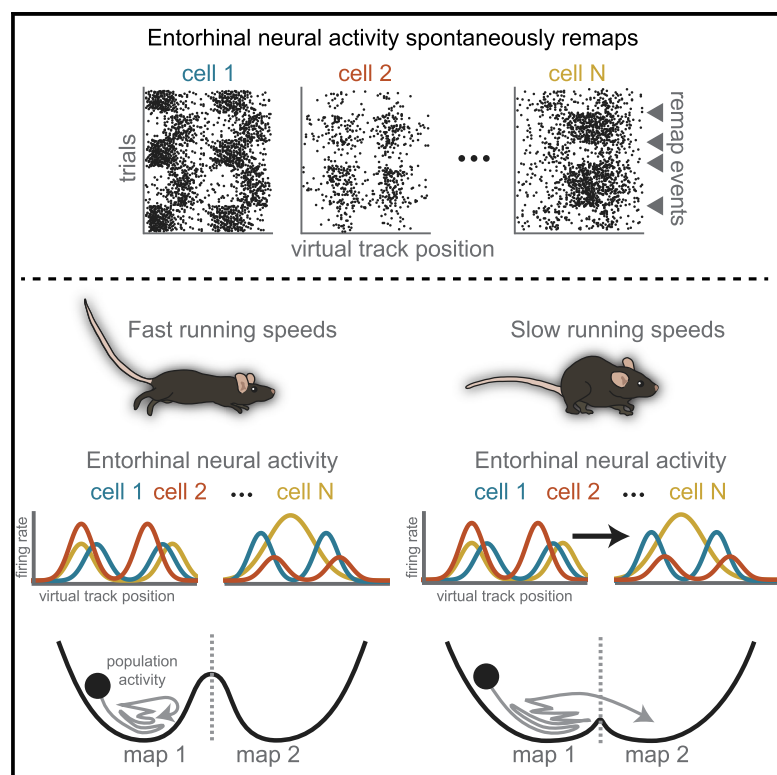# Dynamic and reversible remapping of network representations in an unchanging environment

Graphical abstract

Authors

Isabel I.C. Low, Alex H. Williams,
Malcolm G. Campbell,
Scott W. Linderman, Lisa M. Giocomo

Correspondence

ilow@stanford.edu (I.I.C.L.),
giocomo@stanford.edu (L.M.G.)

In brief

Low et al. record from hundreds of neurons in the entorhinal cortex and reveal transitions between distinct neural maps of space in an unchanging task and environment. These transitions are reversible, synchronized across neurons, and correlated with running speed. Alignment of the two activity manifolds enables reliable position decoding across maps.

Highlights

- Entorhinal neurons rapidly and reversibly remap between distinct spatial maps

- Remapping is synchronized across almost all co-recorded neurons

- Alignment of the maps' activity manifolds enables simple decoding mechanisms

- Remapping and related neural variability correlate with slower running speeds

Neuron

CellPress

Article

Dynamic and reversible remapping of network representations in an unchanging environment

Isabel I.C. Low,[1,2,*] Alex H. Williams,[2,3] Malcolm G. Campbell,[1,4] Scott W. Linderman,[2,3] and Lisa M. Giocomo[1,2,5,*]
[1]Department of Neurobiology, Stanford University School of Medicine, Stanford, CA, USA
[2]Wu Tsai Neurosciences Institute, Stanford University, Stanford, CA, USA
[3]Department of Statistics, Stanford University, Stanford, CA, USA
[4]Present Address: Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA
[5]Lead contact
*Correspondence: ilow@stanford.edu (I.I.C.L.), giocomo@stanford.edu (L.M.G.)
https://doi.org/10.1016/j.neuron.2021.07.005

SUMMARY

Neurons in the medial entorhinal cortex alter their firing properties in response to environmental changes. This flexibility in neural coding is hypothesized to support navigation and memory by dividing sensory experience into unique episodes. However, it is unknown how the entorhinal circuit as a whole transitions between different representations when sensory information is not delineated into discrete contexts. Here we describe rapid and reversible transitions between multiple spatial maps of an unchanging task and environment. These remapping events were synchronized across hundreds of neurons, differentially affected navigational cell types, and correlated with changes in running speed. Despite widespread changes in spatial coding, remapping comprised a translation along a single dimension in population-level activity space, enabling simple decoding strategies. These findings provoke reconsideration of how the medial entorhinal cortex dynamically represents space and suggest a remarkable capacity of cortical circuits to rapidly and substantially reorganize their neural representations.

INTRODUCTION

As an animal engages in complex behaviors, it must dynamically integrate external sensory features with internal behavioral state changes. Arousal (Hulse et al., 2017; Salay et al., 2018; Vinck et al., 2015), satiety (Jennings et al., 2019), attention (Kentros et al., 2004), and running speed (Bennett et al., 2013; Hardcastle et al., 2017; Hulse et al., 2017; Niell and Stryker, 2010; Vinck et al., 2015) can affect cortical processing and influence how an animal interacts with its surroundings (Calhoun et al., 2019; Salay et al., 2018). As an animal navigates the world, encountering a continuous stream of sensory features while experiencing behavioral state changes, navigational centers in the brain may integrate these external and internal factors to encode unique episodes or contexts. The extent to which these navigational centers incorporate changes in an animal's internal behavioral state, as opposed to reflecting the external sensory features in the world, remains incompletely understood.

One key navigational center is the medial entorhinal cortex (MEC), which is hypothesized to support navigation and memory through neurons that encode the animal's position and orientation relative to sensory features such as environmental boundaries and objects (Diehl et al., 2017; Gil et al., 2018; Hafting et al., 2005; Høydal et al., 2019; Sargolini et al., 2006; Solstad et al., 2008). When environmental features change, MEC position

and orientation cells can "remap" through changes in firing rate and rotations or shifts in firing field locations (Barry et al., 2007; Diehl et al., 2017; Fyhn et al., 2007; Keene et al., 2016; Krupic et al., 2015; Marozzi et al., 2015; Munn et al., 2020; Solstad et al., 2008). This flexibility in MEC neural coding is hypothesized to be an integral part of navigational and memory processes. It may also interface with a similar phenomenon in the hippocampus (Fyhn et al., 2007), a neural substrate involved in memory formation, where neurons also remap in response to changes in sensory features or task context (Bostock et al., 1991; Frank et al., 2000; Jezek et al., 2011; Markus et al., 1995; Moita et al., 2004; Muller and Kubie, 1987; Wood et al., 2000). Recent work in the hippocampus has revealed remapping in response to other latent factors, which may include changes in an animal's behavioral state (Keinath et al., 2020; Sheintuch et al., 2020; Ziv et al., 2013), suggesting that the MEC may be responsive to these factors as well. Consistent with this possibility, changes in task demands can evoke MEC remapping (Boccara et al., 2019; Butler et al., 2019). However, studies have yet to identify the specific effect of latent factors such as behavioral state on MEC remapping. Further, it is not yet known what MEC remapping dynamics might look like when environmental features or task demands are not delineated into distinct episodes.

Until recently, technological constraints limited most studies to small numbers of simultaneously recorded MEC neurons. As
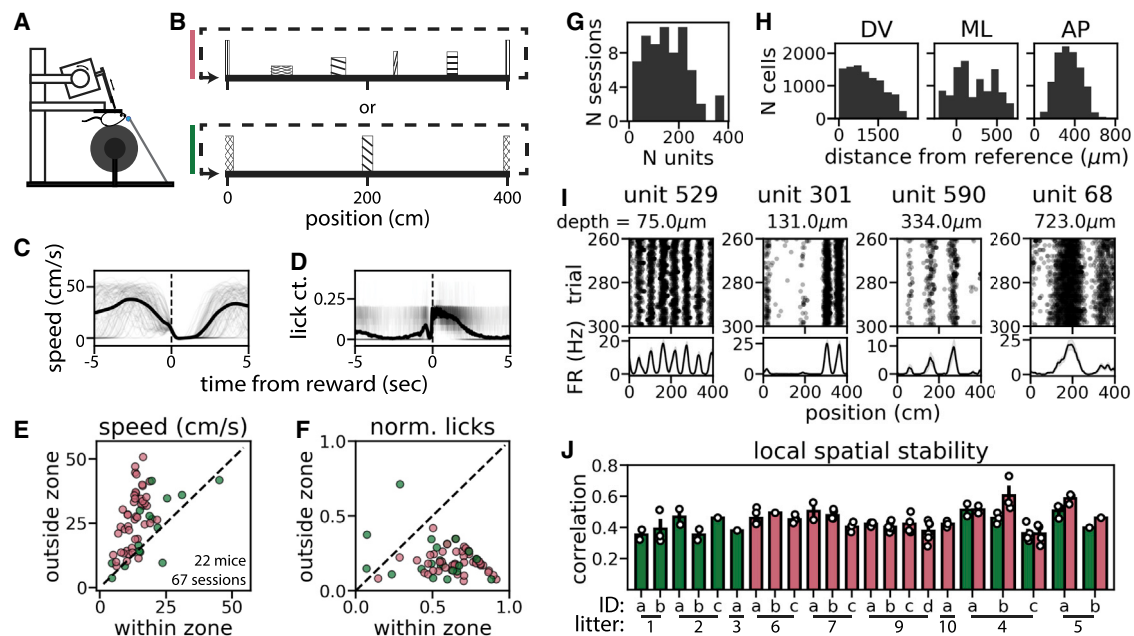
Figure 1. Acute neuropixels recordings during a VR random foraging task

(A) Schematized recording setup.

(B) Schematized side view of the cue-rich (top, pink) and cue-poor (bottom, green) environments (colors maintained throughout).

(C) Running speed near rewards for an example 100 trials (gray traces, each trial; black line, average).

(D) As in (C) for smoothed lick count.

(E) Running speed in versus outside of reward zones (mean difference ± standard error of the mean [SEM], 11.7 ± 1.2 cm/s; Wilcoxon signed-rank test, p = 3.1 × 10^{-11}) (points, individual sessions; for double-track sessions, color indicates which track was first).

(F) As in (E) but for fraction of licks (mean difference in lick number ± SEM, 8.0 ± 0.2; Wilcoxon signed-rank test, p < 1 × 10^{-11}; N = 16,083 reward trials).

(G) Number of MEC units recorded in each session.

