Human Hippocampal Replay as Search

Mishaal Kandapath (mishaalhere@gmail.com)[1] **& Can Serif Mekik (can.mekik@utoronto.ca)**[2]
[1]Department of Computer Science, University of Toronto
[2]Department of Psychology, University of Toronto

Abstract

Human-like systematic generalization remains a challenge for artificial systems. Recent work suggests hippocampal replay as a mechanism for online planning that facilitates generalization in novel, dynamic, and complex environments. Recent models, based on spatial tasks and rodents, present a serial notion of planning. In contrast, recent work on humans finds replay sequences traversing multiple trajectories at the same time on a compositional generalization task. Such a possibility for parallel execution does not interface well with serial planning. Here, we present a novel model of replay as planning with parallel search capabilities, extending an existing serial model of replay. Our model displays human-level generalization faster than the serial baselines. Furthermore, we find that models actively using parallel search produce internal dynamics qualitatively resembling neural replay sequences in humans performing the task. Our findings present search as a powerful general computation leveraged during replay for quickly learning good policies from little data.

Keywords: Hippocampal replay; Machine Learning; Computational Neuroscience; Cognitive Modeling; HPC-PFC Network

Introduction

Humans learn so much from so little. We also generalize effectively. While explosive progress in deep learning has attained or even exceeded human-level performance in some naturalistic tasks, efficient generalization still evades modern deep learning systems (Cherian et al., 2023; Lake et al., 2017; Moskvichev et al., 2023). This contrast is perhaps starkest in systematic compositional generalization tasks (Lake & Baroni, 2023; Moskvichev et al., 2023). Many have proposed that such an ability is rooted in the formation and utilization of rich hierarchical internal world models (Behrens et al., 2018; Lake et al., 2015). Computationally, this hypothesis poses two main questions: how do we form accurate world models from experience, and how might we use them to make sense of the present and imagine the future? This paper is primarily concerned with the latter: Our ability to plan before we act in novel, combinatorial, and compositional environments.

For this question, the phenomenon of *hippocampal replay* is of special interest. While replay is classically viewed as a mechanism of memory consolidation (Nádasdy et al., 1999; Scoville & Milner, 1957; Skaggs & McNaughton, 1996), subsequent evidence finds that it stitches together separately encountered experiences (Gupta et al., 2010) and even traverses entirely novel trajectories (Pfeiffer & Foster, 2013). As a result, replay is now viewed as a process of sampling from a world model that is not plainly restricted to past experience. Consequently, hippocampal replay sequences are also thought to play a central role in Model Based Reinforcement Learning (MBRL), where they can be used for offline training of value and policy networks but also for online planning (Ólafsdóttir et al., 2018; Roscow et al., 2021; Wittkuhn et al., 2021), enabling flexibility critical for generalization within complex and dynamic environments (Hafner et al., 2023).

To serve such functions, hippocampal mechanisms need to be tightly coupled with MBRL roles of the PFC, such as the encoding of abstract and flexible world models (Samborska et al., 2022) and the subsequent derivation of policy and value information from internal plans (Hampton et al., 2006; Miller & Venditto, 2021). However, relatively little is known about the neural circuitry of planning (Jensen et al., 2024; Mattar & Lengyel, 2022), and most work on hippocampal replay deals almost exclusively with rodents and spatial tasks. As a result, recent attempts at modeling forward replay as serial planning within a tightly coupled HPC-PFC circuit also deal with neural correlates and performance in such spatial tasks.

Recent advances have enabled the recording and characterization of replay sequences in humans and in non-spatial tasks (e.g., Kurth-Nelson et al., 2016; Liu et al., 2019). Notably, Schwartenbeck et al. (2023) have found that humans replay sequences appear to simultaneously traverse multiple possible hypotheses and planning trajectories in an abstract combinatorial block construction task. Serial models of replay do not trivially accommodate these results.

In this paper, we examine whether an architectural extension of Jensen et al. (2024) enabling parallel search can capture human performance and neural replay signals in Schwartenbeck et al. (2023)'s block task. In this way, we explore a novel possible neural circuit mechanism implementing replay and planning, building on existing modeling work in reward-based learning circuits involving the PFC (Wang et al., 2018) and the role of replay and HPC in such a system (Jensen et al., 2024). In doing so, we closely follow arguments in Kurth-Nelson et al. (2023) presenting hippocampal replay as compositional computation searching over large spaces, bringing us closer to a formalization of circuit mechanisms required for such a role.

