Supplementary information for "The hippocampus as a predictive map"

Kimberly L. Stachenfeld[1,2,*], Matthew M. Botvinick[1,3], Samuel J. Gershman[4]
[1]DeepMind, London, UK
[2]Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA
[3]Gatsby Computational Neuroscience Unit, University College London, London, UK
[4]Department of Psychology and Center for Brain Science, Harvard University, Cambridge, MA, USA
[*]*stachenfeld@google.com*

1 Predictive temporal codes and the SR

Many models of prospective coding in the hippocampus have drawn inspiration from the well-documented ordered temporal structure of firing in hippocampus relative to the theta phase[1–3], and considered the many ways in which replaying hippocampal sweeps during sharp wave ripple events might be used for planning[4–9]. The firing of cells in hippocampus is aligned to theta such that cells encoding more distant places fire later during a theta cycle than immediately upcoming states (a phenomenon referred to as theta precession). States fire in a sequence ordered according to when they will next appear, suggesting a likely mechanism for forward sequential planning[2,10].

However, precession alone is probably not sufficient to enact backward expansion of place fields in CA1, since NMDA antagonists that disrupt the persistent, backward expansion of place fields leave theta precession intact[11]. Furthermore, precession in CA1 likely originates outside of the hippocampus, as it arises in MEC independently[12], and depends crucially on input from surrounding areas such as MEC and CA3[13,14]. Thus, we think that it is worthwhile to consider the possible contributions of this backward expansion to planning in addition to the contributions of the hippocampal temporal code examined by this prior work.

The type of prospective coding implemented by theta precession and sharp wave ripple events is reminiscent of model-based, sequential forward planning[3]; many experiments and theoretical proposals have looked at how replaying these sequences at decision points and at rest can underlie planning[4–6,8,9]. By integrating the reward reactivated at each state along a sweep through upcoming states, the value of a specific upcoming trajectory can be predicted.

The SR is a different type of prospective code, with different tradeoffs. The SR marginalizes over all possible sequences of actions, making predictions over an arbitrarily long timescale in constant time. This results in a loss of flexibility relative to model-based planning, but greater computational efficiency. Thus, the SR cannot replace the full functionality of model-based sweeps. However, as we allude to in the article, the SR can provide a useful adjunct to the sweeps.

In Fig. S15, we illustrate how performing sweeps in the successor representation space (Fig. S15F) or performing sweeps that terminate on a successor representation of the terminal state (Fig. S15G) can extend the range of these predictions, making the hippocampal representations a more powerful substrate for planning. This is tantamount to a "bootstrapped search" algorithm, variants of which have been successful in a range of applications[15,16].

2 Why are the eigenvectors gridlike?

In this section, we discuss the mathematical properties of the successor representation (SR) that cause many eigenvectors to be gridlike.

An eigenvector of a matrix M is defined as a vector v such that $Mv = \lambda v$ for some scalar λ (the eigenvalue). The eigenvectors of the SR matrix M are the same as the eigenvectors of the transition

probability matrix, T. This is because M can be written as a discounted sum of transition matrices taken to different exponents (Equation 6 in the article).

$$M = \sum_{t=0}^{\infty} \gamma^t T^t \tag{1}$$

where γ is the discount factor. When two matrices A and B share an eigenvector v, that vector will also be an eigenvector of their sum and their product. Thus, M and T share eigenvectors.

For a random walk policy, multiplying a vector by the matrix T applies one timestep of a random walk to the vector v, diffusing the activity at each point in the vector to its neighbors. An eigenvector of T must therefore be stationary under the transition policy described by T. The vector Tv can only have a larger or smaller magnitude, determined by the eigenvalue λ. It may not be rotated, skewed, or translated. This means that wherever the vector has high positive values, there must be an adjacent, equally negative area to balance it out. In this way, positive activity diffusing from the positive areas is balanced by negative activity moving from the negative areas, and there is no net movement of any of the eigenvectors' peaks or troughs under a random walk. On a linear track, a sinusoidal wave solves this problem. A checkerboard, gridlike pattern satisfies this problem in a rectangle environment.

The eigenvectors need to align to the boundaries in such a way that this positive/negative balance is maintained. On a linear track, this constraint is satisfied when the eigenvectors are at their maximum or minimum value. Otherwise, there will not be enough positive or negative activity to balance the adjacent negative or positive peak, respectively. For a rectangular room, it is satisfied when the eigenvectors are at the maximum or minimum value over the perpendicular cross-section. Because the boundary must align with a peak or trough, there must be some exact integer number of half-peaks and half-troughs across the environment for the track or rectangle. In more geometrically complicated rooms, such as in Figure 6, 7, and 8, the arrangement of peaks is harder to characterize, but can still be understood as arranging peaks of activity so as to maintain this positive/negative balance. Thus, the eigenvectors will be found at discrete spacings relative to the size of the enclosure.

As an aside, these eigenvectors are the discrete mathematical equivalent of standing waves over the topogy described by the transition matrix. In fact, the problem of finding these eigenvectors is related to an old mathematical puzzle about "hearing the shape of a drum" (i.e., finding the modes of vibration of an arbitrarily shaped drum)[17]. This is because the transition matrix shares its eigenvectors with the graph Laplacian, a relationship we explore in the Supplemental Section "Partitioning the state space into subgoals with normalized min-cut."

3 Learning eigenvectors by stochastic gradient descent

Grid cells can be learned by stochastic gradient descent without directly computing the eigendecomposition of the SR. Here we formulate this problem as a convex optimization problem that can be learned by stochastic gradient descent.

The problem of computing the K eigenvectors with the largest eigenvalues of the successor representation matrix is equivalent to choosing an embedding matrix Y such that embedded states change slowly under the transition policy[18]. The spectral interpretation of grid cells is therefore related to the idea that grid fields can be learned using slowness as an objective[19]. If the probability of transitioning between states s and s' is high, $Y(s,i)$ and $Y(s',i)$ should be similar to each other when the i^{th} eigenvalue is large. As the eigenvalues decrease, $Y(:,i)$ is free to have higher frequency spatial variation over states. This is why the eigenvectors in Figures S7 and S8, which are arranged in descending order of eigenvalue,

Nature Neuroscience: doi:10.1038/nn.4650

have increasingly high spatial frequencies. The problem of finding the matrix Y is therefore equivalent to minimizing the following cost function:

$$C(Y) = \sum_s \sum_{s'} E(y_s, y_{s'}) T(s, s'), \tag{2}$$

where $y_s \in \mathbb{R}^K$ is the embedding of state s and $E(y_s, y_{s'}) = \gamma ||y_s - y_{s'}||^2$. Intuitively, this cost function favors embeddings that are similar for states which are likely to be visited sequentially. The optimal embedding can be viewed as the solution to a kernel PCA problem with M as the kernel[20].

The cost function can also be viewed as an expectation under the Markov chain induced by the transition function T:

$$C(Y) = \mathbb{E}_{s' \sim T(s,\cdot)} \left[E(y_s, y_{s'}) \right]. \tag{3}$$

We can therefore minimize the cost function online by sampling transitions ($s \to s'$) from the Markov chain and stochastically following the gradient:

$$y_i^{n+1} = y_i^n - \alpha^n \nabla_{y_i} E(y_s^n, y_{s'}^n), \tag{4}$$

where α^n is the step-size (learning rate) at time n, and the gradient is given by:

$$\nabla_{y_i} E(y_s^n, y_{s'}^n) = 2\eta\gamma \frac{y_s^n - y_{s'}^n}{||y_s^n - y_{s'}^n||}, \tag{5}$$

$$\eta = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = s' \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In addition, the embedding vectors are subject to a normalization constraint such that $\|y_s\| = 1$ for every state.

4 Smoothing with Spectral Regularization

For Fig. S12A-B, spectral regularization was implemented by reconstructing the SR from the truncated k-dimensional eigendecomposition at each timestep. This means that we set projections onto all but the eigenvectors with the k highest eigenvalues to 0. To be more precise, the SR matrix M can be eigendecomposed as:

$$M = U \Lambda U^T \tag{7}$$

Where U is a matrix with the eigenvectors of M as its columns and Λ is a diagonal matrix with eigenvalues along the diagonal. The SR M can be approximately reconstructed as M^k using only the first k eigenvectors and eigenvalues, which we will denote as U^k and Λ^k respectively.

Nature Neuroscience: doi:10.1038/nn.4650

$$M^k = U^k \Lambda^k U^{kT}. \tag{8}$$

Note that the superscript k does not refer to exponentiation, but to the matrix rank (the number of non-zero eigenvector components it contains). The superscript T signifies the matrix transpose. The subscript will be used later on to index by timestep, for consistency with the equations in the main text.
Spectral regularization can be implemented online by projecting the full-rank, unsmoothed M_t onto the top k eigenvectors of M after each update, and then computing the reconstructed M_t^k from the k components with largest eigenvectors:

$$M_t^k = U^{kT} U^k M_t U^{Tk} U^k. \tag{9}$$

We demonstrate how spectral regularization can improve SR approximation in a multicompartment spatial task in Figure S12. Figure S12A demonstrates that spectral regularization improves SR approximation in a noisy environment. As the noise increases, the reconstructed M^k that correlates maximally with the ground truth M is constructed from fewer components.

Importantly, the smoothing implemented by spectral regularization is topologically sensitive, meaning that smoothing respects boundaries of the environment. Regularization using a Fourier decomposition does not share this property, and will smooth over boundaries (Fig. S12B). This impairs reconstruction of the SR matrix in multicompartment environments.

Spectral reconstruction can also be used to approximate SR matrices corresponding to large discount factors from the eigenvectors of SR matrices with small discount factors, which can be learned much faster with TD learning (Fig. S12C). This is because SR matrices for different discounts all share the same eigenvectors, and have systematically shifted eigenvalues. We explore this more in Supplemental Section 6. TD learning can take a long time to converge when the discount factor is large. Spectral regularization can allow the SR to support planning over a longer timescale after significantly less training.

5 Partitioning the state space into subgoals with normalized min-cut

We briefly discuss in the article how the SR eigenvectors can be used for subgoal discovery by identifying "bottleneck states" that bridge large, relatively isolated clusters of states[21,22].

The eigenvectors can be used to partition the state space into subgoals using a form of the normalized cuts algorithm[23]. Specifically, the normalized min-cut can be approximated by thresholding the eigenvector with the 2nd largest eigenvalue (the Fiedler vector, u_2) such that states i for which $u_{2i} < 0 - \varepsilon$ go to 0, and states for which $u_{2i} > 0 + \varepsilon$ go to 1 for some very small ε. This decomposes the state space into two sets such that there are as few transition points between states in different sets as possible, while keeping the sets as large as possible[23]. States such that $-\varepsilon \leq u_{2i} \leq \varepsilon$ will go to 0.5; these correspond to subgoals, which connect the two sets of states. If no such states emerge, states on either side of an edge connecting the partitions can be used.

Partial group membership can be implemented by rescaling the eigenvector to the [0,1] range rather than thresholding. Subgoals then can be found at the states nearest 0.5.

An increasingly fine partition of the task space can emerge by including more thresholded eigenvectors. This can be performed iteratively using the initially identified subgoal to partition the environment, and then computing a new set of eigenvectors over the partitioned subspace. This recursive cutting leads to a segmentation of the task, but requires multiple eigendecomposition computations (although the computational complexity of the problem is reduced by half each time).

Nature Neuroscience: doi:10.1038/nn.4650

Traditionally, the eigenvectors of the normalized graph Laplacian, a matrix frequently invoked in spectral graph theory[24], are used in spectral methods such as this one (rather than the eigenvectors of the SR). However, the eigenvectors of the normalized graph Laplacian matrix and the random walk SR matrix are approximately equivalent in cases where most nodes have the same degree, such as in spatial domains. We sketch proof below.

Given an undirected graph with symmetric weight matrix W and given D is a diagonal degree matrix with $D(s,s) = \sum_{s'} W(s,s')$, the graph Laplacian is given by $L = D - W$. The normalized graph Laplacian is given by $\mathscr{L} = I - D^{-1/2} W D^{-1/2}$. For a random walk over this graph, the transition matrix is given by $T = D^{-1}W$. If ϕ is an eigenvector of the matrix $I - \gamma T$ and the degree is constant for all nodes, then ϕ is an un-normalized eigenvector of the normalized graph Laplacian. In a spatial domain, where the degree is constant everywhere except boundaries (almost everywhere as the discretization of the environment approaches continuity), the eigenvectors of $I - \gamma T$ will approach those of the normalized Laplacian as the discretization becomes increasingly fine. Since the SR, M, is equal to $(I - \gamma T)^{-1}$ and a matrix shares its eigenvectors with its inverse, the eigenvectors of the SR will also be similar to the eigenvectors of the normalized graph Laplacian.

6 Eigendecomposition of SR matrices with different discounts

As described in Supplemental Section 2, the SR matrix M can be written as a weighted sum of exponents of the transition matrix T, and therefore shares its eigenvectors with T. Since T is not parameterized by a discount factor γ, it follows that γ parameterizes the eigenvalues, and not the eigenvectors, of M. Here we will derive how γ parameterizes the eigenvalues of the SR matrix. We include a subscript γ to indicate that the SR is parameterized by some discount factor, γ.

Because the product of two matrices with the same eigenvectors will also have those eigenvectors, T and T^t have the same eigenvectors for any t. We demonstrate this below. Consider the eigendecomposition, where each column of U is some eigenvector of T and Λ is a diagonal matrix where the diagonal element λ_i is the i^{th} eigenvalue:

$$T = U \Lambda U^T. \tag{10}$$

Then we can write T^t as:

$$T^t = \left(U \Lambda U^T\right)^t = \left(U \Lambda U^T\right)\left(U \Lambda U^T\right) \ldots \left(U \Lambda U^T\right) = U \Lambda U^T U \Lambda U^T \ldots U \Lambda U^T \tag{11}$$

Since the eigenvectors U are orthonormal (each column is orthogonal to each other column, and each column has a Euclidean norm of 1), each pair of adjacent $U^T U$ will equal the identity matrix I. They therefore cancel out, leaving us with:

$$T^t = U \Lambda \Lambda \ldots \Lambda U^T = U \Lambda^t U^T \tag{12}$$

Thus, T and T^t share their eigenvectors U, and for each eigenvalue λ_i of T, the corresponding eigenvalue of T^t will be λ_i^t.

