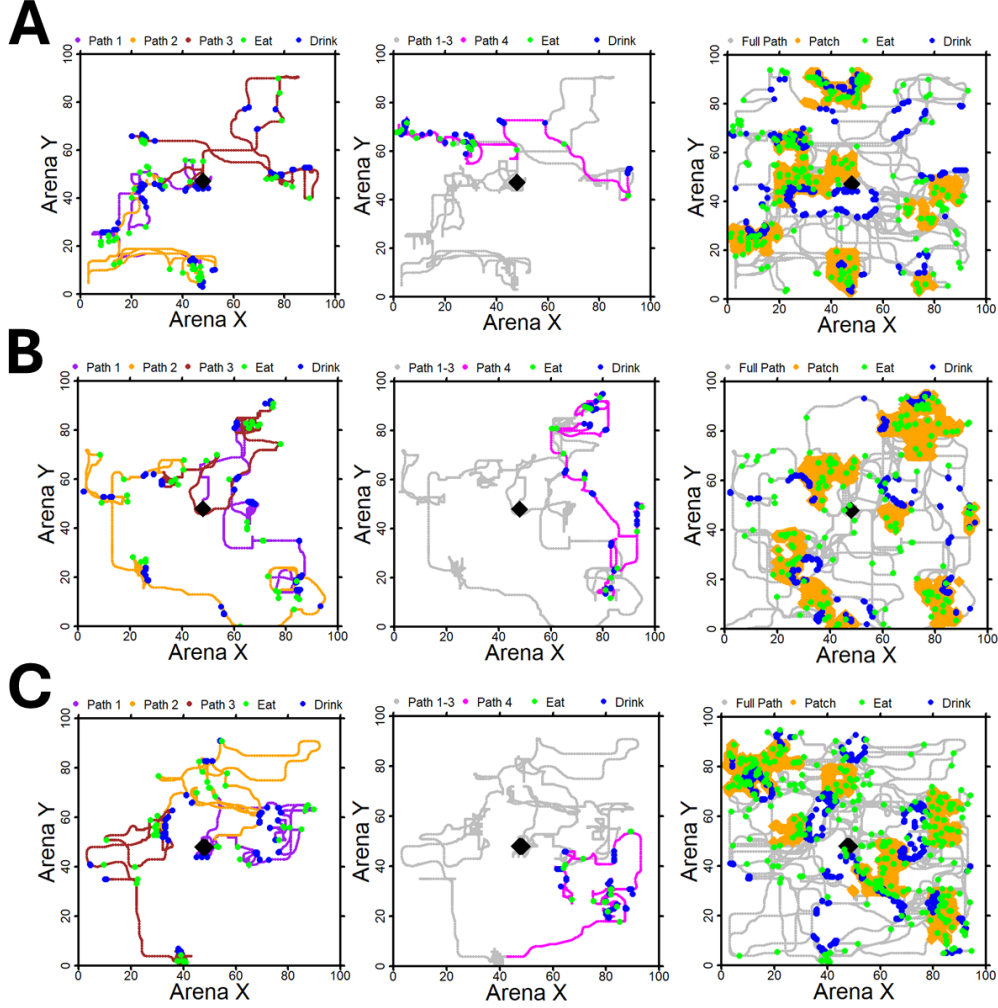


Figure 6: *Agents show stereotyped early exploration and sustained revisitation behavior in novel environments.* Representative trajectories from three episodes. Each row (A–C) illustrates three sequential stages from a single episode: early exploratory loops (left), a path sampled during the transition to revisitation (middle), and the full episode trajectory with revisited patch regions highlighted in orange (right). These examples demonstrate consistent exploratory patterns—such as outwardly expanding loops—and show that exploration persists even during revisitation, often aiding predator avoidance. This figure complements the summary view in Figure 1.

B.5 Training Dynamics and Entropy Evolution

To better understand how behavior evolves over training, we tracked the average policy entropy, Figure 9 shows how decision confidence increases over time, as reflected in the entropy dynamics that indicate increasing policy stability and confidence in selecting goal-directed actions.

These dynamics complement the behavioral and representational changes reported in section 4, further reinforcing the conclusion that high-level behavioral structure and planning capacity emerge progressively during training.

B.6 Memory and Predator-Related Ablations

To ensure our results were not driven by incidental dynamics or shallow heuristics, we conducted ablation experiments targeting memory mechanisms and predator response strategies. Figure 10

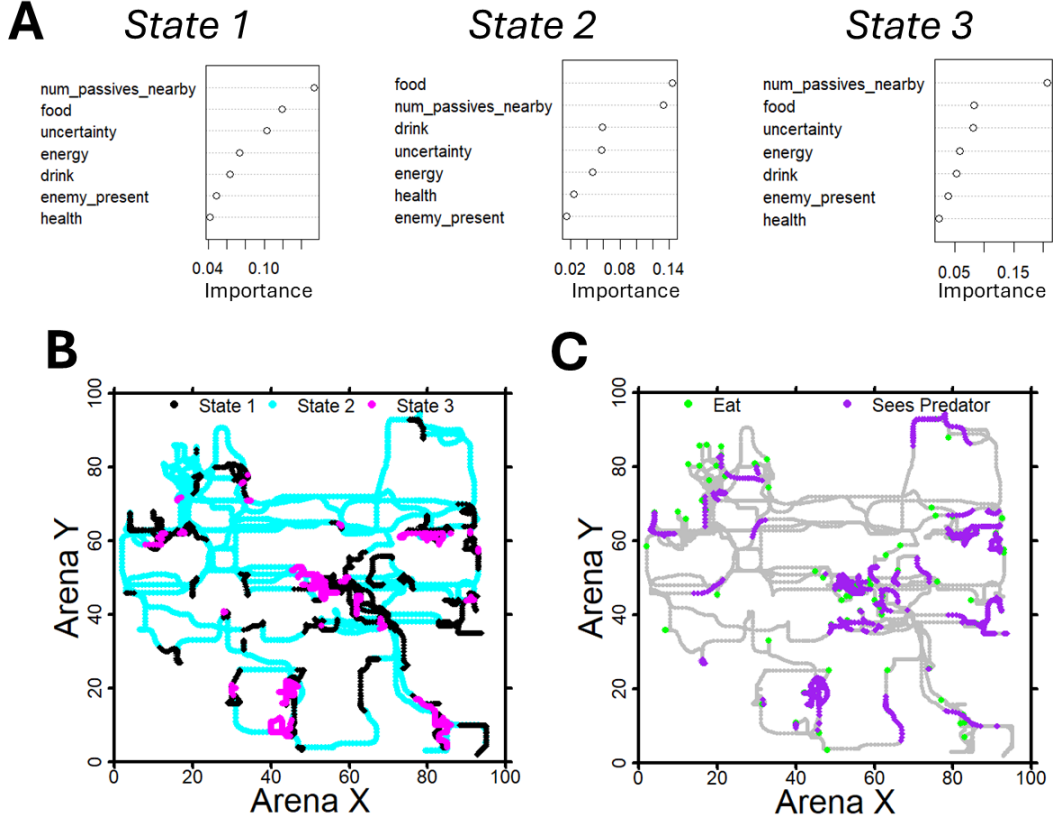


Figure 7: *Agents exhibit distinct movement patterns aligned with task variables such as food, fatigue, and predator presence.* (A) Variable importance from conditional inference trees [96] predicting transitions between three movement states identified via unsupervised clustering [95]: short-, mid-, and long-range locomotion. Key predictors include hunger level (*food*), positional uncertainty (*uncertainty*), predator presence (*enemy_present*), and nearby cow density (*num_passives_nearby*). (B) A representative episode segment colored by inferred movement state. (C) The same trajectory overlaid with predator encounters and eat events. State 1 (mid-range) frequently coincides with predator presence, while state 3 (short-range) clusters near food and eat actions. These results suggest that DRL agents flexibly shift between movement regimes in response to internal drives and environmental context—without requiring weight updates as these results were explored on test arenas with fixed weights.

shows a representative trajectory from a PPO agent without recurrence (i.e., no memory). These agents became trapped in inefficient local loops and had markedly reduced survival times, highlighting the critical role of memory in supporting long-horizon foraging.

In a second ablation, we ablated the auxiliary position-prediction objective (path integration). As shown in Figure 11, this had little effect on observed movement patterns. However, it significantly degraded decoding performance and eliminated structured spatial representations (see Figure 17 in the neural analysis section).

Lastly, we tested a predator-blind variant in which agents could not damage predators. As shown in Figure 4, agents exhibited similar movement strategies and achieved comparable survival times, indicating that predator evasion—not combat—is the primary learned strategy.

