A SOME PROOFS IN SECTIONS 3

Proposition A.1 *Given our assumption of periodic boundary condition, the transition matrix, $T \in \mathbb{C}^{N\times N}$ (eq. 3), is indeed a circulant matrix.*

Proof It is easy to see that the proposition holds trivially for transition matrices with only one-step translations but without Gaussian spread. Hence here we only show the proof for the case where the transition structure includes both Gaussian spread and one-step translations.
Consider for an arbitrary transition matrix T for a 2D rectangular environment with length L and width W, and the underlying transition velocity is $v = (v_x, v_y)$, remembering that $\mathbf{T}$ is the $LW \times LW$ 2D matrix formed from concatenating rows from what would be the 4D matrix of transitions between all pairs of states in a $L \times W$ 2D state space. An arbitrary entry on the kth lower subdiagonal is $T_{i,i-k} = \mathbb{P}(x(t+1) = i - k | x(t) = i)$ for any suitable state i given k (i.e., $i \geq k$). If the Gaussian spread is radially symmetric with constant variance across states, the value of $T_{i,i-k}$ only depends on the distance between state $i-k$ and the state i_v, where i_v is the translated state of state i given the effect of the velocity v. The states $i-k$ and i_v are equivalent to the states $((i-k)//L, (i-k) \bmod L)$ and $(i//L + v_x, i \bmod L + v_y)$ in the two-dimensional spatial domain respectively (where $a//b$ denotes the integer part of a/b). Note that we need to have the velocity $v \in [\pm L/2, \pm W/2]$ so that the translation leaves the actual distance d unchanged. The distance between the state $i-k$ and the expected state i_v in the 2D state space is then

$$d = \sqrt{((i-k)//L - i//L - v_x)^2 + ((i-k) \bmod L - i \bmod L - v_y)^2} \tag{16}$$

For any arbitrary $i' \neq i$ such that $i' = i + m$, we could compute similarly the distance between state $i' - k$ and its corresponding expected state (Gaussian center) $i' + v$. After some algebra, we have that the distance between states $i-k$ and $i+v$ equals the distance between states $i'-k$ and $i'+v$.

$$\begin{aligned} d' &= \sqrt{((i'-k)//L - i'//L - v_x)^2 + ((i'-k) \bmod L - i' \bmod L - v_y)^2} \\ &= \sqrt{((i+\delta-k)//L - (i+\delta)//L - v_x)^2 + ((i+\delta-k) \bmod L - (i+\delta) \bmod L - v_y)^2} \end{aligned} \tag{17}$$

Now if we look at the two square terms within the square root separately, we have

$$\begin{aligned} &((i+\delta-k)//L - (i+\delta)//L - v_x)^2 \\ &= ((i-k)//L + \delta//L - i//L - \delta//L - v_x)^2 \\ &= ((i-k)//L - i//L - v_x)^2 \end{aligned} \tag{18}$$

$$\begin{aligned} &((i+\delta-k) \bmod L - (i+\delta) \bmod L - v_y)^2 \\ &= (((i-k) \bmod L + \delta \bmod L) \bmod L - (i \bmod L + \delta \bmod L) \bmod L - v_y)^2 \\ &= ((i-k) \bmod L + \delta \bmod L - i \bmod L - \delta \bmod L - v_y)^2 \\ &= ((i-k) \bmod L - i \bmod L - v_y)^2 \end{aligned} \tag{19}$$

The second equality holds due to the fact that $(i-k) \bmod L + \delta \bmod L \leq L$ since this is simply the x-position of state $i+\delta-k$, which is never larger than L, hence $((i-k) \bmod L + \delta \bmod L) \bmod L = (i-k) \bmod L + \delta \bmod L$. Similarly $(i \bmod L + \delta \bmod L) \bmod L = i \bmod L + \delta \bmod L$. Hence we have

$$d' = \sqrt{((i'-k)//L - i'//L - v_x)^2 + ((i'-k) \bmod L - i' \bmod L - v_y)^2} = d \tag{20}$$

Hence all entries on the kth lower subdiagonal are identical, i.e. $T_{i,i-k} = T_{i',i'-k}$ for all $1 \leq k \leq LW - 1$. And by similar arguments, we could show that all entries on the kth upper subdiagonal are identical (for $1 \leq k \leq LW - 1$), and equals to the corresponding entries on the $LW - k$th lower subdiagonals. And the fact that all the main diagonal entries are identical is immediate from the problem setting. Hence our target transition matrix is indeed a circulant matrix. (Note that in simulations the transition matrix will only be approximately circulant due to normalisation and numerical issues.)

