Not so griddy: Internal representations of RNNs path integrating more than one agent

William T. Redman[1, 2 *], Francisco Acosta[3], Santiago Acosta-Mendoza[2], and Nina Miolane[4]

[1]Intelligent Systems Center, Johns Hopkins University Applied Physics Lab
[2]Interdepartmental Program in Dynamical Neuroscience, UC Santa Barbara
[3]Department of Physics, UC Santa Barbara
[4]Department of Electrical & Computer Engineering, UC Santa Barbara
[*]Will.Redman@jhuapl.edu

Abstract

Success in collaborative and competitive environments, where agents must work with or against each other, requires individuals to encode the position and trajectory of themselves and others. Decades of neurophysiological experiments have shed light on how brain regions [e.g., medial entorhinal cortex (MEC), hippocampus] encode the self's position and trajectory. However, it has only recently been discovered that MEC and hippocampus are modulated by the positions and trajectories of others. To understand how encoding spatial information of multiple agents shapes neural representations, we train a recurrent neural network (RNN) model that captures properties of MEC to path integrate trajectories of two agents simultaneously navigating the same environment. We find significant differences between these RNNs and those trained to path integrate only a single agent. At the individual unit level, RNNs trained to path integrate more than one agent develop weaker grid responses, stronger border responses, and tuning for the *relative* position of the two agents. At the population level, they develop more distributed and robust representations, with changes in network dynamics and manifold topology. Our results provide testable predictions and open new directions with which to study the neural computations supporting spatial navigation.

When navigating real-world environments, individual agents (e.g., animals) must adjust their trajectories based on the observed trajectories of others [1, 2]. This becomes especially important in competitive and collaborative settings, where the positions, movement directions, and speeds of others contain information necessary for goal-based decision making. For example, an agent competing with others for food must keep track of where "opponent" agents are and where they are headed, in order to effectively update its strategy. While decades of research has shed fundamental light on the neural circuitry involved in spatial navigation in the absence of others ("single agent" navigation), little is known about how the brain enables spatial navigation in the more general "multi-agent" scenario.

A core component of spatial navigation is *path integration* [3], the ability to update an estimate of position in an environment from a sequence of movement directions and speeds (Fig. 1A). Medial entorhinal cortex (MEC) is believed to play a critical role in supporting path integration via functional classes of neurons [4], such as grid cells [5, 6], border cells [7], and band cells [8]. Recent research has shown that MEC is activated when human subjects observe others navigate environments [9, 10], suggesting that it may additionally be involved in computations supporting multi-agent path integration. That neural activity in MEC could be shaped by trajectories of others is further supported by recent work demonstrating that populations of cells in hippocampus, a brain region heavily interconnected with MEC, encode spatial information related to the positions of other agents [11–16]. However, what representations exist in MEC to support multi-agent path integration and how they differ from the representations that have been found in single agent environments, remains largely unexplored. This is of additional importance as it is unclear whether continuous attractor networks [17–19], which are believed to underlie the grid code in MEC [20–24], can support the encoding of multiple individuals [25, 26].

To begin to investigate this open research direction, we train a recurrent neural network (RNN) model to perform path integration of two agents ("dual agent" path integration) and interrogate the representations that emerge. These RNN models [27, 28], when trained on single agent path integration, have been shown to develop properties analogous to MEC. In particular, they exhibit units with grid, border, and spatially periodic "band" responses [27–30] and population dynamics consistent with continuous toroidal attractors [31, 32]. While their biological plausibility remains unclear [33, 34], we choose to analyze these RNN models for three reasons. First, they contain inter-connectivity between "hippocampus"-like and "MEC"-like layers, enabling us to model aspects of the experimentally reported multi-agent hippocampal responses [11–16]. Second, the lack of neural data from MEC in multi-agent settings makes the outputs of these RNN models unbiased predictions that can be used to further assess their validity. And third, beyond their connection to neuroscience, these RNN models represent legitimate machine learning systems that exhibit improved goal-based navigation when paired with reinforcement learning [27]. As autonomous agents become increasingly deployed in everyday environments, their ability to navigate in multi-agent settings is increasingly important [35] and understanding how the representations learned by RNN models trained on single and dual agent path integration compare can provide new insight into how to further improve this capability.

We find that, with only a few modifications, we can train existing RNN models [27, 28] to achieve good performance on dual agent path integration ("dual agent RNNs" – Sec. 1). These dual agent RNNs are flexible enough to generalize to single agent path integration, whereas RNNs trained on single agent path integration ("single agent RNNs") are not capable of performing dual agent path integration (Sec. 2.1). Investigating the learned representations, at the individual unit level, we find that the need to path integrate more than one agent leads to the emergence of stronger border representations and weaker grid representations (Sec. 2.2), a result consistent with recent human neurophysiological experiments [9, 10]. Measuring the representations in a reference frame that captures the joint position of both agents, we find that units in dual agent RNNs develop tuning for "relative space" (Sec. 2.3). Utilizing topological and dynamical tools, we characterize differences between single and dual agent RNNs at the population level (Sec. 2.4), finding no evidence for continuous toroidal attractors underlying the activity of dual agent RNNs.

Contributions. We study, *for the first time*, how representations that emerge in RNNs trained on dual agent path integration differ, at the individual unit and population level, from the representations of single agent RNNs. This provides support of recent neurophysiological experiments [9, 10] and testable predictions that can guide future experimental design.

1 RNN model

Prior work has considered the setting where a single agent traverses a square environment through finite length paths [27–29]. The agent starts a given path at position $\mathbf{z}(0) = [x(0), y(0)]^\intercal \in [-L/2, L/2] \times [-L/2, L/2]$ and subsequently moves to positions $\mathbf{z}(0) \to \mathbf{z}(1) \to ... \to \mathbf{z}(T)$. This path can alternatively be defined by $\mathbf{z}(0)$ and a sequence of movement directions, $\theta(t) \in [-\pi, \pi]$, and speeds, $v(t) \in [0, \infty)$, such that $x(t+1) = x(t) + v\Delta t \cos\theta(t)$ and $y(t+1) = y(t) + v\Delta t \sin\theta(t)$, for small Δt (Fig. 1A). The accurate updating of $\mathbf{z}(t+1)$ from only $\mathbf{u}(t) = [\theta(t), v(t)]^\intercal$ is formally what is meant by **path integration**.

To learn a RNN model that can perform single agent path integration, prior work has utilized the following architecture (Fig. 1B). The inputs $\mathbf{u}(t)$ are projected, via weights $\mathbf{W}^{\text{in}} \in \mathbb{R}^{n_G \times 2}$, into a recurrent layer of n_G units. This recurrent layer has connectivity defined by weights $\mathbf{W}^{\text{rec}} \in \mathbb{R}^{n_G \times n_G}$. The units in the recurrent layer are used to drive the activation of n_P output units, via weights $\mathbf{W}^{\text{out}} \in \mathbb{R}^{n_P \times n_G}$. More explicitly, the RNN model evolves via the dynamics

$$\mathbf{r}(t+1) = \sigma\left[\mathbf{W}^{\text{rec}}\mathbf{r}(t) + \mathbf{W}^{\text{in}}\mathbf{u}(t)\right], \tag{1}$$

$$\hat{\mathbf{p}}(t+1) = \mathbf{W}^{\text{out}}\mathbf{r}(t+1), \tag{2}$$

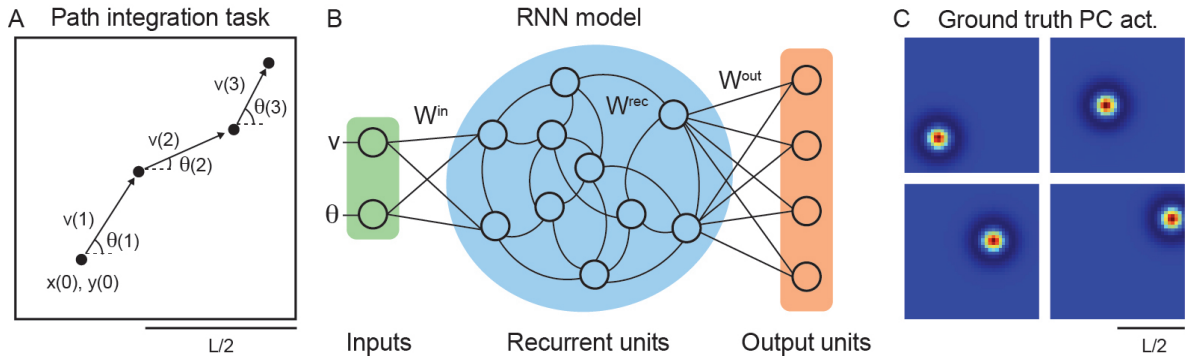

Figure 1: **Overview of RNNs trained to perform single agent path integration.** (A) Illustration of path integration task. The agent starts at known position $x(0), y(0)$ and makes a sequence of movements through space given by the movement directions $\theta(t)$ and speeds $v(t)$. From this, $x(t+1), y(t+1)$ must be estimated. (B) Schematic of single agent RNN model architecture [27, 28], which takes input $v(t)$ and $\theta(t)$, recurrently processes it, and drives activations of an output layer. (C) Example ground truth place cell (PC) activations used as targets for the RNN output units.

where $\mathbf{r}(t) \in \mathbb{R}^{n_G}$ and $\hat{\mathbf{p}}(t) \in \mathbb{R}^{n_P}$ are the activations of the recurrent and output units at time t, and $\sigma(\cdot)$ is an activation function.

The RNN model is trained by comparing the output layer activations, $\hat{\mathbf{p}}(t)$, with "ground truth" activations, $\mathbf{p}(t)$, using the cross entropy loss,

$$\mathcal{L}(t) = -\hat{\mathbf{p}}(t) \log \mathbf{p}(t) - \lambda ||\mathbf{W}^{\text{rec}}||_2. \tag{3}$$

The second term, $\lambda ||\mathbf{W}^{\text{rec}}||_2$, is a weight decay regularization term that penalizes the L_2 norm of the recurrent weights, by strength λ. This was found to be beneficial for generalization [32]. The form $\mathbf{p}(t)$ takes has been shown to determine properties of the representations learned by the single agent RNNs [28]. We follow Sorscher et al. (2019) and set the ground truth activations to be "place cell"-like [36], with the place fields having a difference of Gaussian (DoG) structure [28] (Fig. 1C),

$$p_i(t) = \text{DoG}[\mathbf{z}(t), \mathbf{c}_i, \sigma_1, \sigma_2] = \exp\Big(-||\mathbf{z}(t) - \mathbf{c}_i||_2^2/2\sigma_1^2\Big) - \exp\Big(-||\mathbf{z}(t) - \mathbf{c}_i||_2^2/2\sigma_2^2\Big) \tag{4}$$

where $p_i(t)$ is the ground truth activation of unit i at time t, $\mathbf{c}_i \in [-L/2, L/2] \times [-L/2, L/2]$ is the center of its place field, and $\sigma_1 < \sigma_2$ define the place field's size. The initial activation in the recurrent layer is set using the ground truth place cell activations, $\mathbf{r}(0) = \mathbf{W}^{\text{back}}\mathbf{p}(0)$, where $\mathbf{W}^{\text{back}} \in \mathbb{R}^{n_G \times n_P}$.

To assess the RNN model's performance in single agent path integration, the position of the agent is decoded from the activations of the output layer. Prior work has used a simple top-n_d decoder, where the predicted location of the agent is found by averaging the place field centers, $\mathbf{c}_i$, corresponding to the $n_d \in \mathbb{N}$ units in the output layer with the highest activations. That is, $\hat{\mathbf{z}}(t) = \frac{1}{n_d}\sum_{i=1}^{n_d} \mathbf{c}_{\alpha_i}(t)$, where the α_i correspond to the ranked order of the output layer activations [i.e., $\hat{p}_{\alpha_1}(t) > \hat{p}_{\alpha_2}(t) > ... > \hat{p}_{\alpha_{n_P}}(t)$]. The decoding error is then defined as $||\mathbf{z}(t) - \hat{\mathbf{z}}(t)||_2$.

To enable the RNN model to perform dual agent path integration, we make four modifications (see Appendix A and Table S1, for more details):

- **Additional inputs:** We set $\mathbf{u}(t) = [\theta_1(t), v_1(t), \theta_2(t), v_2]^\intercal$, where $\theta_i(t)$ and $v_i(t)$ correspond to the ith agent's movement direction and speed at time t.

- **Summed place cell activations:** We set the ground truth place cell responses to be the sum of their responses to the location of each agent independently. That is, if $\mathbf{z}_i(t)$ corresponds to the position of the ith agent at time t, then $p_i(t) = \text{DoG}[\mathbf{z}_1(t), \mathbf{c}_i, \sigma_1, \sigma_2] + \text{DoG}[\mathbf{z}_2(t), \mathbf{c}_i, \sigma_1, \sigma_2]$ (Eq. 4). This choice is inspired, in part, by "social" place cells [11–16].