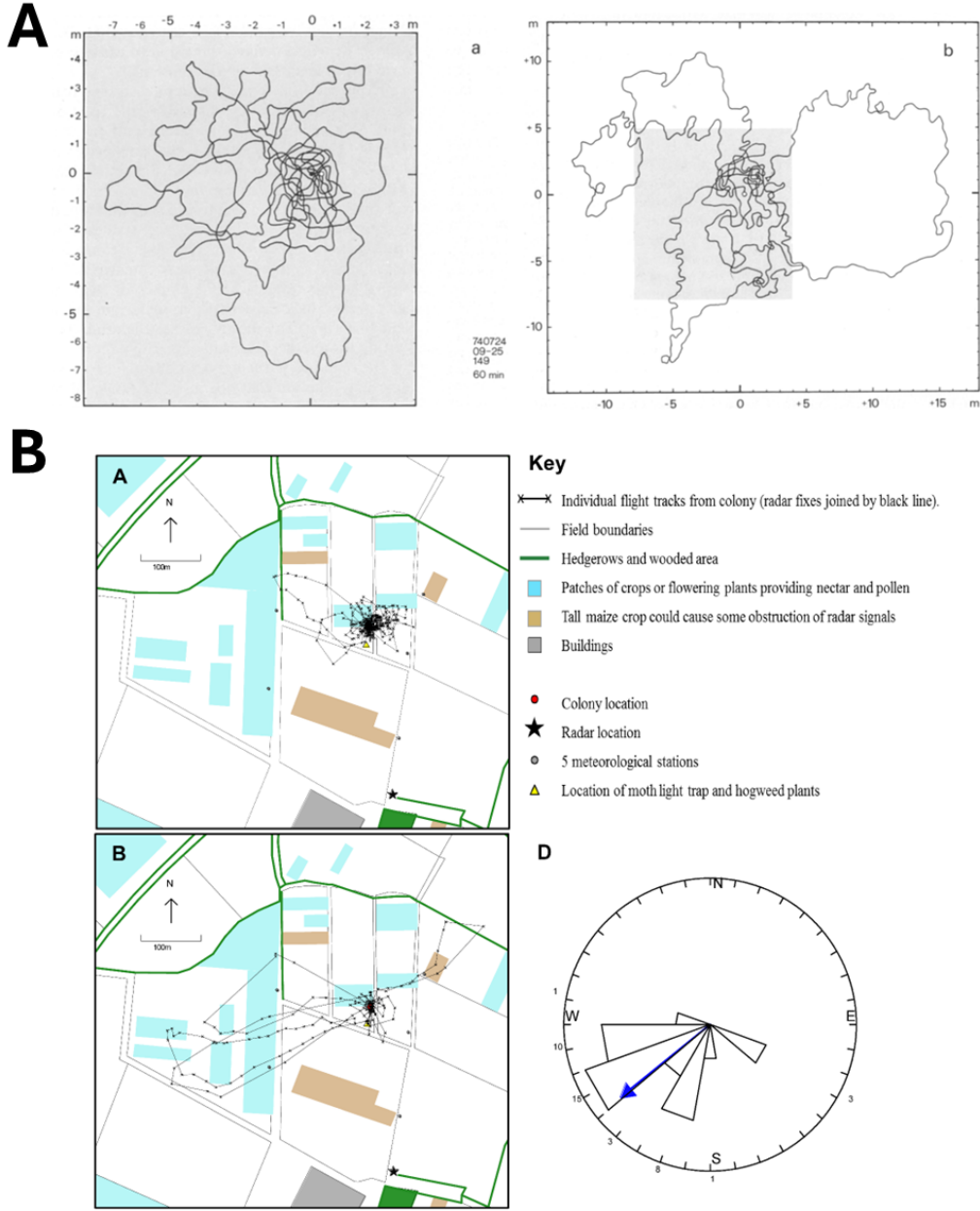


Figure 8: *Agent exploration trajectories qualitatively resemble real-world search patterns in ants and bees.* (A) Adapted from [79]: an ant’s search trajectory following nest displacement, showing outwardly expanding loops over time. (B) Adapted from [97]: bee trajectories during their first, second, and third departures from the nest. In both cases, the animals exhibit looping paths that rotate azimuthally and expand in spatial scale—closely matching the trajectories observed in DRL agents during early foraging. These parallels suggest that similar search strategies may emerge across biological and artificial systems when operating under uncertainty and partial observability.

C Interpreting Agent Representations: Supplementary Analyses

This section presents supplementary analyses and figures that support the neural decoding results in section 4. We examine how pruning and auxiliary objectives influence spatial encoding, and provide additional insight into the structure and modularity of the learned internal representations.

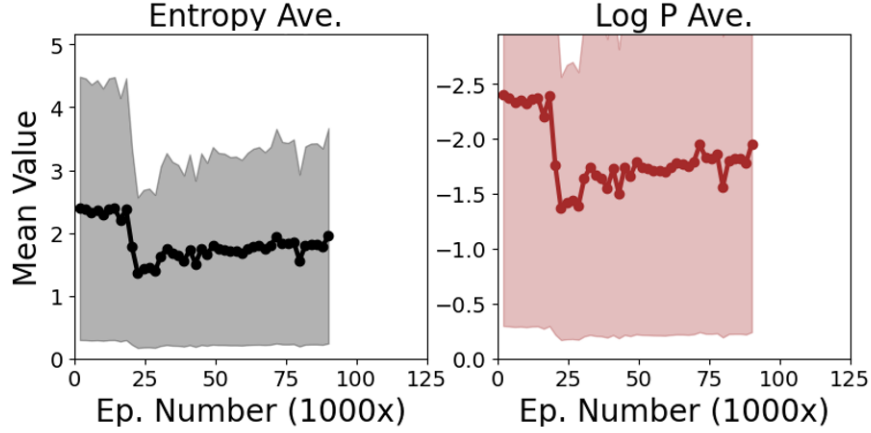


Figure 9: *Entropy dynamics reveal a transition from undirected exploration to confident, goal-directed behavior.* Evolution of average entropy over training for PPO-RNN agents. As training progresses, entropy declines and the log probability of the selected actions increases, reflecting increased confidence and policy refinement. Entropy slowly increases later in training as the agent discovers ways to maximize both task performance and policy entropy, reflective of the loss function.

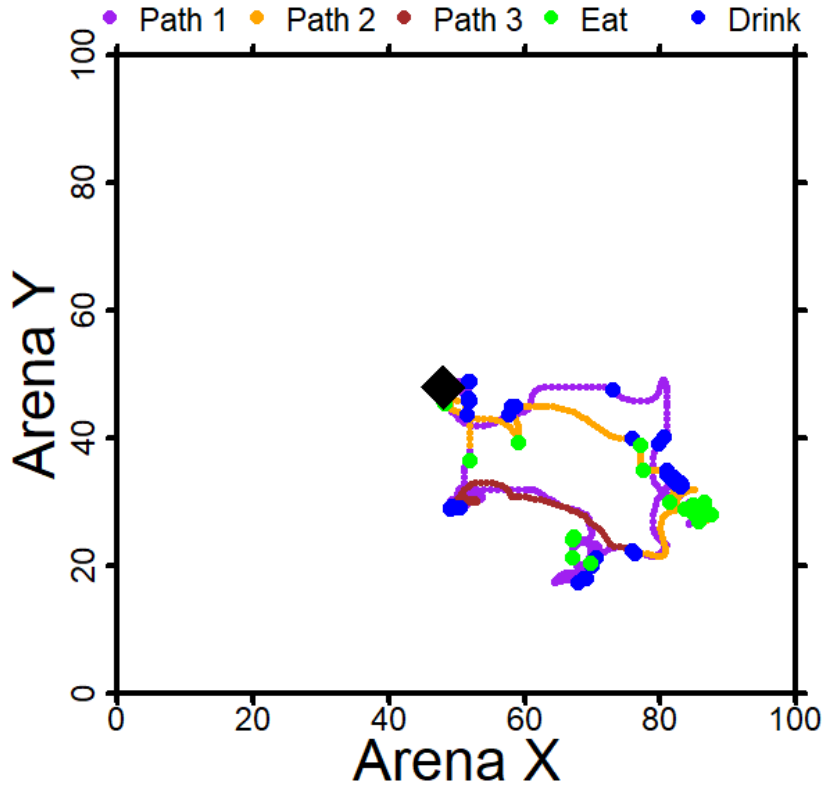


Figure 10: *Agents without recurrent memory exhibit poor spatial coverage and reduced survival.* Trajectory from a PPO agent without a recurrent architecture. Memoryless agents often became trapped in local loops and survived significantly fewer timesteps than agents with recurrence.