Now we consider the corresponding $(LW-k)$th upper subdiagonal (to the kth lower subdiagonal), by similar arguments, we have that for any $T_{i'',i''+k} = \mathbb{P}(x(t+1) = i''+k | x(t) = i'')$ for suitable i'' (i.e. $i''+k \leq L$), the distance between the state $i''+k$ and expected next state $i''+v_t$ are the same as d, which is equivalent to $T_{i'',i''+k} = T_{i,k-i}$. Hence all entries on the $(LW-k)$th upper subdiagonal are identical and equal to the entries on the kth lower subdiagonal. This holds for arbitrary $1 \leq k \leq LW-1$. □

Proposition A.2 *For any circulant matrix $T \in \mathbb{C}^{N\times N}$ as shown in eq. 3, its kth eigenvector takes the form:*

$$\mathbf{v}^k = \frac{1}{\sqrt{N}} \left[1, \omega_k, \omega_k^2, \cdots, \omega_k^{N-1}\right]^T \tag{21}$$

where $\omega_k = \exp\left(\frac{2\pi k i}{N}\right)$ is the kth Nth root of unity, and the set of eigenvalues equals to the set of DFTs of an arbitrary row/column of T.

Proof Firstly, note that the product between the circulant matrix T and an arbitrary vector $\mathbf{v}$ is equivalent to a convolution.

$$\mathbf{w} = T \cdot \mathbf{v} = \begin{bmatrix} T_0 & T_{N-1} & \cdots & T_2 & T_1 \\ T_1 & T_0 & T_{N-1} & \cdots & T_2 \\ \vdots & T_1 & T_0 & \ddots & \vdots \\ T_{N-2} & \cdots & \ddots & \ddots & T_{N-1} \\ T_{N-1} & T_{N-2} & \cdots & T_1 & T_0 \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix} \tag{22}$$

And we immediately have that

$$w_k = \sum_{j=0}^{N-1} T_{j-k} v_j \tag{23}$$

This is true due to the periodicity of the entries given by the circulant structure. Then if we take the dot product of T and an arbitrary vector $\mathbf{v}^m$ of the form shown in eq. 21, the lth entry of the output vector has the following form.

$$\sum_{j=0}^{N-1} T_{j-l} \omega_j^m = \omega_l^m \sum_{j=0}^{N-1} T_{j-l} \omega_{j-l}^m \tag{24}$$

where the equality holds since $\omega_j^m = \exp\left(\frac{2\pi i}{N} jm\right) = \exp\left(\frac{2\pi i}{N}(j-l)m\right) \exp\left(\frac{2\pi i}{N} lm\right) = \omega_l^m \omega_{j-l}^m$. Note that the last sum in Eq. 24 is independent of the choice of l since both T_j and ω_j are periodic hence any change in l is simply rearranging the terms in the summation. Also we have that $\omega_l^m = \omega_m^l$ is the lth entry of the eigenvector v_m. Hence we have

$$T\mathbf{v}_m = \lambda_m \mathbf{v}_m \tag{25}$$

where

$$\lambda_m = \sum_{j=0}^{N-1} T_j \omega_j^m \tag{26}$$

for $m = 0, \ldots, N-1$. Hence for an arbitrary $N \times N$ circulant matrix T, the eigenvalues take the form as shown in eq. 26 and the corresponding eigenvectors take the form as shown in eq. 21, and the eigenvalues are equivalent to the DFT of the first row of the circulant matrix immediately follows from eq. 26 and the definition of DFT (Bracewell [4]). □

The predicted phase change in the eigenvalues over the eigenvalues of the baseline symmetric transition matrix computed with Fourier modes computed via Fourier shift theorem (eq. 8) under our formulation perfectly captures the actual phase changes caused by the one-step translations in the eigenvalues between the symmetric and and asymmetric transition matrices, as shown in Fig. 6A. However, when the transition dynamics is a combination of diffusion and one-step translations, the predicted phase changes in eigenvalues will no longer perfectly match the actual phase changes observed as shown in Fig. 6B, and the oscillation is caused by the diffusion process. Namely, although the expected translation is indicated by the velocity, the actual translation spans a range of states depending on the width of the diffusion field.

