A Biologically-Inspired Dual Stream World Model

Arthur Juliani
Department of Psychology
University of Oregon
ajuliani@uoregon.edu

Margaret Sereno
Department of Psychology
University of Oregon
msereno@uoregon.edu

Abstract

The medial temporal lobe (MTL), a brain region containing the hippocampus and nearby areas, is hypothesized to be an experience-construction system in mammals, supporting both recall and imagination of temporally-extended sequences of events. Such capabilities are also core to many recently proposed "world models" in the field of AI research. Taking inspiration from this connection, we propose a novel variant, the Dual Stream World Model (DSWM), which learns from high-dimensional observations and dissociates them into context and content streams. DSWM can reliably generate imagined trajectories in novel 2D environments after only a single exposure, outperforming a standard world model. DSWM also learns latent representations which bear a strong resemblance to place cells found in the hippocampus. We show that this representation is useful as a reinforcement learning basis function, and that the generative model can be used to aid the policy learning process using Dyna-like updates.

1 Introduction

Humans are able to recall and imagine long, temporally extended sequences of events. This capability has been referred to as the brain's 'construction system,' because of the way in which these coherent sequences are constructed during memory recall, imagination, planning, and when dreaming (Hassabis and Maguire, 2009). The capacity to represent coherent temporal sequences of events is tied closely to the ability to skillfully navigate the world around us, due to the existence of what has been referred to as a cognitive map of space in mammals (Tolman, 1948). Both abilities have been localized to hippocampus and surrounding structures, collectively referred to as the medial temporal lobe (MTL) (Tulving and Markowitsch, 1998; O'keefe and Nadel, 1978; Morris et al., 1982).

The spatial context represented by the cognitive map in the hippocampus has been proposed to provide an index for the experiential content of the memory itself, which is stored elsewhere in the cortex (Teyler and DiScenna, 1986). These index representations themselves spontaneously activate in coherent sequences which mirror those of animals during actual experience (Foster, 2017). It has been hypothesized that these capabilities underpin numerous cognitive abilities, from planning to memory consolidation (Pezzulo et al., 2017). There have been a wealth of proposed theoretical models for the MTL (McNaughton et al., 1991; Hasselmo, 2009; Schapiro et al., 2017; Whittington et al., 2019) (See (Behrens et al., 2018) for a recent review).

In parallel, within the field of AI research, generative temporal models, often referred to as 'world models' (Ha and Schmidhuber, 2018) have been developed which are able to emulate the ability of the medial temporal lobe to generate temporally extended sequences of experience. Often these models are used within the context of model-based Reinforcement Learning, where the outcomes of simulated events in the model are used to either learn a value function, or improve a policy (Sutton and Barto, 2018; Hafner et al., 2018).

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Recently proposed world models such as Generative Temporal Model with Memory (GTM-M) and Memory Based Predictor (MBP) have incorporated differentiable memory stores (Gemici et al., 2017; Wayne et al., 2018). Such memory systems allow for adaptation to changes in the environment within a given episode, and can be seen as relating directly to the function of the hippocampus in the MTL (Wayne et al., 2018). Rather than having a single latent variable represent a given state, another class of models, including the Generative Temporal Model with Spatial Memory (GTM-SM) and the Tolman Eichenbaum Machine (TEM), split the representation into context and content variables (Fraccaro et al., 2018; Whittington et al., 2019). This enables the re-use of structural knowledge when faced with novel content within an environment.

Taking inspiration from the construction hypothesis, as well as the recent innovations described above, we propose a novel method which utilizes separate context and content streams, a differentiable memory store, and a forward model over context variables. We refer to this model as a Dual Stream World Model (DSWM), and demonstrate that it outperforms a single stream world model on a series of generative modeling tasks in environments with shared structure but novel content. In addition, it also learns a latent representation which bears a strong resemblance to that of place cells. We demonstrate that this learned representation serves as a useful basis function for downstream reinforcement learning tasks. Furthermore, by utilizing the generative model to perform additional offline learning using the Dyna algorithm (Sutton, 1991; Russek et al., 2017), agents using this state space are able to learn to solve navigation tasks in only a few exposures to the environment.