(H) Locations for all recorded MEC units relative to their dorsal-ventral (DV), medial-lateral (ML), and anterior-posterior (AP) location.

(I) Rasters (top) and tuning curves (bottom) for four example cells from one session (black lines, average firing rate; shading, SEM).

(J) Average spatial correlation of nearby trials (mean moving average correlation, 5 nearest trials ± SEM, 0.441 ± 0.009; mean spatial correlation range, 0.28–0.73) (points, individual sessions; bars, individual mice; error bars, SEM across sessions). Double-track sessions show cue-rich and cue-poor blocks separately (26 blocks from 13 sessions).

(E–H and J) N = 10,440 cells in 67 sessions across 22 mice. The example session in (C) and (D) is mouse 6a, session 1009_1, and in (I) is mouse 6a, session 1010_1. See also Figure S1.

a result, previous studies of flexibility in MEC neural coding often focused on how the firing fields of single functionally defined neurons (e.g., grid, border, or head direction cells) responded to changing environmental features. However, not all MEC neurons fall into discrete functional classes (Hardcastle et al., 2017; Hinman et al., 2016), and many MEC coding properties are proposed to emerge at the population level (Burak and Fiete, 2009; Couey et al., 2013; Fuhs and Touretzky, 2006; McNaughton et al., 2006; Ocko et al., 2018; Pastoll et al., 2013). These features of MEC coding raise the question of how the MEC circuit as a whole transitions between contextual representations.

Here we investigated how large populations of MEC neurons transition between spatial representations in an invariant environmental context. Using silicon probes, we simultaneously recorded from hundreds of MEC neurons while mice navigated a virtual reality (VR) environment. We found that remapping events occurred synchronously across the MEC population—recruiting a large swath of cells with heterogeneous coding properties—without any change in environmental features or task demands. Further, we demonstrated that each map corresponded to a

distinct activity manifold. Running speed correlated with transitions between these manifolds, and the manifolds for different maps were geometrically aligned, which enabled consistent position decoding in spite of remapping. Our findings demonstrate that a single population of neurons can preserve unchanging information (e.g., fixed environmental features) while also encoding changes in internal context, demonstrating a remarkable capacity for reliability and flexibility in MEC navigational coding.

RESULTS

Spatial representations remap in an invariant VR environment

Head-fixed mice foraged for water rewards along an infinite VR track with landmark cues that repeated every 400 cm (Campbell et al., 2020; Figures 1A and 1B; Figure S1). 11 mice navigated a cue-rich track (5 landmarks; Figure 1B, top; n = 43 sessions; i.e., cue-rich sessions), 6 mice navigated a cue-poor track (2 landmarks; Figure 1B, bottom; n = 11 sessions; i.e., cue-poor sessions), and 5 mice navigated alternating blocks of cue-rich and

cue-poor trials within each session (n = 13 sessions; i.e., double-track sessions). Mice could lick for water in a visually marked reward zone, which appeared at random locations along the track (Figure S1). In reward zones, mice slowed (difference in running speed, 11.7 ± 1.2 cm/s; p = 3.1 × 10^{-11}) and licked for water (difference in lick number, 8.0 ± 0.2; p < 1 × 10^{-11}), demonstrating familiarity with the task (Figures 1C–1F).

To record neural activity during behavior, we acutely inserted Neuropixels silicon probes (Jun et al., 2017) into the MEC for up to six recording sessions per mouse (up to three sessions per hemisphere). Each session was associated with a unique probe insertion, and sessions from the same hemisphere were recorded in distinct parts of the MEC (Figure S1; STAR Methods). Thus, we recorded simultaneously from hundreds of neurons across a large portion of the MEC dorsal-to-ventral axis in individual mice (N = 10,410 cells) (Figures 1G and 1H; Figure S1). To quantify spatial coding stability across co-recorded neurons, we estimated each neuron's position-aligned firing rate on each trial and computed the correlation between all spatial representations for each pair of trials (STAR Methods). Network-wide position coding was typically stable across neighboring trials (moving average correlation, 5 nearest trials: 0.441 ± 0.009; Figures 1I, 1J, and 2A; STAR Methods). In some cases, population-wide neural representations appeared untethered from landmarks for part or all of the session (interquartile range min to max, 0.034–0.394; Figures S1 and S2).

To characterize the stability of population-wide spatial coding across all trials for each session, we collected the spatial correlations for all trial pairs into a trial-by-trial similarity matrix of network-wide spatial representations (Figures 2B–2F, right; Figure S2). In many recording sessions, we observed clear changes in these similarity matrices and in the spatial firing patterns of single neurons distributed across the MEC (i.e., remapping events) (Figures 2B–2F; Figure S2). These remapping events occurred without any change to environmental sensory cues or task demands. Importantly, remapping events did not reflect recording probe movement (Figures S1 and S3). MEC cells often switched abruptly between stable spatial representations (i.e., maps), with cells returning repeatedly to one of several distinct maps within a single session, resulting in a checkerboard pattern of trial-by-trial similarity (Figures 2B–2F; Figures S1 and S2). In other sessions, spatial representations underwent a single transition between stable maps or transitioned between spatially stable and unstable coding regimes (Figures 2B and 2F; Figure S2). Although the frequency and stability of remapping was thus heterogeneous across mice and sessions, in all cases, remap events appeared to recruit co-recorded neurons all along the dorsal-to-ventral MEC axis (Figures 2B–2F, right; Figures S1 and S2).

Data-driven detection of network-wide remapping
To characterize neural remapping not related to changing visual cues, we focused our main analysis on sessions from mice who only experienced one of the two tracks (single-track sessions, n = 54 sessions in 17 mice). Double-track sessions recapitulated all major single-track findings, indicating that the general properties of network-wide remapping were consistent across different VR tasks (Figure S4). To group trials with similar network-wide spatial activity, we applied k-means clustering to each session

(Figure 3A). The k-means model assigns a single cluster label to each trial, and these cluster labels often visibly matched the checkerboard pattern in trial-by-trial similarity matrices (Figure 3B; Figures S2 and S5). Despite making the strict assumption that each trial belongs to a single spatial map, a 2-cluster k-means model consistently approached the performance of a less constrained uncentered principal component analysis (PCA) model, which allows each trial to contain a blend of multiple spatial maps (Figure 3C). These results suggest that remapping events were well approximated by discrete transitions between spatial maps.

Using the relative performance of k-means to PCA, we identified 28 of 54 single-track sessions that were adequately fit by a 2-cluster k-means model (Figure 3D; n = 23 cue-rich sessions, 5 cue-poor sessions; 4,984 cells; 13 mice). In many of these sessions, neural activity repeatedly returned to the same set of maps (13 of 28 sessions exhibited three or more remap events; Figure S3). Trials within a given map were more similar to each other than to trials from the other map (p = 3.79 × 10^{-6}; Figure 3E). We focused subsequent analysis on these 28 "2-map sessions" because they were the simplest and most common case. The remaining 26 sessions showed heterogeneous remapping that was not well captured by this two-factor model (Figure 2; Figures S1, S2, and S5).

How do these network-wide events recruit single neurons? In 2-map sessions, the majority of cells throughout the MEC (82.4%) changed their firing properties in precise agreement with population-derived remapping events (i.e., consistent remappers, average distance to cluster centroid < 1; Figures 3B, bottom, and 3F; STAR Methods). There was some regional variability in the proportion of these consistent remappers, but they always comprised the bulk of the population (Figure 3F). Thus, most MEC neurons remapped abruptly and synchronously.

Entorhinal neurons remap reversibly and heterogeneously
We next investigated how remapping influences the coding properties of single MEC neurons. In 2-map sessions, many single neurons exhibited distinct spiking patterns and spatial tuning curves within each k-means-identified map (Figure 4A; see Figure S5 for sessions with more than 2 maps). To quantify these tuning changes, we calculated the change in peak firing rate (i.e., rate remapping) and the spatial dissimilarity (i.e., global remapping) across the two maps for each cell (Knierim et al., 1998; Muller and Kubie, 1987; O'Keefe and Conway, 1978; STAR Methods). Across all cells (spatial and non-spatial), we observed diverse remapping responses with no obvious clustering in type of remapping (Figures 4B and 4C). Many cells showed some rate remapping (median absolute change in peak firing rate, 1.28-fold) but largely retained their spatial tuning across maps (median spatial dissimilarity, 0.031; Figures 4A–4C). Nevertheless, the dissimilarity distribution was heavy tailed (Figure 4B), indicating a subpopulation of cells with large changes in spatial tuning across maps (5% of cells had a dissimilarity greater than 0.223; e.g., Figures 4A and 4C, cool colors).