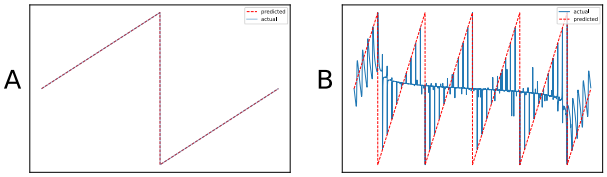

Figure 6: **Application of Fourier shift theorem for predicting changes in eigenvalues**. The blue lines and red dotted lines represent the actual and predicted shifts in complex phases of the eigenvalues of transition matrices given some drift velocity and the symmetric transition matrix where the transition dynamics are **A**: translation only (5 units rightward); **(B)**: diffusion with one-step translations (5 states rightward + diffusion).

Proposition A.3 *The updated SR given the insertion of a barrier is*

$$S = S_0 - C(I + RC)^{-1}RS_0 \tag{27}$$

where S_0 and S are the initial and updated SR, $R = S_0[J, :]$ and $C = S_0[:, J]$ are the J-th rows and columns of S_0 respectively, where J is the index set of states adjacent to the inserted barrier.

Proof This derivation is inspired by Piray and Daw [35]. Given the definition of the SR, we have

$$S = (I - \gamma T)^{-1}, \quad S_0 = (I - \gamma T_0)^{-1} \in \mathbf{R}^{N\times N} \tag{28}$$

where N is the number of states. Given the insertion of a barrier, S and S_0 only differ in their j-th rows for $j \in J$ where J is the index set of states adjacent to the barrier. Hence we could write:

$$R = T[J, :] - T_0[J, :] \in \mathbb{R}^{|J|\times N} \tag{29}$$

Then if we have $E \in \mathbb{R}^{|J|\times N}$ with zeros everywhere but ones on the j-th rows for $j \in J$, then by setting $W = I - T$ and $W_0 = I - T_0$, we could write:

$$W = W_0 + ER \tag{30}$$

The Woodbury inversion formula is usually use in cases whn we are trying to compute the inverse of a matrix given a low-dimensional perturbation (Riedel [37]).

$$: (A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \tag{31}$$

Hence by applying the Woodbury inversion formula, we have:

$$\begin{aligned} W^{-1} &= W_0^{-1} - EW_0^{-1}(I + REW_0^{-1})^{-1}RW_0^{-1} \\ \Rightarrow S &= S_0 - C(I + RC)^{-1}RS_0 \end{aligned} \tag{32}$$

where $C = ES_0$ are the j-th columns of S_0 for $j \in J$. □

B Some proofs in Section 4

Proposition B.1 *The "sense of direction", θ^*, is given by the form shown in eq. 9*

Proof Essentially, we wish to find value of θ such that under the drift velocity $\mathbf{v}_\theta = (v\cos(\theta), v\sin(\theta))$, given the start and target states, $\mathbf{s}_0$ and $\mathbf{s}_G$, the future discounted occupancy of $\mathbf{s}_G$ starting from $\mathbf{s}_0$ (or $W[\mathbf{s}_0, \mathbf{s}_G]$, where W is the SR matrix), is maximised. Under our formulation, W can be calculated as follows:

$$W = F\text{diag}(1/(1 - \gamma\Lambda^{\mathbf{v}_\theta}))F^{-1} \tag{33}$$

where F is the DFT matrix (eq. 4), and $\Lambda^{\mathbf{v}_\theta}$ is the set of eigenvalues of the transition matrix given velocity $\mathbf{v}_\theta$. From our analysis based on Fourier shift theorem (eq. 8), for each $\lambda_i^{\mathbf{v}_\theta} \in \Lambda^{\mathbf{v}_\theta}$, we have that:

$$\lambda_i^{\mathbf{v}_\theta} = D_i\omega^{\mathbf{v}_\theta\cdot\mathbf{k}_i} \tag{34}$$

where D_i is the ith eigenvalue of the symmetric (baseline) diffusion transition matrix, and $\mathbf{k}_i$ is the wavevector for the ith Fourier mode. Then using linear algebra, we immediately arrive at the expression in eq. 9. □