The paper is organized as follows. We first contextualize our work by presenting a brief review of the Jensen et al. (2024) model and the abstract combinatorial block construction task

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

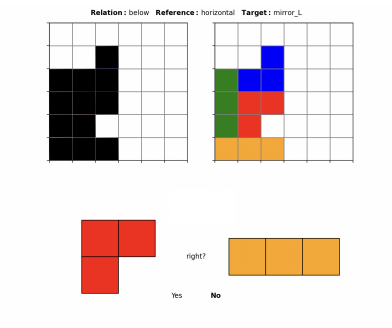

Figure 1: An illustration of the task in Schwartenbeck et al. (2023). On the left is the silhouette to be reconstructed. On the right, a construction in progress. Below, a question appears after the completion of construction, probing the participant on a relation between blocks in the task. Above, the last sampled action from our model is displayed: the model decided to place the horizontal shape below the half-T-like shape.

in Schwartenbeck et al. (2023). We then present our proposed model and report on the performance and replay signals in this same task. We close with a discussion of our results, their implications, and directions for future work.

Background

The Jensen et al. (2024) model targets navigation-based tasks and consists of a GRU (Cho et al., 2014) network that operates in two phases: planning and acting. The inputs to the model mainly consist of the current state, previous action, previous reward, maze wall locations, and elapsed time. Model outputs include logits over the action space and a value prediction. The action space of the model includes all the actions possible in its environment plus whether or not to enter the planning phase. In the planning phase, the model sequentially generates a plan up to a maximum depth or until it reaches its goal state. A learned world model is used to perform planning-phase rollouts, and each planning step is appended as input to the subsequent acting phase. In this model, replay is characterized as on-policy rollouts in the planning phase, with rollouts iteratively improving subsequent acting phase decisions. Importantly, these rollouts correlate with human thinking times and the content of forward replay signals in rodents in (different) maze navigation tasks.

The Schwartenbeck et al. (2023) study targets a different setting. Here, participants are presented with a task consisting of a 6x6 grid, 4 distinct shapes (blocks) to be placed on the grid, and some geometric silhouette to reconstruct (see Figure 1 for a recreation). Each instance of the task has a well-defined solution formed by exactly one set of blocks. In the experiment, participants first complete a training phase, where a target silhouette is presented for 6 seconds per trial, giving participants time to plan. Participants then reconstruct the grid on a screen, after which they are provided with binary correctness feedback. Participants are also presented with questions regarding relational positions on the grid (see Figure 1 for an example). In the test phase, participants are given 3.5 seconds to infer a construction plan and answer questions, but without any feedback nor the ability to lay out constructions on a screen. Rather, they are evaluated on the basis of questions like those in the training phase.

Test silhouettes are composed of exactly three unique blocks and, across instances within a trial, share exactly one block, which is called the *stable* block. As a result, participants could predict one block in each proceeding presentation before actually encountering it. Furthermore, since test grids are composed of three blocks, Schwartenbeck et al. (2023) label blocks attached to the *stable* block as *present*, and those not attached to *stable* but attached to *present* as *distant present*. The block not in a solution is labeled *absent*. Thus, by virtue of the structure of the task and test items, a solution to the task consists of a sequence of block and relation selections that match the given silhouette.

Schwartenbeck et al. (2023) analyze replay sequence data collected using MEG during the inference of construction plans in the test phase. A given replay sequence can itself be characterized as a sequence of length-2 ($A \rightarrow B$) sequences or a length-3 ($A \rightarrow B \rightarrow C$) sequence. Crucially, they find that replay sequences appear to be temporally ordered in a systematic manner, suggesting that the underlying circuits may be conducting hypothesis testing. Concretely, length-2 sequences of blocks reflected a strategy where each of the leftover 3 blocks' relation to the stable block was first tested. This was followed by sequences testing the relation between the two remaining blocks. There was also an additional sequence from *present* to *stable* succeeding these two. They also presented length-3 sequence effects presenting that multiple possible trajectories were explored simultaneously. A reproduction[1] of their analysis is presented in Figure 3.