2 Dual-Stream World Model

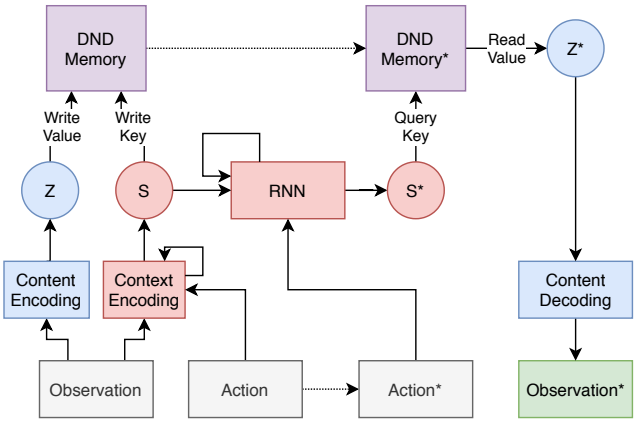

Figure 1: Dual Stream World Model diagram. Blue represents content information. Red represents context information. Purple represents joint content and context information. White represents model inputs. Green represents model outputs. Nodes marked with a $*$ indicate information at the next time step of the simulation.

The DSWM consists of four main components. A content auto-encoder, a context encoder, a forward model (RNN), and a differentiable neural dictionary (DND). Specifically, we utilize a variational encoder with a gumbel-softmax distribution for both the context and content components (Kingma and Welling, 2013; Jang et al., 2016). We implement the forward model using a gated recurrent unit (GRU) (Chung et al., 2014), and use as input both the latent context state s as well as the current action a. The DND is similar to that introduced by Pritzel et al. (2017) and uses the latent context state s as keys, and the latent content state z as values. The lookup process (DND Memory*) uses cosine similarity between a query key ($s*$) and the stored keys to determine a similarity score. The top 5 stored values are then weighted by their similarity scores using a softmax function to derive the retrieved z ($z*$).

This process can be seen as roughly mapping onto the lateral entorhinal cortex (content encoding) (Deshmukh and Knierim, 2011), medial entorhinal cortex (context encoding) (Hafting et al., 2005), and hippocampus (differentiable look-up and forward model) (Hassabis and Maguire, 2009). The model can be seen as an instantiation of the memory indexing theory, whereby the context variable is

used to index the content variable, which itself is an abstracted representation of a high-dimensional observation, which can be thought of as a cortical state (Teyler and DiScenna, 1986).

The following series of steps take place in a given time-step. First a new observation is observed from the environment, and encoded as the latent content variable z_t. In parallel, the observation o_t and the previous action a_{t-1} are used to encode the latent context s_t variable. The inferred context variable s_t and content variable z_t are then stored together as a key-value pair in the DND M_t. The forward model is then unrolled using both the next action a_t the agent takes, and the current inferred context variable s_t to produce a new context variable s_{t+1} that is used to query the memory to read a new content variable z_{t+1}, which is decoded into a predicted observation o_{t+1}. This process is described in Figure 1. Specific details of the network model architecture are presented in A.1.

The DSWM is trained to minimize four objectives. Observation prediction: mean squared error between actual and predicted observations $L_{Obs} = \frac{1}{n}\sum_{n=1}^{N} |o_t^q - o_t^p|^2$. Spatial context prediction: mean squared error between true and predicted position $L_{Pos} = \frac{1}{n}\sum_{n=1}^{N} |pos_t^q - pos_t^p|^2$. Sequence coherence: Kullback–Leibler (KL) divergence between inferred and generated context variables $L_S = D_{KL}(p(s_t|o, s_{t-1})||q(s_{t+1}|s_t, a_t))$. Latent variable regularization: the negative entropy of the context and content variable distributions, which acts as a regularization term.

We use a form of supervision to train the context variable s, based on agent position. We note however that other fully unsupervised loss functions are possible in cases where the environment is not inherently spatial, such as retrieval error during the look-up process.

3 Evaluation Methods

3.1 Generative Modeling Methods

In order to examine the capabilities of the DSWM to predict coherent trajectories of observations, we use a set of environments with a complex topographic structure, partially visible observations, and variability in appearance. Each environment consists of a 2D gridworld, from which the agent can move in the four cardinal directions, but cannot move through walls. Each environment is composed of $11x11$ square units. We use images drawn from a sliding window over a larger visual pattern map juxtaposed on the environment. See Figure 2 for an example of these environment topographies, the pattern maps, and the derived observations.

Each pattern map is generated by randomly selecting a green or red pixel to be placed in each unit of the environment that does not contain a wall. The agent is provided with observations which consist of a $5x5$ window around its current location, which displays the content of the pattern map as well as the location of any walls within the environment. We use environments with four different topographies. These consist of an open area $OpenMaze$, an environment with four connected rooms $RoomsMaze$, an environment with a symmetrical obstacle in the middle $RingMaze$, and an environment with four symmetrical obstacles $HallwayMaze$ (See also Figure 11 for examples of all four topographies). For each of these topographies, we generate 100 different pattern maps to provide a variety of different objects for the agent to observe.