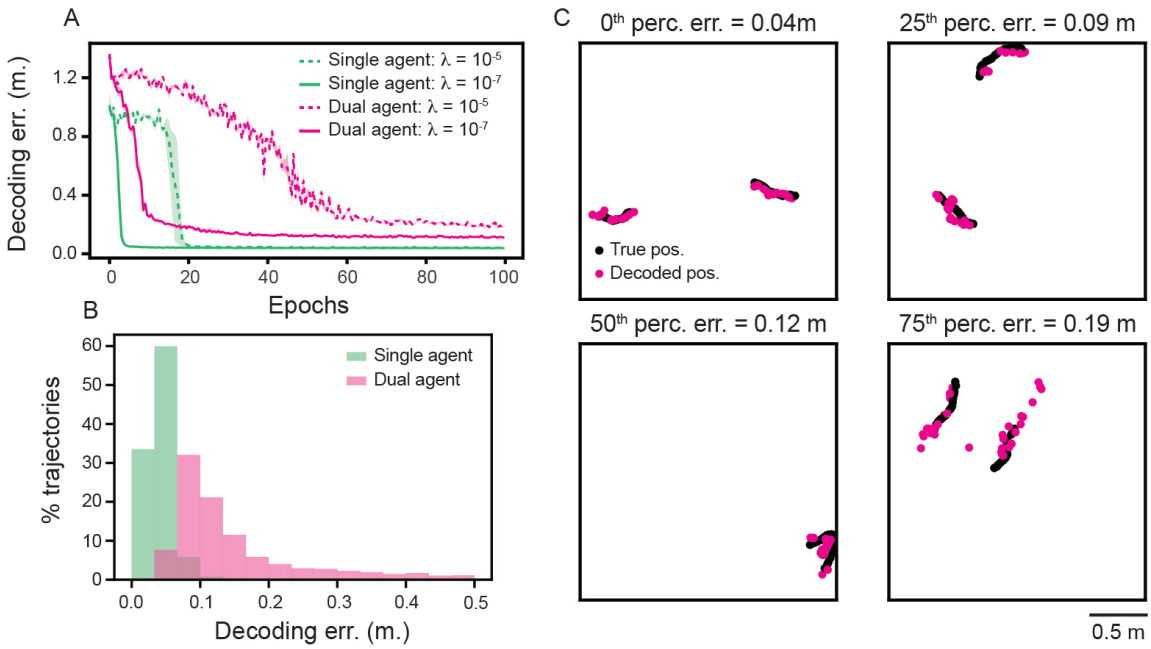

Figure 2: **RNNs can be trained to successfully perform dual agent path integration.** (A) Decoding error, as a function of training epoch, for RNNs trained and tested on single and dual agent path integration. Solid line is mean across 5 independently trained networks and shaded area is maximum and minimum of all 5 networks. (B) Distribution of median decoding error, across 5000 trajectories (1000 per network), for single and dual agent RNNs. (C) Example ground truth and decoded dual agent trajectories. Trajectories were chosen as those closest to the 0th, 25th, 50th, and 75th percentile of the decoding error distribution.

- **k-means clustering for decoding:** We use k-means clustering to separate the place field centers corresponding to the $2n_d$ output layer units with the highest activations into $k = 2$ groups. The predicted positions of the two agents, $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{z}}_2(t)$, are estimated by the center of each cluster.
- **Reduced weight decay regularization:** We reduce the weight regularization strength, λ (Eq. 3), as we expect the added complexity of dual agent path integration to require less sparsity in $\mathbf{W}^{\text{rec}}$.

Training this modified RNN on dual agent path integration, we show – for the *first time* – that it is possible to learn a network that achieves high performance (Figs. 2, S1). While the decoding error of dual agent path integration is greater for dual agent RNNs than the decoding error of single agent path integration is for single agent RNNs (Fig. 2A, B), we find that nearly 50% of all 5000 dual agent trajectories sampled have an error below 0.10 m, which was previously used as a threshold to determine success [33] (Fig. 2B). In addition, visualizing the RNN's prediction on individual dual agent trajectories shows qualitatively similar behavior to the true trajectories, even when decoding error is > 0.10 m. (Fig. 2C). As hypothesized, reducing the strength of weight decay, from $\lambda = 10^{-5}$ to $\lambda = 10^{-7}$, enables an increase in performance on dual agent path integration (Fig. 2A – compare pink solid and dashed lines).

2 Results

2.1 Representations in single agent RNNs are not optimal for dual agent path integration

The discovery that recurrent units in RNNs trained on single agent path integration (Sec. 1) develop responses like those found in MEC [27–29] has been interpreted as a normative demonstration of the optimality of these representations. Are these same representations also optimal for dual agent path integration?

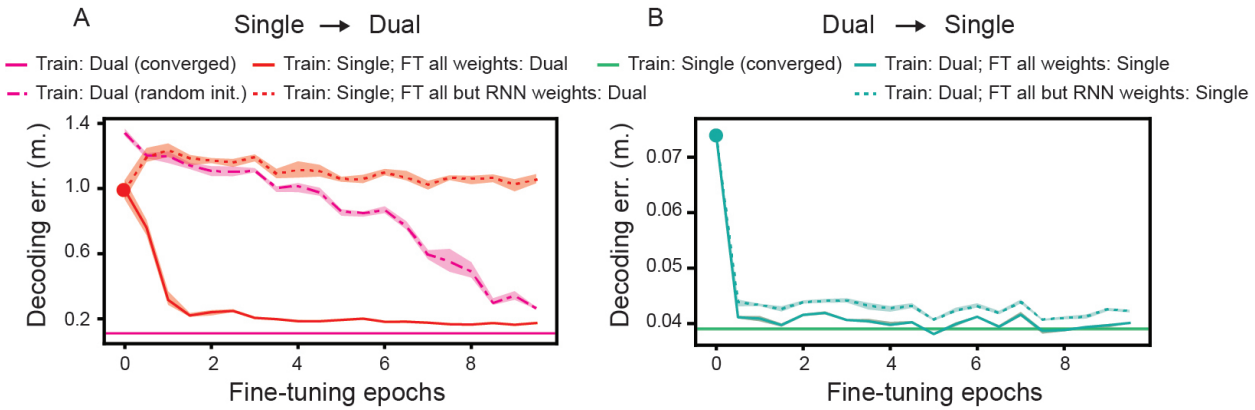

Figure 3: **Dual agent RNNs can generalize to single agent path integration, but not vice versa.** (A) Decoding error of RNNs trained on single agent path integration and tested on dual agent path integration with (red lines) and without (red dot) fine-tuning (FT) on dual agent path integration. Converged performance and performance from random initialization, of dual agent RNNs (dashed and solid pink lines, respectively), are shown for comparison. (B) Same as (A), but for RNNs trained on dual agent path integration and tested on single agent path integration. (A)–(B) Lines denote mean and shaded area denotes maximum and minimum, across 5 independently trained RNNs.

To test this, we apply trained single agent RNNs to the dual agent setting (Appendix B). These networks have a decoding error of ≈ 1.0 m. (Fig. 3A – red dot), nearly a $10\times$ increase in error relative to the trained dual agent RNNs (Fig. 3A – pink solid line). To determine whether this high decoding error is due to poor generalization or to the RNN being sensitive to the precise value of the network weights, we "fine-tune" all the trained single agent RNN weights ($\mathbf{W}^{\text{rec}}$, $\mathbf{W}^{\text{in}}$, $\mathbf{W}^{\text{out}}$, $\mathbf{W}^{\text{back}}$ – Sec. 1) for 10 epochs on the dual agent path integration task (Appendix B). We find that this leads to an improvement in performance on dual agent path integration, although the end decoding error remains higher than the decoding error of the trained dual agent RNN (Fig. 3A – compare solid red and pink lines). However, this improvement is dependent on updating the trained weights in the recurrent layer, as freezing $\mathbf{W}^{\text{rec}}$ and fine-tuning the other weights again leads to high decoding error (Fig. 3A – red dashed line). Taken together, these results demonstrate that the representations learned by the recurrent units in trained single agent RNNs are not optimal for dual agent path integration.

In contrast, trained dual agent RNNs can perform single agent path integration with decoding error < 0.10 m. (Fig. 3B – cyan dot). Fine-tuning the weights on single agent path integration further improves performance, even when $\mathbf{W}^{\text{rec}}$ is frozen (Fig. 3B – cyan dashed line). This is aligned with recent experimental and computational work showing that fixed MEC connectivity and slow time-scale plasticity between MEC and hippocampus enables generalizable representations of new environments [37]. These results demonstrate that dual agent RNNs have internal representations that are flexible enough to enable generalization to single agent path integration.

2.2 Single and dual agent RNNs develop different representations at the individual unit level

To begin dissecting the representations that emerge in dual agent RNNs, we examine the functional properties of individual units in the recurrent layer. Previous work has found that single agent RNNs develop responses analogous to three functional classes of neurons in MEC: grid cells [5, 6], border cells [7], and band cells [8]. The distribution of these properties can be quantified by computing scores for each unit (Appendix C.2). While grid and border scores are standard metrics, to our knowledge no band score exists. We therefore developed a metric to quantify "banded-ness" (Appendix C.2)

We find statistically significant differences in the distribution of these functional properties between single and dual agent RNNs (Fig. 4A, B). In particular, we find dual agent RNNs develop units with lower grid scores, but higher border and band scores (Fig. 4A). Visualizing rate maps associated with the dual

agent RNN units that have the highest grid scores, we find that many have responses that are not solely constrained to vertices of triangular lattices, but are instead "honeycomb"-like (Fig. 4B). This suggests that dual agent path integration leads to a weaker grid code. In contrast, dual agent RNN units with high border and band scores have more selective tuning, as compared to units of single agent RNNs, which have additional grid structure (Fig. 4B). This suggests dual agent path integration leads to a strengthened border and band code. These results are consistently observed across all 5 independently trained RNNs (Fig. S2).

Prior work has found all three of these classes to be important for single agent path integration [27, 29, 31, 32]. To understand their importance in dual agent path integration, we perform targeted ablations, removing units with the highest grid, border, and band scores (Appendix C.3). We compare the decoding error of these RNNs to RNNs with randomly removed units, as this has been found to be a strong baseline [31, 38]. Unlike single agent RNNs, which have a significant increase in decoding error relative to the random baseline with the ablation of all functional classes, dual agent RNNs are less sensitive to the ablation of any one functional class (Fig. 4C). This suggests that dual agent RNNs develop a more distributed and robust representation.

To determine the extent to which our results are due to the nature of dual agent path integration, as opposed to specific choices of RNN architecture and hyper-parameters, we perform several control experiments. First, we consider the possibility that, by having twice as many agents to track, dual agent RNNs have half the effective size of single agent RNNs. In such a case, we might expect a smaller single agent RNN to generate responses similar to those of dual agent RNNs. However, half-sized single agent RNNs have representations that are distinct from the full-sized dual agent RNNs (Fig. S3A). Second, dual agent rate maps are computed using the position of only one of the two agents (the other agent being "averaged out" – Appendix C.1). We re-compute the rate maps of dual agent RNNs, removing one of the agents completely. Thus, while the network was trained in the dual agent setting, the rate maps were only affected by a single agent. We find this to have little effect on the representations (Fig. S3B). And third, we measure the distribution of grid, border, and band scores for the RNNs trained with greater weight decay regularization (Fig. 2A – dashed lines). We find that the overall trend of dual agent path integration leading to lower grid scores and higher border scores, relative to single agent path integration, holds (Fig. S4).

Taken together, these results demonstrate that the addition of another dynamic variable (i.e., a second agent) leads to differences in the distribution of functional properties, at the individual unit level, between single and dual agent RNNs.

2.3 Dual agent RNNs develop tuning in relative space

Recent neural recordings have demonstrated that, when mice are trained to navigate to reward sites that move in physical space, grid cells perform path integration in reference frames that are anchored to the locations of these rewards [39]. These results highlight the fact that hippocampus and MEC can perform computations in different coordinate systems [40]. Inspired by these results, we examine whether dual agent RNNs develop grid responses in another reference frame. One candidate coordinate system is defined by the relative positions of the two agents ("relative space" – Fig. 5A, Appendix D). We do not find evidence for units having grid responses in relative space (Fig. 5B, C), demonstrating that dual agent RNNs are not simply using the same representations as in the single agent setting, but in a different reference frame. However, we find that units classified as border and band cells in the absolute (i.e., allocentric) reference frame have strong tuning in relative space (Fig. S5A). Units with high border score have greatest activations when both agents are near borders (Fig. S5A, B) and units with high band score have greatest activations when both agents are positioned at periodic distances, along parallel lines (Fig. S5A, B). The latter could enable easy representation of parallel movement, where the agents are naturally separated. In contrast, units with high grid score have activations that are largely invariant to relative space (Fig. S5A, B).