Model

Here, we propose a model of hippocampal replay that supports parallel planning as suggested by results in Schwartenbeck et al. (2023). Our model is closely inspired by models in Jensen et al. (2024) and Wang et al. (2018). We extend the base architecture of these models to accommodate compositional reasoning and parallel planning. For simplicity, we do not explicitly train a world model, but instead use a non-differentiable model of the environment.

Let G be a set of goal states, S be a set of world states, H be a set of hidden (model/brain) states, A be a set of actions, V be a set of state values, and R be a set of reward values. We define an n-dimensional *frontier state* as a member of the set $S^n \times V^n \times A^n \times R^n$ and our model operates by traversing a sequence of n-dimensional frontier states. The frontier state evolves in a manner defined by the functions $\omega : S \times A \mapsto S$, $\upsilon : G \times S \mapsto R$, $\kappa : G \times S^n \times V^n \times A^n \times R^n \times H \mapsto H$, and $\sigma : S^n \times H \mapsto V^n$, and the distributions $\rho(s, v, r, h)$ over one-hot $n \times n$ matrices and $\pi(s, h)$ over A^n.

[1] with permission

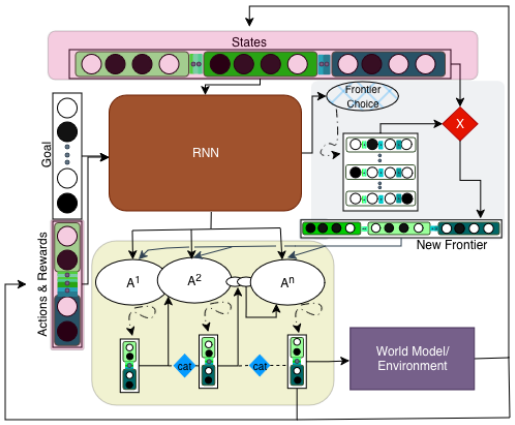

Figure 2: The schematic here presents our model formulation in its full generality. Components in the yellow box represent factored action generation with any number, m, of factors. Colored boxes within larger white boxes denote various vectors with each colored vector belonging to one of n heads being tracked while planning. Each color coded head constitutes information as to the state being tracked in that head and the previous action, reward, and value. Anything contained within a pink box signifies such parallel information tracking. Components within the gray box present a way to sample (with repetition) from and thus manage the frontier for a true search-like capacity.

The function ω is a deterministic world model which maps state-action pairs to new states, υ is a static reward function which maps goal-state pairs to reward values, κ is a learned model kernel which generates a new hidden state from the current goal, frontier state, and hidden state, σ is a learned valuation function which generates a value prediction for a given state frontier, ρ is a routing distribution, derived from the current frontier and model state, selecting a permutation of frontier state, and π is a learned policy distribution which selects actions given a world state and hidden state pair.

The core dynamic of our model is defined by the kernel function κ as follows and is implemented by a GRU as in Jensen et al. (2024).

$$h_t = \kappa(g, s_t, v_{t-1}, a_{t-1}, r_{t-1}, h_{t-1}) \tag{1}$$

In this equation, t indexes the timestep on the current problem, g represents the goal state on the current problem, h_t represents the model's hidden state on timestep t, and $s_t = [s_{t1}, ..., s_{tn}]$, $v_t = [v_{t1}, ..., v_{tn}]$, $a_t = [a_{t1}, ..., a_{tn}]$, $r_t = [r_{t1}, ..., r_{tn}]$ are, respectively, n-headed state, value, action, and reward structures representing the frontier state associated with timestep t. Furthermore, s_{ti} is a state vector, v_{ti} is a scalar state value, a_{ti} is an action vector, and r_{ti} is a scalar reward value.