Proposition B.2 *The equations governing the dynamics of the normative prediction model (eq. 11) and the mechanistic CAN model (eq. 12 are equivalent.*

Proof Substituting eq. 10 into eq. 11, we have:

$$\frac{dg}{dt} = \begin{cases} -\alpha g + \frac{2\pi i}{N}\gamma \mathbf{T}^0 g + (\frac{2\pi i}{N}\gamma \sum_{j=1}^{G} \lambda_j w_j f_j(\langle v, \hat{\mathbf{e}}_j \rangle - 1) + \mu), & g > 0 \\ -\alpha g + \frac{2\pi i}{N}\gamma \mathbf{T}^0 g + \sigma(\frac{2\pi i}{N}\gamma \sum_{j=1}^{G} \lambda_j w_j f_j(\langle v, \hat{\mathbf{e}}_j \rangle - 1) + \mu), & g = 0 \end{cases} \tag{35}$$

Now check with eq. 12 by setting $\tau = \frac{1}{\alpha}$, $\mathbf{W} = \frac{2\pi i}{N}\frac{\gamma}{\alpha}\mathbf{T}^0$, $b(v) = (\frac{2\pi i}{N}\gamma \sum_{j=1}^{G} \lambda_j w_j f_j(\langle v, \hat{\mathbf{e}}_j \rangle - 1) + \mu)/\alpha$, where $\hat{\mathbf{e}}_j$ is the unit vector in the direction of the j-th eigenvector, and checking that when $g > 0$, $\sigma(\frac{2\pi i}{N}\gamma \mathbf{T}^0 g + \sigma(\frac{2\pi i}{N}\gamma \sum_{j=1}^{G} \lambda_j w_j f_j(\langle v, \hat{\mathbf{e}}_j \rangle - 1) + \mu)) = \frac{2\pi i}{N}\gamma \mathbf{T}^0 g + \sigma(\frac{2\pi i}{N}\gamma \sum_{j=1}^{G} \lambda_j w_j f_j(\langle v, \hat{\mathbf{e}}_j \rangle - 1) + \mu)$, we see that under non-zero velocity inputs, by appropriately adjusting the additive velocity input term, $b(v)$, the equations governing the dynamics for the normative and mechanistic models are equivalent. □

C 2D Fourier modes

We know that the Fourier basis vectors from eq. 4 form plane waves as shown in Fig. 5. From standard Fourier analysis in 2D space, the 2D Fourier modes form an orthonormal basis, and takes the following form.

$$\mathbf{v}_{\mathbf{u}}[\mathbf{x}] = \exp\left(2\pi i \mathbf{u} \cdot \mathbf{x}\right) \tag{36}$$

where the 2D Fourier basis vectors are encoded by the position vectors $\mathbf{u} = (u_1/L, u_2/W) \in [0,1] \times [0,1]$ (position vectors of each location in the $L \times W$ environment projected onto $[0,1] \times [0,1]$). The direction of the encoder position vector $\mathbf{u}$ represents the direction of the plane wave and the frequency of the plane wave is the unnormalised direction vector $||\mathbf{u}'||$ (where $\mathbf{u}' = \mathbf{u} \times (L, W)$), note that $\mathbf{u}'$ is also the wavevector for the plane wave. This is a slightly different formulation comparing to the formulation given in eq. 4, which consider the state space as a 1-dimensional flattened vector of the 2-dimensional environment, hence the Fourier basis vectors are the corresponding 1-dimensional Fourier modes. Though both formulation give us the same set of Fourier basis vectors, under the definition in eq. 36, we could easily track the frequency and direction of the plane wave formed from the 2D Fourier modes. And the phase shift via the Fourier shift theorem 8 equivalently applies for this 2D Fourier formulation.