To further characterize the extent to which dual agent RNNs encode relative position, we compute the spatial information [41] of all recurrent units relative space rate maps (Appendix D.2). We find that, while most units contain little spatial information in relative space (corresponding to spatial information ≈ 0), the distribution is heavy tailed (Fig. 5D). Visualizing the relative space rate maps with the highest spatial

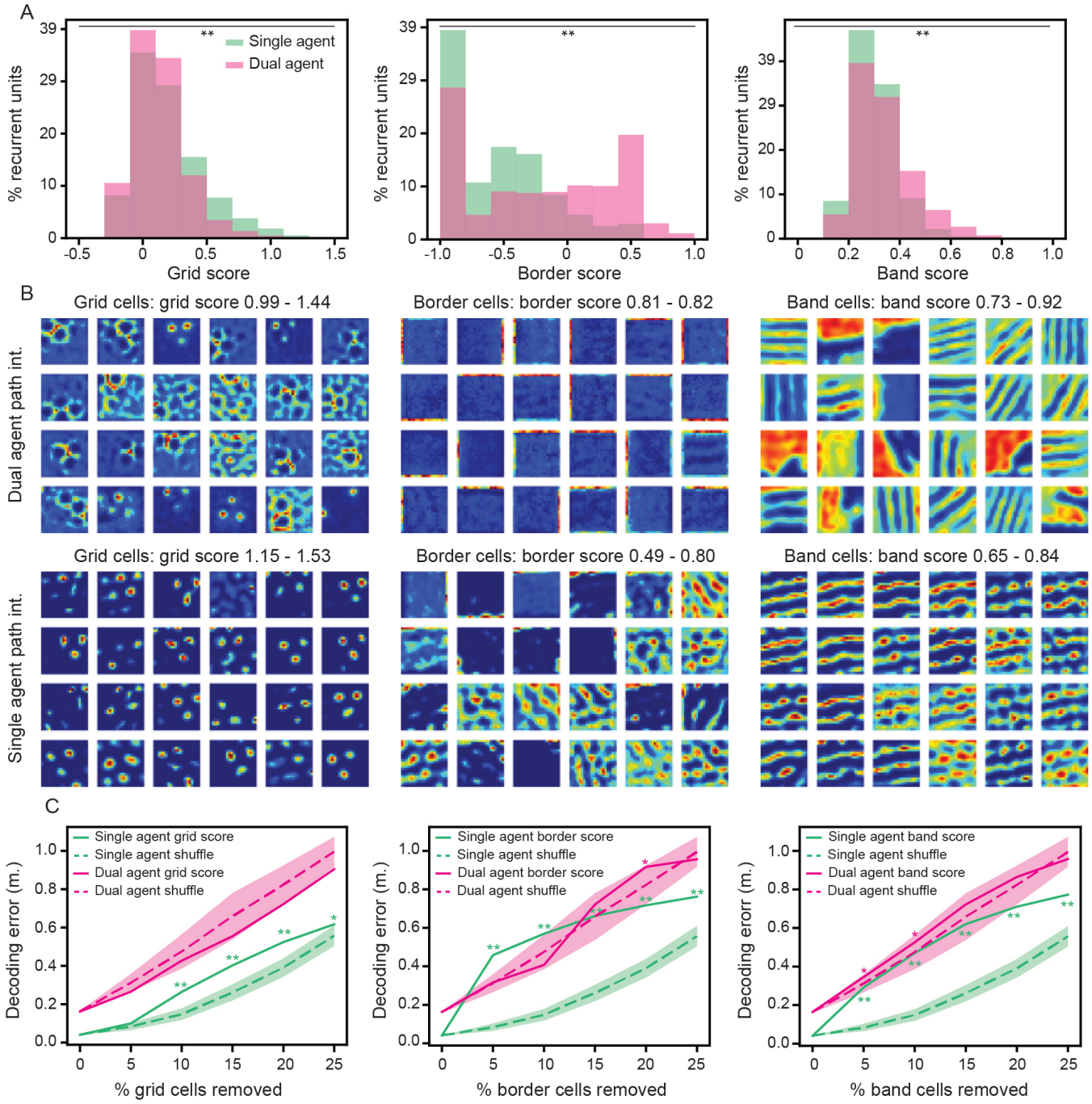

Figure 4: **Single and dual agent RNNs differ in their distribution of functional properties.** (A) Distribution of grid, border, and band scores, computed for all units of single and dual agent RNNs. Distribution includes 5 independently trained RNNs. Kolmogorov-Smirnov (KS) test used to compare distributions ($** : p < 0.01$). (B) Visualization of rate maps (Appendix C.1) for units, from individual RNNs, with highest grid, border, and band scores. (C) Decoding error of trained RNNs with the units corresponding to the highest grid, border, and band score ablated (solid line: mean across 5 independently trained RNNs), compared to trained RNNs with random units ablated (dashed lines: mean across 10 randomly sampled choices of ablated units for each independently trained RNN; shaded area: $\pm$ standard deviation) (Appendix C.3). Mann-Whitney test is used to compare distributions ($* : p < 0.05$ and $** : p < 0.01$ – see Table S2 for p-values).

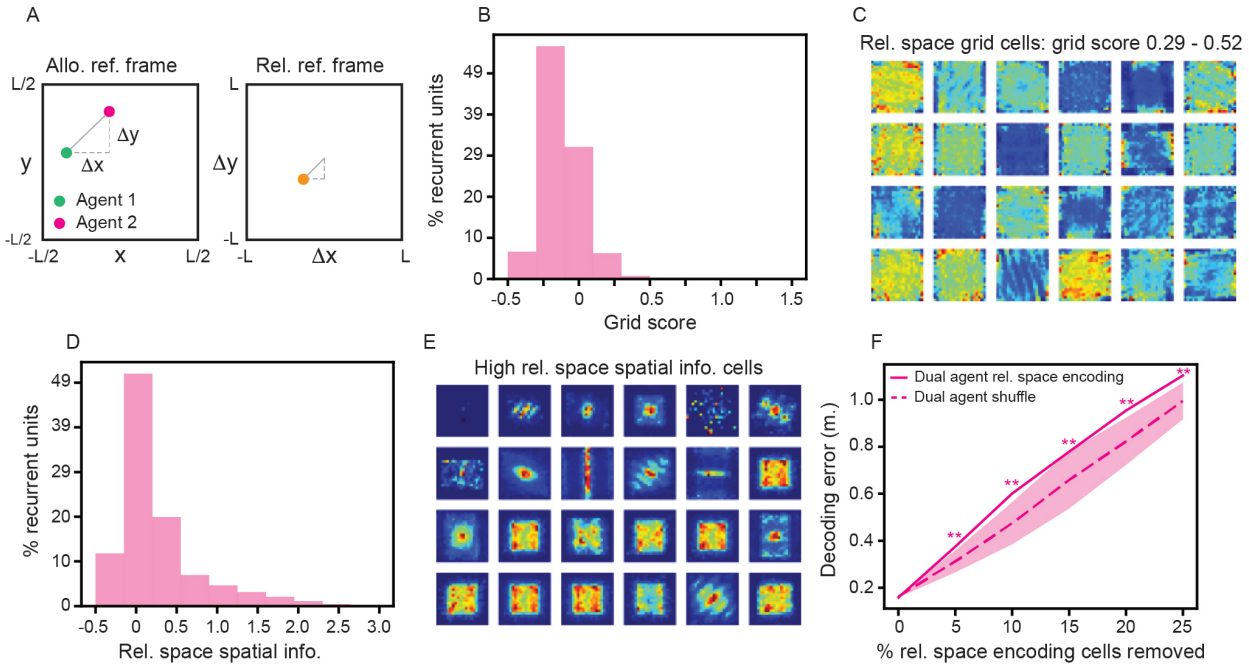

Figure 5: **Dual agent RNNs develop tuning to relative space.** (A) Schematic illustration of a transformation from the allocentric reference frame to the relative position reference frame. (B) Distribution of grid scores (across 5 independently trained RNNs), computed on the relative space rate maps (Appendix D.1). (C) Visualization of relative space rate maps, from an individual dual agent RNN, for units with the highest relative space grid scores. (D) Distribution of spatial information, computed on the relative space rate maps. Distribution includes 5 independently trained RNNs. (E) Visualization of relative space rate maps, from an individual dual agent RNN, with the highest relative space spatial information. (F) Decoding error of dual agent RNN with units having the highest relative space spatial information ablated. Dashed line and shaded area same as Fig. 4C. Mann-Whitney test is used to compute p-values ($** : p < 0.01$ – see Table S3 for statistics).

information, we find that many are localized, with greatest activation when the two agents are near each other (Fig. 5E – center pixels). We hypothesize that, because accurate dual agent path integration becomes more challenging as the two agents get closer together, ablating units with high spatial information in the relative space reference frame would lead to an increase in decoding error. In-line with this hypothesis, we find that removing the units with the highest relative space spatial information leads to a statistically significant decrease in decoding performance, relative to the random ablation baseline (Fig. 5F, Appendix D.3). That this ablation leads to stronger deficits than the ablation of grid, border, and band cells (Fig. 4C), emphasizes of their importance in supporting dual agent path integration.

2.4 Single and dual agent RNNs learn different representations at the population level

Our comparison of the representations that emerge in single and dual agent RNNs has thus far been focused on the activations of individual recurrent units. Our findings that dual agent RNNs develop different distributions of functional classes than single agent RNNs, and that dual agent RNNs encode relative space spatial information, are suggestive that the network structure underlying single and dual agent RNNs are distinct. To further explore this hypothesis, we analyze the recurrent unit activations at the population level.

Topological data analysis (TDA) [42] has become a widely used tool for studying neural population activity [24, 43–47]. A core technique in TDA is persistent homology (Appendix E.1), which can be used to estimate the topology of the underlying neural manifold. Applying persistent homology, we find evidence that the population activations of single agent RNNs are constrained to a manifold having topology consistent with a two-dimensional torus (Fig. 6A, left – black arrows point to features expected from a

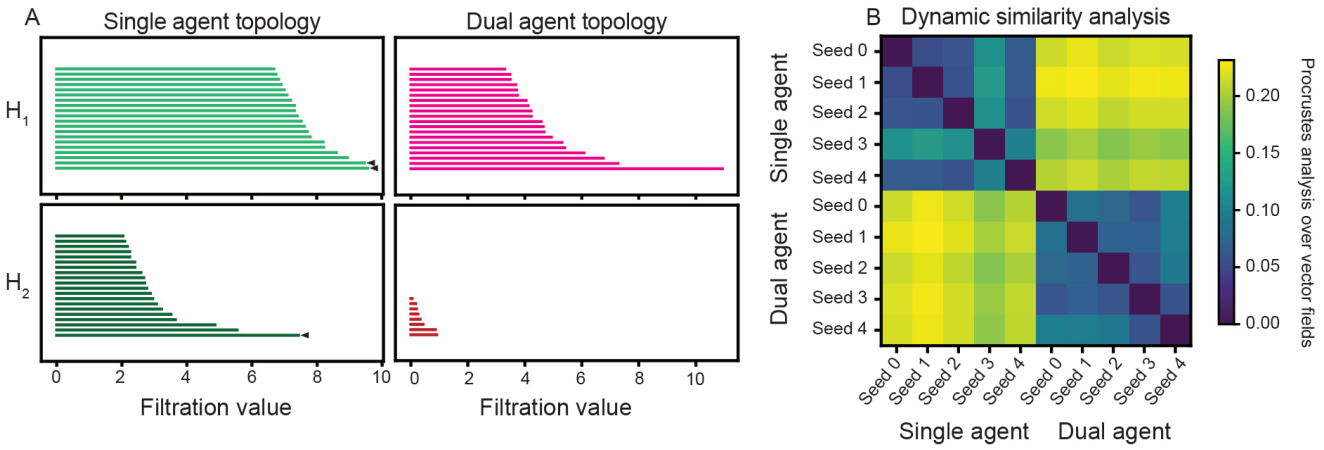

Figure 6: **Population level activations of single and dual agent RNNs differ in their topology and dynamics.** (A) Persistence diagrams (Appendix E.1) of the population activations of example single and dual agent RNNs. H_1 bars of high filtration value correspond to existence of loops and H_2 bars correspond to existence of two-dimensional cavities. Black arrows denote features of the single agent persistence diagram that are consistent with two-dimensional toroidal attractor. (B) DSA (Appendix E.2) applied to the activations of independently trained single and dual agent RNNs. Color denotes Procrustes analysis over vector fields metric.

torus). Measuring the distance [48] between the topological characteristics computed from applying TDA to the single agent RNN activations and applying TDA to an idealized torus, we find a sharp decrease during training (Fig. S6B). These results are consistent with similar analysis that was performed previously on single agent RNNs [31, 32]. When performing the same analysis on the population activations of dual agent RNNs, we find differences in the persistent homology. In particular, the topological signatures of a two-dimensional torus are not clearly present (Fig. 6A, right). Additionally, the topology does not become increasingly closer to an ideal torus with training (Fig. S7B). This suggests that dual agent RNNs do not have the same continuous attractor structure that single agent RNNs have [31, 32].

The topology and geometry of neural manifolds do not capture all properties of the underlying neural circuit. Indeed, RNNs with the same dynamical properties (e.g., existence of fixed point attractors, limit cycles) can be constructed to have different population activation geometries [49]. Similarly, RNNs with the same population activation geometries can be constructed to have different dynamics [49]. Recent work has developed a data-driven dynamical systems framework [dynamic similarity analysis (DSA)] for computing compact representations of RNN dynamics [50]. Therefore, we complemented our TDA results by applying DSA (Appendix E.2) to the population activations of single and dual agent RNNs. We find that independently trained single agent RNNs are dynamically more similar to each other than they are to independently trained dual agent RNNs, and vice versa (Fig. 6B). Therefore, at both the topological and dynamical level, the representations learned by the single and dual agent RNN are different, suggesting that the addition of a second agent fundamentally changes the network structure that emerges.

3 Discussion

Inspired by recent work demonstrating that brain regions believed to encode the self's position and trajectory (e.g., MEC, hippocampus) are modulated by the position and trajectories of others [9–15], we trained a class of RNN models [27, 28] to path integrate two agents and investigated the representations that emerged. Given that these RNN models, when trained to path integrate single agents, develop properties that are similar to those found in MEC at the individual unit (e.g., grid, border, and band responses) and population (e.g., network structure with signatures of continuous toroidal attractors) level, this is a natural point for starting to address the larger question of how the MEC supports representation of multiple dynamic individuals.