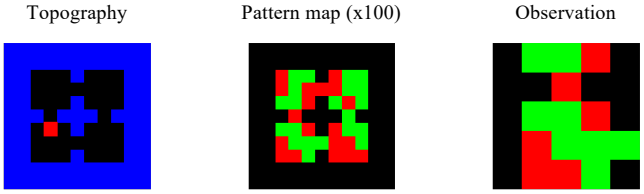

Figure 2: Variable content environment. Left: example environment topography. Blue corresponds to walls. Red corresponds to agent position. Middle: Randomly generated pattern image used to derive observations based on agent location. Right: Agent observations provide a 5x5 window around the agent position.

The datasets used to train each model were collected by running a semi-random behavioral policy for 1000 episodes of 50 steps each. In this case, we create four different datasets, one for each unique topography, and randomly select one of 100 pattern maps to use for each episode.

We compare the proposed DSWM to a single stream world model implementation with similar latent distribution types and capacity (Ha and Schmidhuber, 2018). See Figure 10 in the Appendix for a diagram of this model. Note that during training we reset the DND of the DSWM between each episode, so that stored memories do not carry over. See Table 1 for the complete hyperparameters used to train both the WORLD and DSWM models.

3.2 Reinforcement Learning Methods

We employ a goal-directed navigation task which involves the agent searching for a hidden goal in one of the states of the environment. The reward function consists of a $+1$ reward when the agent reaches the goal state, and 0 reward in all other states. The agent begins each episode in the same location. Halfway through a given training session, in this case, 50 episodes into training, the location of the goal changes to a new position. We use the same set of goal locations for all topographies in order to allow for the consistent comparison between results. See Figure 3 for a visual representation of the goal locations before and after the change for each environment topography.

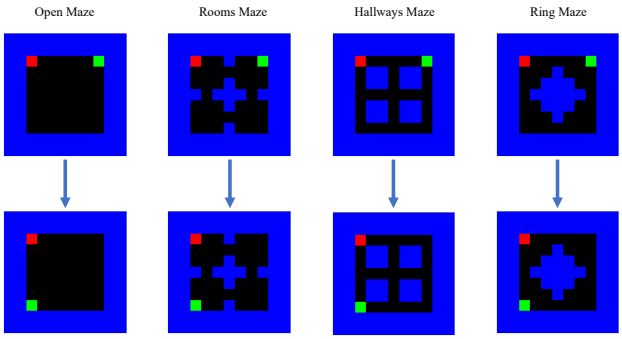

Figure 3: Four different environment topographies, each showing the initial goal location for the first 50 episodes (top) and the second goal location for the following 50 episodes (bottom).

We train agent policies using a modification of the successor feature algorithm (Barreto et al., 2017), chosen for its connection to biological representations within the hippocampus Stachenfeld et al. (2017), as well as its ability to enable rapid adaptation to changes in goal, similar to what is found in animals. In order to address probabilistic state spaces such as those induced by using a distributional representation for s, we replace the dot product between the successor features ψ and the reward function w with a cosine similarity function to derive the Q and V functions. This enables the use of a wider class of representations, including those which are probabilistic over state occupancy. For additional details, see Section A.2.

Each trained agent uses one of three different basis functions: the z distribution from a world model (WORLD), the s distribution from the DSWM, and a pre-computed onehot encoding. We also include an additional variant where we augment the DSWM state space agent with an additional offline learning procedure based on the Dyna algorithm (Sutton, 1991). This offline algorithm uses the forward model of the DSWM to generate trajectories of s states.

Agents are trained for 100 episodes each, with a maximum of 100 steps per episode using an environment from the test set of pattern maps. We reset the DND of the DSWM model between each episode. Each training session is repeated with five separate agent initialization seeds in order to better understand learning dynamics. See Table 2 for all hyperparameters used in these experiments.

4 Experimental Results

4.0.1 Generative Modeling Results

We first compare the prediction accuracy of the models' auto-regressive rollouts in a novel environment. We use a separate set of five held-out pattern maps to create five novel environments for each of the four different topographies to evaluate the models. We collect predictions based on first allowing the agent to run for 30 time-steps within an environment, and then auto-regressively predict the next 20 observations.

We find that for all tested environments the DSWM is able to more accurately predict sequences of observations in these novel environments which were not part of the dataset used for training (DSWM $Mean = 6.025, Std = 6.573$, WORLD $Mean = 8.752, Std = 4.594$, $p < 0.001$). See Figure 4 for the individual losses within each environment. These results suggest that DSWM does indeed have additional generalization ability compared to the WORLD model.