Since M_γ is written as a weighted sum of matrices with the same eigenvectors as T, it will have the same eigenvectors as T. This means that the eigenvectors of M depend entirely on the transition probability matrix under the current policy, and not on the discount. The eigenvalues, on the other hand, will depend on the discount, in the following way:

$$M = \sum_{t=0}^{\infty} \gamma^t T^t = \sum_{t=0}^{\infty} \gamma^t U \Lambda^t U = U \left(\sum_{t=0}^{\infty} \gamma^t \Lambda^t \right) U^T = U \left(I - \gamma \Lambda\right)^{-1} U^T = U (I - \gamma^{-1} \Lambda^{-1}) U^T \tag{13}$$

Nature Neuroscience: doi:10.1038/nn.4650

Where the last step is permitted because I and Λ are both diagonal matrices. Thus, for an eigenvector of T with eigenvalue λ_i, the same eigenvector will have eigenvalue $1 - 1/\gamma\lambda_i$ for M_γ. This means that in general, M_γ will have larger eigenvalues as γ increases. Furthermore, the increase in eigenvalue with discount will be disproportionately felt by eigenvectors with smaller eigenvalues λ_i, which are the eigenvector grids with larger spatial scales. Thus, as that as γ increases, M_γ shifts its weight to the slowly varying eigenvectors with large spatial scales. The overall eigenvalue increase occurs because a larger discount means the discounted expected visits to each state will be larger. The shift toward large spatial scale eigenvectors can be understood as a shift toward the coarser, lower dimensional representation that is required under a longer planning horizon. We visualize how the SR eigenvalues change with discount in Figure S10. In order to look specifically at the shift toward lower spatial scale eigenvectors, rather than the overall increase in eigenvalue with γ, the SR matrices in this plot are normalized by dividing each entry by the maximum element in the SR matrix.

The fact that these eigenvectors are shared among SR matrices with different discounts and that their eigenvalues can be systematically related can be exploited to learn SRs for large discounts more quickly. Typically, TD learning will converge very slowly for SR matrices of large discounts, since visitation information needs to propagate over large distances. However, since the eigenvectors and eigenvalues can be derived from the eigenvectors and eigenvalues of an SR matrix with a much smaller discount, the SR matrix can be approximately reconstructed much earlier from these components. This allows for faster convergence of large discount SR matrices (Fig. S12C), which in turn permits SR planning over larger spatial scales.

7 Eigenvector modularity

While place fields tend to increase gradually in size along the longitudinal axis of hippocampus[25], grid fields increase in scale in discrete increments[26]. Each "module" of grid cells contains cells with fields of approximately the same spatial scale. The ratio between adjacent scales was reported to be approximately $\sqrt{2} = 1.4$ regardless of scale[26].

The grid fields in our model will also have discretized spacing, as explained in Supplemental Section 2 ("Why are the eigenvectors gridlike?"). In the 1D environment, adjacent eigenvectors will have an integer number of half-peaks and half-troughs, meaning the spatial scales will be 1/2, 1/3, 1/4, ... of the length of the track. This gives adjacent "modules" a ratio of 1.5, 1.33, 1.25, etc (Fig. S11A). This departs from the findings of Stensola and colleagues in that the ratio between adjacent modules is not constant. However, it is reassuringly in the vicinity of 1.4 for experimentally plausible scales.

The same scale ratio occurs in the 2D rectangular environment as well. Looking at the spatial scale of our grid fields in the rectangle requires us to define a spatial scale, since our grid fields in this environment are not radially symmetric. In a perfect rectangle discretized with equal spacing, the grid field waveforms can be decomposed into two waveforms: one aligned to the x-axis (s_x) and one aligned to the y axis (s_y). Each of these waveforms can have a different spatial scale. We define the 2D spatial scale s to be the mean of these two scales for each field, so that $s = (s_x + s_y)/2$ (Fig S11B). As shown in Fig. S11C, the 2D eigenvectors can be clustered into modules with the same average spatial scale. These modules have the same scale ratio between adjacent modules as do the 1D eigenvectors.

Another reasonable way to compute scale s would be to take the elliptical mean of an ellipse fit to the 6 nearest peaks. The elliptical mean returns the average radius of an ellipse. This would mean computing spatial scale as $s = \sqrt{a^2 + b^2}/2$, where a and b are the major and minor axes of the ellipse and $a = 2\max(s_x, s_y)\sqrt{/3}$ and $b = \min(s_x, s_y)$.

While the spatial scales returned by the two metrics are somewhat different, the scale ratio between

Nature Neuroscience: doi:10.1038/nn.4650

modules is exactly the same under both metrics. We used the first because it seemed somewhat simpler to conceptualize and made the same prediction. However, the elliptical measure may be easier to fit to recorded grid fields, which can often be skewed or sheared but rarely seem to decompose neatly along orthogonal axes.

8 Parameters

A key parameter of the SR is the discount, γ, which scales the planning horizon of the computed value. If γ is 0, the value of each state will simply be the expected reward at that state, with no consideration of future reward. The SR matrix in this case will be the identity matrix, since all predictions beyond the current state will be discounted to 0. As γ increases, the discounted expected reward takes into account increasingly distal rewards, and the SR fields spread out to cover increasingly distal states. As γ approaches 1, the value of each state approaches the total expected reward over infinite time. If all states are reachable from all other states with nonzero probability, the SR fields will swell to cover all reachable states. Without normalization, the SR (and most likely the value function) will diverge to infinity. In Figure S4, we explore the effects of changing parameters, including the discount factor, on SR fields for a 1D track and for a 2D arena.

In our simulations, we use the same discount factor to compute the optimal policy as we do to compute the successor representation. However, these can technically be treated as different terms. The policy discount, γ_π, would describe the planning horizon the agent has been using. The SR discount, γ_M, modulates spatial scale of the SR field and determines what planning horizon that SR can be used to plan on in the future. This distinction is useful for considering how the discount factor can be used to describe the gradient of spatial scales found along the hippocampal longitudinal axis. We imagine all of these parallel SRs would be trained on the same transition history, which will have been based on the same γ_π. In order to enable future planning on different spatial scales, each SR would be modulated by a different discount γ_M. As far as interpreting the figures in the article, assume $\gamma_\pi = \gamma_M$ unless otherwise stated.

The other main parameter of the model is the softmax inverse temperature parameter, β, which parameterizes the entropy of the policy. This determines whether the policy will be more random (low β) or more deterministic (high β). A high β (more deterministic policy) will produce SR fields with more asymmetric expansion. In Figure S4, we explore the effects of changing β in conjunction with γ for SR fields.

The discount factor γ and the softmax parameter β interact to determine how localized the effect of reward on the grid cells will be. In order for a place field to be skewed, the reward-preferential bias must be appreciably larger than the random noise in the policy permitted by β. If the discounted expected number of visitations to a distal reward from the current location is discounted to near zero and β is finite, the SR at this state will be unaffected by the reward.

When simulating the effects of reward on the SR, a variety of simulation options were introduced. One involved how long the animal should remain at the rewarded location. The SR effects predicted by our model occur because the animal will learn to move toward a reward and remain near the reward for some length of time. In practice, the length of time will depend on how long it takes the animal to find and deplete the reward. In simulation, the length of time can be calibrated by changing the absorption probability at the rewarded location. In Figure S5, we show the effect of using different absorption probabilities on the distribution and shape of place fields in 1D and 2D, and how it depends on the SR discount γ_M. While this is a parameter that can be altered, in all of our other simulations, we used an absorbing probability of 0.

Under a sufficiently large discount, the SR model predicts that firing fields centered near rewarded

Nature Neuroscience: doi:10.1038/nn.4650

locations will expand to include the surrounding locations and increase their firing rate under the optimal policy. The animal is likely to spend time in the vicinity of the reward, meaning that states with or near reward are likely to be common successors. SR place fields in and near the rewarded zone will cluster because it is likely that states near the reward were anticipated by other states near the reward (Fig. S3). For place fields centered further from the reward, the model predicts that fields will skew opposite the direction of travel toward the reward, due to the effect illustrated in Fig. 2: a state will only be predicted when the animal is approaching reward from some more distant state. Given a large potentially rewarded zone or a noisy policy, these somewhat contradictory effects are sufficient to produce clustering of place fields near the rewarded zone (Fig. S3, S4). The punished locations will induce the opposite effect, causing fields near the punished location to spread away from the rarely-visited punished locations (Fig. S3F).

Another simulation decision consisted of how to discretize the environment. This decision could potentially affect the generated grid cells, which are sensitive to the topology of the underlying graph. For various reasons, we decided to use a triangular lattice because it is the densest tiling of points in which all adjacent points are equidistant. Furthermore, unlike with a square lattice, the triangular lattice cannot be stretched or sheared in 2D without changing the length of edges in the graph. However, the generated eigenvectors do not actually depend very strongly on the type of lattice, as we show in Figure S18. In Figure S18A-B, we show the first 64 eigenvectors recovered using a square lattice (A) and a triangular lattice (B). In Figure S18C, we use uniformly distributed points rather than a lattice. The weights in the adjacency matrix are proportional to the diffusion distance between points s and s', $\exp\left(\|x(s)-x(s')\|^2/\sigma^2\right)$. This measure corresponds to the transition probability for continuous Brownian motion, and is comparable to sampling random points in a continuous diffusion process. These eigenvectors are distorted by the heterogenous density of the points, but retain a grid structure. As the spacing between points becomes more irregular, the eigenvectors become increasingly distorted.

The parameters used for each simulation are described below, organized by figure (including both main text and supplement).

Figure 1: Updating value following change in reward

A. Discount $\gamma = 0.75$.

Figure 2: SR illustration and comparison to other models.

A-C. 1D tracks discretized to 500 states. 2D barrier environments discretized with 40×40. Parameters described below as fraction of boundary length, so 0.5 refers to a half of the track or wall length. The spatial scale parameters were chosen so that σ and γ lead to Gaussian and SR place fields, respectively, with approximately the same width at half maximum.

A. $\mu_{\text{track}} = 0.75$, $\sigma_{\text{track}} = 0.04$; $\mu_{\text{barr}} = \langle 0.50, 0.45 \rangle$, $\sigma_{\text{barr}} = 0.066$.

B. $\mu_{\text{track}} = 0.75$, $\sigma_{\text{track}} = 0.04$; $\mu_{\text{barr}} = (0.50, 0.45)$, $\sigma_{\text{barr}} = 0.066$.

C. $x_{\text{track}}(s) = 0.75$, $\gamma_{\text{track}} = 0.084$; $x_{\text{barr}}(s) = (0.50, 0.45)$, $\gamma_{\text{barr}} = 0.13$.

D-E. None

Figure 3: Behaviorally dependent changes in place fields.

B. *Experience dependent asymmetric backward expansion*. Discretized to 300 states. $\gamma = 0.9$, $p_{\text{right}} = 0.66$, $p_{\text{left}} = 0.34$.

F-H. *Alvernhe et al. (2011) recordings from Tolman detour maze.* Discretized maze enclosed by 20×10 box, channel width = 1, discount $\gamma = 0.95$, softmax inverse temperature parameter $\beta = 5$.

I-L. *Simulated SR with reward.* Track discretized to 121 states. The right transition probability $p_R(s) = C_{\text{right}}/D(s)$, left transition probability $p_L(s) = C_{\text{left}}/D(s)$, stay transition probability $p_S(s) =$

Nature Neuroscience: doi:10.1038/nn.4650

$C_{\text{stay}}/D(s)N(s-s_R,100)$. $N(s-s_R,100)$ is the probability density function of the normal distribution centered at $s-s_R$ with standard deviation 100, s is the index of the state, s_R is the index of the rewarded state ($s_R=251$), and $\sigma=100$ parameterizes how much the agent slows near the reward. $D(s)$ is the degree of state s and serves as the normalizing constant so that transition probabilities sum to 1, with $D(s)=p_R(s)+p_L(s)+p_S(s)$.

I-J. $C_{\text{right}}=1$, $C_{\text{left}}=0$, $C_{\text{stay}}=1$, Discount $\gamma=.98$. A larger discount was used for Fig. 6A-B than for Fig. 6C-D to demonstrate the effects of the SR model more clearly – the difference in field shape shown in Fig. 6B is quite small at the smaller discounts.

K-L. $C_{\text{right}}=1$, $C_{\text{left}}=0$, $C_{\text{stay}}=1$, Discount $\gamma=.9$. Smaller discount used for Fig. 6C-D than in Fig. 6A-B so that the SR would be near 0 at distal locations, which allows the shift to be demonstrated more clearly.

Figure 4: Hippocampal representations in non-spatial task[27]

SR learned by TD learning with 500 step random walk, learning rate $\eta=0.1$, discount $\gamma=0.98$.

Figure 5: Hippocampal representations in spatio-temporal task[28]

Discount $\gamma=0.98$. Representational similarity between states is computed as the correlation between the two rows of the SR indexed by those states (the population vectors corresponding to those states). Spatial distance refers to the Euclidean distance between states using the x,y locations of states using the embedding shown in Figure 8D. Temporal distance refers to the number of states along the shortest path between states, and "both" refers to the sum of spatial and temporal distance. The following pairs were the top five states with the lowest (above) and smallest (below) spatial distance, temporal distance, and both (spatial + temporal distance) between them.

Spatial pairs
Low distance: [5,1; 5,4; 10,7; 6,5; 4,1]
High distance: [9,3; 10,1; 10,4; 7,3; 10,3]
Temporal pairs
Low distance: [5,1; 5,4; 8,1; 9,3; 4,1]
High distance: [5,2; 4,2; 6,2; 3,2; 9,2]
Both pairs
Low distance: [5,1; 5,4; 4,1; 6,5; 10,7]
High distance: [5,2; 6,2; 4,2; 3,2; 9,2]

Figure 6: Grid cells in different geometric environments

Square. Discretization 40×42. Eigenvector 37 shown. Split in half down the slightly longer edge (between 21 and 22) for the split halves analysis.

Hexagon. Diameter (distance between opposite vertices) 40. Eigenvector 51 shown.

Circle. Diameter 40. Eigenvector 31 shown.

Trapezoid. Base 40, height 80, top 8. Eigenvector 52 shown. Split two thirds of the way between the short edge and long edge of the trapezoid (between 53 and 54) for the split halves analysis to match results to Krupic *et al.* (2015).

Orientation histograms. Includes all first 120 eigenvectors for which ellipses could be fit to the 6 central peaks of the spatial autocorrelation (defined as the 6 peaks nearest the center). This eliminated fields with very large scales (not enough peaks) or fields where the fit ellipse was a degenerate line (all peaks in a row).

Nature Neuroscience: doi:10.1038/nn.4650

Histogram used 1001 bins, which meant that no two eigenvector grid fields counted toward the same bin unless their orientations were exactly the same. The histogram was then smoothed with a small kernel with width $\sigma = 5$ bins (0.01π radians) and window size of 101 bins (0.1π radians) for display.