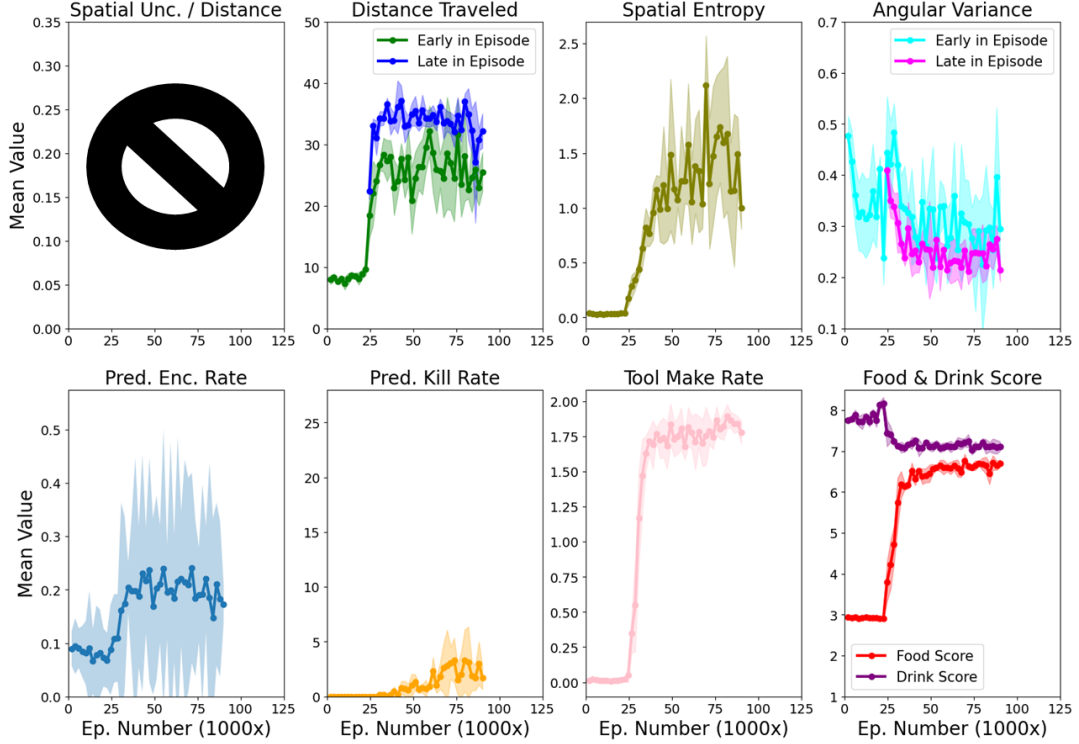


Figure 11: *Disabling the auxiliary position-prediction objective limits the predator killing abilities of agents.* Training dynamics and movement patterns for PPO-RNN agents without the auxiliary path integration objective. The absence of structured spatial uncertainty plots (top left) is due to the removal of the predicted position output normally produced by the auxiliary loss, since predicted current position is not outputted by the network in this experiment. The primary difference in learning history dynamics relative to the baseline model is that without the position-prediction objective, the emergence of predator killing abilities of agents is worse. This ability requires agents crafting a tool, with a sword being the highest damage tool, and reducing the predator health to 0. Predator killing is difficult because the predators can also kill the agent.

C.1 Impact of Auxiliary Objectives on Decoding

Figure 17 shows that removing the auxiliary path integration objective eliminated above-chance decoding of allocentric position from the RNN state. This indicates that the auxiliary loss was not merely a regularizer, but essential for inducing spatially interpretable internal representations.

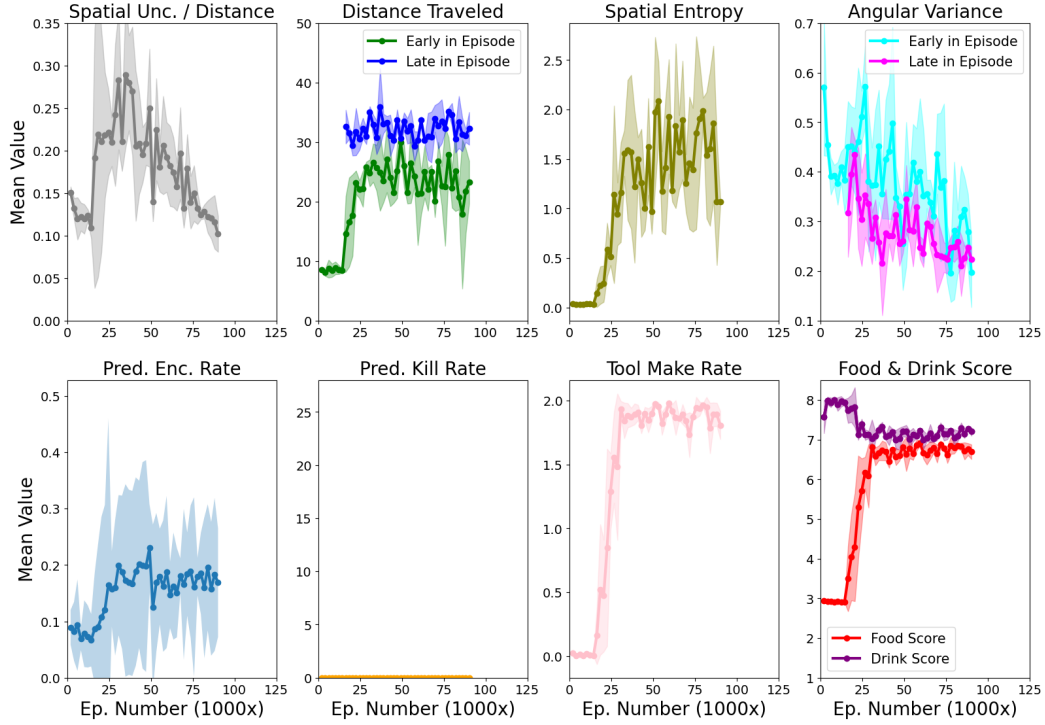


Figure 12: *Predator evasion dominates over combat in learned foraging behavior.* PPO-RNN agents trained without the ability to damage predators exhibited similar movement patterns and survival outcomes. This indicates that effective predator evasion—rather than combat—is the primary strategy developed by the agent.

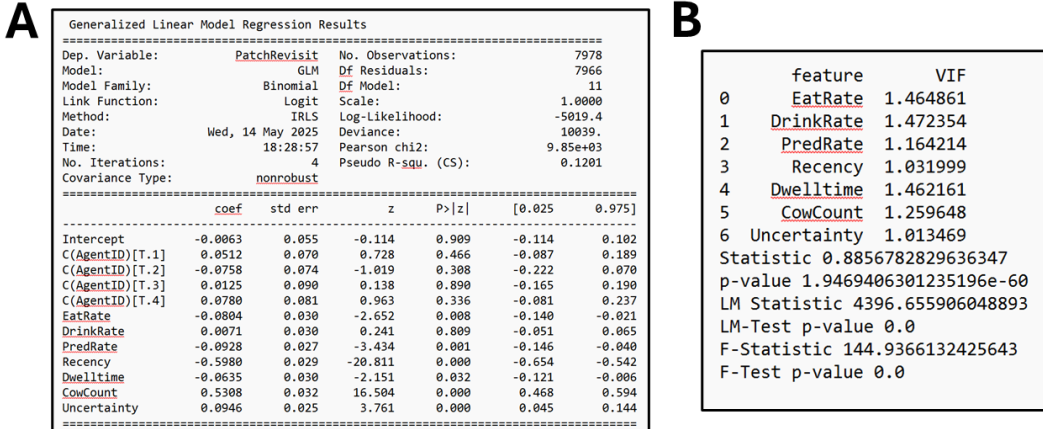


Figure 13: *GLM results confirm that patch revisitation decisions are shaped by uncertainty, reward history, and predator presence.* (A) Full GLM output underlying Figure 3, obtained using `statsmodels.formula.api` in Python. The model classified patches as chosen (1) or not (0) based on average historical variables, using agent ID as a fixed effect. A single model was fit jointly across five PPO-RNN agents, spanning 7978 revisitation decisions. (B) Variance inflation factor (VIF) scores show no excessive multicollinearity among predictors (all VIF < 10).

Table 2: *Variables logged at each timestep to support downstream behavioral and neural analyses.* These include internal state variables, agent decisions, environmental context, predictive outputs, and metadata, and are used to compute all behavioral metrics and decoding analyses reported in the paper.

Variable	Description
Action	Action selected by the agent
Health	Current health of the agent
Food	Current food level of the agent
Drink	Current water level of the agent
Energy	Current energy level of the agent
Done	Did the episode end?
Is Sleeping	Is the agent currently sleeping?
Is Resting	Whether the agent is in a resting state (disabled in all experiments presented)
Player Position	Agent’s absolute X/Y position relative to the top-left corner of the 96×96 arena (origin at 0,0)
Recover	The current state of the timer for the agent to recover a point of health
Hunger	The current state of the timer for the agent to lose a point of food
Thirst	The current state of the timer for the agent to lose a point of drink
Fatigue	The current state of the timer for the agent to lose a point of energy
Light Level	The light level variable, which modifies predator spawn rates (higher when lower, i.e. at night)
Distance to Melee	L1 distance to the nearest melee predator
Melee on Screen	Is there a melee predator currently on screen?
Distance to Passive	L1 distance to the nearest cow/food animal
Passive on Screen	Is there a cow currently on screen?
Distance to Ranged	L1 distance to the nearest ranged predator
Ranged on Screen	Is there a ranged predator currently on screen?
Num Melee Nearby	How many melee predators are currently on screen?
Num Passives Nearby	How many cows are currently on screen?
Num Ranged Nearby	How many ranged predators are currently on screen?
Delta X & Y (Relative Position)	Agent’s position relative to its starting location (center of arena)
Predicted Delta X & Y	Agent’s internal prediction of its position relative to its starting location
Num Monsters Killed	How many predators has the agent defeated this episode so far?
Has Sword	Whether the agent has crafted a sword to improve combat effectiveness this episode
Has Pick	Whether the agent has crafted a pickaxe to remove stone obstacle tiles
Held Iron	Number of iron resources collected (used to craft improved weapons)
Value	The agent’s value function output
Entropy	The entropy of the agent’s policy for the current timestep
Log Probability	Log probability of the selected action under the policy at the current timestep
Episode ID	Random ID for the current episode

Table 3: Model and environment hyperparameters used for the *baseline* configuration.