Parallel Trajectories

Parallel planning in the model is implemented as the traversal of a sequence of n-dimensional frontiers. Values and rewards allow for the effective management of the frontier by the model, where the model can selectively weigh and then dedicate more/less capacity to certain nodes. This is done by first sampling a one-hot *router* matrix, R, followed by a permutation of f_{t-1}:

$$R \sim \rho(f_{t-1}, h_t) \tag{2}$$

$$\hat{f_{t-1}} = R f_{t-1} \tag{3}$$

The gray box in Figure 2 represents this sampling of R followed by the potential reordering of f_{t-1}. In doing so, the model maintains and modifies what is effectively a state-queue, conducting search in a manner similar to algorithmic implementations of $A*$ search (Russell & Norvig, 2010)

At $t = 0$, the model starts in planning mode, denoted with an input of 1 to the model. The initial entry s_0 is copied into all n frontier slots, along with placeholder v_{-1}, a_{-1}, r_{-1} values, and initial hidden state h_{-1}. Equation 1 is stepped to produce h_0. A value prediction is made for each state in this frontier:

$$v_t = \upsilon(s_{t-1}, h_t) \tag{4}$$

Subsequently, we use Equation 2 and then 3 to produce the re-arranged frontier, and then predict an action to take for each state in each of the n frontier nodes:

$$a_{ti} \sim \pi(\hat{s_{(t-1)i}}, g, h_t) \tag{5}$$

Given the new actions, we can now determine the new states using ω:

$$s_{ti} = \omega(\hat{s_{(t-1)i}}, a_{ti}) \tag{6}$$

and obtain resulting rewards:

$$r_t = \eta(g, s_t) \tag{7}$$

The model runs in the planning phase until some pre-specified depth, after which processing switches to the acting phase. The model is provided with a binary input of 0. As for the remaining inputs, since the acting phase can only evaluate one trajectory at a time, input values are copied into all of the n frontier slots. Equations 2 and 3 are skiped and actions are sampled by averaging over the n distributions in Equation 5.

Note that the input size grows linearly in n, and as a result, we do not provide entire planning trajectories to the model as in Jensen et al. (2024).

Action Space

We want to construct a model for success at inherently compositional tasks. Thus, we equip the model to deal with actions that are compositional in nature.

A compositional set of actions A is represented as a cross product of m, the number of compositional factors, component sets $A^k; 1 \leq k \leq m$. Sampling from a categorical distribution over A can be similarly formulated as conditional sampling over m factorized probability distributions defined over A^k. This breaks up Equation 5 into:

$$a_t^k \sim \begin{cases} p(a_t^k | \{a_t^j\}_{j=1}^{j=k-1}, s_t, h_t) & \text{if } k > 1 \\ p(a_t^k | s_t, h_t) & \text{if } k = 1 \end{cases} \tag{8}$$

The yellow box in Figure 2 reflects such a sampling procedure placed within our architecture.

Method

We train our model on the compositional generalization task in Schwartenbeck et al. (2023), present quantitative comparisons of its performance and qualitative comparisons of its neural dynamics against the human data presented in that study.

Design

Our primary experimental variable is model configuration, which has three levels. At the first level, we train RNN models as described in *Model* but without the planning phase, henceforth referred to as *baseline* models. These are compared to similarly sized *planning* model variants, which form the other two levels. The two *planning* variants are trained, respectively, with and without Equations 2 and 3. A model without these equations is very similar to an ensemble, albeit with shared weights for the RNN. So we call the variant that does not make use of these equations *integrated ensemble* and the one that does make use of them *full planning*.

As a secondary factor, models are trained with or without a richer reward structure (see *Reward*). In the case of *planning* models, this means that actions in the acting phase interact with an environment without the richer reward structure, but actions in the planning phase interact with the richer structure.

Materials

We train and evaluate our models on the grid construction phase of the Schwartenbeck et al. (2023) task. We use the same test dataset as Schwartenbeck et al. (2023) (48 unique targets, each composed of 3 unique blocks). This test set is very well-balanced in terms of the relations presented to the participant. In contrast, the training set is not so well-balanced. The original training set consists of 568 training grids. This set proved to be insufficient for good generalization of simple prototype RL models. As a result, here, we generate a synthetic training set of unique grid examples, with grids possessing anywhere between 1 and 4 unique blocks. We then filter out grids that occur in the test set as well as any translations of test grids[2]. We then further filter the dataset so as to represent all relational configurations roughly equally. This final dataset consists of 5710 training examples.

Procedure

States, Actions and the Environment Since participants in Schwartenbeck et al. (2023) are only given a silhouette, the goal state is encoded as a binary 36-dimensional array representing the target silhouette within the grid. Since there are 4 block types, the constructed state vector is represented similarly with entries valued between 0 and 4 inclusive.