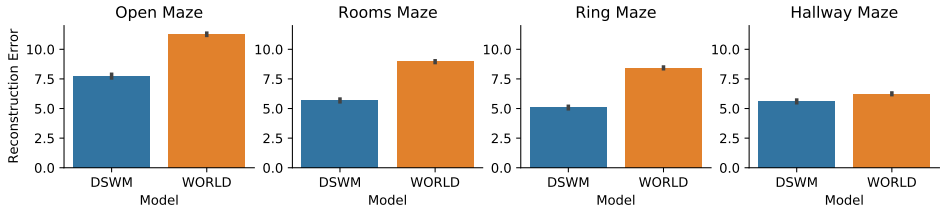

Figure 4: MSE of observation predictions from rollouts of both models in four different environment topographies. Error bars represent standard error. In all environments, DSWM is able to significantly better predict trajectories of future observations than the WORLD model.

We can also inspect qualitatively the predictions produced by each model. Example auto-regressive rollouts from the two models are presented in Figure 5 (Rollout example from all environment topography variations are presented in Figure 8. We can see that while both models are reasonably accurate at predicting the structure of the environment, the WORLD model fails to predict the correct content in novel environments, whereas the DSWM is able to predict both the content and structure. As such, this provides evidence that the DSWM is able to adapt to an environment's novel visual content as long as it retains a familiar topographical structure.

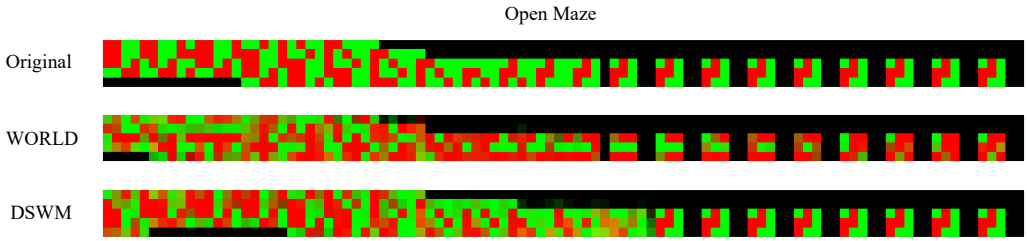

Figure 5: Examples of reconstructed observations from rollouts of both World and DSWM models in the "Open Maze." Environment uses pattern map reserved for testing, and not seen during training. DSWM is able to better predict the true trajectory of future observations within the novel environment.

We next examined the learned latent representations within the DSWM, asking whether the learned representation of the s latent space reflects place-like firing properties. Given the loss function which induces a representation from which the agent position can be decoded, we might expect that such a representation would arise. This is not guaranteed however, since the observations being encoded into s contain both spatial and non-spatial information, and in some cases the non-spatial information dominates the observation.

To answer this question, we qualitatively examine the learned representations of s mapped onto the environment topography. We find that the representations can be best described as indeed being place-like in their firing affinities. See Figure 6 for examples. In particular, we find that the inferred s_t units are highly spatially local, whereas the s_{t+1} units generated by the forward model have wider spatial selectivity. We can draw a hypothetical connection to the CA3 and CA1 regions of the hippocampus, which are hypothesized to be involved in inference and generation, respectively (Teyler and Rudy, 2007).

4.0.2 Reinforcement Learning Results

We present the results of the goal-driven navigation experiments in Figure 7 (Additionally, see Figure 9 for learning curves). This contains the mean and median steps-to-goal of the final 20 episodes

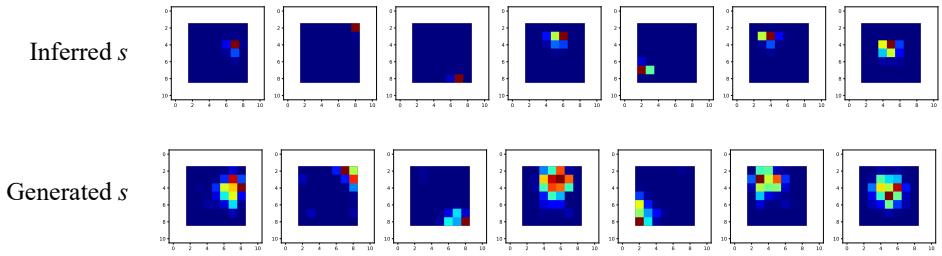

Figure 6: Examples of activations of first fourteen units of inferred and generated s from DSWM model in the “Open Maze" environment topography.

of training for each agent. We find that for all four environments, the state space derived from the DSWM model latent space s is able to match or outperform both the state space derived from the WORLD model latent space s as well as the one-hot state space encoding.