Parameter	Default Value	How Chosen (if applicable)
Model Parameters		
Learning Rate	0.0002	Craftax default
γ	0.99	"
λ_{GAE}	0.8	Same as Craftax default
PPO clipping ϵ	0.2	Same as Craftax default
$\mathcal{W}_V$	0.5	Same as Craftax default
$\mathcal{W}_{\text{entropy}}$	0.01	Same as Craftax default
Activation function	tanh	Same as Craftax default
Number of neurons per layer	512	Same as Craftax default
Steps per PPO iteration	64	Same as Craftax default
Number of parallel environments	1024	Same as Craftax default
Number of epochs per PPO iteration	4	Same as Craftax default
Number of minibatches per epoch	8	Same as Craftax default
Total training timesteps	3,000,000,000	Chosen to ensure convergence or trend visibility within a 24–48 hour wall-clock training window
$\mathcal{W}_{\text{aux}}$	0.025	Parameter sweep of {0.01, 0.1, 0.025, 1.0}, balancing performance and position encoding quality
Pruning type	Magnitude	Only JaxPruner type that did not severely degrade performance in testing
Prune Step	20,000	Mid-training: after baseline agent begins performing reasonably, but before convergence
PQN-specific Parameters		
ϵ Start	1.0	PQN default
ϵ Finish	0.05	Parameter sweep of {0.005, 0.01, 0.05, 0.1}
ϵ Decay	1.0	Parameter sweep of {0.1, 1.0, 2.0}
Total training timesteps	12,000,000,000	Chosen to allow PQN time to explore+converge
Total decay timesteps	6,000,000,000	Parameter sweep of {1B, 6B, 12B}
Num environment steps per update	128	PQN default
λ	0.5	PQN default
Environment Parameters		
Predators	True	
Max Cows	108	Selected from a sweep {48, 72, 108} to balance patch-leaving and revisitation in competent agents
Use full action space	False	Disabled actions not relevant to ForageWorld (subset of original Craftax action space)
Map Size	96	Chosen from among {24, 48, 96} to maximize the navigation and memory challenge for the agent
Directional Vision	False	

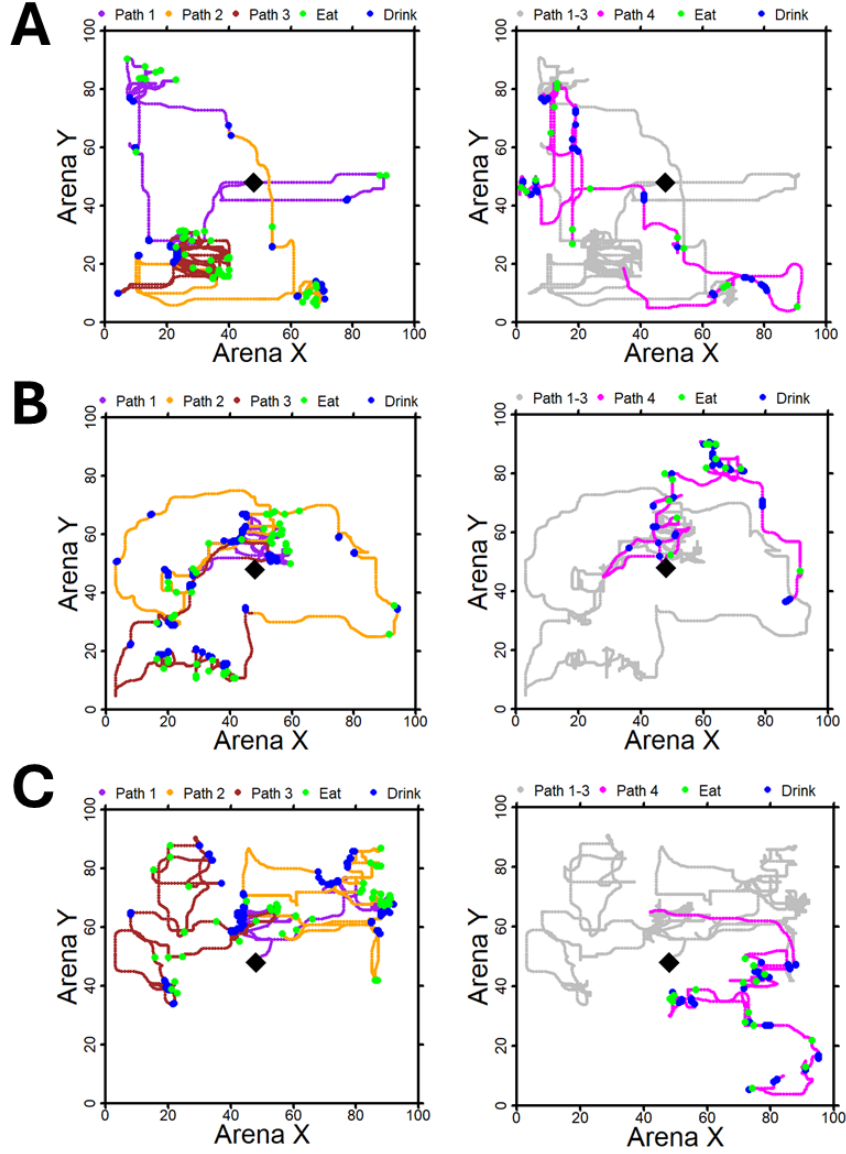


Figure 14: *Front-facing agents exhibit denser local exploration in early training.* Each row shows a representative early-exploration trajectory for an agent with a front-facing field of view (FOV), as opposed to the 360° egocentric view used in the baseline configuration. Front-facing agents tended to perform intermittent localized and dense exploratory behavior during the early phases of exploration.

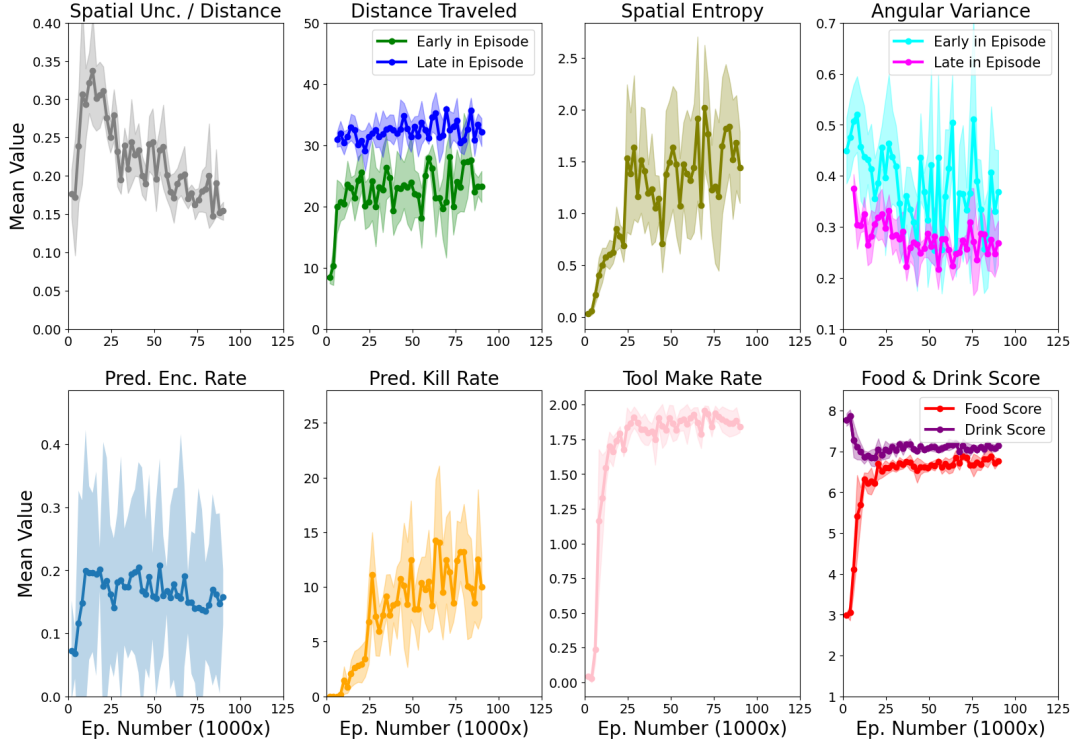


Figure 15: *Front-facing FOV agents exhibit earlier onset of long-range exploration.* Variant of Figure 4 for PPO-RNN agents with a front-facing field of view. These agents showed a faster shift toward long-range exploration compared to baseline agents. This modification may serve as a useful manipulation in future studies of sensory constraints.