Cells also exhibited changes in spatial information across the two maps (e.g., Figure 4D), resulting in many cells (13% ± 1%) that were significantly spatial in one map but not the other
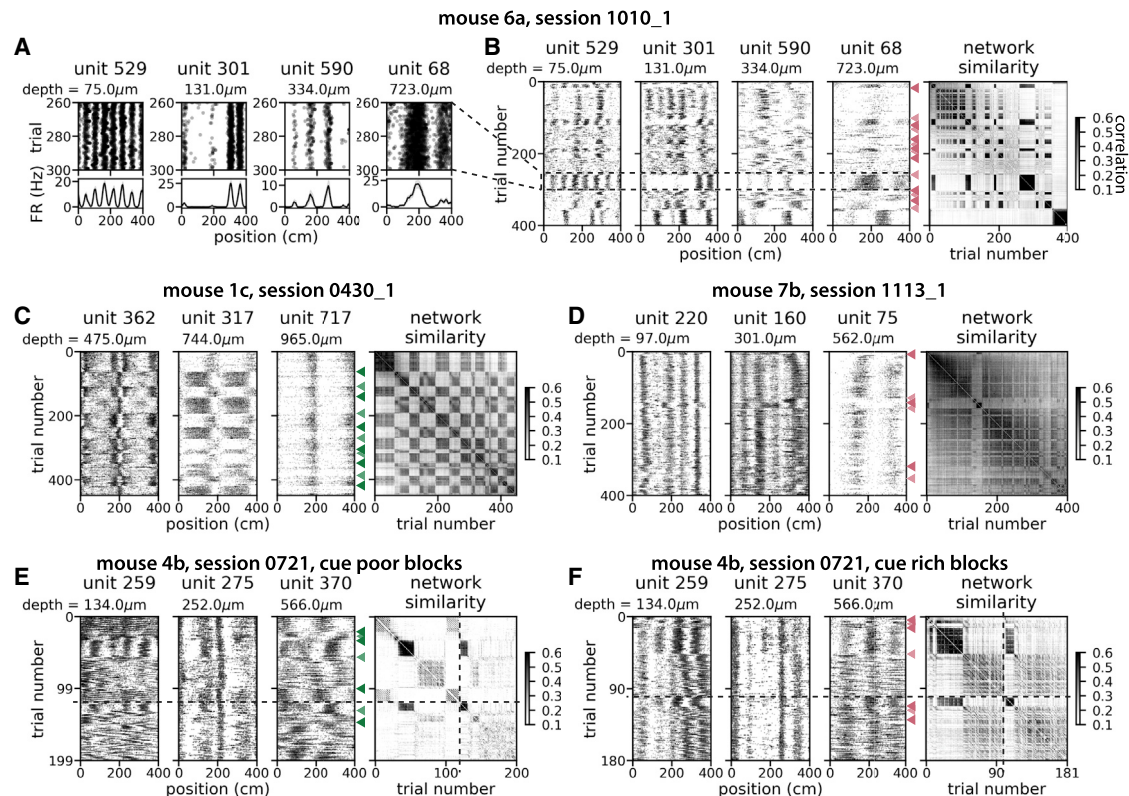
Figure 2. Spatial representations remap in an invariant VR environment
(A) Rasters (top) and tuning curves (bottom) for four example cells (40 example trials) from one session (black lines, average firing rate; shading, SEM; same as Figure 1J).
(B) Full rasters (left) and network-wide trial-by-trial similarity matrix (right) from the same example session (arrowheads, transitions in spatial coding; colors alternate light/dark for discernibility; color bar, trial-by-trial spatial correlation) (n = 142 cells).
(C and D) As in (B) but for a different example cue-poor (C; n = 227 cells) and cue-rich (D; n = 139 cells) session.
(E and F) As in (B)–(D) but for one double-track session split into cue-poor (E) and cue-rich (F) trial blocks (dashed lines indicate breaks between blocks; n = 55 cells).
(B–F) Pink arrowheads indicate cue-rich tracks and green arrowheads cue-poor tracks.
See also Figures S2 and S3.

(p < 0.05; Figure 4E). Cells with high spatial information in at least one map were likely to remap in coordination with the rest of the population (1,449 of 1,545 spatial cells; Figure 4F). Thus, remapping reflected changes in the spatial coding properties of MEC neurons and recruited most spatially informative cells.

Navigationally relevant cell types remap differentially
The MEC contains functional cell types that exhibit distinct spatial coding statistics (e.g., grid, border, object vector, and head direction cells) (Hafting et al., 2005; Høydal et al., 2019; Kropff et al., 2015; Sargolini et al., 2006; Savelli et al., 2008; Solstad et al., 2008), raising the question of how remapping recruits these types. To investigate this question, we leveraged a previous finding that grid cells are responsive to mismatches between visual and motor feedback in 1D VR, whereas border cells remain stable (Campbell et al., 2018). In a subset of 2-map sessions (n = 9 cue-rich sessions), we appended 5 trials in which the mouse had to run twice as far to traverse the same visual landscape (i.e., gain mismatch) (Figure 5A). We classified spatially

stable cells that were highly responsive to gain mismatch as putative grid cells (224 of 1,379 cells) and those that were unaffected by gain mismatch as putative border cells (152 cells) (Figures 5B–5D; Figure S6) (referred to here as grid and border cells; STAR Methods). Using this metric, we found that the proportions of each cell type were similar to what is seen in freely moving experiments (Campbell et al., 2018; Mallory et al., 2018; Miao et al., 2015; Figure 5B; STAR Methods). Grid cells often had spatially periodic firing fields of various spatial frequencies, and border cells often had periodic firing fields that qualitatively reflected the tower spacing (Figure 5G; Figure S6). In keeping with our analysis of all 2-map sessions, most grid, border, and other spatial cells were consistent remappers (Figure S6). Grid cells exhibited more extensive remapping than border cells or other spatial cells, demonstrating more global remapping (p = 1.97 × 10^{-5}; Figure 5E, right) and more rate remapping than the other cell types (p = 4.6 × 10^{-4}; Figure 5E, left). These results indicate that remapping events may differentially affect specific functional cell types.
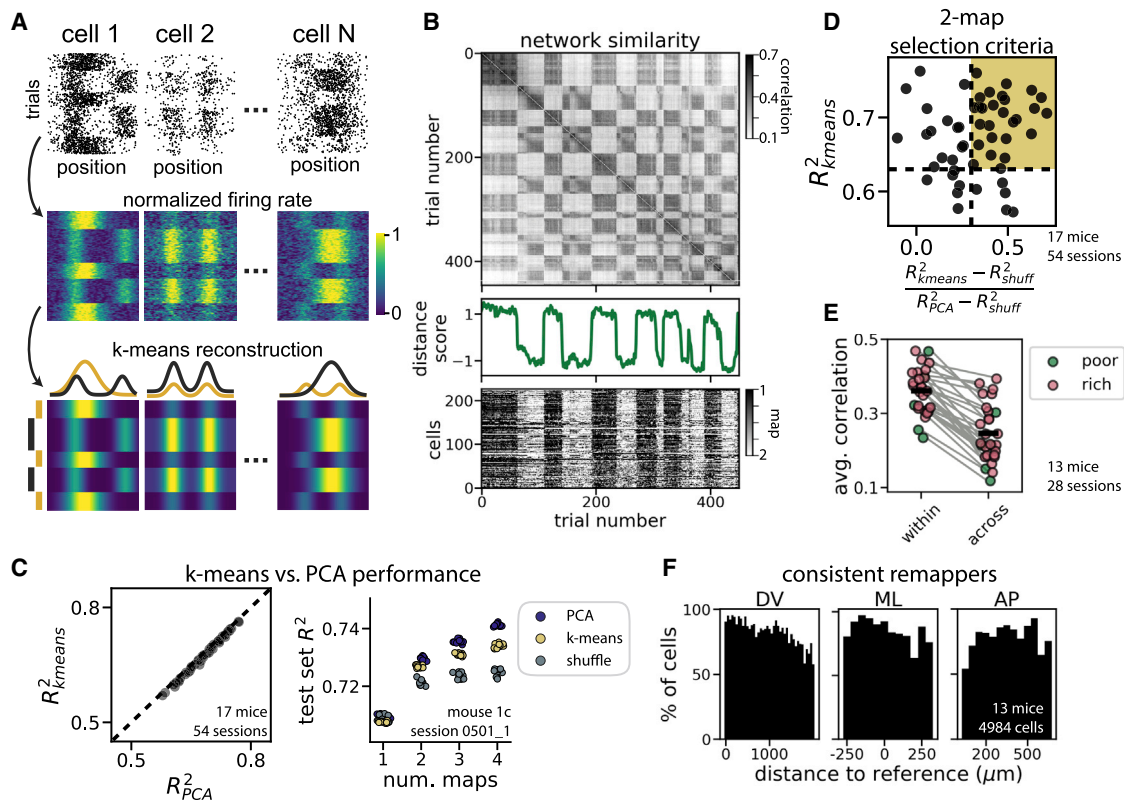
CellPress

Figure 3. Data-driven detection of discrete, synchronous remapping

(A) Schematized k-means clustering strategy: raw spikes (top) are converted to normalized firing rate (center); k-means gives a low-dimensional estimate of the neural activity (bottom).

(B) Trial-by-trial similarity (top, as in Figures 2B–2F), distance score for population-wide neural activity by trial (center; a score of 1 indicates that population activity is in the map 1 k-means cluster centroid; score = −1, map 2 centroid), and single-neuron distance to k-means cluster centroid across trials, sorted by depth (bottom; color bar, distance to cluster: gray, midway between maps; black, at or beyond map 1 centroid; white, map 2) for an example session (mouse 1c, session 0430_1, n = 227 cells).

(C) 2-factor k-means versus PCA performance (uncentered R^2) for all single-track sessions (left) and model performance for uncentered PCA (indigo), k-means (gold), and k-means on shuffled data (gray) for an example session (right) (mouse 1c, session 0501_1, n = 122 cells).

(D) Selection criteria for 2-map sessions (gold shading; performance gap with PCA < 70% relative to shuffle, $R^2_{kmeans} > 0.63$).

(E) Average trial-to-trial spatial similarity for trials from the same map versus across maps for all 2-map sessions (mean change in correlation ± SEM, −32.43% ± 2.57%; Wilcoxon signed-rank test, p = 3.79 × 10^{-6}) (green, cue-poor sessions; pink, cue-rich sessions).

(F) Percentage of all cells that were consistent remappers by location in the MEC (4,108 of 4,984 cells).

(C and D) n = 54 sessions, 17 mice.

(E and F) n = 4,984 cells, 28 sessions, 13 mice.

See also Figures S2 and S5.

Positional information is conserved at a population level across distinct spatial maps

MEC neurons project to multiple brain areas involved in spatial information processing (Kerr et al., 2007). To consider how downstream brain areas might make use of positional information encoded in the MEC in spite of network-level remapping, we investigated whether linear mechanisms can predict position from MEC neural activity by fitting circular-linear regression models ("decoders") (Figures 6A–6C; see Figure S5 for sessions with more than 2 maps; STAR Methods). We found that performance was comparable between models trained and tested on trials from a single map and models trained and tested on trials from both maps (p = 0.65; Figure 6D). Although remapping could in theory disrupt the performance of a simple linear

decoder if each neuron's spatial tuning was sufficiently different across the maps, this finding may arise because only ∼5% of cells showed large changes in spatial tuning across maps (Figure 4B). To simulate an alternative case where each neuron's spatial tuning changed dramatically across maps, we produced a shuffled "synthetic map" for each of the two maps in which spatial coding was greatly altered (STAR Methods). Linear decoders trained jointly on data from each true map, and its synthetic map performed substantially worse than decoders trained on data from the pair of true maps (p < 0.05; Figure 6D).