Topography	Optimal	Statistic	WORLD	DSWM	DSWM+DYNA	ONEHOT
Open	5	Mean	32.1	5.81	5.0	7.76
		Median	7.45	5.0	5.0	7.1
Rooms	7	Mean	99.0	23.93	7.04	8.64
		Median	99.0	7.6	7.0	7.55
Ring	5	Mean	99.0	23.8	5.0	5.0
		Median	99.0	5.0	5.0	5.0
Hallway	5	Mean	79.22	5.0	5.0	5.0
		Median	99.0	5.0	5.0	5.0

Figure 7: Statistics from final 20 episodes of each training session for goal-directed agents. DSWM+DYNA results in most consistent learning, with near optimal performance in all four topographies.

We furthermore find that in all environment topographies, the addition of the Dyna algorithm improves the performance of the DSWM state space-based agents, and results in optimal performance (shortest route) for three out of the four environments, with the “Rooms Maze" performance being nearly optimal. We can interpret these results as a clear sign that the learned latent space in the DSWM model is both useful for predicting trajectories of experience in novel environments and in supporting goal-directed navigation in novel environments. Additionally the DSWM+DYNA model performing best suggests that the DSWM has learned a coherent model of the dynamics of the environment which are able to abstract away the specific content of the environment.

5 Conclusion

In this work, we introduced the Dual Stream World Model, a novel generative temporal model which takes inspiration from the ‘construction system’ of the medial temporal lobe. We analyzed this novel model with respect to the coherent generation of trajectories of experience, and found that it is better able to predict future trajectories of experience than a standard world model. Furthermore, we found that the latent context representation within the model bears a strong resemblance to hippocampal place cells, and validated this latent space by demonstrating its usefulness in supporting goal-directed navigation.

The DSWM can be seen as one of a class of recent generative temporal models, such as the Model-Based Predictor (MBP) (Wayne et al., 2018), the Generative Temporal Model with Spatial Memory (GTM-SM) (Fraccaro et al., 2018), and the Tolman-Eichenbaum Machine (TEM) (Whittington et al., 2019). We believe that the DSWM meaningful addition to this growing ensemble of memory-based models of hippocampal learning. It has clearly demonstrated properties of adaptability to changes in environmental content, both in terms of generating coherent trajectories of experience, and in supporting goal-directed navigation, both key properties of a flexible cognitive map.

References

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. (2017). Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065.

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

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Deshmukh, S. S. and Knierim, J. J. (2011). Representation of non-spatial and spatial information in the lateral entorhinal cortex. *Frontiers in behavioral neuroscience*, 5:69.

Foster, D. (2017). Replay comes of age. *Annual review of neuroscience*, 40:581–602.

Fraccaro, M., Rezende, D. J., Zwols, Y., Pritzel, A., Eslami, S., and Viola, F. (2018). Generative temporal models with spatial memory for partially observed environments. *arXiv preprint arXiv:1804.09401*.

Gemici, M., Hung, C.-C., Santoro, A., Wayne, G., Mohamed, S., Rezende, D. J., Amos, D., and Lillicrap, T. (2017). Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*.

Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. (2018). Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801.

Hassabis, D. and Maguire, E. A. (2009). The construction system of the brain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1263–1271.

Hasselmo, M. E. (2009). A model of episodic memory: mental time travel along encoded trajectories using grid cells. *Neurobiology of learning and memory*, 92(4):559–573.

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

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

McNaughton, B. L., Chen, L., and Markus, E. (1991). "dead reckoning," landmark learning, and the sense of direction: a neurophysiological and computational hypothesis. *Journal of Cognitive Neuroscience*, 3(2):190–202.

Morris, R. G., Garrud, P., Rawlins, J. a., and O'Keefe, J. (1982). Place navigation impaired in rats with hippocampal lesions. *Nature*, 297(5868):681.

O'keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon Press.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Pezzulo, G., Kemere, C., and Van Der Meer, M. A. (2017). Internally generated hippocampal sequences as a vantage point to probe future-oriented cognition. *Annals of the New York Academy of Sciences*, 1396(1):144–165.

Pritzel, A., Uria, B., Srinivasan, S., Badia, A. P., Vinyals, O., Hassabis, D., Wierstra, D., and Blundell, C. (2017). Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2827–2836. JMLR. org.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J., and Daw, N. D. (2017). Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS computational biology*, 13(9):e1005768.

Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M., and Norman, K. A. (2017). Complementary learning systems within the hippocampus: a neural network modelling approach to reconciling episodic memory with statistical learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1711):20160049.

Stachenfeld, K. L., Botvinick, M. M., and Gershman, S. J. (2017). The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Teyler, T. J. and DiScenna, P. (1986). The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2):147.

Teyler, T. J. and Rudy, J. W. (2007). The hippocampal indexing theory and episodic memory: updating the index. *Hippocampus*, 17(12):1158–1169.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4):189.

Tulving, E. and Markowitsch, H. J. (1998). Episodic and declarative memory: role of the hippocampus. *Hippocampus*, 8(3):198–204.