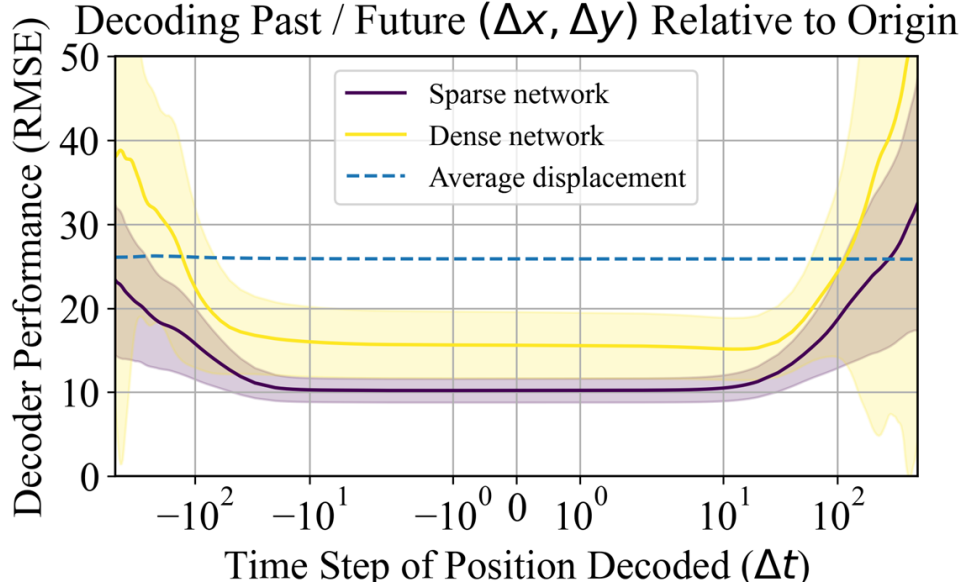


Figure 16: *Pruned networks yield better allocentric position decoding than fully connected networks.* Comparison of allocentric position decoding accuracy (past/future) between sparse (90%) and unpruned PPO-RNN agents. Sparse models show higher accuracy, potentially due to more compact and modular encoding of task-relevant variables. Error bars show 95% confidence intervals.

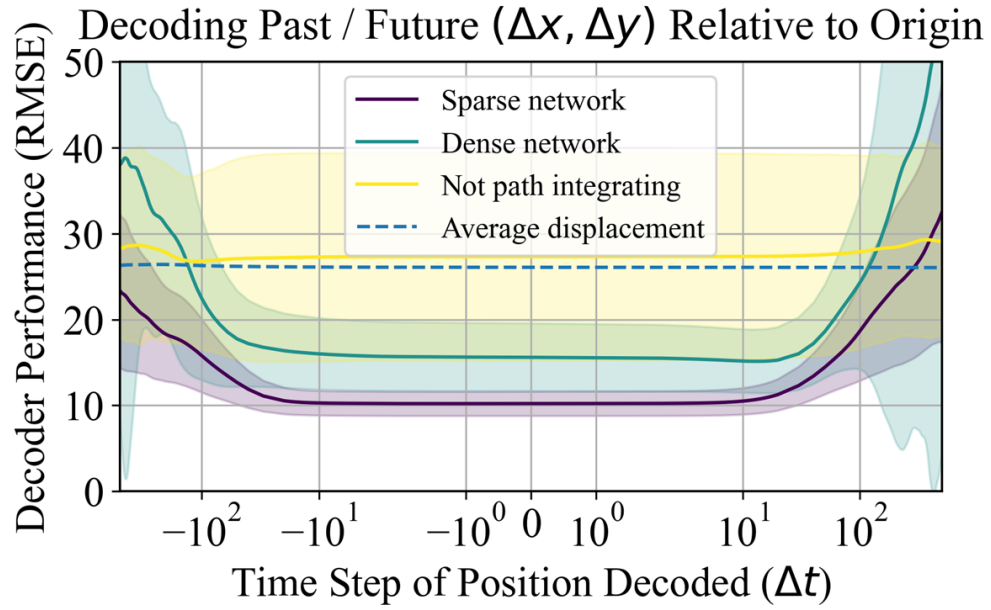


Figure 17: *The auxiliary position-prediction objective is necessary for spatially interpretable representations.* Without the auxiliary path integration loss, allocentric position decoding from the agent’s RNN state drops to chance levels—eliminating the structured spatial encoding observed in the baseline model. Bars show decoding accuracy for past and future positions; error bars reflect 95% confidence intervals.

C.2 Coefficient Structure and Functional Modularity

To better understand how position information is distributed across neurons, we analyzed the regression coefficients learned by decoders trained to predict past and future positions. Figure 18 visualizes these ridge regression weights, aligned by neuron ID, for both sparse and dense PPO-RNN agents. In both architectures, overlapping neuron populations contributed to past and future decoding, but this overlap diminished at longer temporal offsets. Sparse networks exhibited clearer separation between past- and future-coding units, as well as higher overall sparsity in their weight maps—factors that may underlie their superior decoding performance.

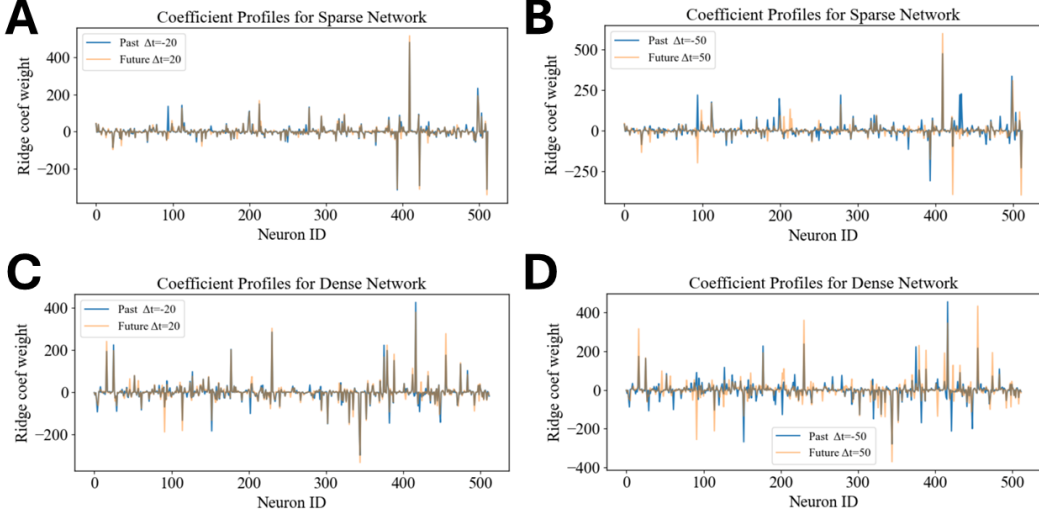


Figure 18: *Sparse agents exhibit modular and separable encoding of past and future spatial positions.* Ridge regression weights for allocentric position decoding, aligned by neuron ID. (A, B) show sparse networks at 20- and 50-timestep prediction horizons; (C, D) show dense networks. While both architectures use overlapping units for past and future decoding, sparse networks show greater separation at longer horizons—consistent with their improved decoding performance.

C.3 Regularization and Train-Test Split for Decoding

Some RNN units remain inactive for entire episodes, making the decoding problem potentially ill-conditioned. Accordingly, we applied ridge regression with an ℓ_2 regularization term:

$$L_{\Delta t}(f) = \sum_{i=1}^N (Y_{t+\Delta t}^i - f(h_t^i))^2 + \alpha \|f\|_K^2$$

This regularization improves model stability while preserving interpretability. A separate decoder was trained for each prediction offset Δt using the following linear model:

$$f(h_t) = Ah_t + b, \quad A \in \mathbb{R}^{2 \times 512}, \quad b \in \mathbb{R}^2$$

We selected ridge regression for its favorable tradeoffs among interpretability, numerical stability, and computational efficiency. With time complexity $O(Np^2)$ and direct access to regression coefficients, this approach enables fine-grained analysis of how individual RNN units contribute to spatial encoding.

Train-Test Split Design

To prevent information leakage due to temporal continuity in h_t , we employed a structured train-test split. Each decoder was trained on the first 75% of timesteps within an episode and evaluated on the remaining 25%. This design ensures that models cannot rely on short-range temporal correlations and must generalize across broader regions of feature space. All decoders were trained across multiple episodes to ensure robustness to variation in arena layouts and agent trajectories.

D Generalized linear model for predicting neural activity from current position

As a supplementary analysis, we explored how position was represented at the single neuron level in the RNNs. NeMoS (Neural Models) was used for fitting generalized linear models (GLMs) that predicted neural activity from the current position of the agent, see <https://github.com/flatironinstitute/nemos> for extended documentation [98]. One GLM was fit per neuron, but multiple episodes (arena configurations) were used for each model fit, typically 5-10 episodes minimum to avoid overtraining to one arena configuration. Prior to setting up the model, the 96×96 arena was coarse-grained into 14×14 position bins, and a binary variable per bin was encoded that reported whether or not the agent was in a given bin (196 total regressor coefficients). Neural data was also shifted into the positive axis (lowest value zeroed), and normalized to the range of 0-1 per neuron to extract the activity changes relative to a given neuron’s baseline for easier cross-neuron comparison.

The GLM fit the first 70% of each episode in the training set of episodes Figure 19. The majority of neurons fit had comparable performance between the test and training sets Figure 20.