The Fourier modes comprises a basis for representing any distribution over the task state space, so we could use a linearly weighted combination of Fourier modes to reconstruct any firing patterns, such as those observed in place cells (Welday et al. [42], Fig. 8). However note that the coincidence detection of small numbers of oscillators with different frequencies will generate periodic patterns, e.g., grid cells, and more oscillators will be needed for those with more local firing fields such as place cells. Note that the total number of Fourier modes equals the number of states in the environment (e.g., LW for the $L \times W$ rectangular environment on a square grid), and it could be infeasible and inefficient to compute and store a large number of such Fourier modes (or neurons with VCO-like firing patterns) in the brain. Hence here we only use the principal modes (taking the top n Fourier basis vectors in terms of the corresponding eigenvalues (frequencies)), within contain the majority of the information is contained, with the number of principal modes depending on the desired reconstruction resolution. We utilised the top 100 principal Fourier modes for most of the simulations in the main text (see Fig. 7 for a typical fixed set of Fourier modes). Fig. 8 demonstrates that the small number of Fourier modes are able to reconstruct grid cells firing fields with various spacings and orientations, and place cells firing fields.

D Transitive Inference

In the main paper we argued that the same set of eigenvectors can be used to predict future occupancy distribution given the transition matrix for symmetrical relations like diffusion between adjacent states and directed transitions (e.g. moving N S E W). Here we briefly discuss the generalisation of our model to non-spatial tasks.

We could apply our method to the one-dimensional transitive inference tasks of this type. e.g., given $A > B$, $B > C$, $C > D$, then infer if $A > D$ (Von Fersen et al. [41])? This would be like having a

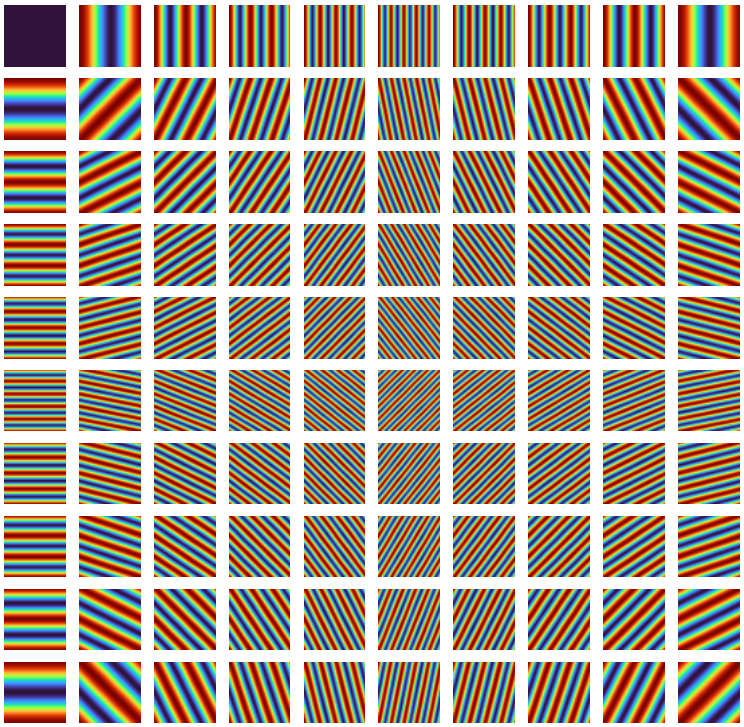

Figure 7: Phase plots of 100 chosen low-frequency Fourier basis vectors with different frequencies and wavevectors.

1D track (and Fourier eigenvectors for 1-step transitions in both directions) corresponding to actions "greater" or "smaller", and using "intuitive planning" to see if using eigenvalues for "greater" will take you from A to B in the discounted future more likely than eigenvalues for "smaller". In order to deal with the non-periodicity of the task, we simulate transitive inference in a small subset of the state space of the torus. As shown in Fig. 9, we see that our framework correctly predicts the transitive relationship between the chosen state x_{129} and states close to x_{129}.

Despite the simplicity of 1D transitive inference (Fig. 9), our model is still and advance on the original intuitive planning method which was restricted to symmetric transition structures, and so indicates distance but not direction.