The model's action space consists of 52 actions which allow the model to specify the placement of four types of blocks. The first four actions are *start* actions for each of the four blocks. The other 48 actions are *relational actions* of the form (Y, Z, rel), where Y is a reference block already on the grid in progress, Z the block to be placed in some relational, rel, configuration with respect to Y. There are four block types are *half-T*, *horizontal*, *vertical*, and *mirror-L*. The four relation types are *above*, *below*, *left-of*, and *right-of*. This space of 52 actions is factorized along two dimensions: blocks (4) and relations (16 values, for each relation of the form (Y, rel), plus a start value) for a total of 21.

Actions may be *valid* or *invalid*. When an *invalid* action is selected, it is treated as an error, incurring a reward penalty, and fails to induce a transition in the environment. *Start* actions are valid only in the initial timestep and repeated block placements are *invalid*, reflecting the fact that each block could be used exactly once in the original task.

Rewards Following Schwartenbeck et al. (2023), the training phase includes binary correctness feedback, represented as $+1$ or -1. A penalty of -1 is provided every time an invalid action is selected. An invalid action may thus be interpreted as a terminal state in the MDP of this task and, consequently, subsequent generation as bactracking. In addition, we also include a penalty of -0.1 to minimize solution length. A secondary reward signal inversely proportional to the distance from the current in-progress grid silhouette to the goal silhouette is also defined. This signal modifies the scoring for intermediate actions, but leaves incorrect and final action scoring unchanged and is included as it was found to help generalization in initial prototype models.

Training and Inference We train the model, following Wang et al. (2018), on outputs of the acting phase using a single-threaded variant of the actor-critic framework Mnih et al. (2016). We constrain the model to perform at most 10 planning steps. Gradient updates are computed based only on acting phase outputs.

We run *planning* models with a frontier queue size of $n = 4$. Like Wang et al. (2018) and Jensen et al. (2024) an entropy bonus $\beta_{ent} = 0.05$ is used, along with a critic loss weighting of $\beta_{critic} = 0.05$, and a discount factor $\gamma = 0.97$. All models are trained using a GRU of 256 hidden dimensions stacked over 6 hidden layers. All model components are trained with layer norm ($\epsilon = 10^{-5}$; Ba et al., 2016) with bias, and dropout (0.2; Srivastava et al., 2014) using the AdamW optimizer ($\alpha = 10^{-4}; \beta_1 = 0.9; \beta_2 = 0.999;\ wd = 0.01; \epsilon = 10^{-8}$; Loshchilov & Hutter, 2017) with a batch size of 64.

Choice processes for action and router selection are implemented using the gumbel-softmax trick, with straight through gradients enabling the sampling of one-hot vectors from respective distributions (Jang et al., 2016; Maddison et al., 2016).

Analytical Measures

The main performance measures are proportion of correctly constructed grids and mean solution length in timesteps. We also examine the number of batched episodes required to reach grid correctness thresholds within suitable mean solution lengths.

[2]Blocks are not rotationally invariant

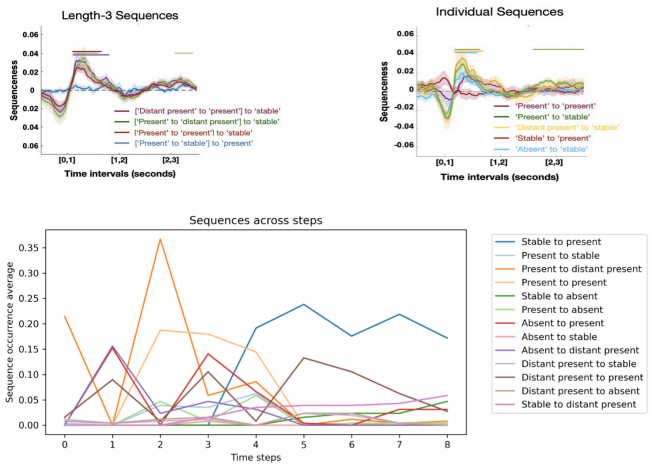

Figure 3: Sequenceness figures from our model and from Schwartenbeck et al. (2023). Bottom presents transitions obtained from the best performing test instance of a planning model variant trained without rich reward structure when acting in the real world. Figures on the top left and top-right present results from Schwartenbeck et al. (2023) on the effects of particular length-2 and length-3 sequences respectively

Correctness thresholds are defined with human performance in mind. In a separate experiment, Schwartenbeck et al. (2023) studied a more complex version of the target task. Since participants do not physically construct arrangements in the target task, we rely on scores from this more complex task. In that task, participants are exposed to increasingly complex grids over trials, and scored well over 90% on levels more similar to our task of interest. So we set our threshold for grid correctness proportion at 90%.