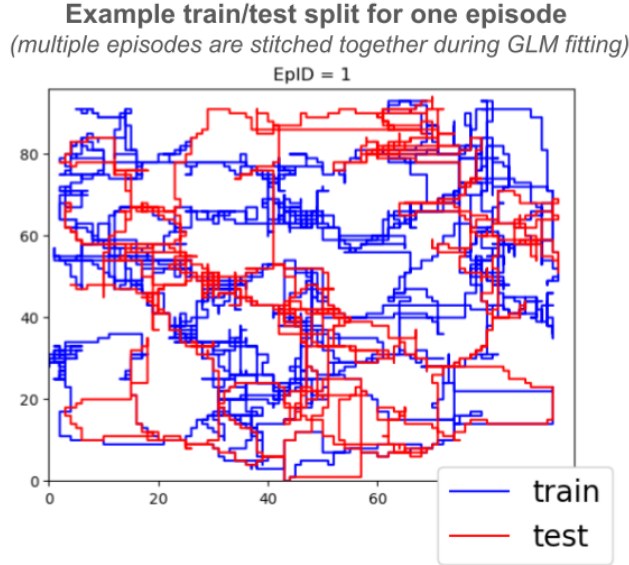


Figure 19: *Example test/train split for a single episode to be used in GLM fitting.* GLM models, one model per neuron, with 5-10 episodes per model, predicted neural activity from position. The first 70% of an episode was trained, while the last 30% was the test split.

Approximately 100/512 neurons had good fit performance using just position regressors across runs, with values typically in the range of 60-120 position-sensitive neurons per training run. An example good fit is shown in Figure 21.

We then plotted the number of neurons with GLM coefficients in the three largest mean-valued k-means clusters out of five clusters, see Figure 22 for the selection. We also plotted the average GLM coefficient per position bin Figure 23. The number of strongly position encoding neurons increased with radial distance from the origin of the arena, and the average GLM coefficient magnitude also increased with radial distance from the origin Figure 24. This trend was seen in all main conditions, PPO-RNN with and without path integration, and PQN-RNN. Thus, an accumulation circuit may ramp with distance from origin and pressure the agent to return to the origin the farther away it gets, partially explaining the periodic origin revisiting behavior observed across conditions. This pattern was present even without the position predicting auxiliary objective Figure 25 and Figure 26, and also in the PQN-RNN runs Figure 27 and Figure 28, suggesting it is fundamental to the task solution. Example scripts to use this analysis method are in the github repository for this paper.

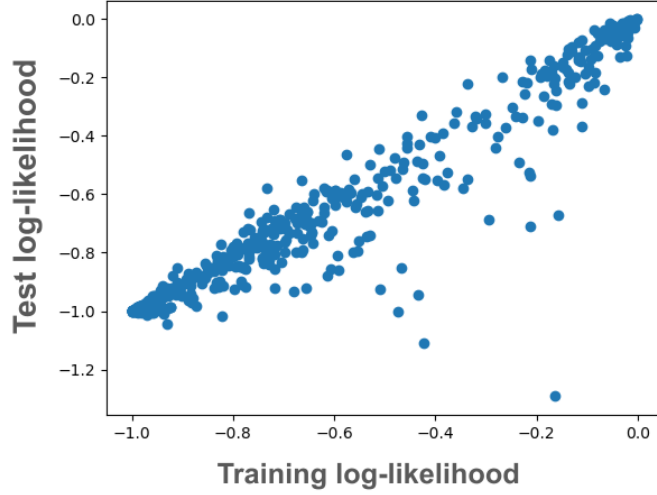


Figure 20: *Log likelihood scores* for the same GLM model (single neuron fit), showing the difference in log likelihood between test and train data, one model per data point. The data here is a PPO-RNN baseline run.

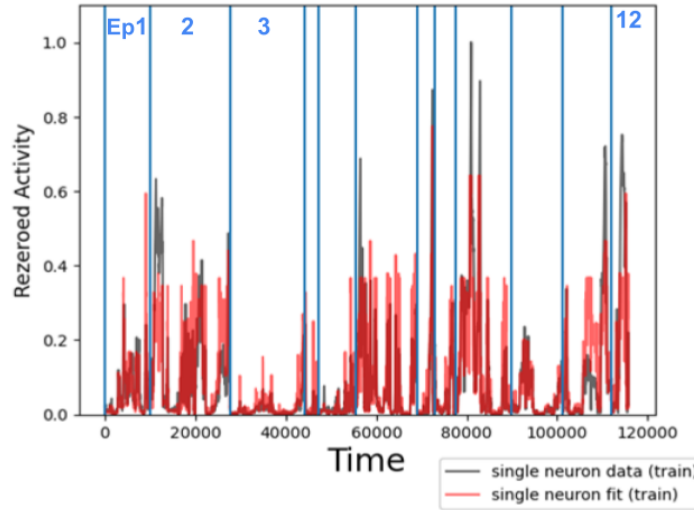


Figure 21: *Example of a single neuron, multi-episode model fit.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. We show an example good fit for a single neuron's data (black), fit to 12 different episodes for one model (fit in red).

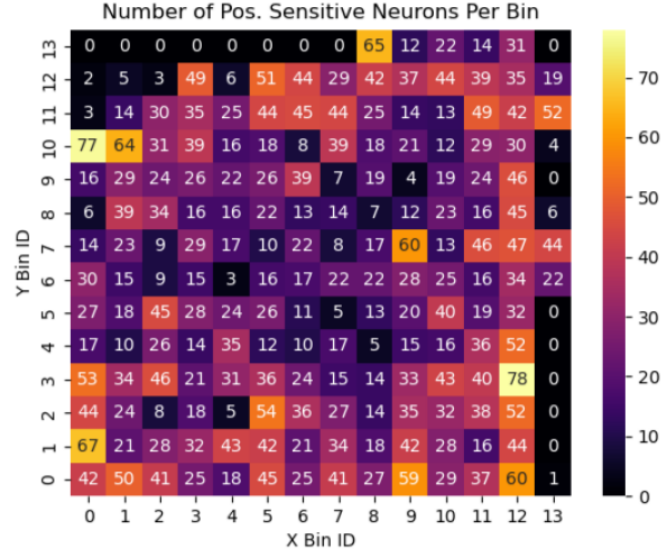


Figure 22: *Number of position-sensitive neurons per position bin in the arena for PPO-RNN.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. The heatmap of the number of well-fit neurons per position bin is shown in the plot. The data here is a PPO-RNN baseline run. The 0's in the uppermost row and rightmost column of bins are due to the agents never going out to those position bins in these runs.

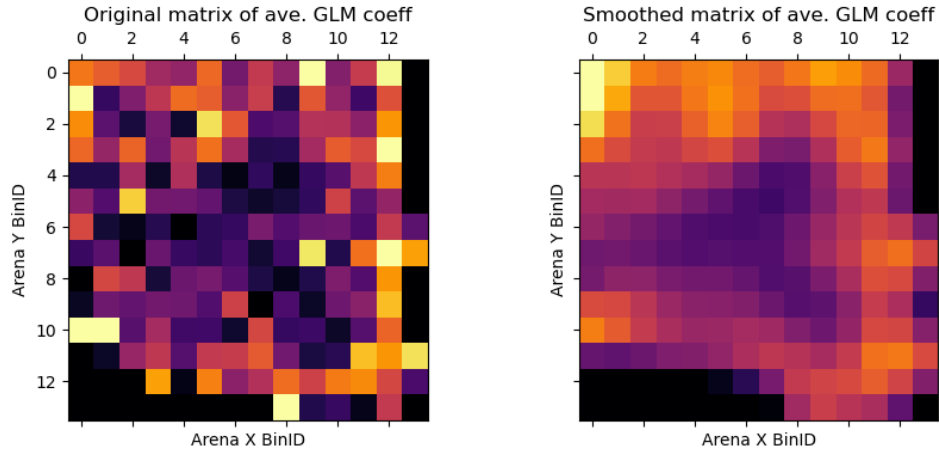


Figure 23: *Heatmap of average GLM coefficient magnitude per position bin for PPO-RNN.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. These plots show the average GLM coefficient value across well-fit neurons per bin (left), and the Gaussian smoothed shape of the same heatmap (right). The data here is a PPO-RNN baseline run. The 0's in the lowermost row and rightmost column of bins are due to the agents never going out to those position bins in these runs.

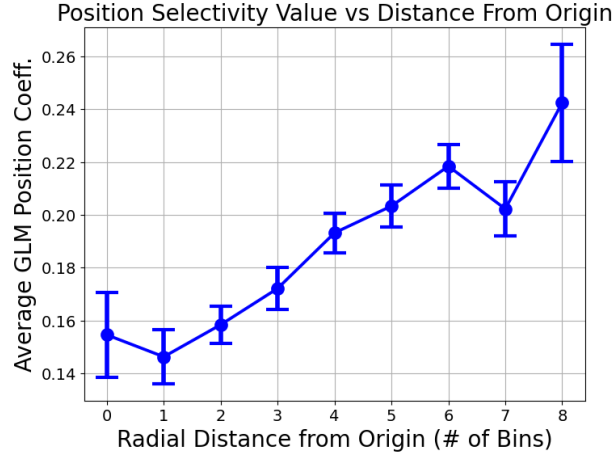


Figure 24: *Average position bin GLM coefficient versus radial distance from the arena origin (center) for PPO-RNN.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. The average GLM coefficient for the well-fit neurons per radial bin distance from the origin was plotted, finding the neural response to a position change increases with distance from the origin. The data here is a PPO-RNN baseline run. Data points are the mean across neurons while error bars are 95% CI.