We find that the representations that emerge in dual agent RNNs significantly differ from those of single agent RNNs. Dual agent path integration leads to weaker grid codes and stronger border codes (Fig. 4). This is consistent with electrophysiological recordings that have found MEC encodes distance of others to boundaries [9] and fMRI recordings that have observed a negative correlation between grid cell strength and performance on multi-agent tasks [10]. Functional class ablations and TDA revealed a distributed and robust representation, with no evidence for the population activity being constrained to a two-dimensional torus (Fig. 6). Despite this, dual agent RNNs were able to generalize to perform single agent path integration, even when the weights in their recurrent layer are frozen (Fig. 3B). This demonstrates the flexibility of their representations [37] and is in-line with work finding that neural networks trained to perform multiple tasks develop abstract and distributed tuning [51–54]. Finally, we found that RNNs trained on dual agent path integration develop tuning in the coordinate system defined by the relative position of the two agents (Figs. 5, S5). These results are aligned with neurophysiological experiments that have uncovered neurons in MEC that encode distance and orientation of self to objects [55].

Limitations. The biological plausibility of the RNN models investigated in this work [27, 28] have been questioned [33, 34] (although, see response by Sorscher et al. (2022) [56]). While our dual agent ground truth place cells were motivated by experimental work measuring hippocampal responses in environments with multiple animals [11–16], it represents only one possible choice. Additionally, it has been found that the RNN model develops weaker modularization in grid properties [27–29], a feature that is believed to be critical for encoding local spatial information [57–59] (although see work suggesting individual grid modules contain more spatial information than previously thought [60–63]). Recent use of loss functions that incorporate more than just path integration have found stronger grid responses and modularization [64, 65]. Future work should investigate the properties that emerge in these models when tasked with dual agent path integration. In our experiments, we did not account for noise in velocity inputs, which have been found to play a dominant role in human path integration error [66], and are presumably higher for the assessment of others' velocities (given the lack of vestibular and proprioceptive feedback). We hypothesize that, as noise is added to the estimate of one agent's position and velocity, dual agent RNNs will develop representations more similar to single agent RNNs, suggesting a trade-off that should be explored.

Predictions. Recent advances in experimental paradigms, where rodents must update their goal-based planning given the trajectories of others [67–69], offer the possibility of comparing the *in vivo* representations of MEC with those generated by the dual agent RNN. Our results make several testable predictions about what responses might be found in MEC. First, we expect grid responses to be weakened, with the cells that do have high grid score exhibiting deviations from standard grid cells ("honeycomb"-like responses – Fig. 4). Second, we expect strengthened boundary and border responses (Fig. 4), with border cells responding most strongly when both agents are near borders (Fig. S5). Finally, we expect that the relative position between agents will be encoded in the a subpopulation of MEC neurons (Fig. 5) that is disjoint from cells with high grid scores (Fig. S5).

Future directions. There are several impactful directions that our work can be extended. First, analyzing the emergent properties of RNNs trained to path integrate more than two agents can lead to a better understanding of how the MEC performs computations in the multi-agent setting. Second, reinforcement learning can be used to probe how the dual agent RNN representations might support higher order behavior [70, 71] (such as short-cut taking in dynamic chasing [72]). Theoretical work, showing – from first principles – what representations are necessary for single agent path integration [73, 74] can be extended to the dual agent path integration setting. This will provide a thorough treatment of how the complexity of multi-agent environments changes the computations needed to be performed by MEC. Finally, foundational work on MEC has found that grid cells encode abstract relationships beyond purely spatial variables [75]. These include social hierarchies [76], sound frequencies [77], and elapsed time [78]. Our work has implications for how the MEC may simultaneously integrate information across these different cognitive axes.

Acknowledgements

We thank Andy Alexander, Michael Goard, Kechen Zhang, and members of the Goard Lab for helpful discussion surrounding this work. We thank Nora Wolcott, Adele Meyers, Anna-Lena Krause, and Luís Pereira for feedback on this manuscript. WTR acknowledges funding by an Internal Research and Development grant from JHU/APL. FA acknowledges funding from the National Science Foundation, grant 2134241. NM acknowledges funding from the National Science Foundation, grant 2313150.

References

[1] Edmund T Hall and Edward T Hall. *The hidden dimension*, volume 609. Anchor, 1966.

[2] Kazushi Tsutsui, Masahiro Shinya, and Kazutoshi Kudo. Human navigational strategy for intercepting an erratically moving target in chase and escape interactions. *Journal of motor behavior*, 52(6):750–760, 2020.

[3] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, 2006.

[4] Mariana Gil, Mihai Ancau, Magdalene I Schlesiger, Angela Neitz, Kevin Allen, Rodrigo J De Marco, and Hannah Monyer. Impaired path integration in mice with disrupted grid cell firing. *Nature neuroscience*, 21(1):81–91, 2018.

[5] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

[6] Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May-Britt Moser, and Edvard I Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72–78, 2012.

[7] Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-Britt Moser, and Edvard I Moser. Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868, 2008.

[8] Julija Krupic, Neil Burgess, and John O'Keefe. Neural representations of location composed of spatially periodic bands. *Science*, 337(6096):853–857, 2012.

[9] Matthias Stangl, Uros Topalovic, Cory S Inman, Sonja Hiller, Diane Villaroman, Zahra M Aghajan, Leonardo Christov-Moore, Nicholas R Hasulak, Vikram R Rao, Casey H Halpern, et al. Boundary-anchored neural mechanisms of location-encoding for self and others. *Nature*, 589(7842):420–425, 2021.

[10] Isabella C Wagner, Luise P Graichen, Boryana Todorova, Andre Lüttig, David B Omer, Matthias Stangl, and Claus Lamm. Entorhinal grid-like codes and time-locked network dynamics track others navigating through space. *Nature Communications*, 14(1):231, 2023.

[11] Xiang Mou and Daoyun Ji. Social observation enhances cross-environment activation of hippocampal place cell patterns. *Elife*, 5:e18022, 2016.

[12] Teruko Danjo, Taro Toyoizumi, and Shigeyoshi Fujisawa. Spatial representations of self and other in the hippocampus. *Science*, 359(6372):213–218, 2018.

[13] David B Omer, Shir R Maimon, Liora Las, and Nachum Ulanovsky. Social place-cells in the bat hippocampus. *Science*, 359(6372):218–224, 2018.

[14] Jeroen J Bos, Martin Vinck, Pietro Marchesi, Amos Keestra, Laura A van Mourik-Donga, Jadin C Jackson, Paul FMJ Verschure, and Cyriel MA Pennartz. Multiplexing of information about self and others in hippocampal ensembles. *Cell Reports*, 29(12):3859–3871, 2019.

[15] Angelo Forli and Michael M Yartsev. Hippocampal representation during collective spatial behaviour in bats. *Nature*, pages 1–8, 2023.

[16] Xiang Zhang, Qichen Cao, Kai Gao, Cong Chen, Sihui Cheng, Ang Li, Yuqian Zhou, Ruojin Liu, Jun Hao, Emilio Kropff, et al. Multiplexed representation of others in the hippocampal ca1 subfield of female mice. *Nature Communications*, 15(1):3702, 2024.

[17] Mark C Fuhs and David S Touretzky. A spin glass model of path integration in rat medial entorhinal cortex. *Journal of Neuroscience*, 26(16):4266–4276, 2006.

[18] Alexis Guanella and Paul FMJ Verschure. A model of grid cells based on a path integration mechanism. In *Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I 16*, pages 740–749. Springer, 2006.

[19] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291, 2009.

[20] Jonathan J Couey, Aree Witoelar, Sheng-Jia Zhang, Kang Zheng, Jing Ye, Benjamin Dunn, Rafal Czajkowski, May-Britt Moser, Edvard I Moser, Yasser Roudi, et al. Recurrent inhibitory circuitry as a mechanism for grid formation. *Nature neuroscience*, 16(3):318–324, 2013.

[21] KiJung Yoon, Michael A Buice, Caswell Barry, Robin Hayman, Neil Burgess, and Ila R Fiete. Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nature neuroscience*, 16(8):1077–1084, 2013.

[22] Benjamin Dunn, Maria Mørreaunet, and Yasser Roudi. Correlations and functional connections in a population of grid cells. *PLoS computational biology*, 11(2):e1004052, 2015.

[23] Yi Gu, Sam Lewallen, Amina A Kinkhabwala, Cristina Domnisoru, Kijung Yoon, Jeffrey L Gauthier, Ila R Fiete, and David W Tank. A map-like micro-organization of grid cells in the medial entorhinal cortex. *Cell*, 175(3):736–750, 2018.

[24] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022.

[25] Simon M Stringer, Edmund T Rolls, and Thomas P Trappenberg. Self-organising continuous attractor networks with multiple activity packets, and the representation of space. *Neural networks*, 17(1):5–27, 2004.

[26] Sung Soo Kim, Hervé Rouault, Shaul Druckmann, and Vivek Jayaraman. Ring attractor dynamics in the drosophila central brain. *Science*, 356(6340):849–853, 2017.

[27] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

[28] Ben Sorscher, Gabriel Mel, Surya Ganguli, and Samuel Ocko. A unified theory for the origin of grid cells through the lens of pattern formation. *Advances in neural information processing systems*, 32, 2019.

[29] Christopher J. Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *International Conference on Learning Representations*, 2018.

[30] Nikhil Ahuja, Veronika Lobellová, Aleš Stuchlík, and Eduard Kelemen. Navigation in a space with moving objects: rats can avoid specific locations defined with respect to a moving robot. *Frontiers in behavioral neuroscience*, 14:576350, 2020.

[31] Vemund Schøyen, Markus Borud Pettersen, Konstantin Holzhausen, Marianne Fyhn, Anders Malthe-Sørenssen, and Mikkel Elle Lepperød. Coherently remapping toroidal cells but not grid cells are responsible for path integration in virtual agents. *Iscience*, 26(11), 2023.

[32] Ben Sorscher, Gabriel C Mel, Samuel A Ocko, Lisa M Giocomo, and Surya Ganguli. A unified theory for the computational and mechanistic origins of grid cells. *Neuron*, 2022.

[33] Rylan Schaeffer, Mikail Khona, and Ila Fiete. No free lunch from deep learning in neuroscience: A case study through models of the entorhinal-hippocampal circuit. *Advances in Neural Information Processing Systems*, 35:16052–16067, 2022.

[34] Rylan Schaeffer, Mikail Khona, Adrian Bertagnoli, Sanmi Koyejo, and Ila Rani Fiete. Testing assumptions underlying a unified theory for the origin of grid cells. *arXiv preprint arXiv:2311.16295*, 2023.

[35] Kapil Katyal, Yuxiang Gao, Jared Markowitz, Sara Pohland, Corban Rivera, I-Jeng Wang, and Chien-Ming Huang. Learning a group-aware policy for robot navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11328–11335. IEEE, 2022.

[36] John O'Keefe and Lynn Nadel. *The Hippocampus as a Cognitive Map*. Oxford University Press, 1978.

[37] John H Wen, Ben Sorscher, Surya Ganguli, and Lisa Giocomo. One-shot entorhinal maps enable flexible navigation in novel environments. *bioRxiv*, pages 2023–09, 2023.

[38] Aran Nayebi, Alexander Attinger, Malcolm Campbell, Kiah Hardcastle, Isabel Low, Caitlin S Mallory, Gabriel Mel, Ben Sorscher, Alex H Williams, Surya Ganguli, et al. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34:12167–12179, 2021.

[39] Jing-Jie Peng, Beate Throm, Maryam Najafian Jazi, Ting-Yun Yen, Hannah Monyer, and Kevin Allen. Grid cells perform path integration in multiple reference frames during self-motion-based navigation. *bioRxiv*, 2023.

[40] Eduard Kelemen and André A Fenton. Dynamic grouping of hippocampal neural activity during cognitive control of two spatial frames. *PLoS biology*, 8(6):e1000403, 2010.

[41] William Skaggs, Bruce Mcnaughton, and Katalin Gothard. An information-theoretic approach to deciphering the hippocampal code. *Advances in neural information processing systems*, 5, 1992.

[42] Chad M Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PloS one*, 10(5):e0126383, 2015.

[43] Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L Ringach. Topological analysis of population activity in visual cortex. *Journal of vision*, 8(8):11–11, 2008.

[44] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, 2015.

[45] Gard Spreemann, Benjamin Dunn, Magnus Bakke Botnan, and Nils A Baas. Using persistent homology to reveal hidden information in neural data. *arXiv preprint arXiv:1510.06629*, 2015.

[46] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature neuroscience*, 22(9):1512–1520, 2019.

[47] Eliott Robert Joseph Levy, Simón Carrillo-Segura, Eun Hye Park, William Thomas Redman, José Rafael Hurtado, SueYeon Chung, and André Antonio Fenton. A manifold neural population code for space in hippocampal coactivity dynamics independent of place fields. *Cell reports*, 42(10), 2023.

[48] Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

[49] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in neural information processing systems*, 32, 2019.

[50] Mitchell Ostrow, Adam Eisen, Leo Kozachkov, and Ila Fiete. Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. *arXiv preprint arXiv:2306.10168*, 2023.

[51] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

[52] Laura Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *bioRxiv*, pages 2022–08, 2022.