E SIMULATIONS

E.1 FURTHER DETAILS OF THE GC-DQN AGENT

The overall architecture can be found in the graphical illustration in fig. 4. At each timestep, the state values and a specific action value are fed into a neural network for all possible values of actions (Blue box in the bottom left of fig. 4), which outputs $n_{actions} \times n$ output, where n is the number of Fourier modes inputs to the second network. The output can considered as the specific updates to each Fourier modes corresponding to the action in the current location, like the phase shift in the Fourier shift theorem.

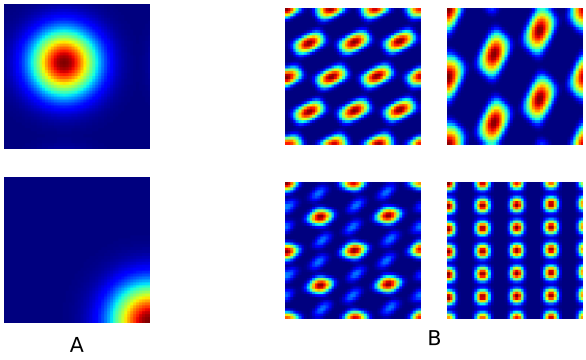

Figure 8: **Constructed place cell and grid cell firing fields from Fourier modes.** **A**: Place cells firing fields constructed from coincidence detection of selected input Fourier modes (bottom plot shows a place field restricted to a small subset of the toroidal state space); **B**: Grid cell firing fields with various spacings and orientations constructed from principal Fourier modes (Fig. 7).

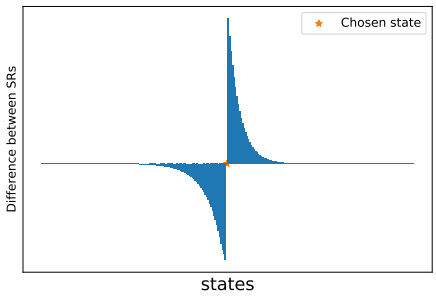

Figure 9: **Generalisation of flexible planning on transitive inference task**. Given $\{x_i\}_{i=0}^{259}$ such that $x_i > x_{i+1}$ for all i (and $x_{259} > x_0$ for ensuring the circulant structure). The bar plot shows that we can correctly infer transitive relations between the chosen state x_{129} (red star) and nearby target states via computing the difference between the discounted future occupancy of the target state under the action-dependent SRs (eq. 2) corresponding to the "smaller" (left) and "greater" (right) actions. The x-axis denotes the states, and the y-axis denotes the difference between the SRs.

The inputs to the grid cell network are the first n Fourier modes, whose dimensions (D) are determined by the size of the state space. When the state variables are continuous, we compute an approximate size of the state space by discretising each state variable. The number of principal Fourier modes (n) is chosen arbitrarily as long as the majority of the information can be reconstructed from the chosen set of Fourier modes. Higher values of n leads to finer details of the prediction, but also induces higher computational costs.

At each timestep, the n Fourier modes is fed as the input to the grid cell network (shown in the middle row of fig. 4(A)). Each action multiplier (outputs from the state-action network) is multiplied with the corresponding column of the weight matrix between the input layer and the hidden layer of the grid cell network. The outputs of the hidden layer is then transposed, and forward propagate to the output layer of the second network. The computations of the grid cell network is considered to be equivalent to using the Fourier modes to construct a weight value for choosing each action at a given state that aids navigation/planning.

The outputs from the grid cell network and the standard DQN agent is then combined to output a vector, that acts as the values for each action that guides action choice in the current timestep.