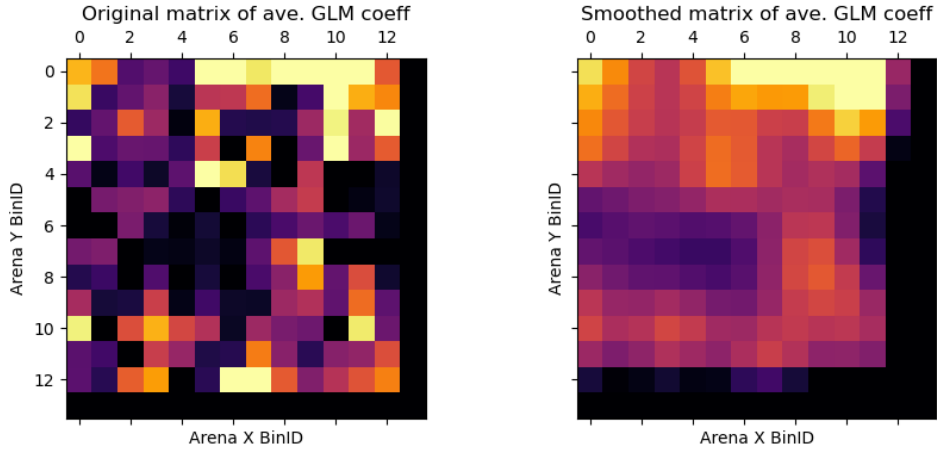


Figure 25: *Heatmap of average GLM coefficient magnitude per position bin for PPO-RNN without path integration.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. These plots show the average GLM coefficient value across well-fit neurons per bin (left), and the Gaussian smoothed shape of the same heatmap (right). The data here is a PPO-RNN without position-prediction (no path integration) run. The 0's in the lowermost row and rightmost column of bins are due to the agents never going out to those position bins in these runs.

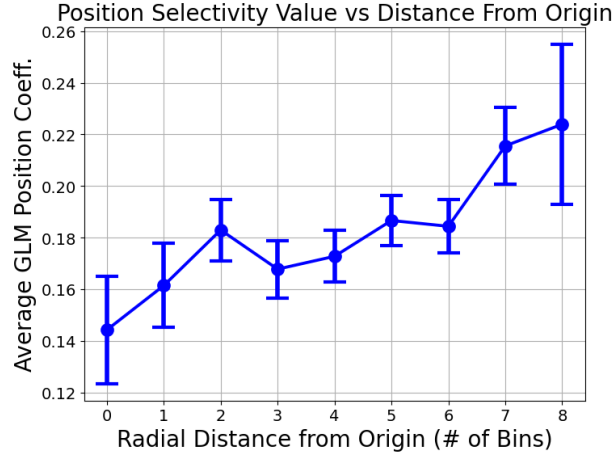


Figure 26: *Average position bin GLM coefficient versus radial distance from the arena origin (center) for PPO-RNN without path integration.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. The average GLM coefficient for the well-fit neurons per radial bin distance from the origin was plotted, finding the neural response to a position change increases with distance from the origin. The data here is a PPO-RNN without position-prediction (no path integration) run. Data points are the mean across neurons while error bars are 95% CI.

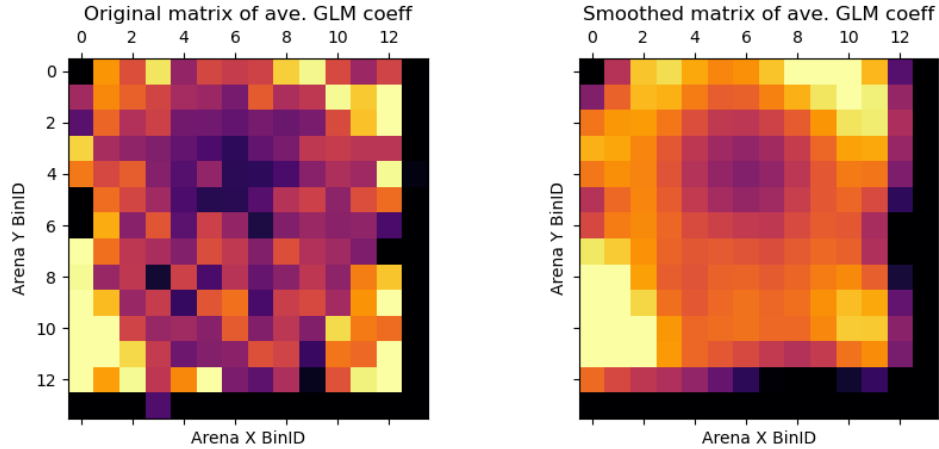


Figure 27: *Heatmap of average GLM coefficient magnitude per position bin for PQN-RNN.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. These plots show the average GLM coefficient value across well-fit neurons per bin (left), and the Gaussian smoothed shape of the same heatmap (right). The data here is a PQN-RNN run. The 0's in the lowermost row and rightmost column of bins are due to the agents never going out to those position bins in these runs.

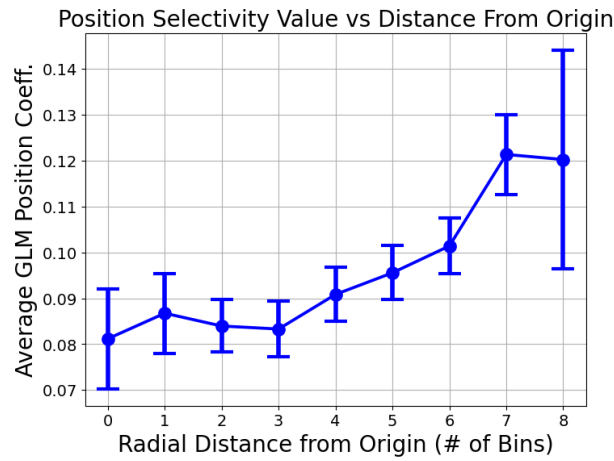


Figure 28: *Average position bin GLM coefficient versus radial distance from the arena origin (center) for PQN-RNN.* Each of the 512 GLM models fit, one per neuron, has one GLM coefficient for each of 196 position bins (14 x 14) that the arena grids were coarse-grain binned into for this analysis. The average GLM coefficient for the well-fit neurons per radial bin distance from the origin was plotted, finding the neural response to a position change increases with distance from the origin. The data here is a PQN-RNN run. Data points are the mean across neurons while error bars are 95% CI.

E Alternative DRL Objective: Parallelized Q-networks (PQN)

Recent work has shown that the off-policy RL algorithm parallelized Q-networks (PQN) can perform well in vanilla Craftax [75]. We explored whether this alternative DRL algorithm performs differently compared to PPO on ForageWorld. We adapted the PQN algorithm to ForageWorld using the LSTM model architecture and default hyperparameters from [75], which differ in some ways from our PPO implementation (for example, using an LSTM instead of a GRU for the RNN). Except where noted in Table 3, we used the same environment and training hyperparameters between PPO and PQN. This experiment provides an orthogonal test of ForageWorld— would a different algorithm, model, and training parameters result in different behavior?

We found that the PQN-LSTM models performed comparably to the PPO-GRU baseline for the subset of PQN models that learned the task (Figure 29), but that only a subset of PQN runs would converge to this level of performance— across the two successful test conditions that varied the exploration hyperparameters of PQN (in particular controlling the rate of decay of ϵ , the chance of taking a random action instead of the optimal action under the current Q value function of PQN), only 30% of PQN runs performed comparably to the PPO model, with the remainder failing catastrophically in a way that was not seen in the baseline PPO runs. We believe this is the result of suboptimal exploration by PQN, i.e. the poorly performing PQN runs learned simple and reliable but suboptimal behaviors early in training and then were slow to deviate from them later on. For the runs that did perform comparably to PPO, we found overall similar learning histories and pathing behavior to the PPO baseline runs (Figure 30, Figure 31), but with the noticeable behavioral differences of much lower rates of predator killing (i.e. reducing predator HP to 0) (Figure 31) in the learning history analysis. Thus, this result suggests that, despite showing comparable overall performance at the task, PPO seems to converge to a predator fighting strategy, while PQN converges to a predator evasion strategy.

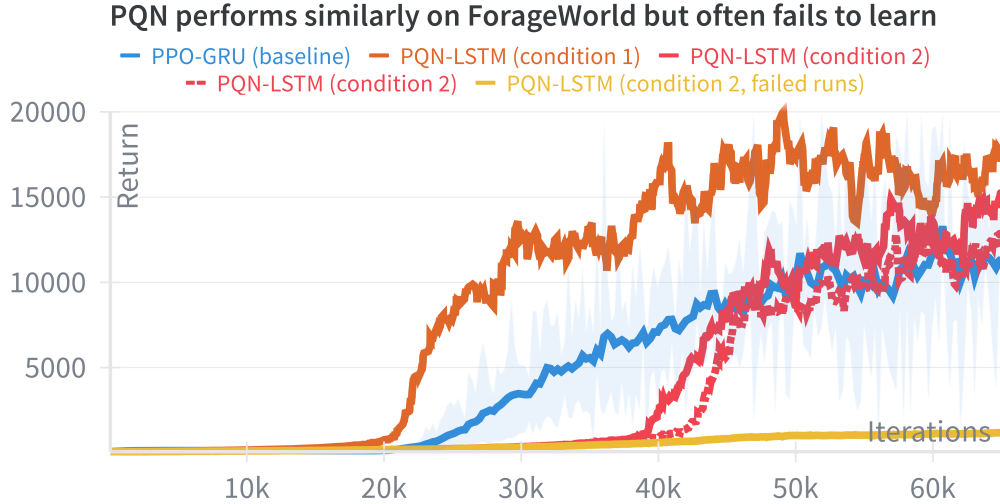


Figure 29: *Performance comparison for PQN versus PPO.* Individual PQN-LSTM runs performed comparably to the PPO-GRU baseline for two different exploration hyperparameter configurations, but the majority of runs failed to learn more than a trivial policy. Condition 1 used the ϵ decay parameters shown in Table 3, while condition 2 adjusted ϵ start, finish, and decay parameters so that ϵ decayed half as fast and ended at a final value of 0.1 (twice the chance of a random action). No condition resulted in uniformly successful runs. Other exploration hyperparameter values were tested, which are detailed in Table 3, but did not result in any high-performing runs.