[53] Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *Nature neuroscience*, 25(6):783–794, 2022.

[54] W Jeffrey Johnston and Stefano Fusi. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nature Communications*, 14(1):1040, 2023.

[55] Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, 2019.

[56] Ben Sorscher, Gabriel C Mel, Aran Nayebi, Lisa Giocomo, Daniel Yamins, and Surya Ganguli. When and why grid cells appear or not in trained path integrators. *bioRxiv*, pages 2022–11, 2022.

[57] Ila R Fiete, Yoram Burak, and Ted Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27):6858–6871, 2008.

[58] Xue-Xin Wei, Jason Prentice, and Vijay Balasubramanian. A principle of economy predicts the functional architecture of grid cells. *Elife*, 4:e08362, 2015.

[59] Martin Stemmler, Alexander Mathis, and Andreas VM Herz. Connecting multiple spatial scales to decode the population activity of grid cells. *Science Advances*, 1(11):e1500816, 2015.

[60] Geoffrey W Diehl, Olivia J Hon, Stefan Leutgeb, and Jill K Leutgeb. Grid and nongrid cells in medial entorhinal cortex represent spatial location and environmental features with complementary coding schemes. *Neuron*, 94(1):83–92, 2017.

[61] Revekka Ismakov, Omri Barak, Kate Jeffery, and Dori Derdikman. Grid cells encode local positional information. *Current Biology*, 27(15):2337–2343, 2017.

[62] Benjamin Dunn, Daniel Wennberg, Ziwei Huang, and Yasser Roudi. Grid cells show field-to-field variability and this explains the aperiodic response of inhibitory interneurons. *arXiv preprint arXiv:1701.04893*, 2017.

[63] William T Redman, Santiago Acosta-Mendoza, Xue-Xin Wei, and Michael J Goard. Robust variability of grid cell properties within individual grid modules enhances encoding of local space. *bioRxiv*, pages 2024–02, 2024.

[64] Ruiqi Gao, Jianwen Xie, Xue-Xin Wei, Song-Chun Zhu, and Ying Nian Wu. On path integration of grid cells: group representation and isotropic scaling. *Advances in Neural Information Processing Systems*, 34:28623–28635, 2021.

[65] Rylan Schaeffer, Mikail Khona, Tzuhsuan Ma, Cristobal Eyzaguirre, Sanmi Koyejo, and Ila R Fiete. Self-supervised learning of representations for space generates multi-modular grid cells. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[66] Matthias Stangl, Ingmar Kanitscheider, Martin Riemer, Ila Fiete, and Thomas Wolbers. Sources of path integration error in young and aging humans. *Nature communications*, 11(1):2626, 2020.

[67] Jisoo Kim, Chaewoo Kim, Hio-Been Han, Cheol Jun Cho, Wooseob Yeom, Sung Q Lee, and Jee Hyun Choi. A bird's-eye view of brain activity in socially interacting mice through mobile edge computing (mec). *Science Advances*, 6(49):eabb9841, 2020.

[68] Alexander T Lai, German Espinosa, Gabrielle E Wink, Christopher F Angeloni, Daniel A Dombeck, and Malcolm A MacIver. A robot-rodent interaction arena with adjustable spatial complexity for ethologically relevant behavioral studies. *Cell reports*, 43(2), 2024.

[69] Chunyu A. Duan, Ivana Orsolic, Qianbo Yin, Mehul Rastogi, Tom Hagley, Bruno Cuz, Andre Almeida, and Jeffrey Erlich. Mice dynamically adapt to opponents in multiplayer games. *Computational and Systems Neuroscience (Cosyne) Conference*, 2024.

[70] Kazushi Tsutsui, Ryoya Tanaka, Kazuya Takeda, and Keisuke Fujii. Collaborative hunting in artificial agents with deep reinforcement learning. *bioRxiv*, pages 2022–10, 2022.

[71] Kazushi Tsutsui, Kazuya Takeda, and Keisuke Fujii. Synergizing deep reinforcement learning and biological pursuit behavioral rule for robust and interpretable navigation. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

[72] Andrew S Alexander, Janet C Tung, G William Chapman, Allison M Conner, Laura E Shelley, Michael E Hasselmo, and Douglas A Nitz. Adaptive integration of self-motion and goals in posterior parietal cortex. *Cell reports*, 38(10), 2022.

[73] John B Issa and Kechen Zhang. Universal conditions for exact path integration in neural systems. *Proceedings of the National Academy of Sciences*, 109(17):6716–6720, 2012.

[74] Will Dorrell, Peter E Latham, Timothy EJ Behrens, and James CR Whittington. Actionable neural representations: Grid cells from minimal constraints. In *The Eleventh International Conference on Learning Representations*, 2022.

[75] Jon W Rueckemann, Marielena Sosa, Lisa M Giocomo, and Elizabeth A Buffalo. The grid code for ordered experience. *Nature Reviews Neuroscience*, 22(10):637–649, 2021.

[76] Seongmin A Park, Douglas S Miller, and Erie D Boorman. Inferences on a multidimensional social hierarchy use a grid-like code. *Nature neuroscience*, 24(9):1292–1301, 2021.

[77] Dmitriy Aronov, Rhino Nevers, and David W Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, 2017.

[78] Benjamin J Kraus, Mark P Brandon, Robert J Robinson, Michael A Connerney, Michael E Hasselmo, and Howard Eichenbaum. During running in place, grid cells integrate elapsed time and distance run. *Neuron*, 88(3):578–589, 2015.

[79] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[80] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[81] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser. py: A lean persistent homology library for python. *Journal of Open Source Software*, 3(29):925, 2018.

[82] Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

[83] V Esposito Vinzi, Wynne W Chin, Jörg Henseler, Huiwen Wang, et al. *Handbook of partial least squares*, volume 201. Springer, 2010.

[84] Omer Hazon, Victor H Minces, David P Tomàs, Surya Ganguli, Mark J Schnitzer, and Pablo E Jercog. Noise correlations in neural ensemble activity limit the accuracy of hippocampal spatial representations. *Nature communications*, 13(1):4276, 2022.

[85] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

[86] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):19, 2017.

[87] I Mezić. Data-driven analysis and forecasting of highway traffic dynamics. *Nature communications*, 11(1):2090, 2020.

[88] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

[89] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

[90] Stephen Wiggins. *Introduction to applied nonlinear dynamical systems and Chaos*. Springer, 1996.

[91] William T Redman, Juan M Bello-Rivas, Maria Fonoberova, Ryan Mohr, Ioannis G Kevrekidis, and Igor Mezić. On equivalent optimization of machine learning methods. *arXiv preprint arXiv:2302.09160*, 2023.

[92] William T Redman, Maria Fonoberova, Ryan Mohr, Ioannis G Kevrekidis, and Igor Mezić. Algorithmic (semi-) conjugacy via koopman operator theory. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6006–6011. IEEE, 2022.

[93] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[94] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

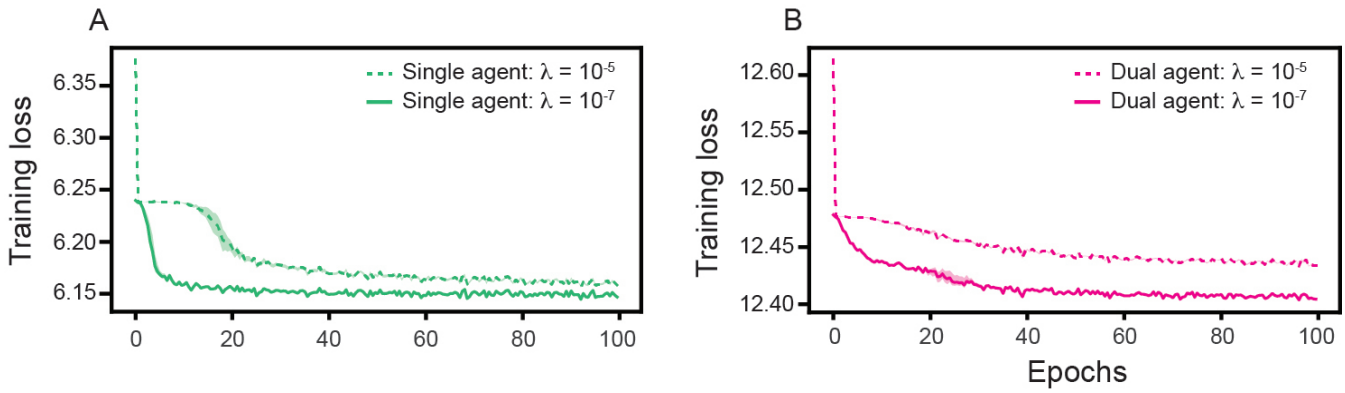

Figure S1: **Training loss for RNNs trained on single and dual agent path integration.** (A)–(B) Training loss, as a function of training epoch, for RNNs trained on single and dual agent path integration, respectively. Solid line is mean across 5 independently trained networks. Shaded area is maximum and minimum of all 5 networks.

A RNN architecture and training details

Code used for training and evaluating RNNs on path integration was pulled from `https://github.com/ganguli-lab/grid-patter` Vanilla RNNs, with parameters set to that reported in Table S1 (unless otherwise noted), were trained on single or dual agent path integration. Code will be made publicly available upon acceptance.

In Sec. 1, we listed the four modifications that were made in enabling RNNs to learn dual agent path integration. Here we provide more details on each of these modifications:

Additional inputs: Trajectories for both agents were sampled using the same generative function [27, 28]. Each time step is characterized by movement direction, $\theta(t)$, and speed, $v(t)$. We expand the architecture of the vanilla RNN to allow for four inputs (agent 1 movement direction, agent 1 speed, agent 2 movement direction, and agent 2 speed) by setting the input variable `input_size` of `torch.nn.RNN` to 4.

Summed place cell activations: The output layer of the RNN model comprises of "place cells" [36], whose "ground truth" activations are tied to the distance of an agent to specific locations ("place fields"). In-line with previous work, we use a difference-of-Gaussian model for the place field responses [28]. The ground truth activations are used as an error signal for training the RNN model in a supervised fashion. To generate the ground truth activations for the dual agent setting, we evaluate the distance of each agent to all place fields. This is used to generate activations for each place cell (for each agent, independently). We sum these responses, for each cell, to generate the "ground truth" activations.

k-means clustering for decoding: For single agent path integration, previous work has decoded the location of the agent by considering the $n_d = 3$ most active place cells and averaging the $x-$ and $y-$coordinates of their place field centers. To decode positions from place cell activations in the dual agent setting, we first consider the $2n_d = 6$ most active place cells. We then take the $x-$ and $y-$coordinates of the corresponding place field centers and apply $k-$means clustering, using the function `kmeans` from the package `scipy.cluster.vq`, setting $k = 2$. The centroids of the clusters are used as the predicted positions for the two agents. We compute the decoding error as the Euclidean distance between the true and predicted positions of the agents (the positions considered as 4−dimensional vectors). Because the order of the centroids is not guaranteed to match the arbitrary ordering of the agents' positions, we find the ordering that minimizes the distance. As the $k-$means clusering incurs additional computational costs, we only perform the decoding twice each epoch (at the beginning and half-way through).

Reduced weight decay regularization: Previous work has introduced weight regularization to the weights in the recurrent layer, $\mathbf{W}^{\text{rec}}$, to encourage sparsity and aid generalization (Eq. 3 [32]. Given that we expect dual agent path integration to be a more challenging task, we hypothesize that the weight decay value used for single agent path integration might lead too much sparsity to perform the dual agent path integration task well. We therefore reduced the value used for weight decay, taken as an input in the Adam optimizer `torch.optim.Adam`, from $\lambda = 10^{-5}$ to $\lambda = 10^{-7}$. We note that previous work reported using $\lambda = 10^{-4}$ [32], however the publicly code had the preset value of `1e-4`, which is equivalent to $\lambda = 10^{-5}$.

The training loss associated with single and dual agent path integration is presented in Fig. S1. We note that there is no equivalent to "test" loss, as the trajectory used to train each iteration was sampled randomly from a distribution [28]. No trajectories were separately maintained in a "test set".

Table S1: **Parameters used to train RNNs on single and dual agent path integration.** Unless otherwise noted (in the text), these are the values the parameters used to train single and dual agent RNNs take. Parameters that take values that are different from the defaults of previous models [28, 32] are bolded. Any values not listed are the same as they were in previous work [28, 32] .

Parameter	Value
Epochs	100
Batch size	200
Batches per epoch	1000
Path length (T)	20
Arena length (L)	2.2 m
Learning rate	10^{-5}
Place cells (n_P)	512
Grid cells (n_G)	4096
σ_1	0.12
σ_2	0.24
Activation	ReLU
Weight decay	10^{-7}
Optimizer	Adam

All training experiments and analysis were performed using an AWS GovCloud g4dn.xlarge instance with 1 GPU and 4 CPUs.