Grid similarity. All first 120 eigenvectors are included. To compute grid similarity, the spatial autocorrelation was first computed for both halves of each environment. The circular window used to compare autocorrelations had radius of 15 (since the dimensions of the autocorrelation are twice that of the grid field, the window never went off the edge of the autocorrelation map). The autocorrelation outside this window was set to zero for similarity comparisons. "Similarity" between two halves of a grid field refers to the Pearson correlation coefficient of the windowed autocorrelations on each half of the maze.

Figure 7: Grid fragmentation in compartmentalized maze[29]

Discretization 40×40, 10 alternating channels of width 4. Eigenvectors (top row) 32, 13, (bottom row) 14, 22 shown. Eigenvectors 20 through 120 used for similarity matrix to exclude low-frequency, non-grid eigenvectors.

Figure 8: Grid fields in multi-compartment environment[30]

Discretization 16×12 with the two square compartments each discretized at 8×8, the connecting passageway at 16×4, and the opening to the passageway 6 nodes wide.

Local model initialization. Corresponding states in the different rooms were mapped to the same index in the adjacency matrix to simulate local state aliasing as the initial condition. SR discount $\gamma = .9$, SR learning rate $\alpha = 0.1$ for the first 500 pre-training epochs and 0.01 for the next 500, SGD learning rate 0.0005, 1000 epochs total of 50 timesteps each.

Global model initialization. The SR was initialized to the local SR model learned, then constraint that corresponding states in the different rooms be the same was removed so that the rooms could slowly differentiate. SR discount $\gamma = .9$, SR learning rate $\alpha = 0.01$, SGD learning rate 0.0005, 1000 epochs total of 50 timesteps each.

Figure S1: Model Free versus SR

Channel width 1, channel lengths 4. Discount $\gamma = 0.98$, learning rate $\eta = 0.01$, noise $\varepsilon \in [-0.05, -0.05]$, 100 epochs of 500 timesteps for each reward change, random walk policy.

Figure S2: SR place fields in Tolman Detour Maze

See parameters for Figure 5. Shown are place fields for every other state (even numbered indices) so that states are sampled evenly from all parts of the maze.

Figure S3: SR place fields

Discretization 32×30, discount $\gamma = 0.99$.

A. Random walk policy.

B. Random walk policy with no transition permitted through the boundaries shown.

C. Random walk policy with no transition permitted through the boundaries shown.

D. Rewarded location = $[14, 18] \times [13, 17]$, softmax policy given inverse temperature parameter $\beta = 1$.

E. Rewarded location 1 = $[20, 24] \times [10, 14]$, rewarded location 2 = $[5, 10] \times [20, 25]$, softmax policy given inverse temperature parameter $\beta = 1$.

F. Punished location 1 = $[20, 24] \times [10, 14]$, softmax policy given inverse temperature parameter $\beta = 1$.

Shown are SR place fields for the following states in the $32 \times 30 = 960$ state environment: 178, 464, 422, 611, 418, 363, 728, 755, 665, 303, 907, 33, 266, 45, 94, 789. The x-coordinate for state s can be recovered with $x = \texttt{floor}(s/30)$ and the y-coordinate with $y = \texttt{mod}(s, 30)$.

Nature Neuroscience: doi:10.1038/nn.4650

Figure S4: Effect of model parameters
SR fields shown for different settings of the discount γ and inverse temperature parameter in 1D and 2D environments.
A. 1D environment discretized with 200 states. 10 evenly spaced SR fields are shown for each condition.
B. 2D environment discretized to 25×25 states. SR fields for states $[104, 306, 413, 433]$ shown, with x, y locations $[(4,5),(6,13),(13,17),(8,18)$. Reward located at state 456, location $(6,16)$.

Figure S5: Effect of model parameters γ and β
SR field arrangement shown for different discount factors and absorption probability at the rewarded location.
A. Environment discretized to 30×32. Rewarded location = $[14,18] \times [13,17]$, softmax policy given inverse temperature parameter $\beta = 1$.
B. 1D environment discretized to 50 states. Reward at state 50, softmax policy given inverse temperature parameter $\beta = 1$.

Figure S6: Multiscale representations
Discretization 20×20. Multicompartment environments have same discretization parameters as described for Figure S6. Points mapped to 2D using multidimensional scaling.

Figure S7: Additional grid fields in compartmentalized environments
See parameters for Figures 9 and 10.

Figure S8: Additional grid fields in non-rectangular geometric environments
See parameters for Figure 9.

Figure S9: Ground truth and learned eigenvectors
See parameters for Figure 11.

Figure S10: Eigenvalues for SR matrices with different discount factors
Discretized 8×8. All other parameters shown.

Figure S11: Eigenvector modularity
1D discretization 100, 2D discretization 100×100. Spatial scale in 2D defined as $(s_x + s_y)/2$, where s_x and s_y are the spatial scales in the x and y directions, as shown in Fig. S11B.

Figure S12: Spectral regularization
Discretization 30×30. Metric used to evaluate learned and reconstructed SRs is correlation with the ground truth SR matrix ($(I - \gamma T)^{-1}$).
A. Discount $\gamma = 0.98$. Noise added as some percent of maximum value of SR matrix (max = 3.01), ranging from 1 - 40%. Reconstructions implemented for k dimensions, where k spans 1 - 900 (where 900 is the full dimensionality of the SR).
B. Discounts ranging from .5 - .98, evenly spaced on log scale. Simulated for dimensionalities between 1 and 900 (where 900 is the full dimensionality of the SR).
C. Discounts ranging from .5 - .98. Eigenvalues and eigenvectors are based on the eigenvectors of the SR with the smallest discount, which gets learned quickly using TD learning. Eigenvalues from $M_{\gamma=0.5}$ are rescaled using the relationship shown in the rightmost graph, shifting weight onto the lower-frequency

Nature Neuroscience: doi:10.1038/nn.4650

eigenvectors, allowing the approximation of SR matrices for longer time scales. This enables SR matrices for large discounts to be approximated much sooner than TD alone permits, since it takes many trials for information to propagate over large spatiotemporal scales with TD alone.

Figure S13: Subgoals for hierarchical RL

Circles marking subgoals were placed where the eigenvector with the corresponding color was 0. Lines were drawn when a whole contour was equal to zero.

A. Multicompartment environment I. Discretized at 20×20. Doorways placed at: (10,10), (5,8), (5,14). Doorway width 2.

B. Multicompartment environment II. Discretized at 20×20. Doorways placed at: (10,4), (10,16), (5,8), (5,14). Doorway width 2.

C. 2-Step tree maze. Channel width 5, first channel length 10, other channel lengths 5.

Figure S14: Context preexposure facilitation

Discretization 10×10, punished location $R(5,5) = -1$, discount $\gamma = .9$; 10 simulation trials shown, 2×10^4 timesteps.

Figure S15: Possible interactions between SR and hippocampal sweeps

D-F. Scatter plots. Noise drawn from normal distribution with $sigma = 0.05$. Discount factor $\gamma = .6$ where applicable (D,F).

Figure S16: SR transfer in tree-maze

Graph structure shown in Fig. S16A at full discretization. Discount $\gamma = .98$. Results shown for softmax inverse temperature parameters $\beta = 0.1, 1, 100$.

Figure S17: Simulation visualization

Discretization 8×8. Discount $\gamma = .95$. SR field 36 $(x,y) = (5,5)$ and eigenvector 4 shown.

Figure S18: Effects of discretization on simulated grid cells

A. Square Lattice. Discretization 29×30.

B. Triangular Lattice. Discretization 29×30.

C. Randomly Scattered points. $N = 862$ points, uniformly scattered in rectangle with dimensions 1×1.1 but with minimum distance between points limited to 0.02. The weights in the adjacency matrix are proportional to the diffusion distance between points s and s', $\exp\left(\|x(s) - x(s')\|^2/\sigma^2\right)$. As mentioned above, this measure corresponds to the transition probability for continuous Brownian motion, and is comparable to sampling random points in a continuous diffusion process.

Figure S19: Effects of rewarded policy on SR eigenvectors

Discretization 32×30, rewarded location $R(16,15) = 1$, discount $\gamma = .98$, softmax inverse temperature parameter $\beta = 3$ for optimal policy. First 64 eigenvectors shown.

Nature Neuroscience: doi:10.1038/nn.4650

References

1. Maurer, A. P. & McNaughton, B. L. Network and intrinsic cellular mechanisms underlying theta phase precession of hippocampal neurons. *Trends in Neuroscience* **30**, 325–333 (2007).
2. Hasselmo, M. E. & Stern, C. E. Theta rhythm and the encoding and retrieval of space and time. *NeuroImage* **85**, 656–666 (2014).
3. Lisman, J. & Redish, A. D. Prediction, sequences and the hippocampus. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **364**, 1193–1201 (2009).
4. Johnson, A. & Redish, A. Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience* **27**, 12176–12189 (2007).
5. Gupta, A. S., van der Meer, M. A. A., Touretzky, D. S. & Redish, A. D. Hippocampal replay is not a simple function of experience. *Neuron* **65**, 695–705 (2010).
6. Gupta, A. S., van der Meer, M. A. A., Touretzky, D. S. & Redish, A. D. Segmentation of spatial experience by hippocampal theta sequences. *Nature Neuroscience* **15**, 1032–1039 (2012).
7. P., M. A., N., B. S., Lipa, P., Skaggs, W. E. & Barnes, C. A. Greater running speeds result in altered hippocampal phase sequence dynamics. *Hippocampus* **22**, 737–747 (2012).
8. van der Meer, M. A. A., Johnson, A., Schmitzer-Torbert, N. C. & Redish, A. D. Triple dissociation of information processing in dorsal striatum, ventral striatum, and hippocampus on a learned spatial decision task. *Neuron* **67**, 25–32 (2010).
9. Pezzulo, G., van der Meer, M. A., Lansink, C. S. & Pennartz, C. M. Internally generated sequences in learning and executing goal-directed behavior. *Trends in Cognitive Sciences* **18**, 647–657 (2014).
10. Sanders, H., Rennó-Costa, C., Idiart, M. & Lisman, J. Grid cells and place cells: An integrated view of their navigational and memory function. *Trends in Neurosciences* **38**, 763–775 (2015).
11. Ekstrom, A., Meltzer, J., McNaughton, B. & Barnes, C. Nmda receptor antagonism blocks experience-dependent expansion of hippocampal "place fields". *Neuron* **31**, 631–638 (2001).
12. Hafting, T., Fyhn, M., Bonnevie, T., Moser, M.-B. & Moser, E. I. Hippocampus-independent phase precession in entorhinal grid cells. *Nature* **453**, 1248–1252 (2008).
13. Schlesiger, M. *et al.* The medial entorhinal cortex is necessary for temporal organization of hippocampal neuronal activity. *Nature Neuroscience* **18**, 1123–1132 (2015).
14. Middleton, S. J. & McHugh, T. J. Silencing ca3 disrupts temporal coding in the ca1 ensemble. *Nat Neurosci* **19**, 945–951 (2016).
15. Sutton, R. S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin* **2**, 160–163 (1991).
16. Silver, D. *et al.* Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
17. Fisher, M. E. On hearing the shape of a drum. *Journal of Combinatorial Theory* **1**, 105 – 125 (1966).
18. Sprekeler, H. On the relation of slow feature analysis and laplacian eigenmaps. *Neural computation* **23**, 3287–3302 (2011).
19. Franzius, M., Sprekeler, H. & Wiskott, L. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology* **3**, 3287–3302 (2007).

Nature Neuroscience: doi:10.1038/nn.4650

20. Ham, J., Lee, D. D., Mika, S. & Schölkopf, B. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning* (ACM, 2004).

21. Şimşek, Ö., Wolfe, A. & Barto, A. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd International Conference on Machine Learning*, 816–823 (ACM, 2005).

22. Solway, A. *et al.* Optimal behavioral hierarchy. *PLoS Computational Biology* **559** (2014).

23. Shi, J. & Malik, J. Normalized cuts and image segmentation. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, 888–905 (IEEE, 2000).

24. Belkin, M. & Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 585–591 (MIT Press, 2002).

25. Kjelstrup, K. *et al.* Finite scale of spatial representation in the hippocampus. *Science* **321**, 140–143 (2008).

26. Stensola, H. *et al.* The entorhinal grid map is discretized. *Nature* **492**, 72 – 78 (2012).

27. Schapiro, A. C., Turk-Browne, N., Norman, K. & Botvinick, M. Statistical learning of temporal community structure in the hippocampus. *Hippocampus* **26**, 3–8 (2016).

28. Deuker, L., Bellmund, J., Schröder, T. & Doeller, C. An event map of memory space in the hippocampus. *eLife* **5**, e16534 (2016).

29. Derdikman, D. *et al.* Fragmentation of grid cell maps in a multicompartment environment. *Nature Neuroscience* **12**, 1325–1332 (2009).

30. Carpenter, F., Manson, D., Jeffery, K., Burgess, N. & Barry, C. Grid cells form a global representation of connected environments. *Current Biology* **25**, 1176–1182 (2015).

31. Skaggs, W. E., McNaughton, B. L., Wilson, M. A. & Barnes, C. A. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* **6**, 149–172 (1996).

Nature Neuroscience: doi:10.1038/nn.4650

Supplemental Figures

Nature Neuroscience: doi:10.1038/nn.4650

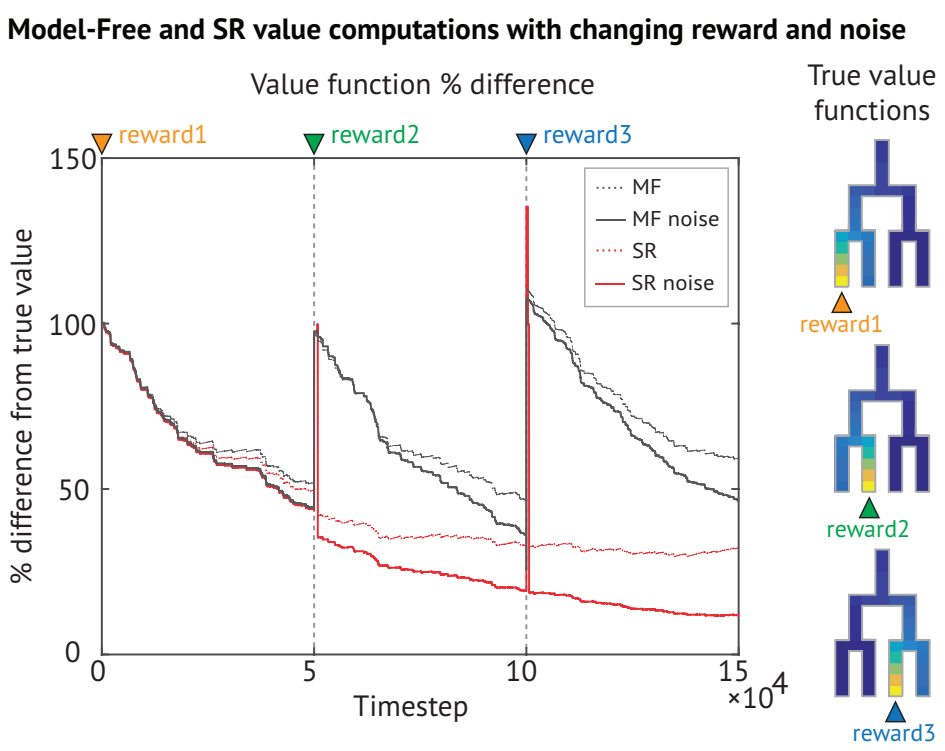

Figure S1. (A) Comparison between learning a value function with model-free (MF) TD learning (black) and with SR TD learning (red) under a random walk policy with changing reward location. For an MF agent, the value function must be entirely relearned each time the reward changes location. For an SR agent, the error will jump when the reward changes, but quickly drops as soon as the new reward is found. Furthermore, the SR provides an advantage over MF learning when there is noise in the state signal. The error surpasses 100% when the animal has neither learned the current value function nor unlearned the previous value function. The value functions corresponding to each reward location are shown on the right.