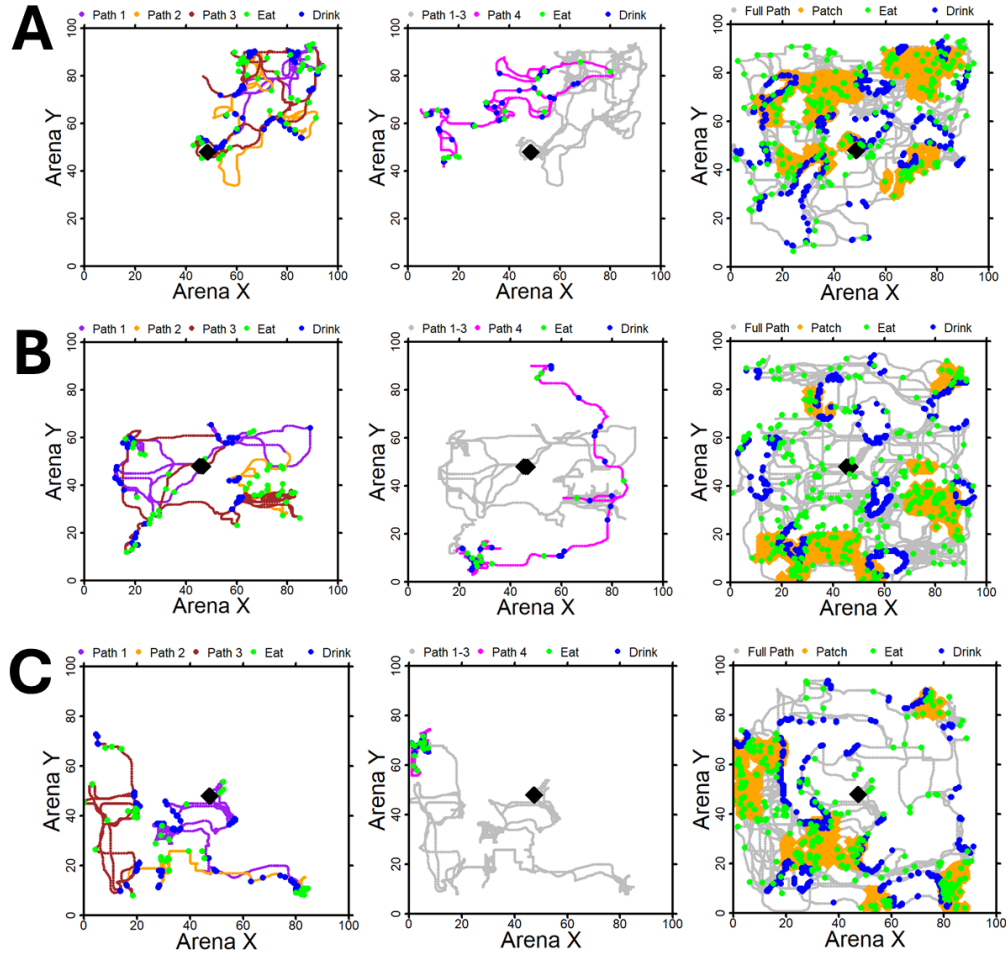


Figure 30: *Representative trajectories from three episodes for PQN-RNN.* Each row (A–C) illustrates three sequential stages from a single episode: early exploratory loops (left), a path sampled during the transition to revisitation (middle), and the full episode trajectory with revisited patch regions highlighted in orange (right).

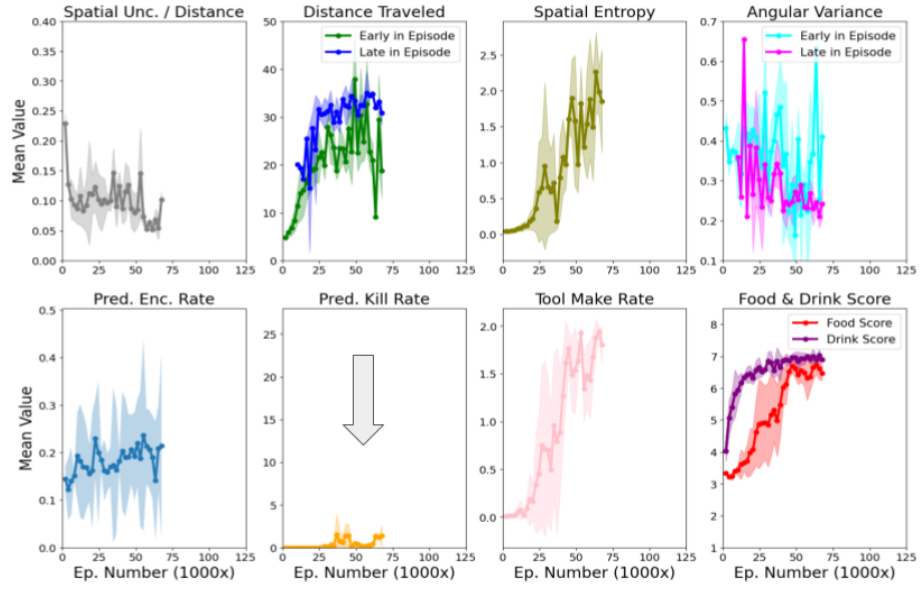


Figure 31: *Training dynamics and movement patterns learning history for PQN-RNN agents with the auxiliary path integration objective. PQN had overall similar patterns to PPO, but was found to not learn predator killing as well (gray arrow).*

F Toward Architectures for Cognitive Map Formation (Future Directions)

While our agents do not yet use structured maps, our findings suggest that such architectures may enhance both behavior and interpretability. Building on these findings about memory, planning, and spatial encoding in model-free agents, we highlight one architectural direction and outline broader opportunities for future work. An important next step is to explore architectures that more closely reflect mammalian spatial navigation capabilities. In biological systems, both allocentric and egocentric representations support planning and memory [99], particularly through grid-based spatial computations [100].

These “cognitive maps” [5, 101–104] extend beyond current position encoding—they support retrospection, prospective planning, and flexible decision-making [105–107]. Recent models have begun incorporating cognitive map-like representations into reinforcement learning frameworks [108].

In animals, spatial maps are modulated by task variables such as goals, landmarks, and reward-related events [109, 110, 20, 111]. These advantages have been recognized in machine learning as well, with grid-based representations increasingly integrated into transformers [112] and actor-critic agents [113] to improve task performance, efficiency, and interpretability.

Future work should examine how grid-based recurrent architectures solve the ForageWorld task—probing what spatial computations emerge and how they differ from standard RNNs. More broadly, our behavioral-neural analysis framework could be applied to other agent architectures and tasks, or used to generate hypotheses about navigation and memory in biological systems.

G Craftax licensing permissions

<https://github.com/MichaelTMatthews/Craftax/blob/main/LICENSE>

Our ForageWorld environment is modified from Craftax. The Craftax license is as follows, taken from the URL above:

“Copyright (c) 2024 Michael Matthews

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software."

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We tried to carefully ensure the strength of evidence matched the strength of claims in our work.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have discussed limitations in section 5

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: We do not have theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Environment, model and analysis codes are provided in the paper's github repository (<https://github.com/RileySE/Craftax-Foraging/tree/foraging>).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes] .

Justification: Environment, model and analysis codes are provided in the paper's github repository (<https://github.com/RileySE/Craftax-Foraging/tree/foraging>). We provide key hyperparameters and other settings to reproduce experiments in Table 3

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

6. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, when needed to justify statistical claims, the authors use 95% CI as error bars unless otherwise specified.

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

7. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes] .

Justification: Yes, see the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

8. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The authors have reviewed the code of ethics and believe our work is not in violation of any rules.

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

9. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, see section 5.

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

10. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: We did not scrape any data or release a model at significant risk of dual-use applications.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

11. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#) .

Justification: Our work modifies the Craftax environment, whose free use license is given in the appendix: Appendix G. Additionally, the proper license terms were followed from the respective host journals for the adapted, previously published figures in Figure 8. We paid for the use of figure [79], and the other figure was published under an unrestricted use license, Creative Commons Attribution License, as long as the author is accredited [97].

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

12. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#) .

Justification: We have released the ForageWorld environment code and accompanying model and analysis code, which are provided on Github at (<https://github.com/RileySE/Craftax-Foraging/tree/foraging>).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

13. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#) .

Justification: We do not perform human research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

14. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA] .

Justification: We do not perform human research.

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

15. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA] .

Justification: We do not use LLMs.

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.