Models also performed poorly when trained on data from one map and tested on data from the other map (p = 7.55 × 10^{-11}; Figures 6E and 6F), although these decoders still largely outperformed a shuffle control (trained on one map, tested on the
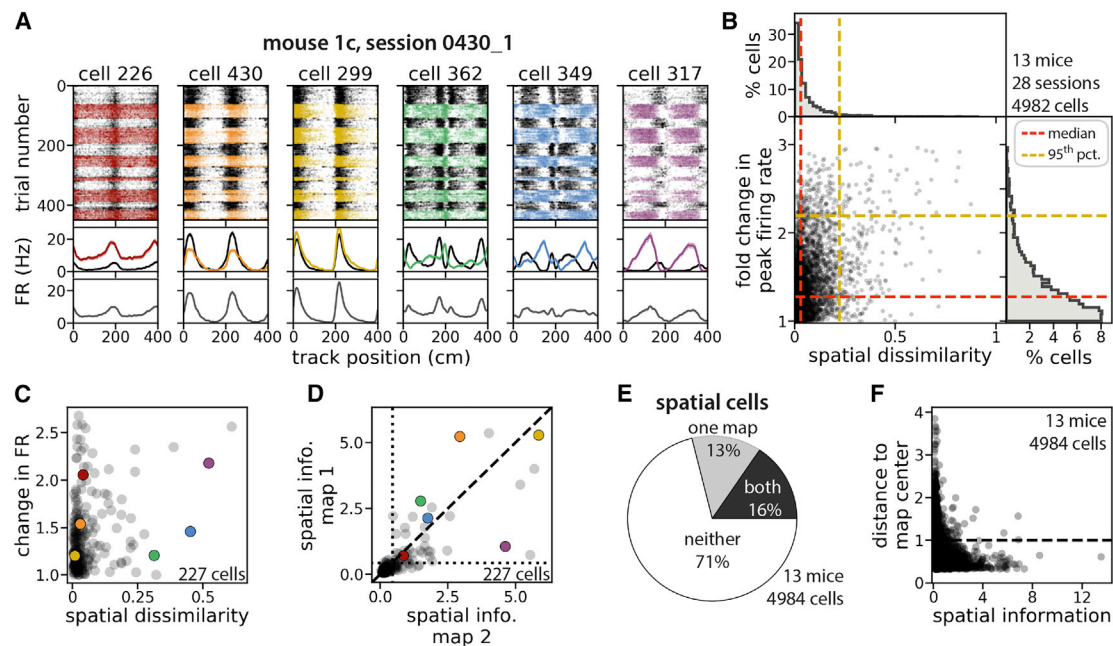
Figure 4. Reversible and heterogeneous remapping of MEC representations

(A) Single-neuron spiking (top) and tuning curves (center) for example cells from a cue-poor, 2-map session, colored/divided by k-means cluster labels (black, map 1; color, map 2), versus averaged over the session (bottom) (solid line, average firing rate; shading, SEM; the color scheme denotes cell identity and is preserved in C and D).

(B) Absolute fold change in firing rate versus spatial dissimilarity across maps for neural tuning curves from all 2-map sessions (n = 4,982 cells) (points, single cells; histograms, density distribution for each variable; red dashes, median; gold dashes, 95th percentile).

(C) Same as (B) but for the example session.

(D) Spatial information for single neurons in map 1 versus map 2 for the example session.

(E) Percentage of MEC neurons from all 2-map sessions that were spatial in one map (gray), both maps (black), or neither map (white) (mean absolute change in spatial information ± SEM, 57.3% ± 1.0%).

(F) Distance to k-means cluster centroid by highest average spatial information across maps (black dashes, threshold for consistent remappers).

The example session in (A), (C), and (D) is from mouse 1c, session 0430_1 (n = 227 cells). In (E) and (F), n = 4,984 cells from 28 sessions in 13 mice. See also Figure S5.

shuffled other map; 3 of 56 model pairs were no better than shuffle; p ≥ 0.05; Figure 6G; STAR Methods). Therefore, the two maps are different—decoders specialized to a single map struggled to generalize to the other map (Figures 6E–6G)—but, given data from both maps, a single linear decoder performed as well as separate, specialized decoders (Figure 6D). These results indicate that, in principle, downstream brain areas can exploit positional information encoded in the MEC in the presence of remapping.

Neural activity transitions between geometrically aligned ring attractor manifolds

MEC representations of space can be flexible (e.g., spontaneous remapping) and reliable (e.g., consistent decoding performance). To reconcile these two aspects of the circuit, we characterized the geometry of position coding in K-dimensional firing rate space (where K denotes the number of simultaneously recorded neurons). For the continuous 1D VR tracks used in this study, spatially driven network activity should follow a 1D ring manifold as the animal moves through space, reflecting the ring-shaped environment (Figure 7A). Attractor models of spatial navigation predict that circuit dynamics are locally stable around

these ring manifolds, enabling persistent internal representations of position (Samsonovich and McNaughton, 1997). Each k-means cluster centroid provides an empirical estimate of this attractor manifold, and, indeed, the low-dimensional linear embedding of each centroid had a qualitative ring structure (Figures 7B and 7C). We derived these ring-shaped manifolds from all co-recorded MEC neurons, making them distinct from the ring structures that arise in attractor models of co-modular grid cells (Burak and Fiete, 2009; Fuhs and Touretzky, 2006; Guanella et al., 2007; Spalla et al., 2019). Manifolds derived from cue-poor environments were tightly wound around themselves, as quantified by an entanglement metric (Figure 7D; n = 10 manifolds from 5 cue-poor sessions, 46 manifolds from 23 cue-rich sessions; STAR Methods), indicating that there may be limited discriminability between the first and second halves of the track in this environment. These results suggest that the number of landmark cues alters the geometry of the spatial map without altering its topology as a 1D ring.

We next used PCA to visualize the two manifolds of each 2-map session in the same low-dimensional space. In cue-rich and cue-poor environments, the manifolds appeared geometrically aligned so that position coding was translated along a
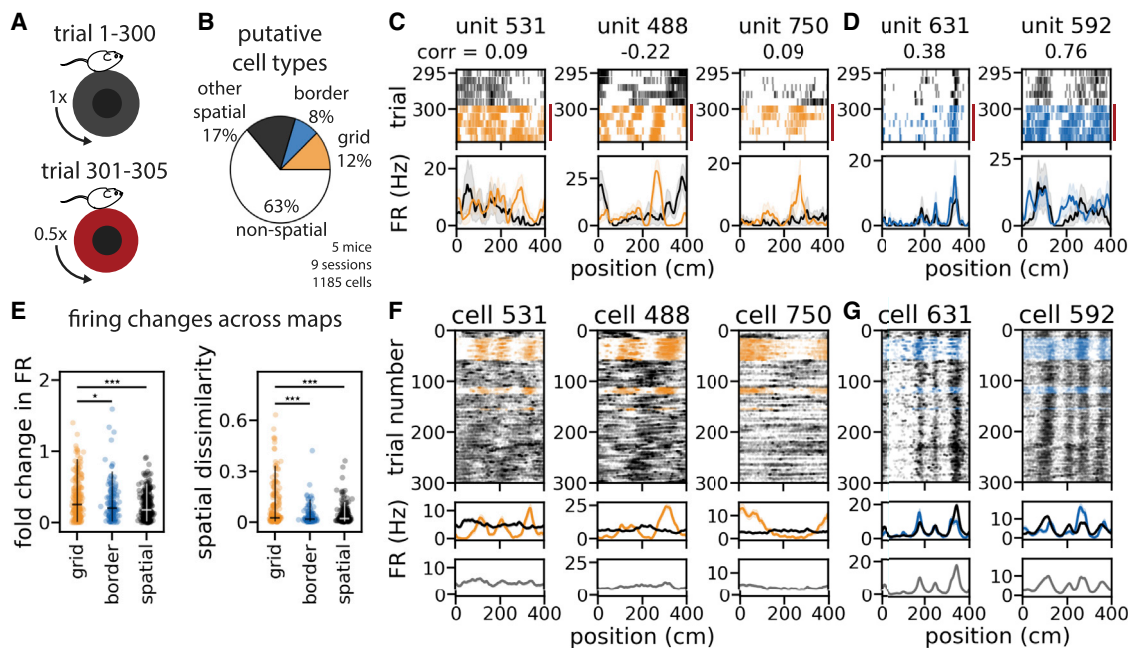
CellPress

Figure 5. Navigationally relevant cell types participate differentially in remapping

(A) Schematized trial structure: 300 trials with a fixed relationship between visual and motor feedback (top, "normal trials") and 5 trials requiring the animal to run twice as far to travel the same visual distance (bottom, "gain change trials").

(B) Distribution of putative cell types (session mean ± SEM: grid cells, 18.26% ± 3.81%; border cells, 13.23% ± 2.07%; spatially stable cells, 24.57% ± 3.12%).

(C) Rasters (top) and tuning curves (bottom) from example grid cells for adjacent normal (black) and gain change trials (red bar, colored spikes) (numbers indicate cell identity and correlation of tuning curves across gain change; solid line, average firing rate; shading, SEM).

(D) Same as (C) but for example border cells.

(E) Absolute fold change in firing rate (left) (median change, 95th percentile: grid cells = 25.5%, 89.2%; border cells = 20.6%, 71.6%; other spatial cells = 18.5%, 55.5%; Kruskal-Wallis H test, $p = 4.6 \times 10^{-4}$; Wilcoxon rank-sum test: border versus grid cells, $p = 0.045$; other spatial versus grid cells, $p = 8.59 \times 10^{-5}$; other spatial versus border cells, $p = 0.217$) and spatial dissimilarity (right) (median, 95th percentile: grid cells = 0.027, 0.334; border cells = 0.019, 0.137; spatial cells = 0.022, 0.119; Kruskal-Wallis H test, $p = 1.97 \times 10^{-5}$; Wilcoxon rank-sum test: border versus grid cells, $p = 4.13 \times 10^{-5}$; other spatial versus grid cells, $p = 1 \times 10^{-4}$; other spatial versus border cells, $p = 0.227$) across maps for all grid, border, and other spatial cell tuning curves (***$p < 0.001$, *$p < 0.05$; horizontal bars, median; vertical bars, 5th to 95th percentile).

(F) Rasters (top) and tuning curves (middle) for example grid cells, colored by k-means cluster labels (color, map 1; black, map 2) versus averaged over the session (bottom) (solid line, trial-averaged firing rate; shading, SEM).