Nature Neuroscience: doi:10.1038/nn.4650

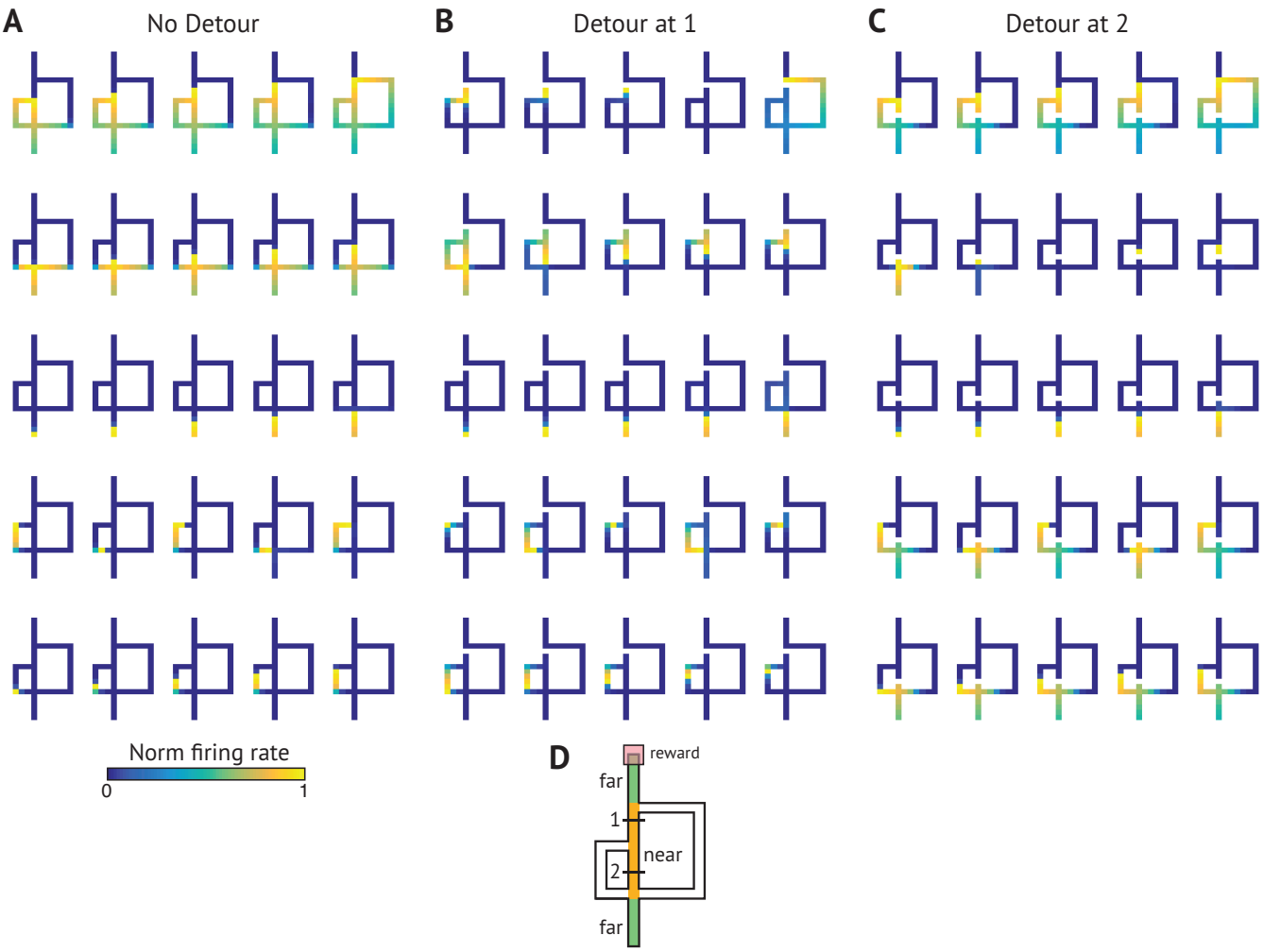

Figure S2. Full set of SR place fields corresponding to different barrier positions in Figure 3 of main document. (A) No barrier. (B) Barrier at position 1. (C) Barrier at position 2. (D) Key showing barrier positions, reward location, and regions classified as "near" and "far" for analysis in Figure 5.

Nature Neuroscience: doi:10.1038/nn.4650

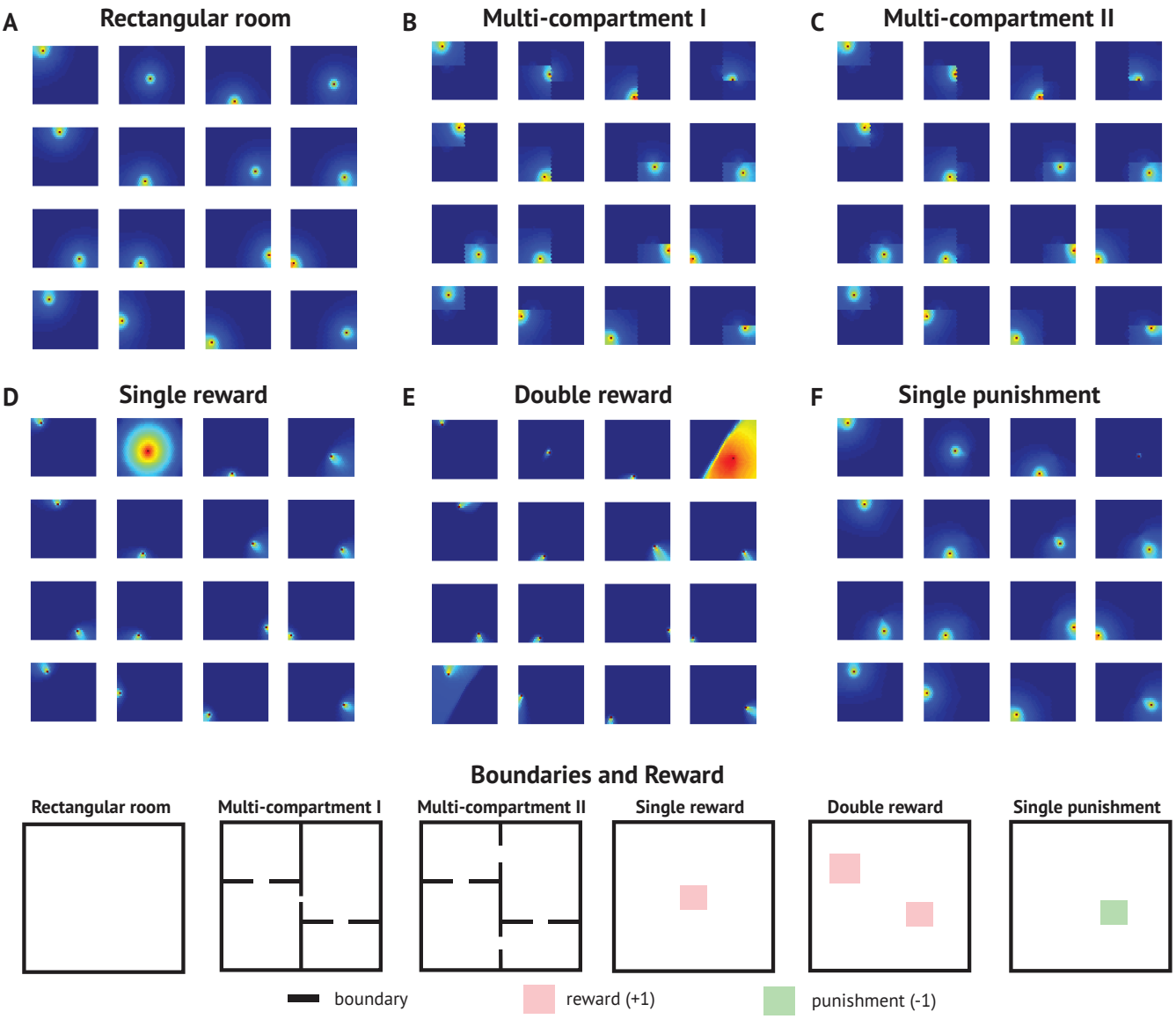

Figure S3. Example SR place fields corresponding to different environments and reward configurations. (A) SR fields are radially symmetric, gradually decaying circles under a random walk. (B,C) SR fields are constrained to remain within the compartments of the environment. (D,E) Place fields near the rewarded locations swell to include the many states that predict them, those further away skew backwards. (F) Most place fields are unaffected, those near the punished region skew slightly towards it to account for fleeing from the punished area.

Nature Neuroscience: doi:10.1038/nn.4650

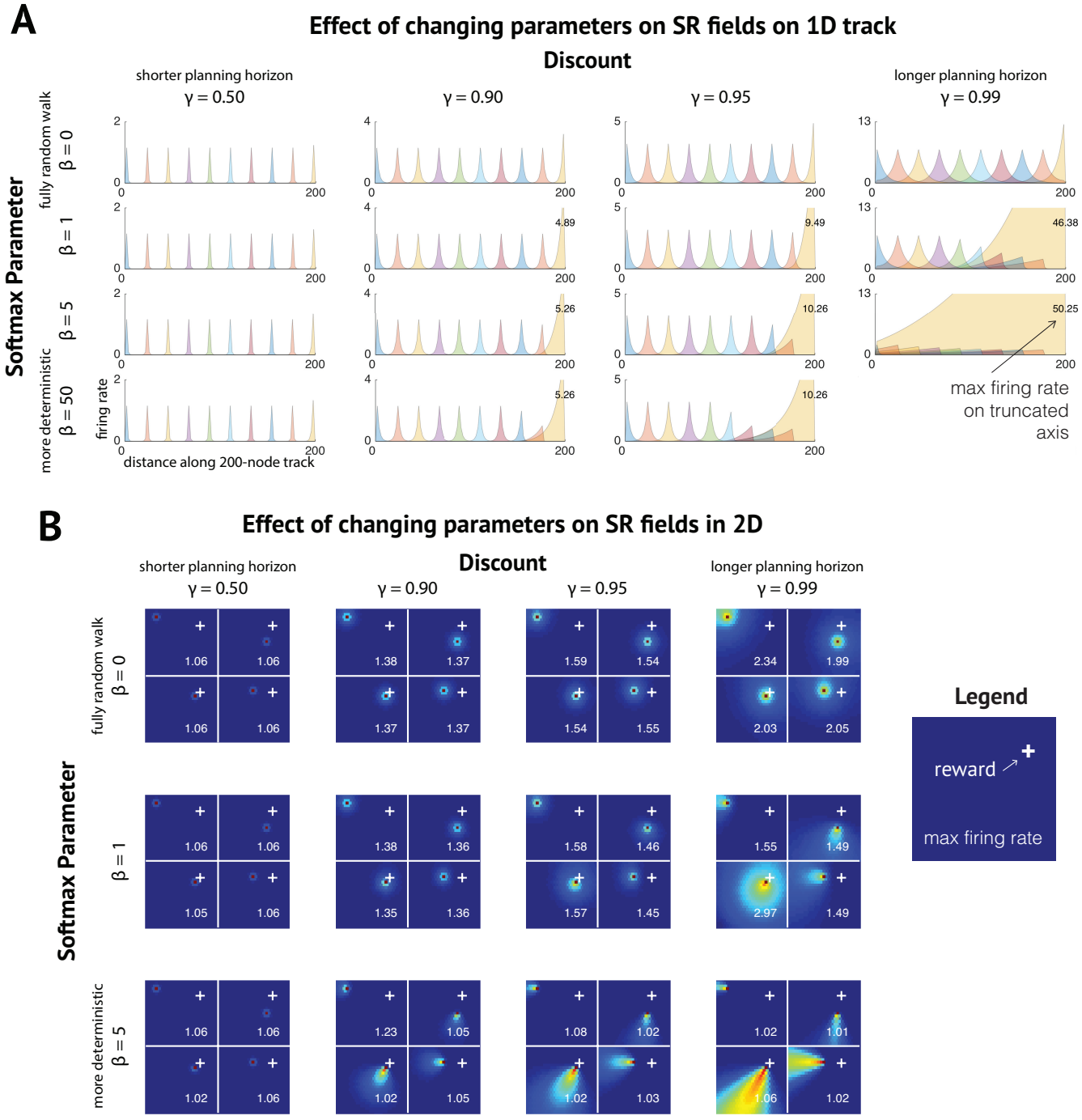

Figure S4. To explore the effects of the discount parameter γ and the softmax inverse temperature parameter β, we plot SR place cells simulated for differing values of these parameters. We place a reward at the right-most point along the track ($x = 200$). (A) SR place fields along a 1D track. Increasing the discount γ increases the spatial scale of the place fields, and in so doing increases the distance from the reward at which place cells are affected by the presence of the reward. The parameter β scales how much randomness will be included in the policy, where 0 is entirely random and ∞ is entirely deterministic. When β is larger and the policy is less random, the place fields will be more skewed opposite the direction of motion, and fields further from the reward will be affected. (B) SR place fields in a 2D rectangle. The parameters have similar effects in 2D. The white + indicates the rewarded location, and the number on each plot is the maximum SR firing rate. Once again, a larger discount factor γ induces wider place fields, and a larger β parameter induces greater skew in a larger region of SR place fields.