B Generalization experiment details

Let $\mathbf{W}^{\text{in}}$ denote the weights projecting from the inputs to the units in the recurrent layer. Given the difference in input dimension between single and dual agent RNNs , modifications to $\mathbf{W}^{\text{in}}$ is necessary to enable an RNN trained on single agent path integration to be tested on dual agent path integration (and vice versa). Going from single to dual agent path integration (Fig. 3A), we concatenate $\mathbf{W}^{\text{in}}$ with itself, forming a new $\mathbf{W}^{\text{in}} = [\mathbf{W}^{\text{in}}, \mathbf{W}^{\text{in}}] \in \mathbb{R}^{n_G \times 4}$. Going from dual to single agent path integration (Fig. 3B), we remove the weights corresponding to the third and fourth inputs. That is, if $\mathbf{W}_{\text{in}} = [\mathbf{W}_1^{\text{in}}, \mathbf{W}_2^{\text{in}}, \mathbf{W}_3^{\text{in}}, \mathbf{W}_4^{\text{in}}]$, where $\mathbf{W}_i^{\text{in}}$ corresponds to the weights projecting from input i to all units in the recurrent layer, then we generate a new set of weights $\mathbf{W}^{\text{in}} = [\mathbf{W}_1^{\text{in}}, \mathbf{W}_2^{\text{in}}] \in \mathbb{R}^{n_G \times 2}$. Because all other training and architecture hyper-parameters are the same between RNN models trained on single and dual agent path integration, no other modifications were considered.

To freeze the weights in the recurrent layer, we set those parameters to have `requires_grad` = `False`.

C Single unit representation details

C.1 Constructing rate maps

The construction of rate maps, which visualize activity of individual neurons , as a function of physical location, is a standard tool used to understand the neural representations underlying spatial navigation. In the context of RNNs trained on single agent path integration, this tool has been used to discover grid, border, and band cells [27–29, 31, 32]. To do this, n_t trajectories, corresponding to the paths of a single agent, are generated and used as inputs to RNN models. The activations of each unit in the recurrent layer and the position of the agent at each time point is saved. The average activation value, for a given position, is then computed and plotted. Position is resolved at a resolution of m bins in the $x-$ and $y-$direction, leading to m^2 bins in the rate map.

For the dual agent setting, we similarly sample n_t trajectories, use them as inputs to dual agent RNNs, and save the activations. The primary difference for constructing rate maps for dual agent RNNs is that only the position of one of the agents is used to construct the maps (and not both). By default, we take the first agent's position. However, since there is no difference between the agents, this choice should not have any major effect.

To better understand the extent to which this choice for visualizing the rate maps affects our interpretations, we also compute rate maps where only a single agent's trajectory is sampled. That is, we trained an RNN on dual agent

path integration, but then sample only a single agent's trajectory when constructing the rate maps, setting the inputs corresponding to the movement direction and speed of the second agent to 0 (we refer to this approach as "Dual agent RNN single agent rate map"). We find that, in general, this leads to similar rate maps (compare Fig. 4B with Fig. S3B).

For all rate maps, we set $n_t = 1000$ and $m = 20$. We note that it is typical for rate maps to be smoothed by applying a filter after their construction. However, we chose not to smooth the rate maps (for either single or dual agent path integration) to ensure that the interpretations of the representations we studied were being driven by filtering.

C.2 Functional class metrics

Visual inspection of rate maps have led to the identification of subsets of neurons and hidden units that have distinct properties (e.g., "grid-like"). To quantify these properties and enable unsupervised identification, metrics have been developed. These are discussed in detail below:

Grid score: Ideal grid cells are characterized by having firing fields distributed along the vertices of a triangular lattice [5, 6]. For neural data, the grid signature has been found to be more strongly apparent when considering not the rate map, but the spatial autocorrelogram (SAC) of the rate map. Let $\mathbf{X}_i$ denote the rate map corresponding to unit i in the recurrent layer, and $\mathbf{S}_i$ denote its SAC. If $\mathbf{S}_i$ has strong triangular lattice structure, rotating it by $60°$ (which we denote by $\mathbf{S}_i^{60}$) should lead to a SAC that is highly correlated with the $\mathbf{S}_i$. This is similarly true for rotations of $120°$ and $240°$. High correlations could also arise when there is a finer symmetry. To identify when this is the case, the correlation between $\mathbf{S}_i^{30}$, $\mathbf{S}_i^{90}$, and $\mathbf{S}_i^{150}$ can be taken. The grid score, gs_i, is computed as

$$\text{gs}_i = \frac{1}{2}\left[\text{corr}(\mathbf{S}_i, \mathbf{S}_i^{60}) + \text{corr}(\mathbf{S}_i, \mathbf{S}_i^{120})\right] - \frac{1}{3}\left[\text{corr}(\mathbf{S}_i, \mathbf{S}_i^{30}) + \text{corr}(\mathbf{S}_i, \mathbf{S}_i^{90}) + \text{corr}(\mathbf{S}_i, \mathbf{S}_i^{150})\right], \tag{5}$$

where $\text{corr}(\mathbf{S}_i, \mathbf{S}_i^{\theta}) = (\sum_j \tilde{S}_{i,j}\tilde{S}_{i,j}^{\theta})/\sum_j(\tilde{S}_{i,j})^2$, with $\tilde{S}_{i,j}$ being the jth bin of the SAC of hidden unit i, with the center peak masked out.

Border score: Ideal border cells are characterized by having their activations selectively aligned along a border. To compute the border score [7], the rate map corresponding to unit i in the recurrent layer, $\mathbf{X}_i$, is first thresholded, with any spatial bin having activation $< 0.3 \cdot \max(\mathbf{X}_i)$ being set to 0. Connected components of this thresholded rate map are found using `scipy.ndimage` and those with area < 200 cm^2 are additionally set to 0 [7]. Let $\tilde{\mathbf{X}}_i = \mathbf{X}_i \odot \mathbf{M}_i$ correspond to this masked out rate map, where $\mathbf{M}_i \in \{0,1\}^{m\times m}$ is the mask and $\odot$ is the element-wise multiplication operation. The maximum coverage along any of the four environment walls, $c_{\mathbf{M}_i}$, is computed by finding the number of non-masked out spatial bins along each wall (i.e., $\mathbf{M}_i(x,y) = 1$) and dividing it by m. Finally, the mean firing distance, $d_{\mathbf{M}_i}$, is computed for each remaining connected component is by taking the mean of the distance of each spatial bin in the connected component to the nearest wall, weighted by the normalized activation at that spatial bin. $d_{\mathbf{M}_i}$ is then defined by taking the average value across all connected components. The border score, bos_i, is the computed as

$$\text{bos}_i = \frac{c_{\mathbf{M}_i} - d_{\mathbf{M}_i}}{c_{\mathbf{M}_i} + d_{\mathbf{M}_i}}. \tag{6}$$

Band score: To the best of our knowledge, no metric that explicitly quantifies the "banded-ness" of neural activity or hidden unit activations exists. We therefore created one to enable the exploration of how adding an additional agent affects the development of units with band structure in their rate maps. As with grid and border scores, let $\mathbf{X}_i$ denote the rate map corresponding to unit i in the recurrent layer. Let $\mathbf{X}(k_x, k_y)$ be a two-dimensional sinusoid with frequency k_x, in the $x-$direction, and k_y, in the $y-$direction. Finally, let $K = \{0, 0.1, ..., 2.0\}$ be possible values for k_x and k_y. The band score, bas_i, is computed as

$$\text{bas}_i = \max_{k_x,k_y\in K} \texttt{corrcoef}\left[\mathbf{X}_i, \mathbf{X}_i(k_x, k_y)\right], \tag{7}$$

where `corrcoef` is `numpy`'s Pearson product-moment correlation coefficient.

C.3 Functional class ablation experiments

To probe how units in the recurrent layer are used in the computations underlying the ability to path integrate one or two agents, we perform functional class targeted ablation experiments (Fig. 4C). To do this, we rank the units based on their functional properties (grid, border, or band score). We then ablate units with the highest $p\%$ of the scores by setting all incoming and outgoing weights to 0. That is, if unit i is determined to have a sufficiently high score and is to be ablated, we set $\mathbf{W}^{\text{rec}}_{i,j} = \mathbf{W}^{\text{rec}}_{j,i} = 0$, for all $j = 1, ..., n_G$. Here, $\mathbf{W}^{\text{rec}}$ are the weights in the recurrent layer.

Figure S2: **Consistent individual unit level representations across independently trained dual agent RNNs**. Same as Fig. 4B (top), for all 5 seeds (each corresponding to an independently trained RNN).

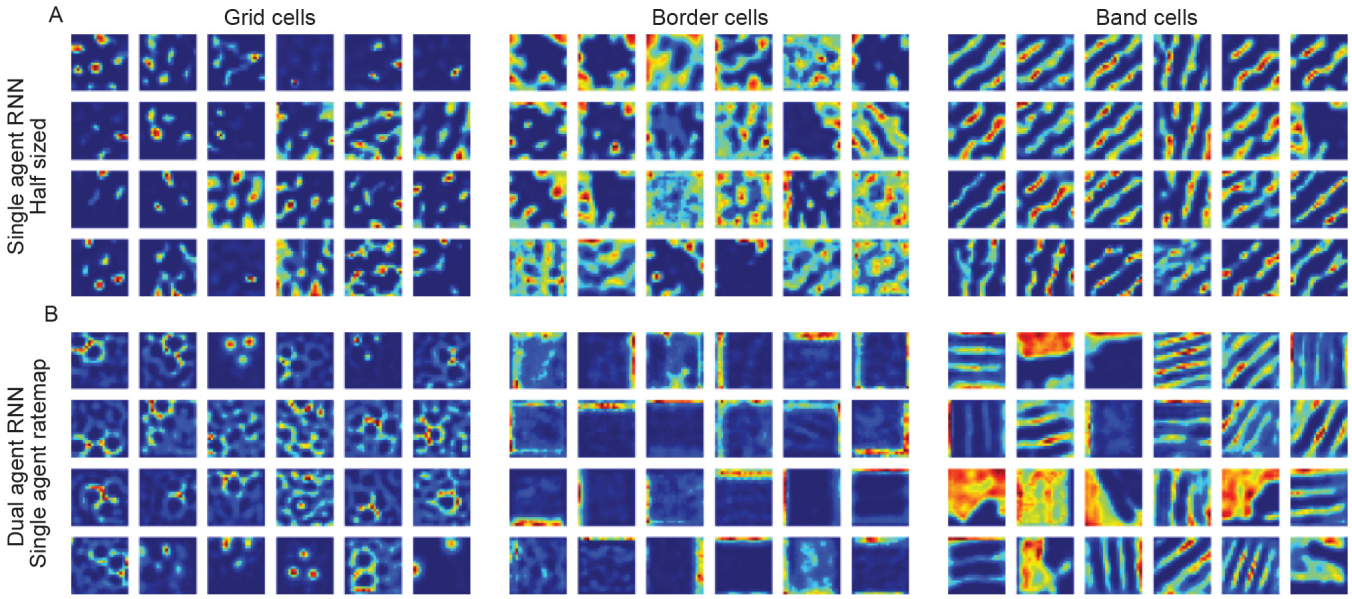

Figure S3: **Control experiment rate maps.** (A) Visualization of rate maps from an RNN with half the number of recurrent and output units trained on single agent path integration, with the highest grid, border, and band scores. (B) Same as Fig. 4B (top), but for rate maps computed from the trajectories of a single agent (Appendix C.1). Note that the RNN was, as in the case of Fig. 4B (top), trained on dual agent path integration.

D Relative position representation details

D.1 Relative position rate maps

To construct the traditional rate maps used to visualize grid, border, and band cells (Fig. 4, Appendix C.1), the $x-$ and $y-$coordinates of the environment were split into bins and the average activity, per spatial bin, was computed using one of the agent's x- and y-position.

To examine whether there was any tuning in the relative position space, we construct rate maps using not x and y but $\Delta_x = x_1 - x_2$ and $\Delta_y = y_1 - y_2$, where (x_1, y_1) and (x_2, y_2) are the locations of the two agents (the ordering is arbitrary, but fixed). This provides a transformation from the allocentric reference frame to the relative space reference frame (Fig. 5A). Note that this transformation is not one-to-one, as multiple configurations of agents position in allocentric space have the same relative space coordinates. Note that relative space reference frame has twice the range, in each dimension, as allocentric space. This is because, while $x, y \in [-L/2, L/2]$, $\Delta_x, \Delta_y \in [-L/2, L/2]$.

Interpreting the rate maps constructed in the relative space can be slightly un-intuitive. To aid in this, we provide illustrations of three relative space rate maps, with specific locations of the rate map visualized in the allocentric reference frame (Fig. S5B). As noted above, because the mapping is not one-to-one, these are not the only possible realizations of the allocentric space. From these examples, we see how a unit in the recurrent layer that is identified from its allocentric rate map as a grid cell has the same average activity across a wide range of relative positions of the two agents (Fig. S5A, left). In contrast, the example border cell responds only when both agents are along the borders (Fig. S5A, middle), which can be seen by the activations being restricted to when Δ_x is at its minimal and maximal values. For an example band cell, again there is specificity in the response related to the value of Δ_x, with activity occurring only at certain intervals (Fig. S5A, right). In between those intervals, there is little activation.