(G) As in (F) but for example border cells. All example cells are from mouse 9b, session 1207_1 (n = 61 grid cells, 9 border cells, 165 total cells).

(B and E) n = 224 grid cells, 152 border cells, 308 other spatial cells of 1,185 total excitatory cells from 9 cue-rich sessions in 5 mice. Orange color denotes grid cells, and blue denotes border cells throughout.

See also Figure S6.

single dimension in neural activity space (Figures 7E and 7F) relative to a shuffle control (Figures 7G and 7H). Using Procrustes shape analysis (STAR Methods), we found that the two manifolds required only modest rescaling (23 of 28 sessions ≤ 10% difference in scale; Figure 7I) and rotation (all were within 7% of the optimal rotation; Figure 7J) to align with one another. This alignment could arise from preserved spatial tuning in some neurons and indicates that single-neuron coding changes (Figures 4A and 4B) did not result in net firing rate differences across maps. Although tuning is correlated within grid cell modules (Hafting et al., 2005), manifold alignment was not dependent on putative grid cells (Figure S6). Thus, remapping largely corresponded to a translation in neural activity space between geometrically aligned ring manifolds.

Because the k-means-identified ring manifolds show only the average neural activity within each map, we next investigated how single-trial variability around these manifolds was struc-

tured. For each session, we applied PCA to the residuals of the k-means model and kept enough components to capture at least 20% of variance in these residuals. These dimensions contain the most "noise" or variability that is not captured by the k-means model. We then computed the angle between the PC (principal component) subspace and the dimension separating the two manifolds (the "remapping dimension"); this angle was smaller than expected under a null distribution of random linear subspaces, indicating that unexplained neural variability was oriented preferentially in the remapping dimension (Figures 7K and 7L). The scale of this variability was limited so that the two manifolds were often well separated (the maps can be considered two separate ring manifolds rather than a single hollow cylinder). This result suggests that variability in network activity could predispose the network to remap (or jump from one ring manifold to the other), with attractor dynamics locally stabilizing the network activity within each map (Video S3).
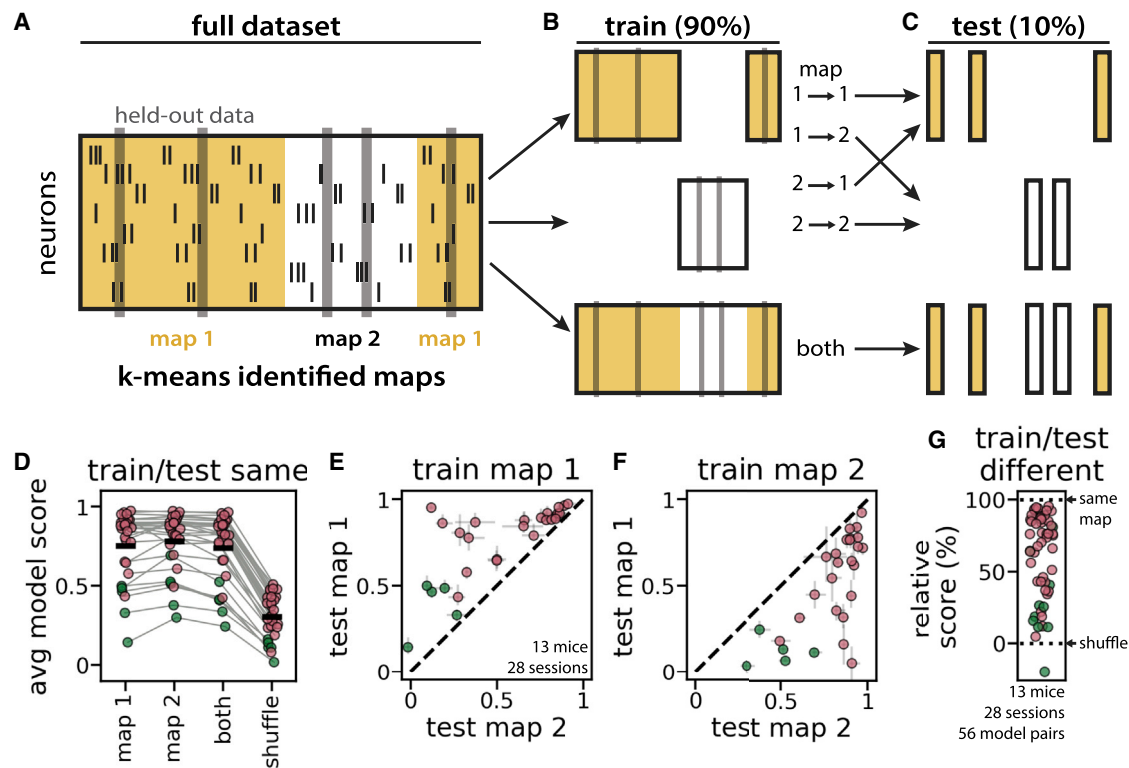
Figure 6. Positional information is conserved at a population level across maps

(A–C) Schematic: decoder training and testing strategy (yellow, map 1; white, map 2).

(A) Population-wide spiking activity (hash marks) was divided into k-means clusters; 10% of data were held out for testing (gray).

(B) Decoders were trained on data from map 1 (top), map 2 (center), or both maps (bottom).

(C) Each model was used to predict position on held-out data from map 1 (top), map 2 (center), or both maps (bottom). Models trained only on data from one map were also used to predict position using only data from the other map (diagonal arrows from B to C; labels indicate test map → train map).

(D) Decoder performance for models trained and tested on data from the same map (map 1 and map 2), from both maps, or from each map and its shuffle for all 2-map sessions (score = 0, chance; score = 1, perfect prediction) (mean performance ± SEM: train/test map 1 = 0.75 ± 0.04; train/test map 2 = 0.78 ± 0.04; train/test both maps = 0.74 ± 0.04; Kruskal-Wallis H test, p = 0.65; mean ± SEM decrease from train/test both to shuffle, 60.84% ± 1.5%).

(E and F) Decoder performance within versus across maps for all sessions (gray bars, interquartile range).

(G) Decoder performance for models trained on one map and tested on the other relative to shuffle (0%) and within-map performance (100%) (mean relative performance ± SEM, 61.81% ± 4.01%; Wilcoxon signed-rank test, p = 7.55 × 10^{-11}; n = 56 model pairs).

(D–F) Points indicate individual sessions; green, cue-poor; pink, cue-rich; n = 4,984 cells, 28 sessions, 13 mice.

A simple model explains the alignment of spatial attractor manifolds

Is the geometric alignment of spatial manifolds (Figures 7E and 7F) expected under existing theories of navigational circuit dynamics? To address this question, we constructed a bistable ring attractor model by linearly combining the weights of a winner-take-all network and a ring attractor network (Figure S7; STAR Methods). This approach is similar to previous models that have linearly combined decorrelated connectivity matrices to support multiple attractor structures (Romani and Tsodyks, 2010; Roudi and Treves, 2008; Samsonovich and McNaughton, 1997; Stringer et al., 2004). We could induce remapping in this model with perturbations that pushed the neural state into the other map's basin of attraction (Monasson and Rosay, 2014, 2015). This model was sufficient to reproduce multiple ring manifolds, but it failed to recapitulate the experimentally observed geometrical manifold alignment (Figure S7). This finding demonstrates that geometrically aligned ring manifolds

are not an obvious consequence of the attractor model framework.

A minor modification to this model—adding a sub-population of "shared neurons" whose spatial tuning was preserved across maps—was sufficient to produce a pair of bistable ring attractors that were highly aligned (Figure S7). This subpopulation of shared neurons is loosely analogous to the experimentally observed neurons with preserved spatial tuning across maps (Figures 4A and 4B). The modified model had two additional noteworthy properties. First, perturbing the neural activity in the direction of the other map induced a remap but preserved the positional representation (Figure S7). Second, remapping was well described by a translation along a single dimension; thus, any spatial decoder that is insensitive to this dimension will be robust to remapping. For a linear decoder (as in Figure 6), this means the remapping dimension should lie in the null space of the decoder weights (Kaufman et al., 2014; Rule et al., 2020). These results suggest that neurons with shared coding across maps may result
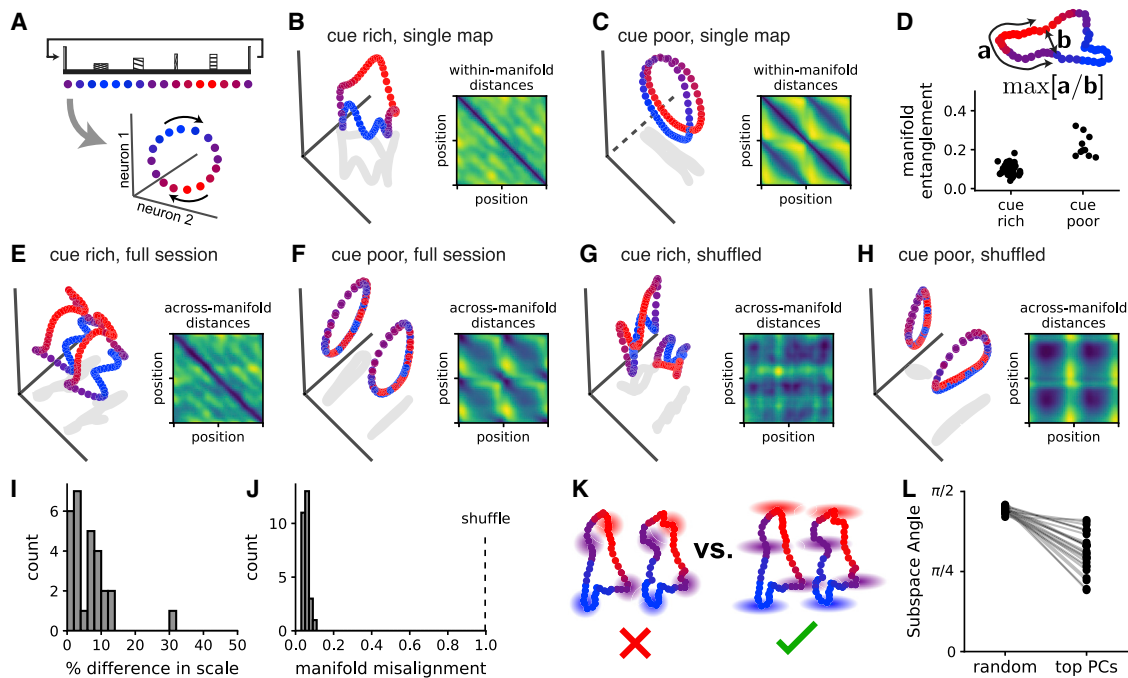
Neuron
Article

CellPress

Figure 7. Neural activity transitions between geometrically aligned ring attractor manifolds

(A) Schematic: the 1D environment (top) should produce a circular 1D trajectory (i.e., ring manifold) in neural activity space (bottom) (the color scheme indicates position and is preserved throughout).