Nature Neuroscience: doi:10.1038/nn.4650

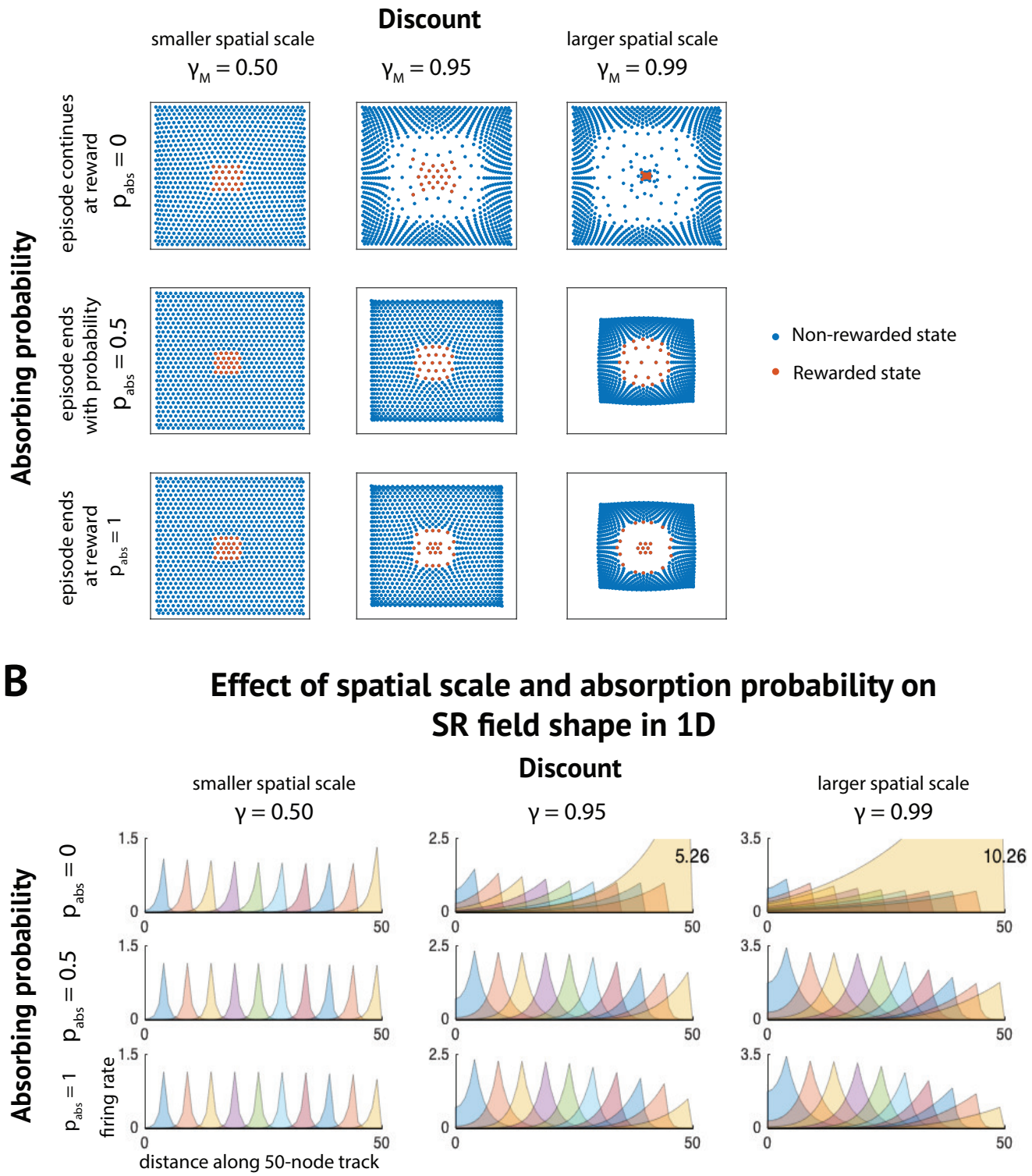

Figure S5. (A) Centers of mass for SR place fields in a 2D rectangular environment with a rewarded area for different discounts and absorption probabilities. (B) SR place fields on a 1D track for different discounts and absorption probabilities.

Nature Neuroscience: doi:10.1038/nn.4650

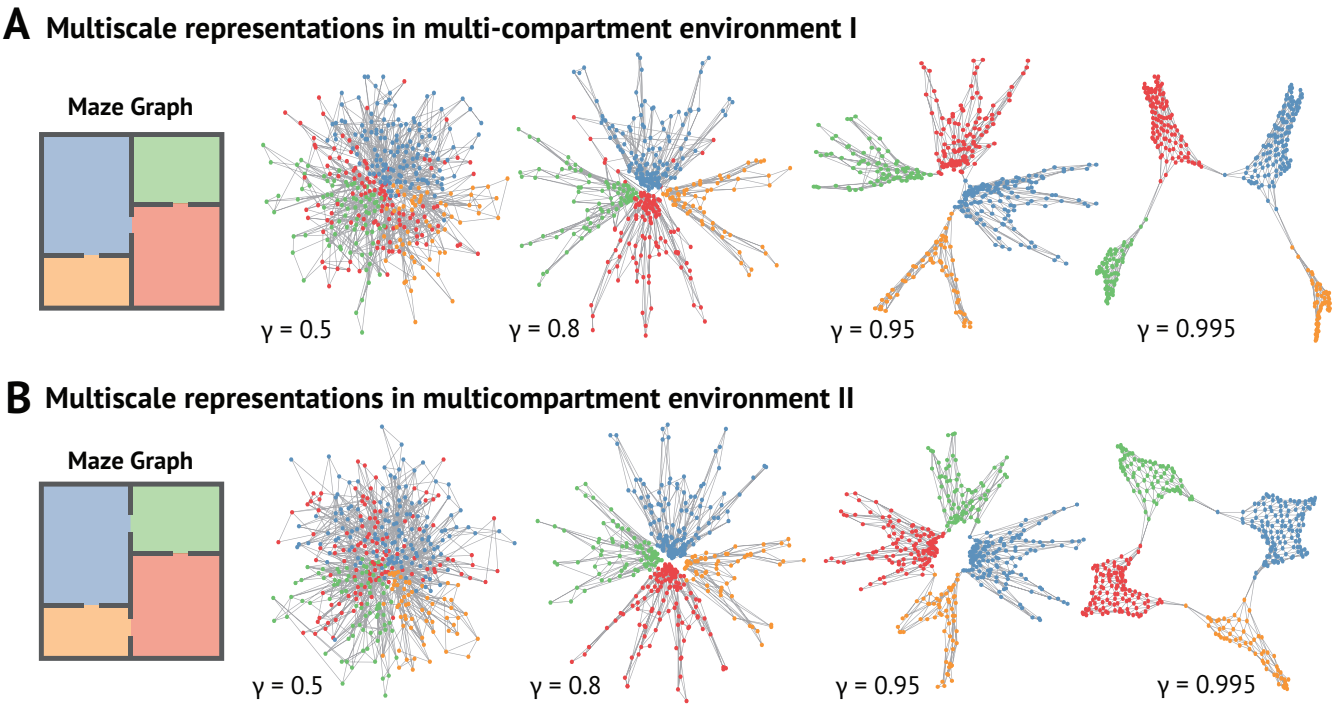

Figure S6. Multi-dimensional scaling projections of the SRs for different states in multicompartment environments. The two environments feature the same compartments connected in different ways. These plots show a 2D projection that maximally preserves Euclidean distances among SR vectors for each state. From left to right, we increase the discount factor γ used to compute the SR. The larger planning horizon ($\gamma = 0.995$) causes states that predict each other on long time scales to cluster. This exposes the long-timescale, low-dimensional organization of the environment. With small values of the discount parameter, the macroscale structure of the maze is not readily apparent in the low dimensional embedding; rather, different states in the same compartment are represented distinctly. Simultaneously representing predictive representations over a range of planning horizons gives rise to a multiscale, hierarchical embedding of states. We can thus visualize how a range of discounts expressed along the hippocampal longitudinal axis may enable hierarchical planning and planning at different timescales.

Nature Neuroscience: doi:10.1038/nn.4650

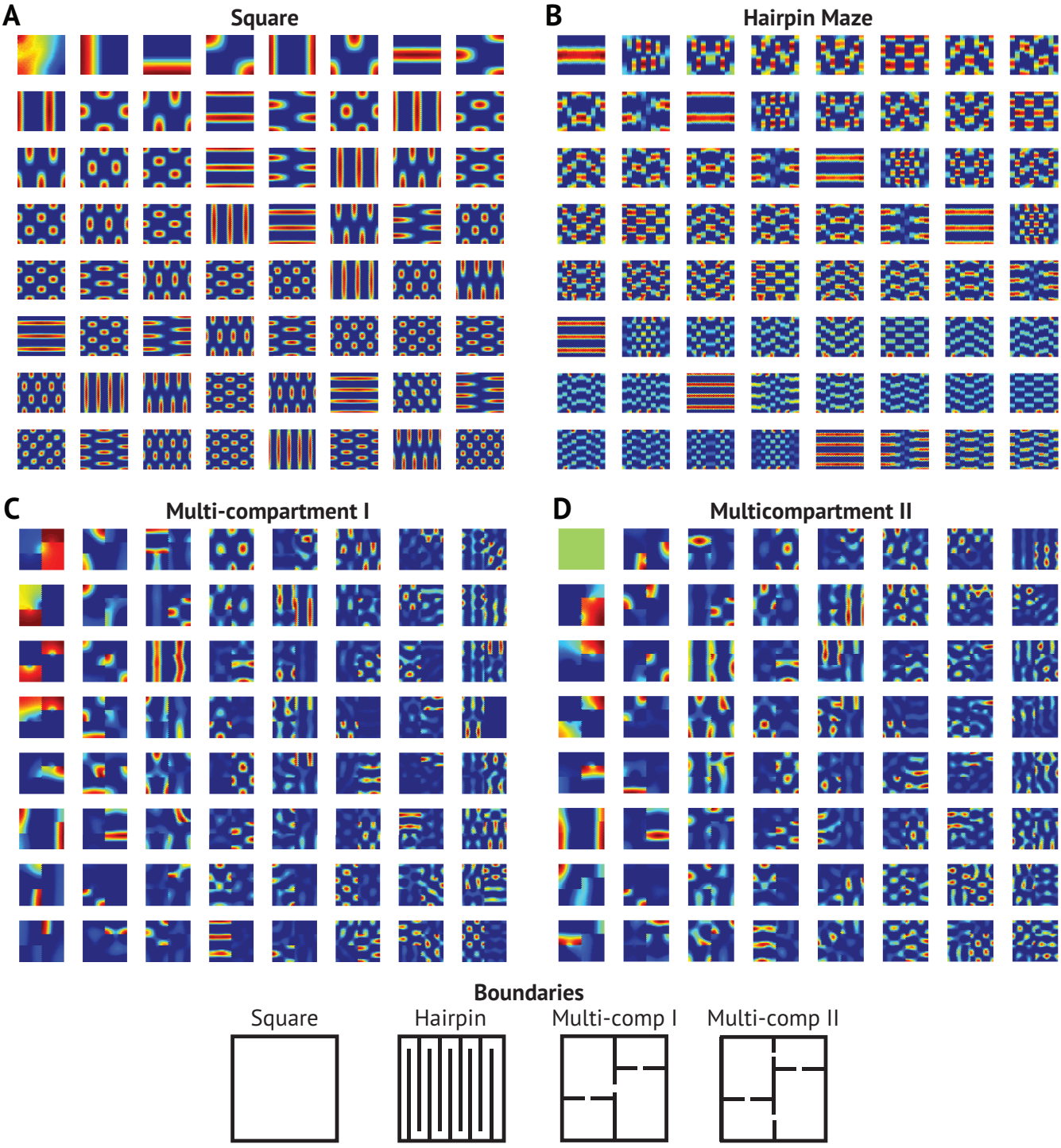

Figure S7. Additional eigenvector grids fields in rectangular and multi-compartment environments (see Fig. 6, 7 in main document and Fig. S12, S13).

Nature Neuroscience: doi:10.1038/nn.4650

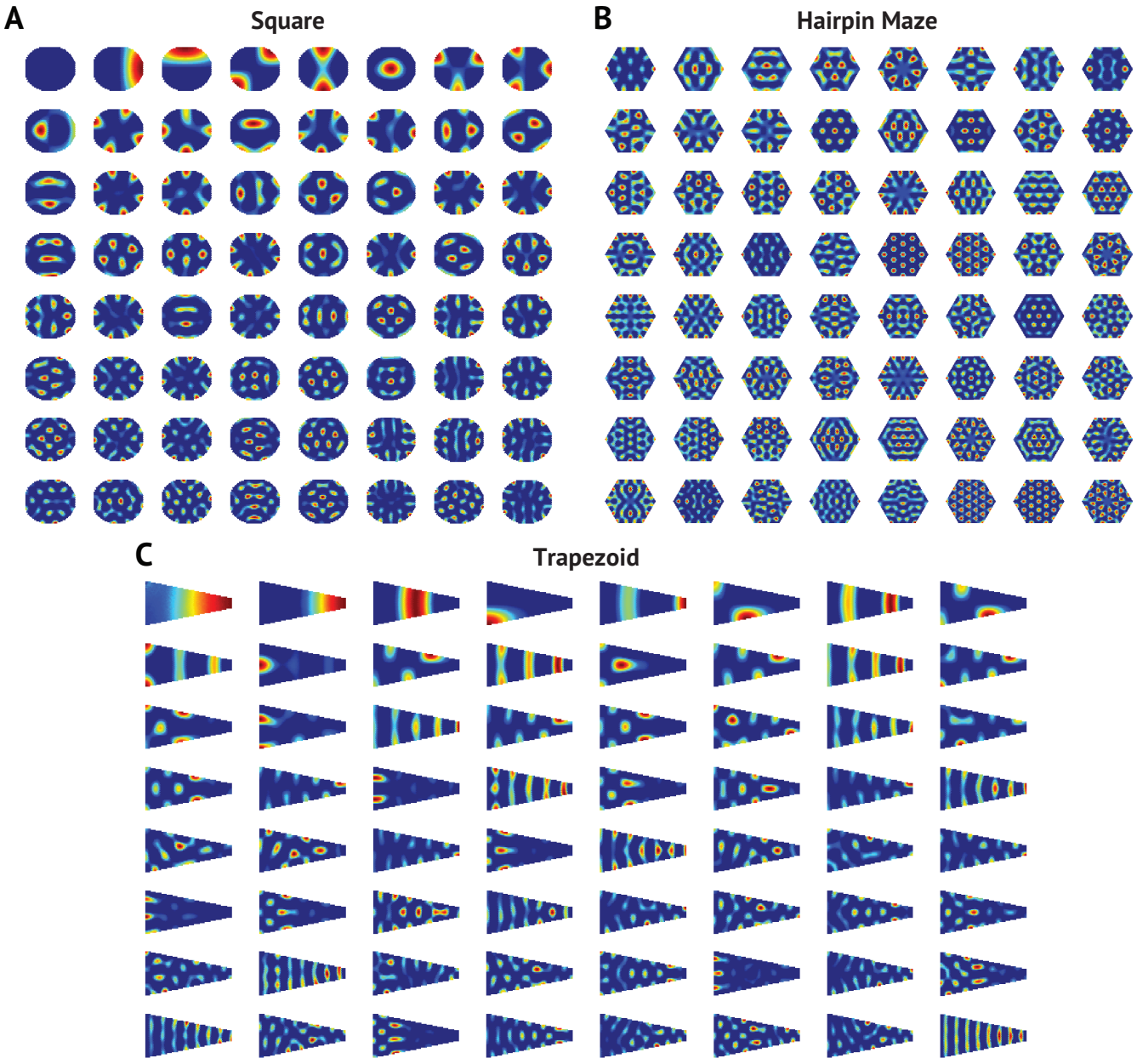

Figure S8. Additional eigenvector grid fields in non-rectangular geometric environments (see Fig. 6 in main document).