Neural Replay Sequences We examine the model's qualitative fit to human neural response signatures in Schwartenbeck et al. (2023). To do so, we characterize blocks according to their relational configuration in the construction using the *stable*, *present*, *distant present*, and *absent* categories defined by Schwartenbeck et al. (2023) (see Background for details). Since we do not retain state between problem instances nor share a block between them, we have no analog for the *stable* block and thus define the first the block placed by the model in a problem instance as the *stable* block. With this scheme, we explore two possible ways to characterize signals from our models as replay: temporal development of trajectories in the acting and planning phase. For the first, we plot the temporal development of the action probabilities over time during the acting phase of the model, averaged over all grids. We use this technique on all models as a baseline for the qualitative fit of neural sequence analogs in the acting phase. In the planning phase, we interpret the plot as the sequence of trajectories explored: plot actions from the planner across all n heads in the planning phase averaged over the number of grids and heads.

Finally, we compute a loss term L_{div} to track the model's use of search in the planning phase. An exploratory search

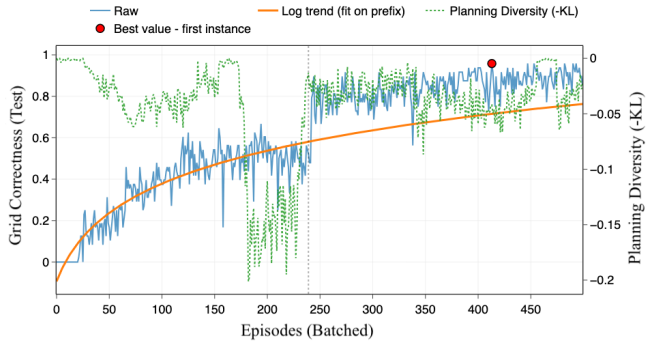

Figure 4: Discrete performance jump correlates with recovery from discrete increase in the pairwise divergence term L_{div}, with performance leaping over the best-fit log trend on the prefix immediately before the jump

policy should result in a larger divergence in the actions chosen among all n action distributions. So we define L_{div} as the summed differences of action distributions, with the KL-divergence $D(.||.)$ as the difference measure and $1 \leq k, l \leq n$.

$$L_{div} = \frac{1}{n^2} \sum_k \sum_l D(\pi(a_t^k | h_t, g, s_{(t-1)k}) || \pi(a_t^l | h_t, g, s_{(t-1)l}) \tag{9}$$

Gradients from L_{div} are not used for training.

Results

All variants attained similar correctness and time-to-criterion on the training set. Most reached the threshold criterion on the test set. However, all *full planning* (n=5) variants reached the performance threshold substantially earlier (median = 614) than their *baseline* (n=5) counterparts (median = 981), with complete separation between groups ($rrb = 1.0$). A Mann-Whitney test was significant ($p < 0.01$), suggesting of a learning-speed advantage for the *full planning* models. This advantage was not observed in the *integrated ensemble* variants (n=5).

Neural signal analog

Neural trajectories obtained from acting phase action probabilities did not display any similarity to hypothesis testing for *planning* and *baseline* variants: mostly flat with peaks around correct transitions over time.

Next, we examine planning trajectories from the planning phase. Plots obtained from tail-end epochs of most models were not representative of actual model solutions, nor any hypotheses testing, but rather static near-flat curves of the same n possibilities over planning time. The only exception to this were plots obtained from models whose internal world models implemented included the richer secondary reward signal. In these models, we observe systematic transition dynamics similar to hypothesis-testing. A representative example is given in Figure 3. Notably, length-3 sequence effects (see Background for details) are well represented in these graphs,

namely *distant present* → *present* → *stable*, *present* → *distant present* →*stable*, and *present* → *present* → *stable*. However, the model is less successful in capturing length-2 sequences.

Temporal Characteristics of L_{div}

An interesting pattern emerges across *planning* variants, and especially in *full-planning* variants. Performance curves initially plateau. Then a sudden increase in L_{div} and a sudden decrease in performance is observed, which is immediately followed by a reduction in L_{div} and a jump in generalization (see Figure 4).