(B) PCA projection of a single map (k-means centroid) from a cue-rich session. Inset: pairwise distances in neural activity for all points (position bins) in the manifold (blue, minimum value; yellow, maximum).

(C) As in (B) but for a cue-poor session. The linear projection uses two principal components and a third orthogonal axis (dashed line) that maximally distinguishes between the first and second half of the track.

(D) Top: schematic of the entanglement measure, which is the maximum ratio of distance along the manifold ("a") to extrinsic distance ("b") (STAR Methods). Bottom: manifold entanglement in cue-rich and cue-poor environments.

(E) PCA projection of two manifolds from a 2-map cue rich session. Inset: across-manifold distances in neural firing rates for every pair of points (colors as in B and C insets).

(F) As in (E) but for a cue-poor session.

(G and H) As in (E) and (F), respectively, but after randomly rotating each manifold.

(I) Percent difference in manifold size for maps from the same session.

(J) Normalized manifold misalignment scores (0, perfectly aligned; 1, shuffle).

(K) Schematics: variability around the manifolds is oriented toward the other manifold (right), not isotropically distributed (left).

(L) Right: angles between the remapping dimension and the top PC subspace for the model residuals (20% of total noise variance). Left: null distribution (random linear subspaces, n = 1,000 samples). All sessions displayed smaller angles than their shuffle control (Wilcoxon rank-sum test, $p < 1 \times 10^{-6}$).

The example in (B), (E), and (G) is session 1005_2 from mouse 6b (n = 149 cells) and in (C), (F), and (H) is session 0502_1 from mouse 1c (n = 227 cells). In (D), (I), (J), and (L), n = 46 manifolds from 23 cue-rich sessions in 10 mice and 10 manifolds from 5 cue-poor sessions in 3 mice.

in geometrically aligned ring manifolds, which, in turn, enables simple mechanisms for remapping and position decoding.

Remapping events and neural variability correlate with slower running speeds

Because the task and environment in our experiments did not change within a given session, we next examined whether single-trial variability and network-wide remapping correlated with aspects of the animal's behavior. We focused on running speed because speed is known to modulate spatial representations in the MEC (Bant et al., 2020; Hardcastle et al., 2017). We first wanted to find out whether the animal's running speed was different on "remap trials" (the two trials book-ending each transition from one map to the other) compared with the intervening "stable blocks" (the block of trials at least two trials away from a

remap event, which all reside in the same map). We restricted our analysis to 2-map sessions that contained at least three remap events and to stable blocks of at least five trials (n = 13 sessions in 7 mice; see Figure S5 for sessions with more than 2 maps; STAR Methods). Across most of these sessions, the animal's average running speed on remap trials was slower compared with its average running speed in the preceding stable block (difference in running speed, 9.8% ± 2.2%; $p = 1.15 \times 10^{-4}$; Figures 8A and 8B). Running speed largely did not vary systematically across the two maps (Figure S8; STAR Methods).

We next investigated the moment-by-moment correlation between neural variability and running speed by binning neural activity and speed into 5-cm position bins within each trial. Given the random reward structure, the running speed distribution for each track position was comparable (Figure S8). For each
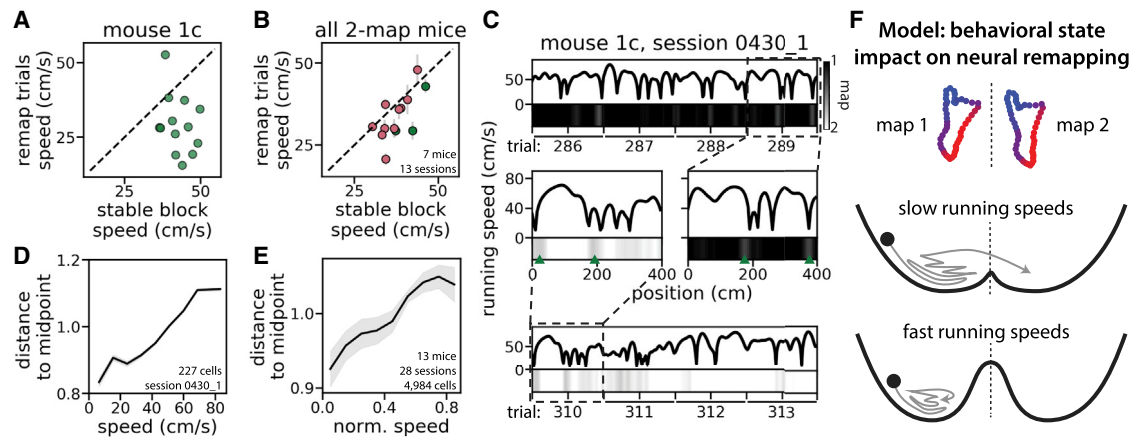
Figure 8. Remapping events and neural variability correlate with slower running speeds

(A) Average running speed in remap trials versus stable blocks for an example session (points, pairs of remap trials/stable blocks; dashed line, unity; n = 13 pairs).

(B) As in (A) but for all 2-map sessions (remap trial speeds were slower in 9 sessions, equal in 2 sessions, and faster in 2 sessions; Wilcoxon signed-rank test, p = 1.15 × 10^{-4}) (points, session average; green, cue-poor; pink, cue-rich; gray error bars, SEM; dashed line, unity; n = 127 pairs from 13 sessions, 7 mice).

(C) Top and bottom: running speed and distance to the midpoint between k-means clusters for trials from two example stable blocks (black trace, running speed; black shading, map 1; white, map 2; gray, between maps). Center: magnification of the first (left) and last (right) example trials, showing neural activity approaching the midpoint (arrowheads) (across sessions, stable bin distance interquartile range: 0.763–1.338).

(D) Average distance to the midpoint versus binned running speed for an example session (distance score = 1, activity is in a cluster centroid; 0, equidistant from each map) (black line, average; gray shading, SEM; distance to midpoint is Z scored).

(E) as in (D) but for all 2-map sessions; speed is normalized within each session (ordinary least-squares regression, R^2 = 0.778, p < 1 × 10^{-6}; n = 10 speed bins per session, 28 sessions, 13 mice, 4,984 cells).

(F) Schematic model showing how running speed might encourage neural activity (ball, activity; arrow, trajectory in state space) to transition between manifolds (top) by shaping the energy landscape (black line) (center, slow speeds; bottom, fast; dashed line, midpoint between clusters).

The example session in (A), (C), and (D) is from mouse 1c, session 0430_1 (n = 227 cells). See also Figures S5 and S8.

position bin, we calculated how close the neural activity was to the midpoint between manifolds, where activity is most likely to switch between maps (STAR Methods). As expected, neural activity was closer to the midpoint in remap trials compared with stable blocks (p < 1 × 10^{-6}; Figure S8). However, we also observed instances where the neural activity approached the midpoint within stable periods, indicating that neural variability does not always provoke a remap event (Figure 8C, arrowheads). Across all position bins, slower running speeds were correlated with a reduced neural distance to the midpoint between manifolds (p < 1 × 10^{-6}; Figures 8D and 8E). These results suggest that neural variability increases in the direction of the other map at slow running speeds, likely increasing the probability of a remap event. If we model the two spatial maps as bistable ring attractors (as discussed in the previous section), then a decrease in running speed could correspond to a reduction in the energetic barrier that separates the two basins of attraction (Figure 8F) or to another change that encourages neural activity to flip to the other attractor. Although the neural distance to the midpoint between clusters was variable across different track positions and appeared to dip slightly in the reward zone, this variability was no greater than expected by chance (p > 0.05; Figure S8).

DISCUSSION

Here we report that MEC representations are capable of remapping in VR environments without any changes to sensory cues (Diehl et al., 2017; Fyhn et al., 2007; Marozzi et al., 2015; Solstad

et al., 2008) or task demands (Boccara et al., 2019; Butler et al., 2019; Keene et al., 2016). Remapping events were coordinated across the MEC population, recruited many different cell types, and often comprised discrete and reversible switches between spatial maps that persisted over long (roughly 1–30 min) timescales. Remapping comprised diverse coding changes across neurons—including up to 3-fold variation in peak firing rate and 50% reconfiguration of spatial coding—and appeared to differentially affect different functional cell types. Although previous reports have found that faster running speed is associated with sharpened spatial tuning in single neurons (Bant et al., 2020; Hardcastle et al., 2017), we found that changes in running speed can also correspond to remapping events that produce large and persistent shifts in MEC coding. Finally, neural activity remapped by transitioning between geometrically aligned manifolds, enabling simple, linear decoding strategies. Our results suggest that a single population of MEC neurons can rapidly switch between multiple stable representations of a single environment.

Although neural activity often alternated between two stable spatial maps, we also observed sessions with more than two stable maps and sessions where activity remapped between spatially stable and unstable coding regimes. We largely focused on single-track sessions with two maps ("2-map sessions"), which allowed us to thoroughly characterize this common form of remapping and provided insight into how neural circuits might store and interpret multiple neural representations more generally. We found that a single decoder could accurately infer position across remapping events, indicating that MEC spatial

Neuron
Article

CellPress

representations contain changing contextual information and stable positional information that is segregated into orthogonal linear subspaces (Kaufman et al., 2014; Rule et al., 2020). Our finding that remapping comprised a translation in neural activity space provides a geometric interpretation of this possibility. This geometric arrangement could allow downstream neurons that are insensitive to the remapping dimension to extract consistent positional information regardless of the map, whereas neurons sensitive to this dimension could extract contextual information.