Nature Neuroscience: doi:10.1038/nn.4650

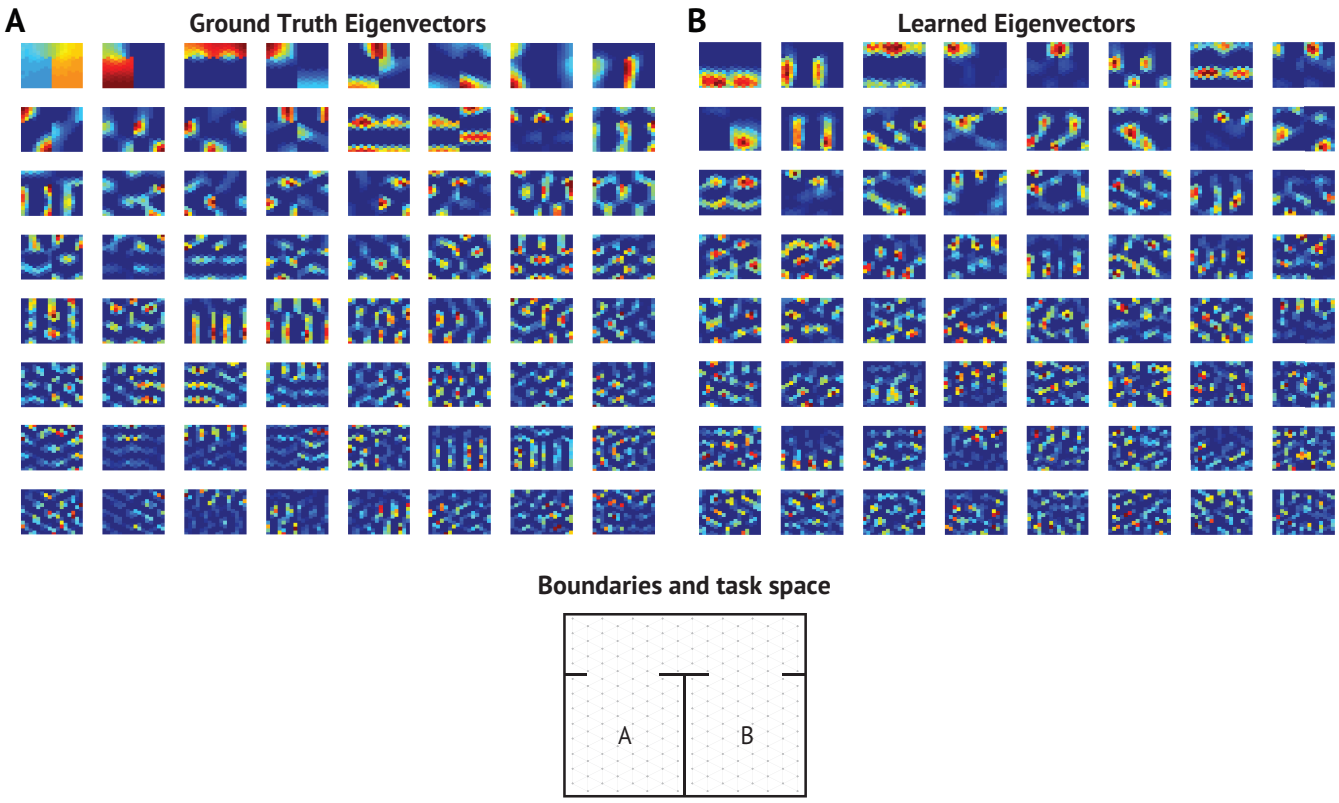

Figure S9. Additional (A) ground truth SR eigenvector grid fields and (B) learned SR eigenvector grid fields for two compartment environment described in Fig. 8 in the main text.

Nature Neuroscience: doi:10.1038/nn.4650

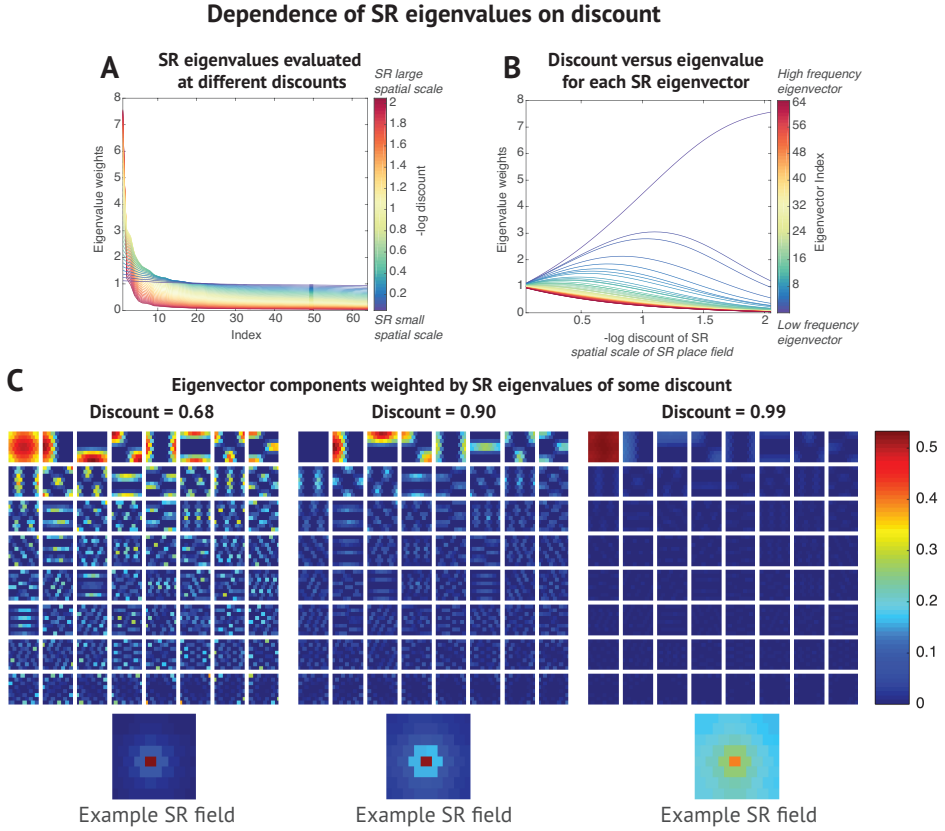

Figure S10. (A) Each line plots the eigenvalues of an SR for a random walk policy in some environment. The color of the line indexes the discount of the SR. All of the SRs have the same discount. When the discount is higher, the eigenvalues on the low spatial frequency eigenvectors dominate. When the discount is lower, the eigenvalues are more evenly distributed over low and high spatial frequency eigenvectors, as these high frequency eigenvectors are required to fit the tighter peaks of the narrower SR place fields. (B) Here we plot the eigenvalues of each eigenvector against the discount factor. The lines are indexed by which eigenvector it refers to, so the k^{th} line shows how much SRs of different discounts project onto the k^{th} eigenvector. This depicts the functional connectivity gradient along the longitudinal axis between hippocampus and MEC predicted by the SR model. (C) SR thresholded eigenvector grid fields, scaled by how much they project onto SRs of different discounts. Each SR was first rescaled by its maximum element so that no values in the SR exceeded 1. An example SR field is plotted below for each discount. This allows us to visualize the contribution of different frequency components to the SR. The low spatial scale fields of SRs with smaller discounts require a larger number of eigenvectors to be accurately reconstructed. The high spatial scale, smoother SRs with larger discounts tend to require fewer components to be reconstructed accurately.

Nature Neuroscience: doi:10.1038/nn.4650

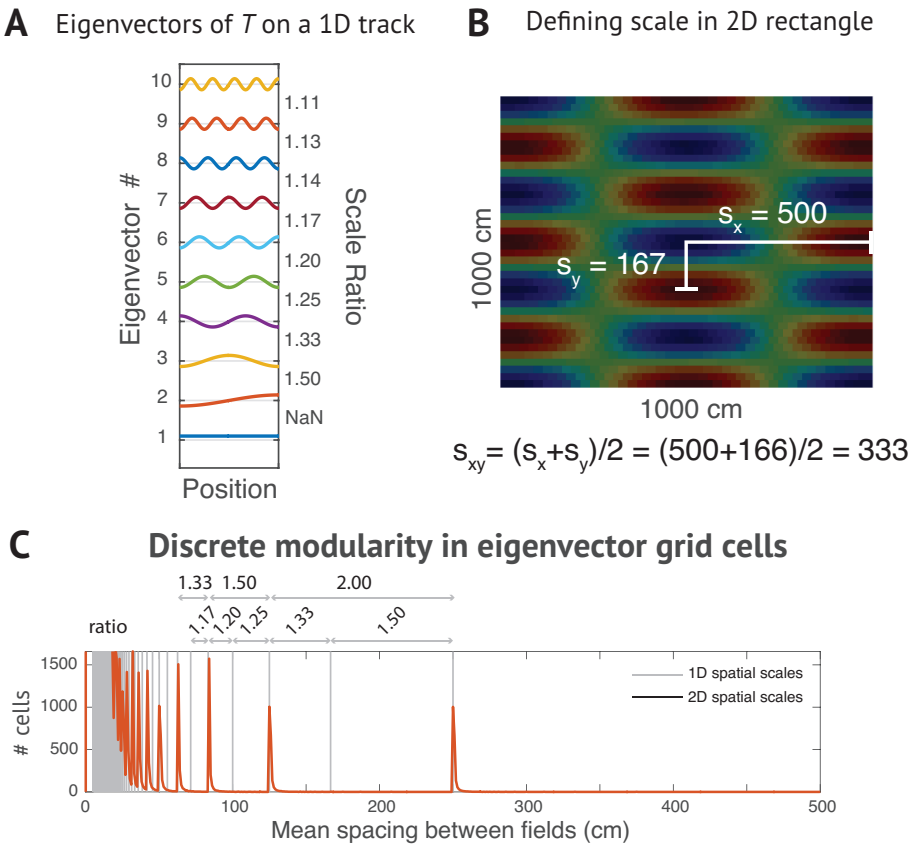

Figure S11. The SR eigenvector model of grid cells predicts that grid cells arise in discrete modules in both (A) a linear track and (B) a 2D rectangular environment. (C) The spatial scales at which these modules occur are shown, as well as the ratio of spatial scales between adjacent modules.

Nature Neuroscience: doi:10.1038/nn.4650

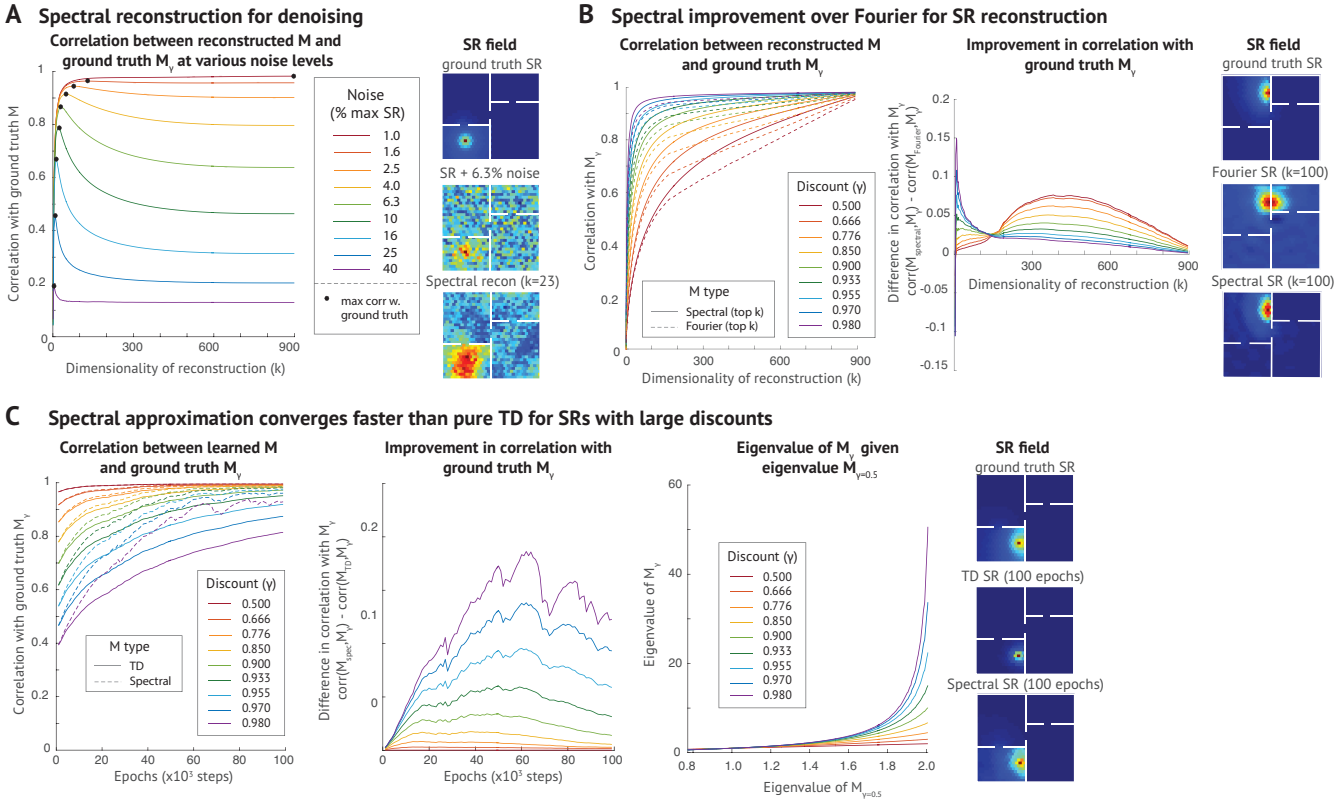

Figure S12. (A) Denoised reconstructions of the SR can be obtained by reconstructing the SR from the top k eigenvectors of the noisy SR. For each noise amount, we show the the correlation of each reconstructed M_k with the ground truth, noiseless M. As the noise corruption decreases, the higher frequency eigenvectors will contain increasingly meaningful information, as opposed to noise, and will contributed positively toward the reconstruction of M. We also show example SR place fields for the ground truth, noisy, and reconstructed M. (B) Here we demonstrate the advantages of the spectral decomposition over the Fourier decomposition for reconstructing M. In a multicompartment environment, the eigenvectors capture more variance than the Fourier basis. The spectral reconstructions are consistently more highly correlated with the ground truth SR than the Fourier reconstructions given the same number of components. Also shown are example SR fields for the ground truth SR, Fourier reconstructed SR, and spectral reconstructed SR. This illustrates another potentially problematic feature of using the Fourier basis for reconstruction: since the Fourier is smooth over barriers in the environment, SRs reconstructed from only the top k Fourier components will be smooth over boundaries rather than discontinuous. (C) For large discounts, the SR takes a while to converge through TD because information must be propagated back many time steps. However, spectral reconstruction allows us to leverage the fact that eigenvectors are shared over SRs with different discounts and that there is a systematic relationship between the eigenvalues of SRs with different discounts. Since the SR will converge relatively quickly for small discounts such as $\gamma = 0.5$, we can use the eigenvectors of $M_{\gamma=0.5}$ and the relationship between eigenvalues of $M_{\gamma=0.5}$ and the eigenvalues of M_γ to approximate an arbitrary M_γ. This allows M_γ to be approximated for large γ much earlier on in training, which can enable planning over longer timescales earlier in training.