Wayne, G., Hung, C.-C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J., Mirowski, P., Leibo, J. Z., Santoro, A., et al. (2018). Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*.

Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T. E. (2019). The tolman-eichenbaum machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv*, page 770495.

A Supplemental Material

A.1 Dual Stream World Model Implementation Details

At each time-step of simulation, the Dual Stream World model operates in two phases, an inference and a generation phase. These phases are governed by the following equations.

Inference phase:

$$z_t \sim p_{enc}(z_t|o_t) \tag{1}$$
$$s_t \sim p_{enc}(s_t|o_t) \tag{2}$$
$$M_t = f_{write}(M_{t-1}, s_t, z_t) \tag{3}$$
$$h_t = f_{forward}(s_t, a_t, h_{t-1}) \tag{4}$$

Generation phase:

$$s_{t+1} \sim q_{forward}(s_{t+1}|s_t, a_t, h_t) \tag{5}$$
$$z_{t+1} \sim q_{read}(z_{t+1}|M_t, s_{t+1}) \tag{6}$$
$$o_{t+1} = f_{decode}(z_{t+1}) \tag{7}$$

This flow, as well as that of the WORLD model used as a comparison baseline are concretely implemented as a fully-differentiable neural networks written using the PyTorch framework (Paszke et al., 2019). In DSWM, $p_{enc}(z_t|o_t)$ and $p_{enc}(s_t|o_t)$ are implemented as three layer multi-layer perceptrons with 256 hidden units each, using the $swish$ activation function after each layer (Ramachandran et al., 2017), except for the final layer, which consists of a gumbel-softmax distribution (Jang et al., 2016). $f_{forward}(s_t, a_t, h_{t-1})$ and $q_{forward}(s_{t+1}|s_t, a_t, h_t)$ are implemented as a gated recurrent unit (GRU) with a hidden layer size of 256 units (Chung et al., 2014). $f_{write}(M_{t-1}, s_t, z_t)$ and $q_{read}(z_{t+1}|M_t, s_{t+1})$ are implemented as a differentiable neural dictionary (DND) (Pritzel et al., 2017), with a cosine similarity look-up function. Lastly, $f_{decode}(z_{t+1})$ is implemented as a three layer multi-layer perceptron with $swish$ activation functions after each layer, except for the last, which utilizes a 'sigmoid' activation function.

A.2 Successor Similarity Algorithm

We use a modified form of the successor feature algorithm described in (Barreto et al., 2017). The traditional formulation of policy learning using successor features consists of two functions, a reward function $w(s')$ and a successor function $\psi(s, s')$. These are updated using temporal difference learning as follows.

$$\delta_w = r_t - w(s') \tag{8}$$
$$w(s')' = w(s') + \alpha_w \delta_w \tag{9}$$

Where α_w corresponds to the reward learning rate.

$$\delta_\psi = s_t + \gamma\psi(s_{t+1}, a_{max}) - \psi(s_t, a_t) \tag{10}$$
$$\psi(s_t, a_t)' = \psi(s_t, a_t). + \alpha_\psi \delta_\psi \tag{11}$$

Where α_ψ corresponds to the successor learning rate. a_{max} corresponds to the action with the highest expected value, derived from the value function $Q(s, a) = \psi(s, a) \cdot w(s')^T$. This equation can also be used to derive a policy, where the Q function can be used to derive a categorical distribution using a softmax function, i.e. $\pi(a|s)$.

While this works well in state spaces where each value in the state vector is independent of all others, it breaks down in cases where there is a mutual dependence. One example of this is over probabilistic state space representations, where a state vector $< s >$ would correspond to a belief state $< b >$ over state occupancies. In this scenario, there is no linear function of the state vector s and a reward vector w which would produce 1 when the agent is in the reward state and 0 in all other states.

In order to address this issue, we replace the dot product with a cosine similarity function ($cos(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$). This allows us to use probabilistic state representations from a generative temporal model as the state space, while bounding the reward and value functions between 0 and 1. We also replace the reward function update rule with one where $w(s')$ is set to s' if a rewarding state is encountered. We additionally set $w(s')$ to a zero vector if the predicted reward was greater than 0.9, but a reward was not received in a given state . We note that this method is only viable in environments in which only a single state is rewarded at a time. Such a requirement however is not a limitation in goal-directed navigation tasks such as the ones performed here or often used in animal research.

A.3 Learning Hyperparameters

Generative Modeling Hyperparameters	
Parameter	Value
z total size	128
z number distributions	8
s total size	49
s number distributions	1
Learning rate	5e-4
h size	256
β_z	0.001
β_s	0.001
Iterations	5000
Batch size	3

Table 1: Hyperparameters used in WORLD and DSWM models in generative modeling experiments.