We further showed that a bistable ring attractor model network (Battaglia and Treves, 1998; Romani and Tsodyks, 2010; Samsonovich and McNaughton, 1997) does not produce aligned rings but that adding a sub-population of "shared neurons" was sufficient to recapitulate the experimentally observed alignment. Previous work has studied transitions between continuous attractors in the context of hippocampal and co-modular grid cell attractor networks (Monasson and Rosay, 2014, 2015; Spalla et al., 2019). Our modeling results differ by (1) considering the full MEC circuit (not only grid cells) and (2) interpreting each continuous attractor as representing the same external environment (not different environments). These results illustrate how our experimental findings might be interpreted and integrated into ongoing theoretical research on MEC and hippocampal circuits.

One feature of remapping between aligned manifolds is that it allows the network to preserve unchanging information while rapidly and synchronously updating changing contextual information. Our findings complement previous studies in the hippocampus (Rubin et al., 2015; Taxidis et al., 2020; Ziv et al., 2013) and MEC (Diehl et al., 2019) that found context delineation in a single environment over longer timescales (hours to days) via representational drift that is asynchronous across neurons. This type of remapping over longer timescales may relate to learning dynamics (Taxidis et al., 2020), grouping of temporally adjacent memories (Rubin et al., 2015), and building an episodic timeline for repeated interactions with a single environment (Diehl et al., 2019; Sheintuch et al., 2020; Ziv et al., 2013). The rapid remapping (under 1 min) we observed across the MEC may interact with these slower processes of representational drift to form a rich internal map of temporal and contextual episodes.

It is possible that the impoverished sensory experience of our virtual environment promoted remapping, perhaps leading the animal to infer multiple different contexts in the same VR track (Ravassard et al., 2013; Sanders et al., 2020). However, complete remapping of hippocampal spatial representations of a single environment was observed recently in freely behaving animals, suggesting that similar remapping can occur in more naturalistic settings (Sheintuch et al., 2020). It is also well established that MEC neurons can remap in freely behaving animals in response to changing sensory features (Diehl et al., 2017; Fyhn et al., 2007; Marozzi et al., 2015; Solstad et al., 2008) or task demands (Boccara et al., 2019; Butler et al., 2019; Keene et al., 2016). Our findings may represent a related form of MEC remapping driven by internal rather than external factors that likely interacts with previously established forms of remapping.

Finally, our finding that remapping events correlated with slower running speeds raises specific hypotheses regarding how behavioral state may drive widespread changes in MEC coding. Running speed can indicate task engagement—e.g., in our task, faster speeds lead the animal to "find rewards" more quickly—or of arousal more broadly (Bennett et al., 2013; Niell and Stryker, 2010; but see Vinck et al., 2015). Behavioral state changes related to running speed and arousal can alter neural processing in complex ways; for example, targeted neuromodulation of GABAergic interneurons modulates sensory neuron responses (Ferguson and Cardin, 2020). It is possible that similar or analogous biophysical mechanisms act on MEC neurons, and it will be of particular interest for future studies to dissect the mechanisms by which MEC neurons are commonly and differentially affected by running speed. Moreover, detailed consideration and tracking of multiple behavioral state variables (e.g., pupil dilation and whisking) will be needed to identify which specific variables control remapping in the navigation circuitry.

We find that the MEC has the capacity to remap in a rapid and reversible fashion, which could support a role of this circuit in dividing the unbroken stream of sensory features encountered during navigation into discrete contextual episodes. Further, our findings align with a larger body of emerging studies demonstrating that cortical activity is highly responsive to behavioral state changes (Jennings et al., 2019; Salay et al., 2018; Stringer et al., 2019). Our results suggest that these behavioral state changes may drive rapid, large-scale reconfiguration of internal representations of the external world in higher-order cortex.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- EXPERIMENTAL MODEL AND SUBJECT DETAILS
  - Mice
- METHOD DETAILS
  - Training and handling
  - *In vivo* survival surgeries
  - *In vivo* electrophysiological data collection
  - Virtual reality (VR) environment
  - Random foraging task
  - Histology and probe localization
  - Offline spike sorting
  - Behavioral data preprocessing
  - Running speed across maps
  - Population analysis and clustering model
  - Manifold alignment analysis
  - Manifold entanglement calculation
  - Distance to cluster calculations
  - Position decoding analysis
  - Spike waveform analysis
  - Tetrode recordings and analysis

- ○ Spatial information calculations
- ○ Rate remapping versus global remapping
- ○ Functional cell type analysis
- ○ Bistable ring attractor model
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ Statistics

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.neuron.2021.07.005.

AUTHOR CONTRIBUTIONS

I.I.C.L., conceptualization, methodology, formal analysis, investigation, writing – original draft, writing – review & editing, and visualization; A.H.W., conceptualization, methodology, formal analysis, writing – original draft, writing – review & editing, and visualization; M.G.C., conceptualization, methodology, writing – review & editing, and visualization; S.W.L., conceptualization, writing – review & editing, supervision, and funding; L.M.G., conceptualization, writing – original draft, writing – review & editing, supervision, and funding.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Allen Institute for Brain Science (2004). Mouse Brain Atlas. https://mouse.brain-map.org/static/atlas.

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. Biol. Cybern. 27, 77–87.

Bant, J.S., Hardcastle, K., Ocko, S.A., and Giocomo, L.M. (2020). Topography in the Bursting Dynamics of Entorhinal Neurons. Cell Rep. 30, 2349–2359.e7.

Barry, C., Hayman, R., Burgess, N., and Jeffery, K.J. (2007). Experience-dependent rescaling of entorhinal grids. Nat. Neurosci. 10, 682–684.

Battaglia, F.P., and Treves, A. (1998). Stable and rapid recurrent processing in realistic autoassociative memories. Neural Comput. 10, 431–450.

Ben-Yishai, R., Bar-Or, R.L., and Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. Proc. Natl. Acad. Sci. USA 92, 3844–3848.

Bennett, C., Arroyo, S., and Hestrin, S. (2013). Subthreshold mechanisms underlying state-dependent modulation of visual responses. Neuron 80, 350–357.

Boccara, C.N., Nardin, M., Stella, F., O'Neill, J., and Csicsvari, J. (2019). The entorhinal cognitive map is attracted to goals. Science 363, 1443–1447.

Bostock, E., Muller, R.U., and Kubie, J.L. (1991). Experience-dependent modifications of hippocampal place cell firing. Hippocampus 1, 193–205.

Burak, Y., and Fiete, I.R. (2009). Accurate path integration in continuous attractor network models of grid cells. PLoS Comput. Biol. 5, e1000291.

Butler, W.N., Hardcastle, K., and Giocomo, L.M. (2019). Remembered reward locations restructure entorhinal spatial maps. Science 363, 1447–1452.

Calhoun, A.J., Pillow, J.W., and Murthy, M. (2019). Unsupervised identification of the internal states that shape natural behavior. Nat. Neurosci. 22, 2040–2049.

Campbell, Malcolm G., Attinger, Alexander, Ocko, Samuel A., Ganguli, Surya, and Giocomo, Lisa (2020). Distinct algorithms for combining landmarks and path integration in medial entorhinal, visual and retrosplenial cortex. bioRxiv. https://doi.org/10.1101/2020.10.05.327106.

Campbell, M.G., Ocko, S.A., Mallory, C.S., Low, I.I.C., Ganguli, S., and Giocomo, L.M. (2018). Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. Nat. Neurosci. 21, 1096–1106.

Chatfield, C. (1984). The Analysis of Time Series: An Introduction (CRC Press).

Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Foster, J.D., Nuyujukian, P., Ryu, S.I., and Shenoy, K.V. (2012). Neural population dynamics during reaching. Nature 487, 51–56.

Couey, J.J., Witoelar, A., Zhang, S.-J., Zheng, K., Ye, J., Dunn, B., Czajkowski, R., Moser, M.-B., Moser, E.I., Roudi, Y., and Witter, M.P. (2013). Recurrent inhibitory circuitry as a mechanism for grid formation. Nat. Neurosci. 16, 318–324.

Diehl, G.W., Hon, O.J., Leutgeb, S., and Leutgeb, J.K. (2017). Grid and Nongrid Cells in Medial Entorhinal Cortex Represent Spatial Location and Environmental Features with Complementary Coding Schemes. Neuron 94, 83–92.e6.

Diehl, G.W., Hon, O.J., Leutgeb, S., and Leutgeb, J.K. (2019). Stability of medial entorhinal cortex representations over time. Hippocampus 29, 284–302.

Eckart, C., and Young, G. (1936). The approximation of one matrix by another of lower rank. Psychometrika 1, 211–218.

Elsayed, G.F., and Cunningham, J.P. (2017). Structure in neural population recordings: an expected byproduct of simpler phenomena? Nat. Neurosci. 20, 1310–1318.

Ferguson, K.A., and Cardin, J.A. (2020). Mechanisms underlying gain modulation in the cortex. Nat. Rev. Neurosci. 21, 80–92.

Fisher, N.I., and Lee, A.J. (1992). Regression Models for an Angular Response. Biometrics 48, 665–677.

Frank, L.M., Brown, E.N., and Wilson, M. (2000). Trajectory encoding in the hippocampus and entorhinal cortex. Neuron 27, 169–178.

Fuhs, M.C., and Touretzky, D.S. (2006). A spin glass model of path integration in rat medial entorhinal cortex. J. Neurosci. 26, 4266–4276.

Fyhn, M., Hafting, T., Treves, A., Moser, M.-B., and Moser, E.I. (2007). Hippocampal remapping and grid realignment in entorhinal cortex. Nature 446, 190–194.

Gil, M., Ancau, M., Schlesiger, M.I., Neitz, A., Allen, K., De Marco, R.J., and Monyer, H. (2018). Impaired path integration in mice with disrupted grid cell firing. Nat. Neurosci. 21, 81–91.

Gower, J.C., and Dijksterhuis, G.B. (2004). Procrustes Problems (OUP Oxford).

Guanella, A., Kiper, D., and Verschure, P. (2007). A model of grid cells based on a twisted torus topology. Int. J. Neural Syst. 17, 231–240.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E.I. (2005). Microstructure of a spatial map in the entorhinal cortex. Nature 436, 801–806.

Hardcastle, K., Maheswaranathan, N., Ganguli, S., and Giocomo, L.M. (2017). A Multiplexed, Heterogeneous, and Adaptive Code for Navigation in Medial Entorhinal Cortex. Neuron 94, 375–387.e7.