Nature Neuroscience: doi:10.1038/nn.4650

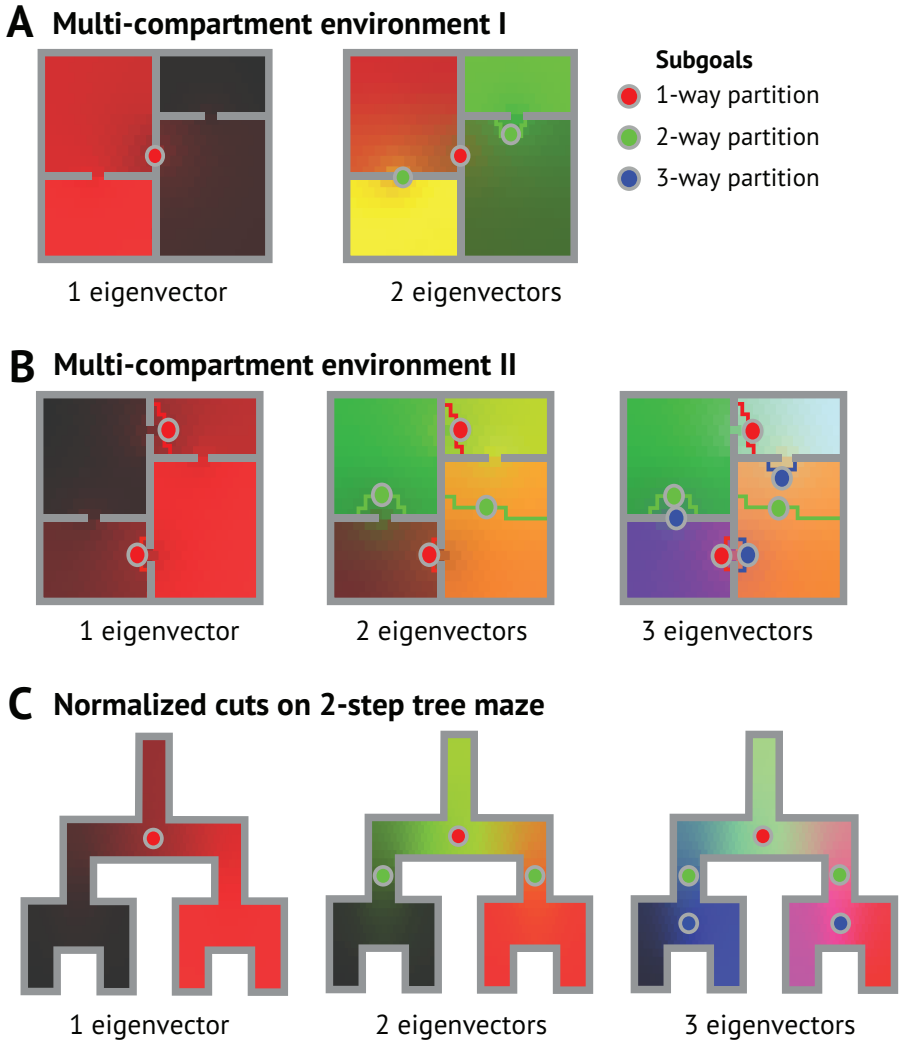

Figure S13. *Subgoals for hierarchical RL.* (A) The eigenvectors with largest eigenvalues can be used to partition the task space, with boundaries between the partitions producing useful subgoals. The remaining eigenvectors are shown in Fig. S7. In compartmentalized environments, subgoal partitions tend to fall at or near doorways. The environment is colored by setting the RGB values of each point in the maze to the corresponding value of the 1st, 2nd, and 3rd eigenvectors used in the illustrated partition. (B) Different environment topologies give rise to different subgoals. The partition is not as clean when compartments are connected in a loop instead of hierarchically (as in A), but the doorways are still identified by the top 3 eigenvectors. (C) Extracted subgoals correspond to major decision points in a 2-step maze.

Nature Neuroscience: doi:10.1038/nn.4650

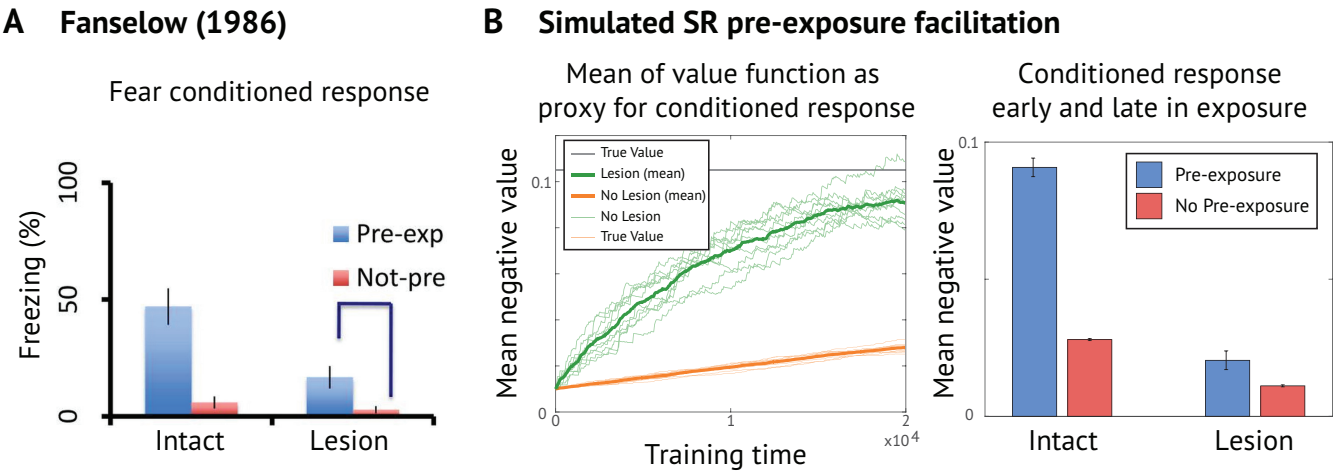

Figure S14. (A) Fear-related freezing response in rats following fear conditioning has been shown to be stronger if the animal is "pre-exposed" to the conditioning environment. This effect is much weaker following hippocampal lesions. (B) Under an SR interpretation, exploring the environment allows the hippocampus to learn a predictive representation. Since value is computed by multiplying the SR by the reward function, an under-developed SR will produce an underestimate of the negative value in the room. As the animal learns a model of how states in the environment predict each other, the value signal can be approximated more fully. Model-free learning does not predict this, since the reward signal cannot propagate through the environment without experience preceding the time of shock.

Nature Neuroscience: doi:10.1038/nn.4650

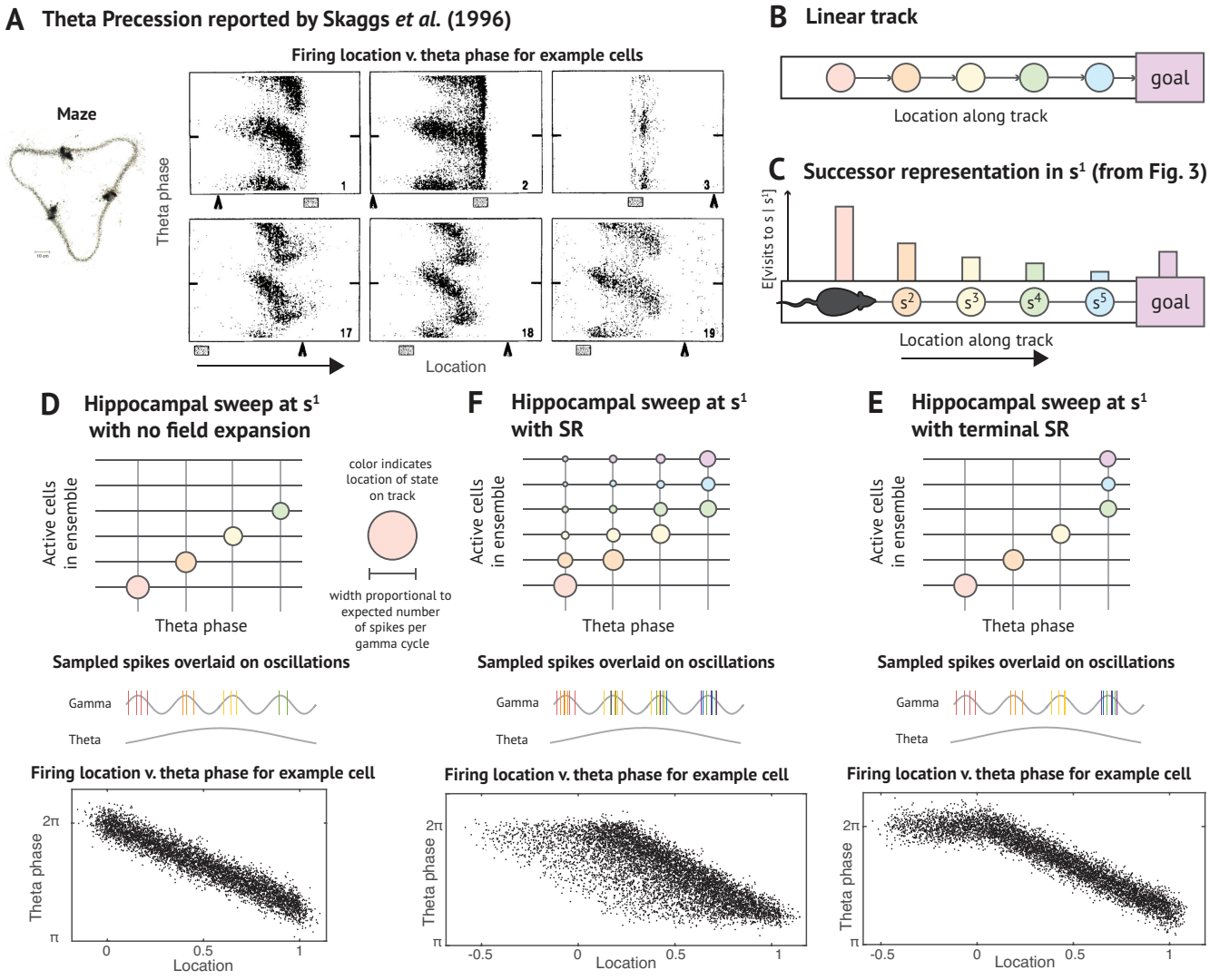

Figure S15. (A) Illustration of theta phase precession reported by Skaggs and colleagues[31]. (B) A depiction of the linear track environment used for the illustration. (C) Successor representation when the animal is in the first state from the left, s^1 (copied from Fig. 2D). (D-F) Illustrations of different ways for a hippocampal sweep to interact with place cell expansion. The top panel illustrates which states have cells in the active ensemble at point in the theta phase (discretized into gamma cycles), and how active each place cell is. The cartoon below shows how spikes would be sampled from the ensemble at each gamma cycle. The bottom panel illustrates the predicted relationship between a cell's firing location and theta phase under this model during the retrieval portion of the phase[2]. (D) A sweep through upcoming state indices. (E) A sweep through a sequence of successor representations of upcoming states. (F) A sweep through upcoming states terminating in a successor representation of the last state reached by the sweep.