D.2 Spatial information

To quantify the amount of information neural responses encode about specific variables, the metric spatial information was developed [41]. In particular, the spatial information, SI_i, encoded by the rate map corresponding to unit i in the recurrent layer, $\mathbf{X}_i$, is computed by

$$\mathrm{SI}_i = \frac{1}{\bar{\mathbf{X}}_i} \sum_{x,y} p(x,y) \mathbf{X}_i(x,y) \log_2 \left(\frac{\mathbf{X}_i(x,y)}{\bar{\mathbf{X}}_i} \right), \tag{8}$$

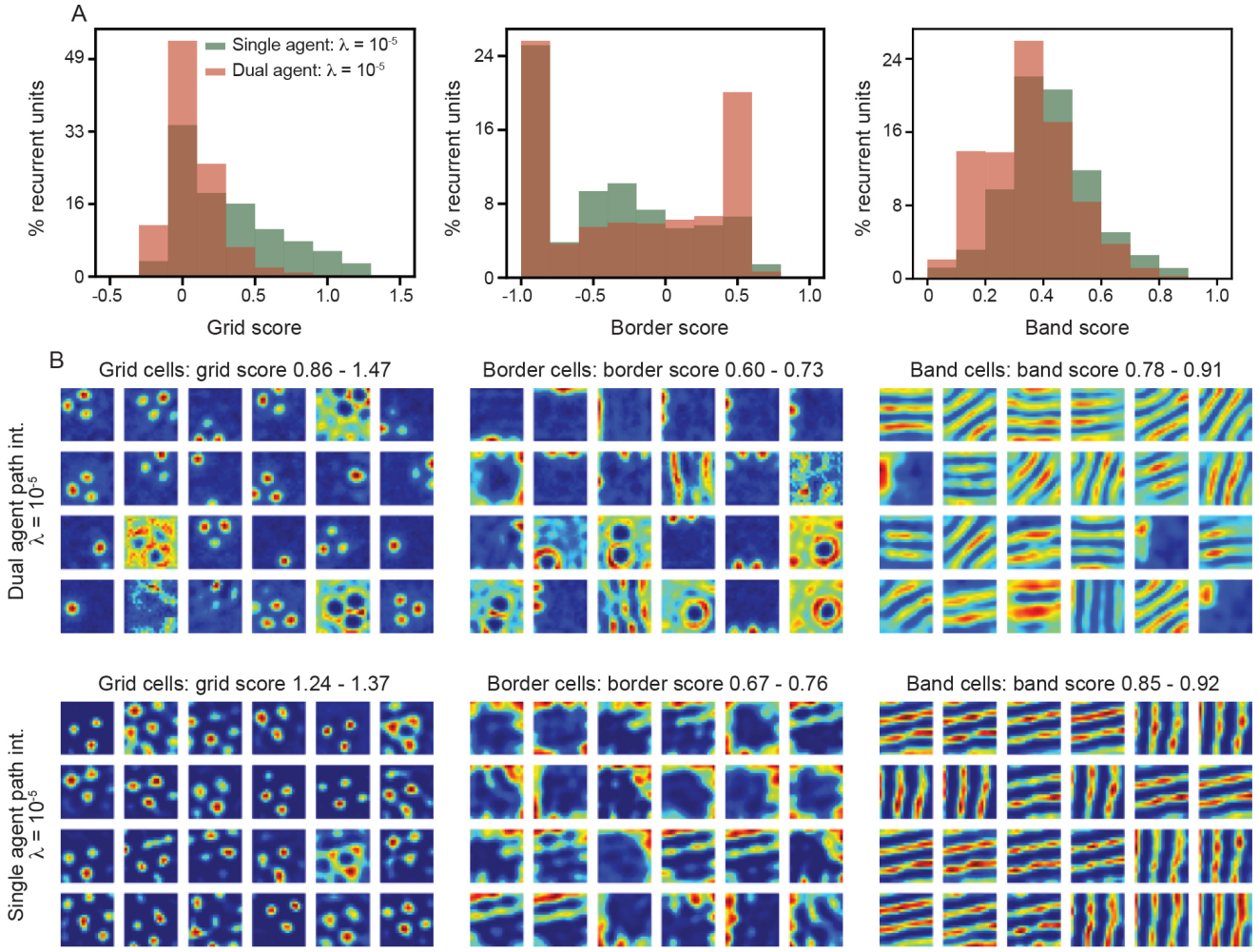

Figure S4: **Difference in functional classes that emerge in single and dual agent RNNs persists when training using greater weight decay.** Same as Fig. 4 (A)–(B), but for RNN model trained with weight decay $\lambda = 10^{-5}$. Results computed from 3 independently trained RNNs.

Table S2: **Functional class ablation statistics.** Mann-Whitney test p-values for RNN model decoding error (Fig. 4C). Statistics, computed using `scipy.stats.mannwhitneyu`, correspond to testing the null hypothesis that the decoding error of the RNN model with functional classes ablated is not greater than the RNN model with random units ablated. Bolded p-values denote those less than 0.05.

% units removed	Functional class	Single agent p-value	Dual agent p-value
5	Grid cells	0.11	0.99
10	Grid cells	$\mathbf{6.4 \cdot 10^{-4}}$	0.87
15	Grid cells	$\mathbf{2.0 \cdot 10^{-3}}$	0.98
20	Grid cells	$\mathbf{3.4 \cdot 10^{-6}}$	0.98
25	Grid cells	$\mathbf{1.3 \cdot 10^{-2}}$	1.0
5	Border cells	$\mathbf{2.9 \cdot 10^{-7}}$	0.24
10	Border cells	$\mathbf{2.9 \cdot 10^{-7}}$	0.98
15	Border cells	$\mathbf{2.9 \cdot 10^{-7}}$	$8.0 \cdot 10^{-2}$
20	Border cells	$\mathbf{2.9 \cdot 10^{-7}}$	$\mathbf{1.4 \cdot 10^{-2}}$
25	Border cells	$\mathbf{2.9 \cdot 10^{-7}}$	0.90
5	Band cells	$\mathbf{2.9 \cdot 10^{-7}}$	$\mathbf{4.6 \cdot 10^{-2}}$
10	Band cells	$\mathbf{2.9 \cdot 10^{-7}}$	$\mathbf{2.9 \cdot 10^{-2}}$
15	Band cells	$\mathbf{2.9 \cdot 10^{-7}}$	$9.0 \cdot 10^{-2}$
20	Band cells	$\mathbf{2.9 \cdot 10^{-7}}$	0.25
25	Band cells	$\mathbf{2.9 \cdot 10^{-7}}$	0.82

Table S3: **Relative space ablation statistics.** Mann-Whitney test p-values for dual agent RNN model decoding error (Fig. 5E) for units with the highest relative space spatial information ablated (Appendix D.3). Details same as Table S2.

% units removed	p-value
5	$\mathbf{1.3 \cdot 10^{-3}}$
10	$\mathbf{6.4 \cdot 10^{-4}}$
15	$\mathbf{6.8 \cdot 10^{-3}}$
20	$\mathbf{4.6 \cdot 10^{-4}}$
25	$\mathbf{2.2 \cdot 10^{-3}}$

where $\bar{\mathbf{X}}_i$ is the mean value of rate map i over all space, x and y correspond to the m spatial bins in each direction that are used to construct the rate map, and $p(x, y)$ is the probability that the agent was in the spatial bin (x, y).

We compute the spatial information on the relative rate maps (Figs. 5, Appendix D.1). While a useful tool, the spatial information can be skewed by units that (spuriously) respond highly to only a single spatial bin (see Fig. 5E, top left rate map). Additionally, the multiplicative dependence on $p(x, y)$ can lead to high spatial information in specific bins, when the probability is skewed. Indeed, we find that $p(x, y)$, in the relative space, is heavily distributed towards the center of the rate maps, corresponding to the two agents being near each other. Thus, while a useful tool for enabling us to identify a new functional class of units, *these results do not rule out other interesting responses in relative space (or other reference frames).*

D.3 Relative space ablation experiments

To probe the importance of the units in the hidden recurrent layer that encode relative space, we perform the same ablation studies as described in Appendix C.3, but ranking units based on the spatial information present in their relative space rate maps, as opposed to their grid, border, or band scores.

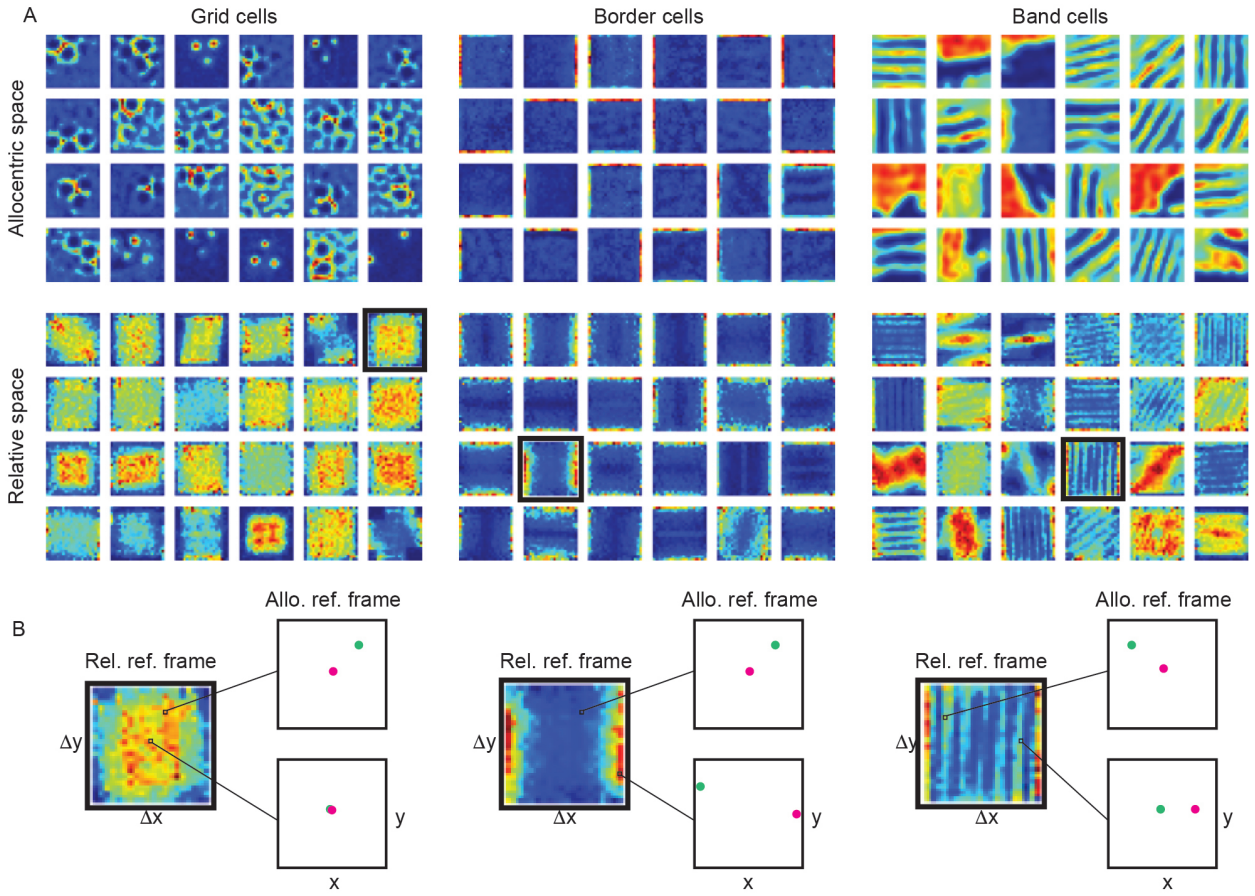

Figure S5: **Border and band cells additionally encode relative space information, but not grid cells.** (A) Comparison of rate maps in allocentric space (top) and relative space (bottom), for units, from an individual dual agent RNN, that had high grid, border, and band scores. Top row is the same as Fig. 4B (bottom). (B) Example relative space rate maps (with the functional class their allocentric ratemaps had high scores for), with schematic illustrations of possible configurations for different locations in relative space.

E Population representation details

E.1 Topological data analysis

TDA [42] has become a widely used tool in neuroscience [24, 43–47]. In particular, it has been leveraged to provide evidence for the population activity of grid cells (from the same module) being constrained to a two-dimensional toroidal manifold [24]. A core tool in TDA is persistence (co)homology, which is used to identify topological structure present in data. Persistence (co)homology, applied to neural activity, can loosely be understood by the following overview. First, the activity of all N neurons, at each time point, is viewed as a point in the space of all possible population activations. This leads to a point cloud in N-dimensional space. Balls, of dimension N, are then centered on each point in the cloud. Gradually, the radii of these balls is increased. When two balls intersect, they are considered a connected component. The topological properties of all connected components is then computed. In particular, the existence of loops (H_1) and two-dimensional cavities (H_2) are extracted and the persistence of these features is measured. This can be visualized in a format called a persistence diagram.