Secondly, we note that L_{div} had large variance over each training run, with L_{div} tending to 0 towards the end. Most models maintained performance thresholds on the test set only after they plateaued on the training set. Although some sporadically reached similar performance levels earlier, we trained them for longer given the large variance in performance. As a result, L_{div} had settled near 0 when model weights for generating search trajectory plots were obtained.

Discussion

In this work, we expand upon Jensen et al. (2024)'s model of forward hippocampal replay to architecturally allow for parallel trajectory evaluation. We evaluate our models on their performance and the capacity to reproduce replay sequences resembling findings in a compositional generalization experiment presented in Schwartenbeck et al. (2023). We find that *full planning* variants learn more efficiently than *baseline* models, though both achieve required generalization. Search trajectories extracted from tail-end epochs of *planning* variants with a richer reward structure in the planning phase reflect structured hypothesis testing as in Schwartenbeck et al. (2023).

Given that most models were trained until tail-end L_{div} was near-zero, the static trajectory plots obtained from these models are symptomatic of the model not actively using search for its performance around this time. In contrast, trajectory plots obtained from the *planning* variant without rich reward structures in its environment reached perfect test generalization well before training set rewards approached the optimal expected value. L_{div} was relatively higher at this point and we thus got graphs bearing sembalance to those in Schwartenbeck et al. (2023). Indeed, examining epoch intervals with high L_{div} produced hypothesis-testing-like search trajectories in *all planning* variants. This suggests that with long training, the model had automatized solving the task and no-longer relied on search for performance (Kahneman, 2011). Similar findings were reported in Jensen et al. (2024), where models trained for longer switched to more reflexive policies. Human performance and replay sequences may thus also be indicative of experience with the task. This is in-line with replay mechanisms playing a major role in our adaptive capacity to generalize from so little.

The temporal characteristics of L_{div} curves also present strong evidence that search was used systematically to improve performance and generalization. During planning, the model briefly explored many different trajectories, and the information gathered during this period appeared to substantially improve the policy used in the acting phase. At the same time, test performance temporarily dipped below the overall trend during periods of extensive search, before recovering once search was actively leveraged. This pattern was unexpected, but closely resembles findings from Gray and Lindstedt (2017). They describe similar dynamics in human skill learning, where short-term declines in performance often precede major improvements. These declines tend to occur during periods of experimentation and strategy discovery, which later produce jumps in performance. Our results may provide a mechanistic explanation of improvements in skill accumulation given models that architecturally allow for search.

Furthermore, search trajectories produced in Figure 3 see a bump after timestep 3 for particular transitions (*stable-present*, *distant-present*, *stable-distant present*). Given the performance of the model at this point and the number of blocks in a test grid, this is the expected number of steps to reach a solution. Given the absence of an explicit stop signal, the model may have appropriated some transitions for this purpose. Indeed, *planning* models that had been trained for long displayed the very same transititons repeated over all timesteps with the same relative frequency. Future work should examine planning with explicit stop signals and the trajectories thus obtained.

Finally, it is not immediately clear how sequences generated by the model should be aligned with the temporal resolution of the signals reported in Schwartenbeck et al. (2023). Consequently, we limited our analysis to a broad characterization of the neural trajectories produced by the model. Future work should average replay sequences across different stages of training and account for sequence timing more carefully in order to enable quantitative comparisons between model-derived signals, the findings of Schwartenbeck et al. (2023), and replay-related signals more generally.

In conclusion, we present a search-based formalization of neural replay in humans. Like Jensen et al. (2024), this is also a theory of planning in the hippocampal-prefrontal network, similarly relying on theories of the PFC as a meta-reinforcement learning system (Wang et al., 2018). With the addition of a general and powerful computation such as search, our results present how relatively little information about the environment may be leveraged through increased computation for faster and more efficient learning. Future work should explicitly examine the adaptability of planning (with or without search) in novel environments with shared structure from past experiences. It may also be worthwhile to examine general pretraining strategies for neural search and any subsequent efficiency benefits to reinforcement learning in structured and complex settings.