Nature Neuroscience: doi:10.1038/nn.4650

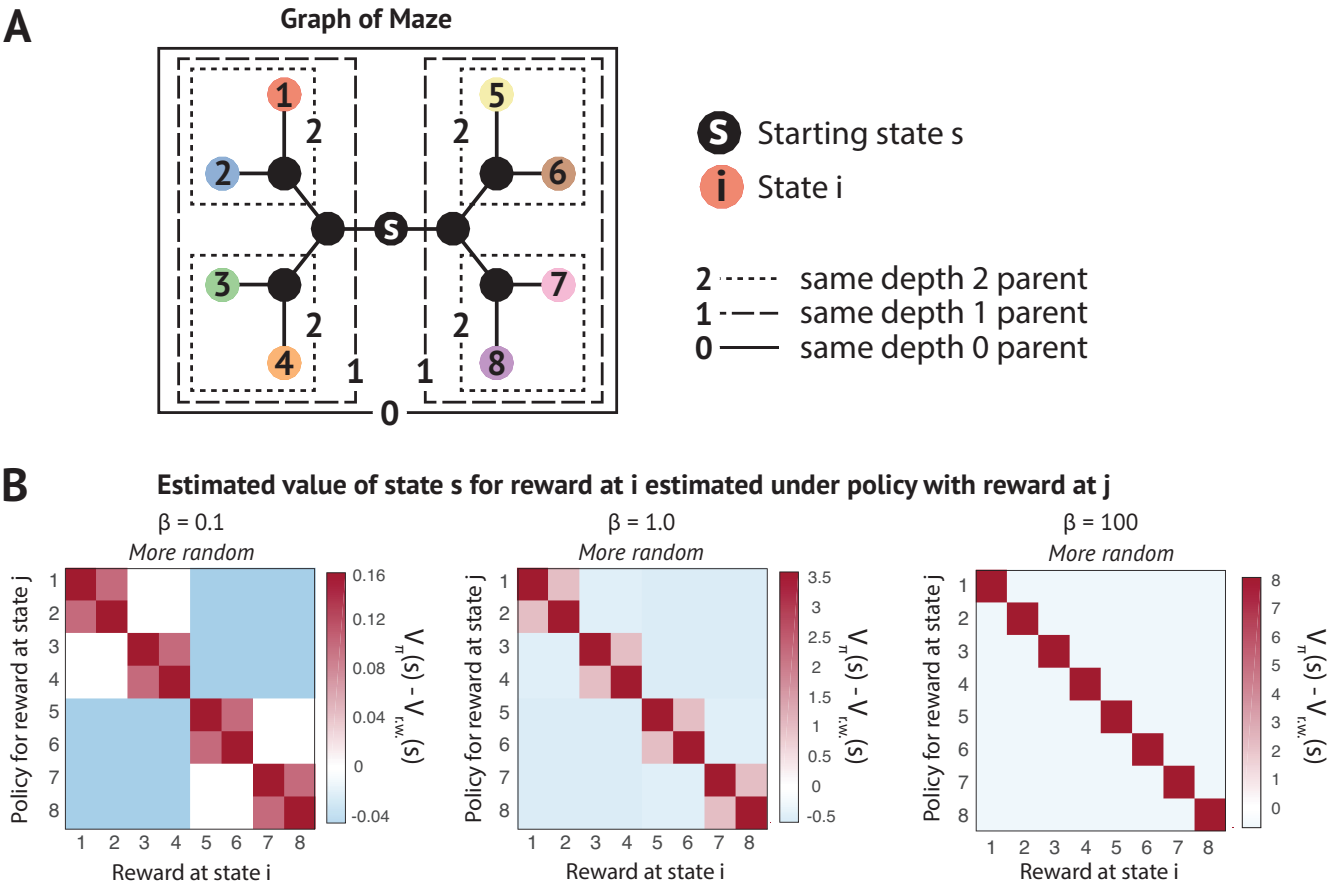

Figure S16. *Testing the generalization of the SR when reward is moved.* Here we illustrate how the quality of SR generalization depends on the policy stochasticity (parameterized by the softmax inverse temperature β), and how similar the optimal paths are for the old and new rewarded location. When there is greater stochasticity (closer to the random walk policy), the SR's generalization to highly dissimilar locations is less impaired, but there is also a reduced generalization advantage when the reward ends up nearby. The random walk SR is used as a baseline. By diffusing value through the graph in accordance with the task's underlying adjacency structure, this representation always generalizes better than re-initializing to a state index representation. The animal should maintain support for random actions until it is very certain of the optimal path. Spectral regularization can promote this by smoothing the SR. (A) Task graph for a depth-2 tree maze, in which the starting state is the root node. Rewards can be presented at the 8 colored "leaf" nodes. The agent begins at state s. (B) Here, we demonstrate how much value function estimation deteriorates when the reward is moved from state j to state i when using the SR for a policy learned when the reward was at state j. To do this, we compute $V(s) = \sum_{s'} M_{\pi_i}(s,s')R_j(s')$ for every combination. The generalization is compared to a baseline of how well the SR for a random walk does, which corresponds to a $\beta = 0$.

Nature Neuroscience: doi:10.1038/nn.4650

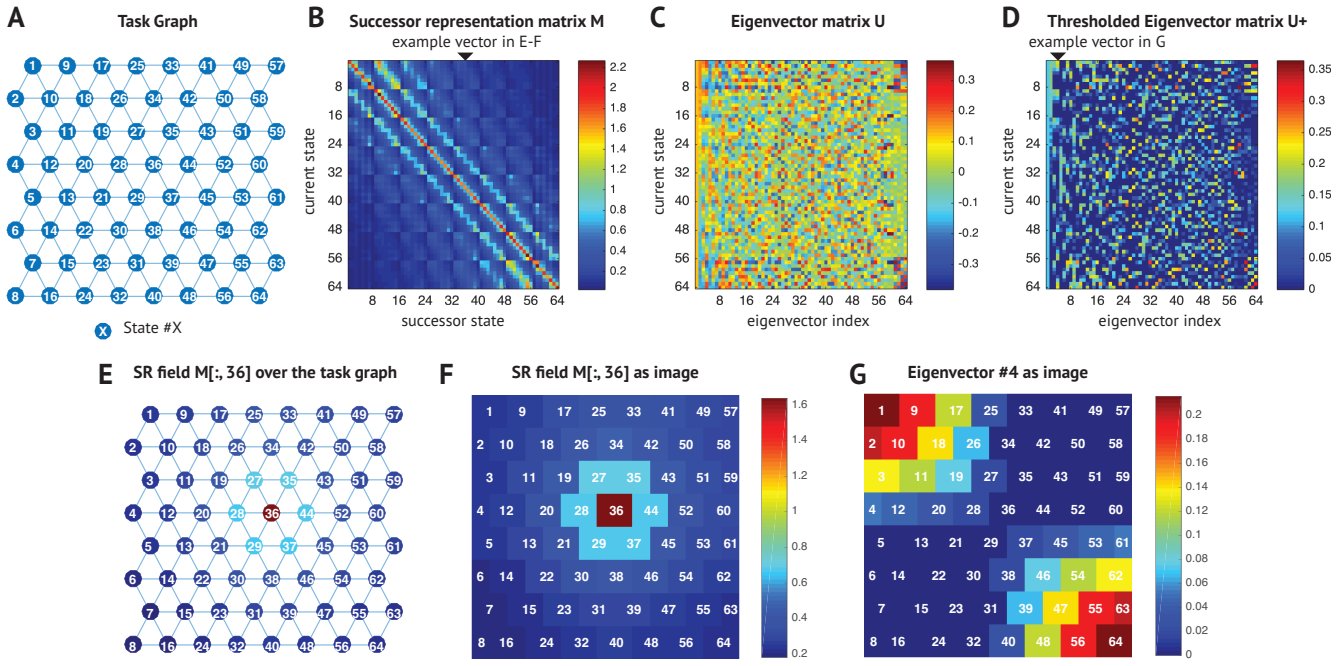

Figure S17. (A) A rectangular space discretized as a triangularly latticed discrete MDP graph. Each node is numbered according to its index in the SR and eigenvector matrix. (B) The full SR matrix M computed for a random walk policy. Each row $M(s,:)$ is a successor representation for some state s, and each column $M(:,s')$ is the receptive field for a neuron encoding state s'. (C) The matrix U containing eigenvectors of the SR matrix M. (D) Eigenvectors were thresholded at 0 to preclude the possibility of negative firing rates. (E) To go from the computed matrices to plotting the receptive fields of simulated neurons, the corresponding column of the matrix was reshaped and each node was colored according to the value of the vector at each state. Here we demonstrate that on the SR receptive field for the neuron encoding state 36. (F) Here we demonstrate how we go from coloring each node in the task graph as shown in (E) to an image. (G) The same principle can be applied to plot simulated eigenvector grid cells, as we demonstrate with the 4^{th} eigenvector.

Nature Neuroscience: doi:10.1038/nn.4650

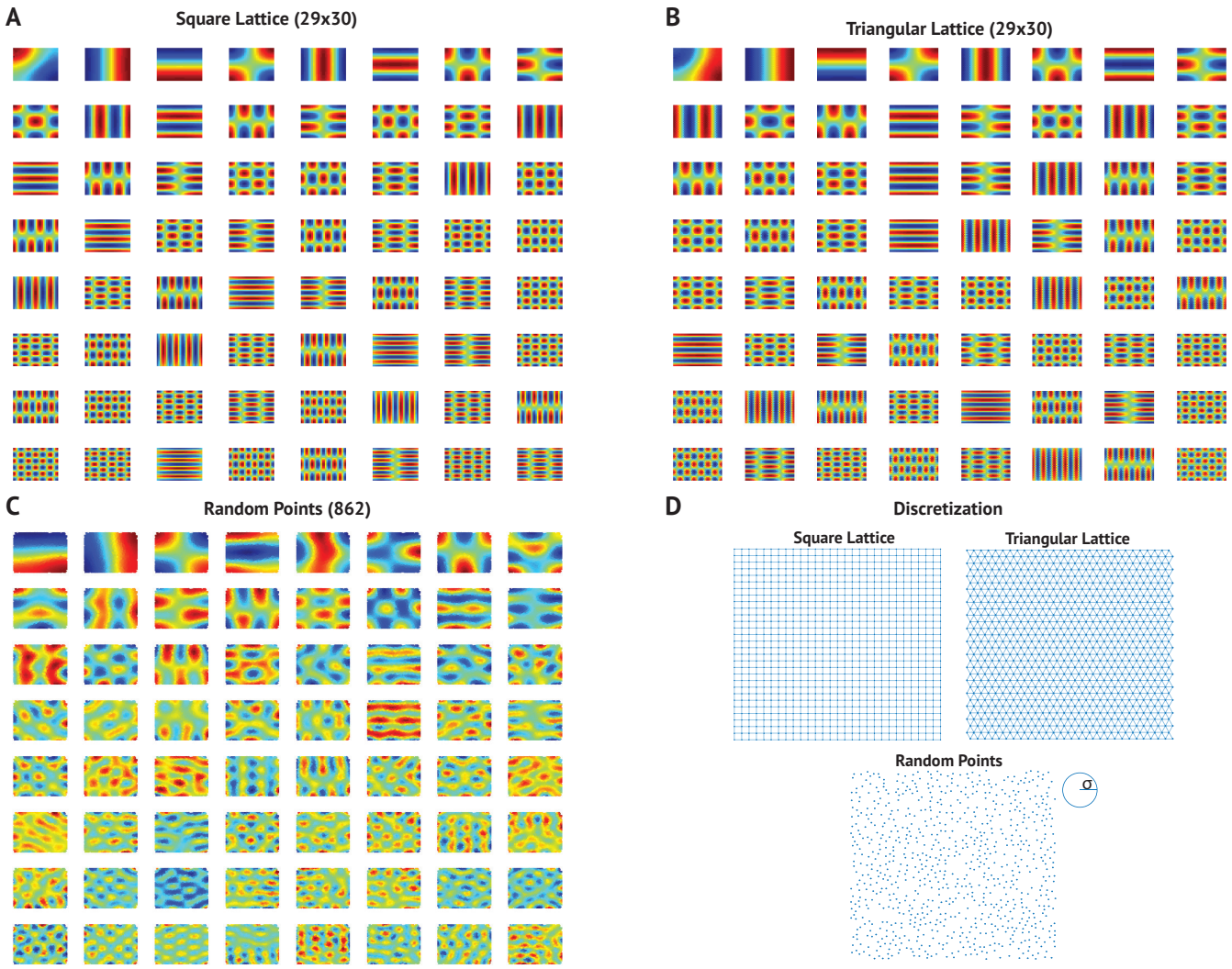

Figure S18. Here we generate SR eigenvector grid fields for differently discretized environments to compare the effects of different discretizations on the generated grid cells. The policy was a random walk in all cases. (A) Square lattice. (B) Triangular lattice (this was used for all spatial simulations unless otherwise specified). (C) Points are randomly, pseudo-uniformly scattered about the environment, subject to a minimum distance between points of 0.02. The edge weight on the adjacency matrix is equal to the diffusion distance between points, $\exp(-||x(s) - x(s')||^2/2\sigma^2)$, which is proportional to the probability density of the agent transitioning from location x to y under a continuous random walk in 2D (Brownian motion).

Nature Neuroscience: doi:10.1038/nn.4650

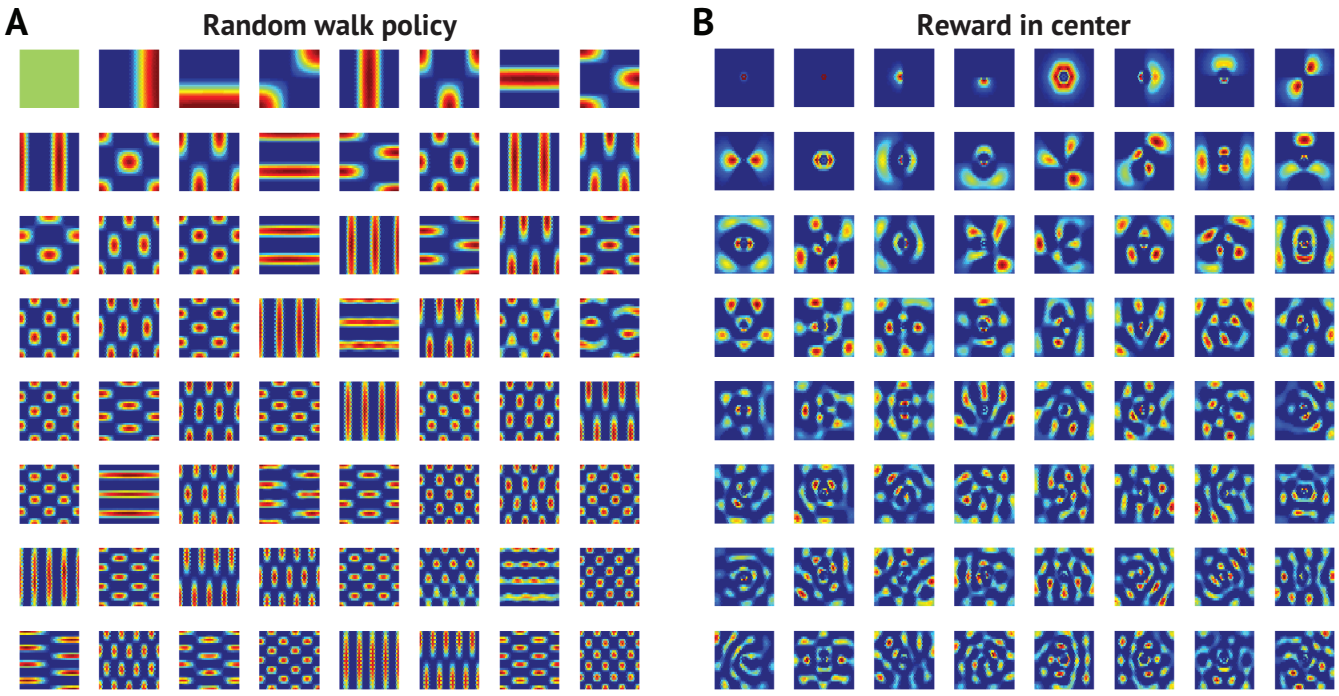

Figure S19. (A) SR eigenvectors under random walk policy and (B) under optimal softmax policy with reward in center of room. The attractor state induced in the policy by the rewarded location warps the SR eigenvectors.

Nature Neuroscience: doi:10.1038/nn.4650