E.2 Simulation details

All simulations were implemented in Python. The simulation details for each task is as follows:

- Fig. 1: The state space is assumed to be a $1D$ ring with 20 states, with the transition probabilities $\mathbb{P}(s_{t+1} = i + 1|s_t = i) = \mathbb{P}(s_{t+1} = i - 1|s_t = i) = 0.5$, and discounting factor$\gamma = 0.9$ for generating the resolvent (eq. 2).
- Fig. 2: Variance of each (Gaussian) firing field (representing the strength of diffusion) is 3; **B**: $(0, 5)$ drift velocity with increasing diffusion (variance increase by 3 per step); **C**: $(3, 3)$ drift velocity with 0 diffusion; **E**: The successor representation is computed using the Fourier modes and corresponding eigenvalues, with the discounting factor $\gamma = 0.9$.
- Fig. 3: **A**: The wind effect causes (0, 2) (2 units southward) displacement at each timestep; **B, D**: The successor representation is computed given a transition matrix that assumes the variance at each (Gaussian) firing field is 1.5, followed by directed actions under the wind effects (with 0 diffusion), the discounting factor is $\gamma = 0.9$; **F, G**: The optimal following the ascending values of the successor representation, without any wind effect. The SR is computed given a transition matrix that assumes the variance at each (Gaussian) firing field is 1.5, followed by directed actions, the discounting factor is $\gamma = 0.9$; All computations are done by working directly with the Fourier modes instead of the transition matrices.
- Fig. 4: The environment is the CartPole task (Barto et al. [3]), and is simulated using the OpenAI gym environment (Brockman et al. [5]). The state value consists of 4 variables: (Cart position, Cart velocity, Pole angle, Pole angular velocity), the action value is an integer takes value from $\{0, 1\}$, where 0 represents moving left, and 1 represents moving right. For constructing the Fourier modes, we discretised each state variable into 8 bins, hence resulting in 8^4 number of states, and we chose the top 50 low-frequency Fourier modes as the inputs to the grid cell network. The standard DQN agent consists of two fully connected hidden layers with standard ReLU activations, with 48 and 24 units, respectively. The target network is updated every 500 timesteps. The deep Dyna-Q agent is a simplified version of the model proposed in Peng et al. [34], with an additional 2-layer neural network learning the environmental transition dynamics, with 64 and 32 units in each hidden layer followed by ReLU activations. At each timestep, the learnt environment model is called to generate K imaginary trajectories that are used for model-based updates to the DQN agent. The number of model-based updates, K, is taken to be 2. The state-action network in the gc-DQN has one hidden layer, with 32 units followed by ReLu activation. The grid cell network has one hidden layers, with hidden size (n, A) followed by ReLU activation, where n represents the number of input Fourier modes, and A represents the number of possible actions. The deep gc-Dyna-Q has similar architecture as the gc-DQN agent, but with an additional environment network that learns the transition dynamics of the environment that is used for model-based updates (with same architecture as the standard deep Dyna-Q agent). All models are learnt using the mean squared error loss function and Adam optimiser (Kingma and Ba [25]) with learning rate 0.001 and no learning rate decay. The exploration strength, ϵ, is set to be 0.8

at the start of each independent run, decreases by 0.05 at each episode, and is bounded below by 0.01. A total of 5 independent runs of 100 episodes are performed for each agent. Note that 100 episodes were simulated for each independent run due to the limited time and computational resources, but the results show that it is sufficient for demonstrating the increase in performance of the gc-DQN agent comparing to the baseline agents. We will, upon acceptance, show simulations with more episodes (up to the points where convergence of the baseline agents are observed) in the camera-ready version. All implementations are performed in the TensorFlow framework (Abadi et al. [1]).

- Fig. 5: A: wavevectors of chosen input Fourier modes: $\mathbf{k}_1 = (4/50, 1/50), \mathbf{k}_2 = (1/50, 4/50), \mathbf{k}_3 = (3/50, -3/50), \mathbf{k}_4 = (-4/50, -1/50), \mathbf{k}_5 = (-1/50, -4/50), \mathbf{k}_6 = (-3/50, 3/50)$; B: real rat trajectory projected onto 50×50 2D spatial domain, firing phase interval of the input Fourier modes: $[-2.5\pi/12, 2.5\pi/12]$, integration time interval: 8, exponential decay rate: 0.2, grid cell firing threshold: 2.95, directional bias: within $\pm\pi/2$ of the head direction (the range of the relative difference between the direction of the wavevector and the head direction, within which the Fourier modes are allowed to fire); C: running direction: $arctan(1/3)$.
- Fig. 9: The discounting factor: $\gamma = 0.3$, the number of states: 26, number of effective transitive inference states: 10.

We will submit codes for reproducibility with the accepted paper.