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Behrens, T. E., Muller, T. H., Whittington, J. C., Mark, S., Baram, A. B., Stachenfeld, K. L., & Kurth-Nelson, Z. (2018). What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, *100*(2), 490–509.

Cherian, A., Peng, K.-C., Lohit, S., Smith, K. A., & Tenenbaum, J. B. (2023). Are deep neural networks smarter than second graders? *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10834–10844.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Gray, W. D., & Lindstedt, J. K. (2017). Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. *Cognitive science*, *41*(7), 1838–1870.

Gupta, A. S., Van Der Meer, M. A., Touretzky, D. S., & Redish, A. D. (2010). Hippocampal replay is not a simple function of experience. *Neuron*, *65*(5), 695–705.

Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2023). Mastering diverse domains through world models, 2024. *URL https://arxiv. org/abs/2301.04104*.

Hampton, A. N., Bossaerts, P., & O'doherty, J. P. (2006). The role of the ventromedial prefrontal cortex in abstract state-based inference during decision making in humans. *Journal of Neuroscience*, *26*(32), 8360–8367.

Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jensen, K. T., Hennequin, G., & Mattar, M. G. (2024). A recurrent network model of planning explains hippocampal replay and human behavior. *Nature neuroscience*, *27*(7), 1340–1348.

Kahneman, D. (2011). Thinking, fast and slow. *Farrar, Straus and Giroux*.

Kurth-Nelson, Z., Behrens, T., Wayne, G., Miller, K., Luettgau, L., Dolan, R., Liu, Y., & Schwartenbeck, P. (2023). Replay and compositional computation. *Neuron*, *111*(4), 454–469.

Kurth-Nelson, Z., Economides, M., Dolan, R. J., & Dayan, P. (2016). Fast sequences of non-spatial state representations in humans. *Neuron*, *91*(1), 194–204.

Lake, B. M., & Baroni, M. (2023). Human-like systematic generalization through a meta-learning neural network. *Nature*, *623*(7985), 115–121.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, *40*, e253.

Liu, Y., Dolan, R. J., Kurth-Nelson, Z., & Behrens, T. E. (2019). Human replay spontaneously reorganizes experience. *Cell*, *178*(3), 640–652.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maddison, C. J., Mnih, A., & Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Mattar, M. G., & Lengyel, M. (2022). Planning in the brain. *Neuron*, *110*(6), 914–934.

Miller, K. J., & Venditto, S. J. C. (2021). Multi-step planning in the brain. *Current Opinion in Behavioral Sciences*, *38*, 29–39.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *International conference on machine learning*, 1928–1937.

Moskvichev, A., Odouard, V. V., & Mitchell, M. (2023). The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *arXiv preprint arXiv:2305.07141*.

Nádasdy, Z., Hirase, H., Czurkó, A., Csicsvari, J., & Buzsáki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *Journal of neuroscience*, *19*(21), 9497–9507.

Ólafsdóttir, H. F., Bush, D., & Barry, C. (2018). The role of hippocampal replay in memory and planning. *Current Biology*, *28*(1), R37–R50.

Pfeiffer, B. E., & Foster, D. J. (2013). Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, *497*(7447), 74–79.

Roscow, E. L., Chua, R., Costa, R. P., Jones, M. W., & Lepora, N. (2021). Learning offline: Memory replay in biological and artificial reinforcement learning. *Trends in neurosciences*, *44*(10), 808–821.

Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Prentice Hall.

Samborska, V., Butler, J. L., Walton, M. E., Behrens, T. E., & Akam, T. (2022). Complementary task representations in hippocampus and prefrontal cortex for generalizing the structure of problems. *Nature Neuroscience*, *25*(10), 1314–1326.

Schwartenbeck, P., Baram, A., Liu, Y., Mark, S., Muller, T., Dolan, R., Botvinick, M., Kurth-Nelson, Z., & Behrens, T. (2023). Generative replay underlies compositional inference in the hippocampal-prefrontal circuit. *Cell*, *186*(22), 4885–4897.

Scoville, W. B., & Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, *20*(1), 11.

Skaggs, W. E., & McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, *271*(5257), 1870–1873.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent

neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, *21*(6), 860–868.

Wittkuhn, L., Chien, S., Hall-McMaster, S., & Schuck, N. W. (2021). Replay in minds and machines. *Neuroscience & Biobehavioral Reviews*, *129*, 367–388.