Reinforcement Learning Hyperparameters	
Parameter	Value
γ	0.99
α	0.1
Dyna rollout length	5
Dyna rollout frequency	0.2
τ	0.001

Table 2: Hyperparameters used for WORLD and DSWM models in reinforcement learning experiments.

A.4 Additional Figures

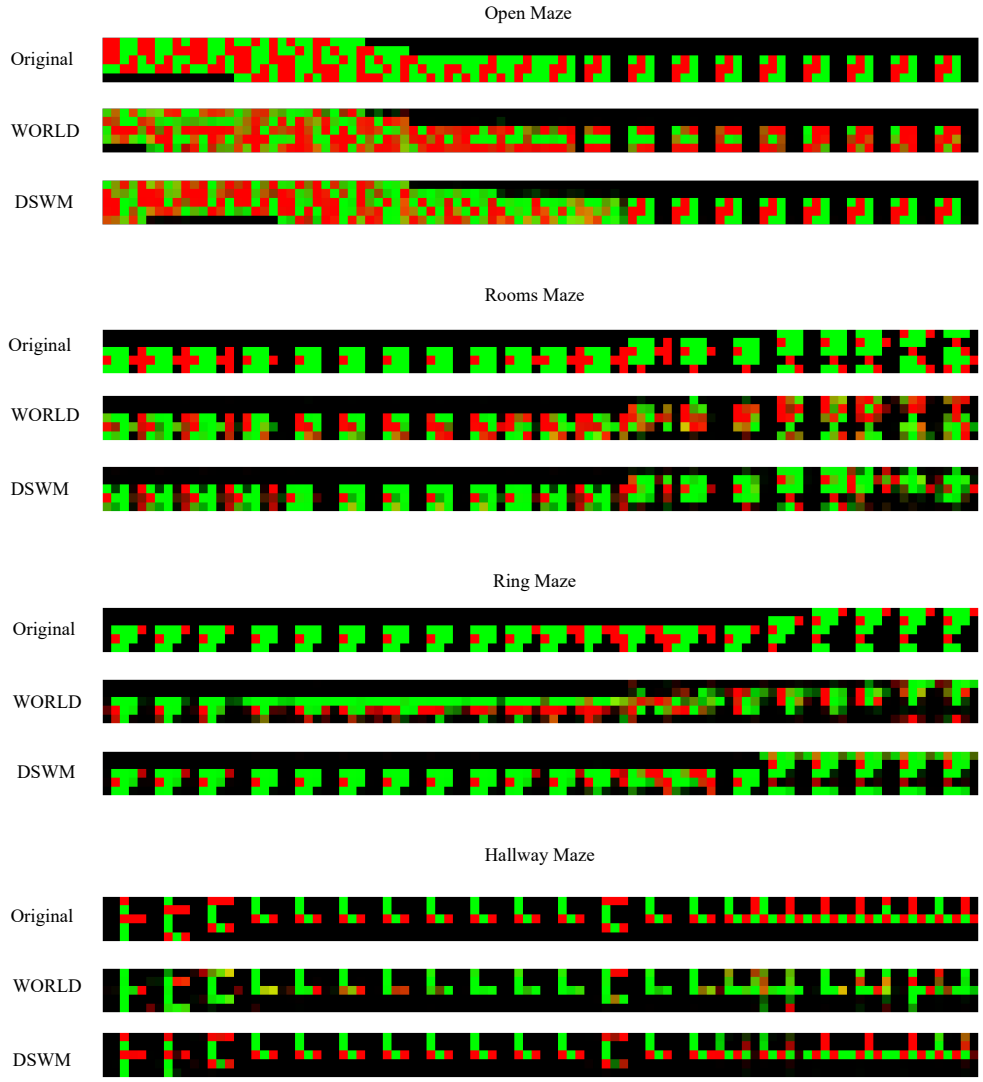

Figure 8: Examples of reconstructed observations from rollouts of both World and DSWM models in all four environment topographies. Environments use pattern maps reserved for testing, and not seen during training. DSWM is able to better predict the true trajectory of future observations within all novel environments.