Because the toroidal topology is expected for each grid module independently [24], we follow the procedure of Gardner et al. (2022) to identify the number of possible modules present in our single and dual agent RNNs. In particular, we utilized UMAP [79] to find a two-dimensional representation of the units of single and dual agent RNNs (Figs. S6A, S7A) and then applied DBSCAN to perform clustering [80]. For both single and dual agent RNNs, we found only a single cluster, consistent with prior observations that the RNN models considered develop weak modularization [65]. We colored the points in the space defined by UMAP by the associated recurrent unit's grid, border, and band score (Figs. S6A, S7A). Qualitatively, we find that units with the lowest border scores tend to be restricted to specific parts of UMAP space. Additionally, there appears to be a gradient across UMAP space defined

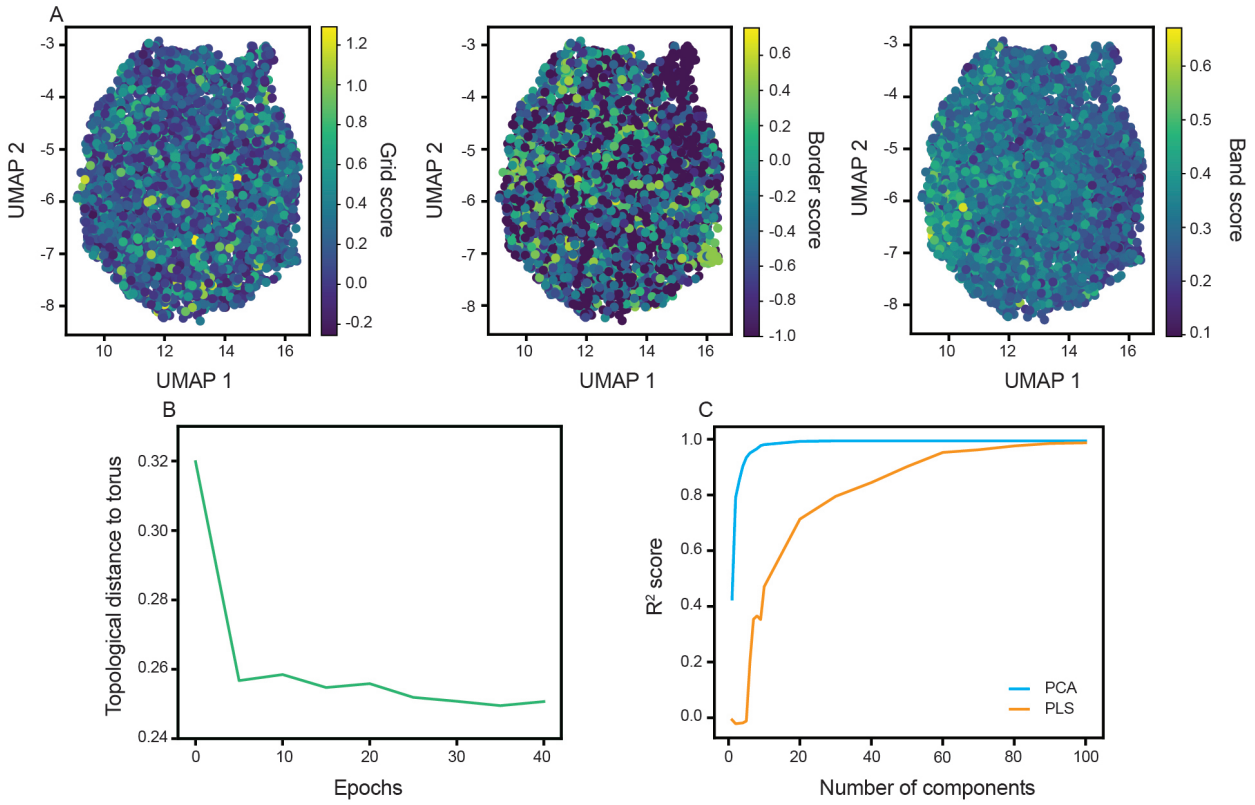

Figure S6: **Computing the topology of single agent RNN activations.** (A) All recurrent units of single agent RNNs projected onto a two-dimensional subspace identified using UMAP [79]. Points are colored by their corresponding units grid, border, and band scores. (B) Topological distance between the persistence diagrams computed from single agent RNN population activations and persistence diagrams computed from an ideal torus. This topological distance is computed across several epochs. (C) Multi-output regression (R^2 score) with agents' positions from the dimensionally-reduced single agent RNN population activations, as a function of number of components used in the dimensionality reduction. Principal component analysis (PCA) and partial least squares (PLS) are considered.

by the band scores.

Having found only a single module, we then computed the Vietoris-Rips persistence diagram corresponding to the activations of all recurrent units for both single and dual agent RNNs. To do this computation, we used the `Ripser` python package [81, 82]. However, persistence diagrams can be subject to noise and are more challenging to compute for high-dimensional spaces. Therefore, to ensure that the differences in topology observed between single and dual agent RNNs was due to different properties underlying the population activations, and not noise, we projected the population activations onto a lower-dimensional subspace and re-computed the persistence diagrams. There are several possible choices for reducing the dimensionality. A natural option is principal component analysis (PCA). However, we find that, for the single agent RNN population activations, the subspace identified by PCA does not enable good multi-output regression with the agent's location (Fig. S6C – orange line). Indeed, nearly 100 principal components are needed to achieve a high R^2. We therefore sought another dimensionality reduction technique that can find lower-dimensional subspaces that are better aligned with features relevant to path integration. Using partial least squares (PLS) [83], which has been successfully applied in neuroscience applications [84], we find that both single and dual agent RNN population activation, captured in a 10-dimensional subspace, can achieved high multi-output regression with the agents' positions (Figs. S6C, S7C – blue lines). Therefore, we use PLS to reduce the dimensionality of the population activations, and then computed the persistence diagram on this reduced subspace (Fig. 6). The PLS was computed using `PLSRegression` from `sklearn.cross_decomposition`.

To understand how the topology of the population level activations of the single and dual agent RNNs evolves with training, we computed the persistence diagram associated with an "ideal" two-dimensional torus and measured the topological distance [48] between the observed persistence diagram and the ideal persistence diagram (Figs. S6B, S7B). We find that the topology of the single agent RNN experiences a large drop in distance to the ideal torus in the first 5 epochs (Figs. S6B), a time-frame similar as the single agent RNNs drop in decoding error (Fig. 2A).

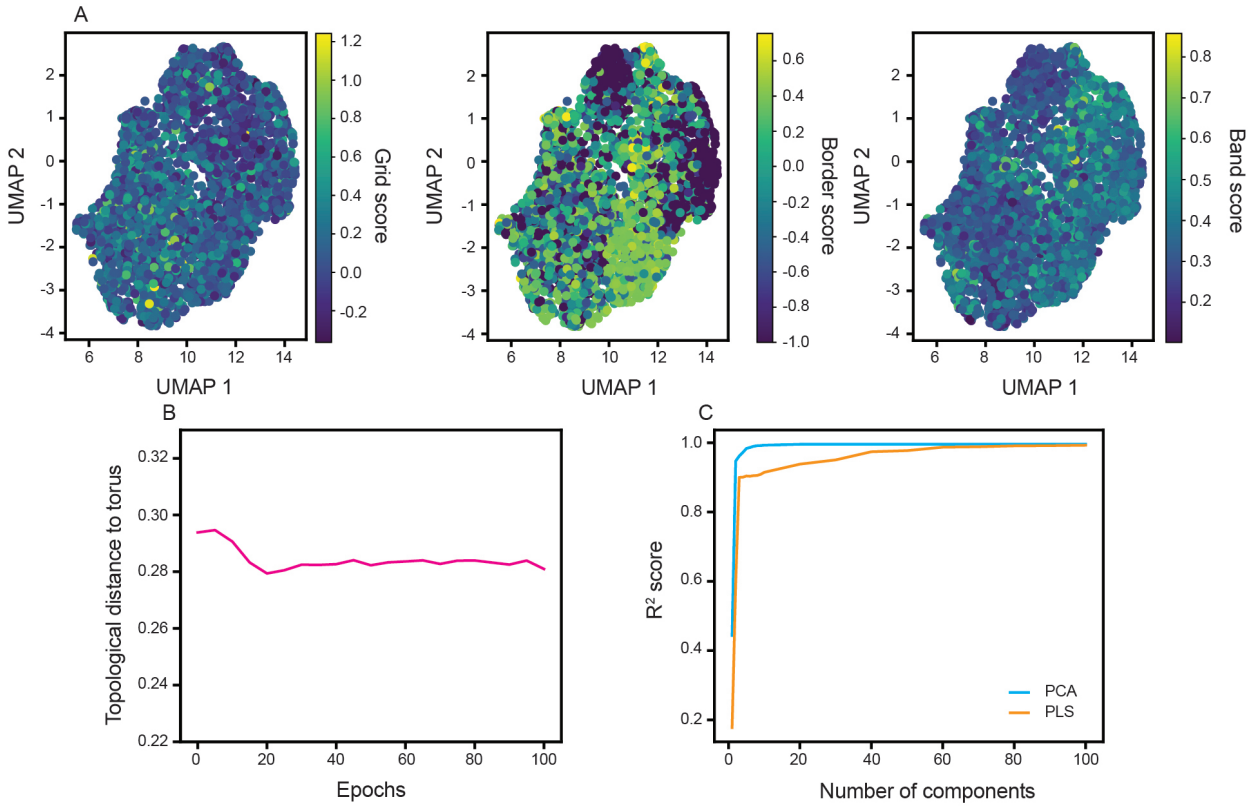

Figure S7: **Computing the topology of dual agent RNN activations.** Same as Fig. S6, but for dual agent RNN.

In addition, the topological distance continues to decrease with increasing training (Figs. S6B). In contrast, the topological distance between the ideal torus and the population activation of dual agent RNNs experiences little change with training time (Figs. S7B).

E.2 Dynamic similarity analysis

The evolution, in time, of the activations of units in RNNs can be viewed as a dynamical system. The features of this dynamical system (e.g., fixed points, limit cycles, quasi-periodic attractors, chaos) are dependent on the inputs and structure of the RNN. Thus, characterizing corresponding dynamical properties offers the ability to gain insight into the organization of the RNN. Unsurprisingly, use of geometric and topological methods to interrogate RNN population activation can fail to properly identify similar and dis-similar dynamics, as they are sensitive to the underlying manifold the activation evolves along [49]. Recent work [50] has used data-driven tools [85–87] from Koopman operator theory [88, 89] to extract properties of RNN dynamics that can be compared using a precise notion of equivalence, called "topological conjugacy" [90]. Similar approaches have been used to study the dynamics associated with training deep neural networks [91], as well as iterative optimization algorithms, more generally [92]. Therefore, we complimented the geometric and topological analysis performed on the structure of the population level representations (Fig. E.1) with dynamical similarity analysis (DSA) [50].

To perform DSA, $K \in \mathbb{N}$ trajectories in physical space, corresponding to K trajectories in population activity space, are sampled. Each trajectory has temporal length $T \in \mathbb{N}$. All the activations are collected into a tensor, $\mathbf{X} \in \mathbb{R}^{K \times T \times n_G}$, where n_G is the number of units in the recurrent layer. We then perform principal component analysis (PCA) on $\mathbf{X}$, reducing the number of dimensions so that we get a new tensor $\mathbf{X} \in \mathbb{R}^{K \times T \times n_M}$, where $n_M < n_G$. Next, $D \in \mathbb{N}$ time-delays are then applied, where $D < T$, and the resulting tensor $\mathbf{H} \in \mathbb{R}^{K \times (T-D) \times n_M D}$ is then flattened along its first dimension. Dynamic mode decomposition [93] is then performed to get a compact representation of the dynamics in terms of the Koopman operator [94], which is restricted to have rank $n_R < n_M D$. The Koopman-based representation between the dynamics of the single and dual agent RNN are then compared, via the "Procrustes over vector field" metric [50].

As can be seen in the brief overview of DSA, there are several hyper-parameters whose value must be chosen.

These include K, D, n_M, and n_R. To be consistent with the geometric and topological analysis, we set $K = 1000$. Because the length of the paths ($T = 20$) is small, we cannot use too many time-delays. Therefore, we set $D = 4$. Because a small number of dimensions were necessary to capture most of the variance of the activations in both single and dual agent RNNs, we considered $n_M = 10$ and $n_R = 30$ (Fig. 6). To determine whether this choice affected the conclusion that single and dual agent RNNs have different dynamical properties, we performed the analysis again, with $n_M = 50$ and $n_R = 100$. Fig. S8 shows a similar conclusion to Fig. 6, with the Procrustes analysis over vector field metric being nearly $2\times$ larger for comparison between single and dual agent RNNs than between single and single agent RNNs or dual and dual agent RNNs. However, because more modes are kept ($n_M = 50$), more noise is present, and the difference is weaker than when fewer modes are kept.

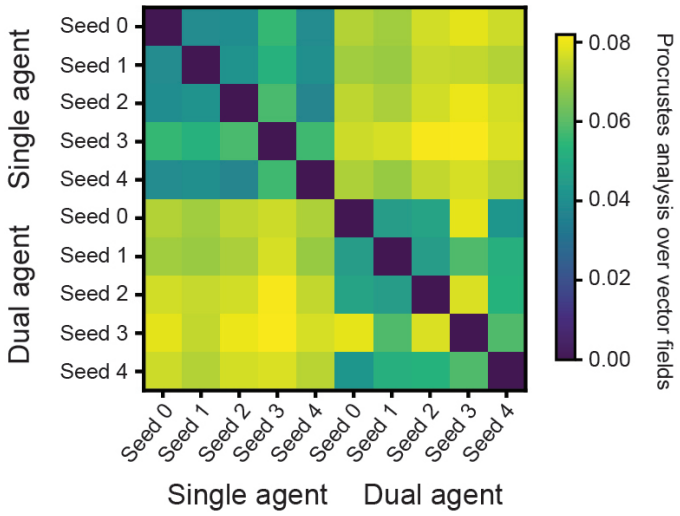

Figure S8: **Difference of activation dynamics between single and dual agent RNNs is consistent with different choice of DSA hyper-parameters.** Same as Fig. 4, but with a different choice of DSA hyper-parameters.