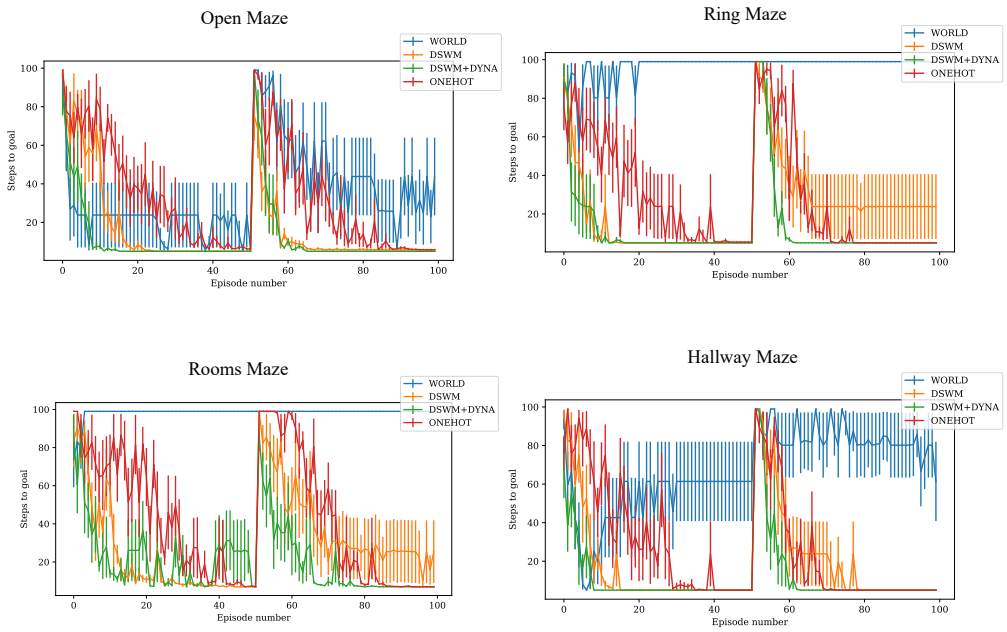

Figure 9: Learning curves in goal-directed navigation task for each of the four unique environmental topographies. Each curve represents the average of five separate initialization seeds for the agent. Error bars represent standard error.

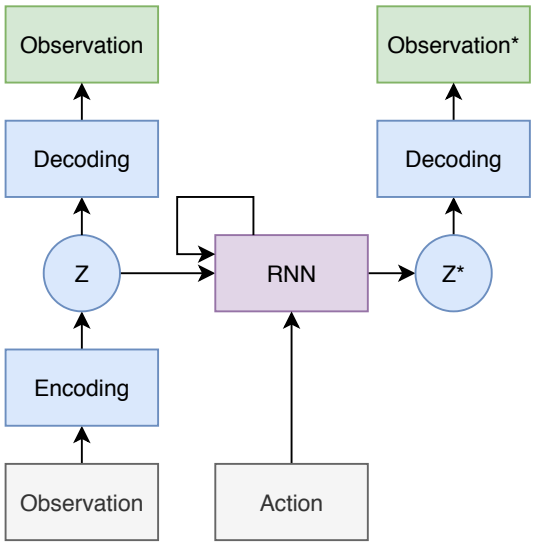

Figure 10: Single Stream World Model diagram. Blue represents content information. Purple represents joint content and context information. White represents model inputs. Green represents model outputs. Nodes marked with a $*$ indicate information at the next time step of the simulation.

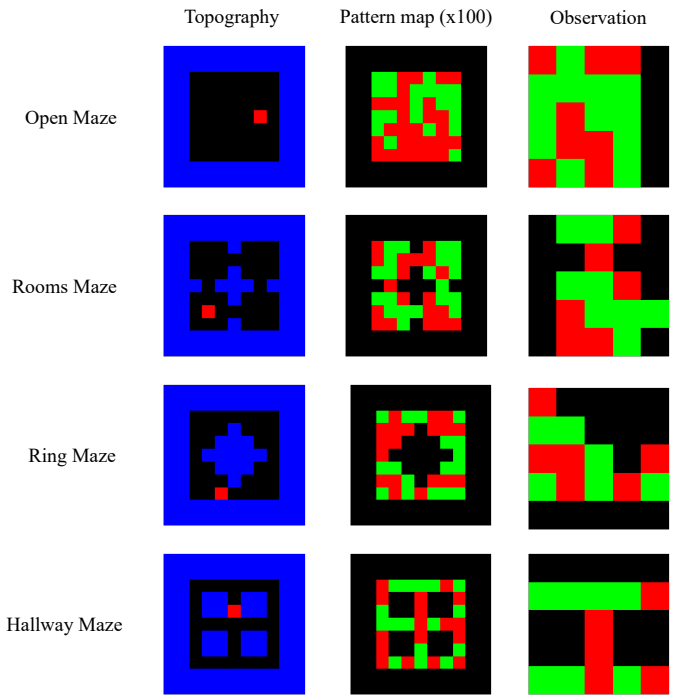

Figure 11: Variable content environments with different topographies. Left: example environment topography. Blue corresponds to walls. Red corresponds to agent position. Middle: Randomly generated pattern image used to derive observations based on agent location. Right: Agent observations provide a 5x5 window around the agent position.