Neuron
Article

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array Programming with NumPy. Nature *585*, 357–362.

Hinman, J.R., Brandon, M.P., Climer, J.R., Chapman, G.W., and Hasselmo, M.E. (2016). Multiple Running Speed Signals in Medial Entorhinal Cortex. Neuron *91*, 666–679.

Høydal, Ø.A., Skytøen, E.R., Andersson, S.O., Moser, M.-B., and Moser, E.I. (2019). Object-vector coding in the medial entorhinal cortex. Nature *568*, 400–404.

Hulse, B.K., Lubenov, E.V., and Siapas, A.G. (2017). Brain State Dependence of Hippocampal Subthreshold Activity in Awake Mice. Cell Rep. *18*, 136–147.

Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. *9*, 90–95.

Jennings, J.H., Kim, C.K., Marshel, J.H., Raffiee, M., Ye, L., Quirin, S., Pak, S., Ramakrishnan, C., and Deisseroth, K. (2019). Interacting neural ensembles in orbitofrontal cortex for social and feeding behaviour. Nature *565*, 645–649.

Jezek, K., Henriksen, E.J., Treves, A., Moser, E.I., and Moser, M.-B. (2011). Theta-paced flickering between place-cell maps in the hippocampus. Nature *478*, 246–249.

Jones, E., Oliphant, T., and Peterson, P. (2001). SciPy: Open source scientific tools for Python. http://www.scipy.org.

Jun, J.J., Steinmetz, N.A., Siegle, J.H., Denman, D.J., Bauza, M., Barbarits, B., Lee, A.K., Anastassiou, C.A., Andrei, A., Aydın, Ç., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. Nature *551*, 232–236.

Kaufman, M.T., Churchland, M.M., Ryu, S.I., and Shenoy, K.V. (2014). Cortical activity in the null space: permitting preparation without movement. Nat. Neurosci. *17*, 440–448.

Keene, C.S., Bladon, J., McKenzie, S., Liu, C.D., O'Keefe, J., and Eichenbaum, H. (2016). Complementary Functional Organization of Neuronal Activity Patterns in the Perirhinal, Lateral Entorhinal, and Medial Entorhinal Cortices. J. Neurosci. *36*, 3660–3675.

Keinath, A.T., Nieto-Posadas, A., Robinson, J.C., and Brandon, M.P. (2020). DG-CA3 circuitry mediates hippocampal representations of latent information. Nat. Commun. *11*, 3026.

Kentros, C.G., Agnihotri, N.T., Streater, S., Hawkins, R.D., and Kandel, E.R. (2004). Increased attention to spatial context increases both place field stability and spatial memory. Neuron *42*, 283–295.

Kerr, K.M., Agster, K.L., Furtak, S.C., and Burwell, R.D. (2007). Functional neuroanatomy of the parahippocampal region: the lateral and medial entorhinal areas. Hippocampus *17*, 697–708.

Knierim, J.J., Kudrimoti, H.S., and McNaughton, B.L. (1998). Interactions between idiothetic cues and external landmarks in the control of place cells and head direction cells. J. Neurophysiol. *80*, 425–446.

Kolda, T.G., and Bader, B.W. (2009). Tensor Decompositions and Applications. SIAM Rev. *51*, 455–500.

Kriegeskorte, N., and Douglas, P.K. (2019). Interpreting encoding and decoding models. Curr. Opin. Neurobiol. *55*, 167–179.

Kropff, E., Carmichael, J.E., Moser, M.-B., and Moser, E.I. (2015). Speed cells in the medial entorhinal cortex. Nature *523*, 419–424.

Krupic, J., Bauza, M., Burton, S., Barry, C., and O'Keefe, J. (2015). Grid cell symmetry is shaped by environmental geometry. Nature *518*, 232–235.

Mallory, C.S., Hardcastle, K., Bant, J.S., and Giocomo, L.M. (2018). Grid scale drives the scale and long-term stability of place maps. Nat. Neurosci. *21*, 270–282.

Markus, E.J., Qin, Y.L., Leonard, B., Skaggs, W.E., McNaughton, B.L., and Barnes, C.A. (1995). Interactions between location and task affect the spatial and directional firing of hippocampal neurons. J. Neurosci. *15*, 7079–7094.

Marozzi, E., Ginzberg, L.L., Alenda, A., and Jeffery, K.J. (2015). Purely Translational Realignment in Grid Cell Firing Patterns Following Nonmetric Context Change. Cereb. Cortex *25*, 4619–4627.

McNaughton, B.L., Battaglia, F.P., Jensen, O., Moser, E.I., and Moser, M.-B. (2006). Path integration and the neural basis of the 'cognitive map'. Nat. Rev. Neurosci. *7*, 663–678.

Miao, C., Cao, Q., Ito, H.T., Yamahachi, H., Witter, M.P., Moser, M.-B., and Moser, E.I. (2015). Hippocampal Remapping after Partial Inactivation of the Medial Entorhinal Cortex. Neuron *88*, 590–603.

Moita, M.A.P., Rosis, S., Zhou, Y., LeDoux, J.E., and Blair, H.T. (2004). Putting fear in its place: remapping of hippocampal place cells during fear conditioning. J. Neurosci. *24*, 7015–7023.

Monasson, R., and Rosay, S. (2014). Crosstalk and transitions between multiple spatial maps in an attractor neural network model of the hippocampus: collective motion of the activity. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *89*, 032803.

Monasson, R., and Rosay, S. (2015). Transitions between Spatial Attractors in Place-Cell Models. Phys. Rev. Lett. *115*, 098101.

Muller, R.U., and Kubie, J.L. (1987). The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. J. Neurosci. *7*, 1951–1968.

Munn, R.G.K., Mallory, C.S., Hardcastle, K., Chetkovich, D.M., and Giocomo, L.M. (2020). Entorhinal velocity signals reflect environmental geometry. Nat. Neurosci. *23*, 239–251.

Niell, C.M., and Stryker, M.P. (2010). Modulation of visual responses by behavioral state in mouse visual cortex. Neuron *65*, 472–479.

O'Keefe, J., and Conway, D.H. (1978). Hippocampal place units in the freely moving rat: why they fire where they fire. Exp. Brain Res. *31*, 573–590.

Ocko, S.A., Hardcastle, K., Giocomo, L.M., and Ganguli, S. (2018). Emergent elasticity in the neural code for space. Proc. Natl. Acad. Sci. USA *115*, E11798–E11806.

Pachitariu, M., Steinmetz, N.A., Kadir, S.N., Carandini, M., and Harris, K.D. (2016). Fast and accurate spike sorting of high-channel count probes with KiloSort. In Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds. (Curran Associates), pp. 4448–4456.

Pastoll, H., Solanka, L., van Rossum, M.C.W., and Nolan, M.F. (2013). Feedback inhibition enables θ-nested γ oscillations and grid firing fields. Neuron *77*, 141–154.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

Pewsey, A., and García-Portugués, E. (2020). Recent advances in directional statistics. arXiv, arXiv:2005.06889 https://arxiv.org/abs/2005.06889.

Presnell, B., Morrison, S.P., and Littell, R.C. (1998). Projected Multivariate Linear Models for Directional Data. J. Am. Stat. Assoc. *93*, 1068–1077.

Ravassard, P., Kees, A., Willers, B., Ho, D., Aharoni, D.A., Cushman, J., Aghajan, Z.M., and Mehta, M.R. (2013). Multisensory control of hippocampal spatiotemporal selectivity. Science *340*, 1342–1346.

Romani, S., and Tsodyks, M. (2010). Continuous attractors with morphed/correlated maps. PLoS Comput. Biol. *6*, e1000869.

Roudi, Y., and Treves, A. (2008). Representing where along with what information in a model of a cortical patch. PLoS Comput. Biol. *4*, e1000012.

Rubin, A., Geva, N., Sheintuch, L., and Ziv, Y. (2015). Hippocampal ensemble dynamics timestamp events in long-term memory. eLife *4*, e12247.

Rule, M.E., Loback, A.R., Raman, D.V., Driscoll, L.N., Harvey, C.D., and O'Leary, T. (2020). Stable task information from an unstable neural population. eLife *9*, e51121.

Salay, L.D., Ishiko, N., and Huberman, A.D. (2018). A midline thalamic circuit determines reactions to visual threat. Nature *557*, 183–189.

Samsonovich, A., and McNaughton, B.L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. J. Neurosci. *17*, 5900–5920.

Sanders, H., Wilson, M.A., and Gershman, S.J. (2020). Hippocampal remapping as hidden state inference. eLife *9*, e51140.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B.L., Witter, M.P., Moser, M.-B., and Moser, E.I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. Science *312*, 758–762.

Savelli, F., Yoganarasimha, D., and Knierim, J.J. (2008). Influence of boundary removal on the spatial representations of the medial entorhinal cortex. Hippocampus *18*, 1270–1282.

Schönemann, P.H. (1966). A generalized solution of the orthogonal procrustes problem. Psychometrika *31*, 1–10.

Seely, J.S., Kaufman, M.T., Ryu, S.I., Shenoy, K.V., Cunningham, J.P., and Churchland, M.M. (2016). Tensor analysis reveals distinct population structure that parallels the different computational roles of areas M1 and V1. PLoS Comput. Biol. *12*, e1005164.

Seung, H.S. (1996). How the brain keeps the eyes still. Proc. Natl. Acad. Sci. USA *93*, 13339–13344.

Sheintuch, L., Geva, N., Baumer, H., Rechavi, Y., Rubin, A., and Ziv, Y. (2020). Multiple Maps of the Same Spatial Context Can Stably Coexist in the Mouse Hippocampus. Curr. Biol. *30*, 1467–1476.e6.

Shlens, J. (2005). A tutorial on Principal Components Analysis. arXiv, arXiv:1404.1100 https://arxiv.org/abs/1404.1100.

Sikaroudi, A.E., and Park, C. (2019). A mixture of linear-linear regression models for a linear-circular regression. Stat. Modelling *21*, 220–243.

Singh, A.P., and Gordon, G.J. (2008). A Unified View of Matrix Factorization Models. In Machine Learning and Knowledge Discovery in Databases, Walter Daelemans, Bart Goethals, and Katharina Morik, eds. (Springer), pp. 358–373.

Skaggs, W.E., Knierim, J.J., Kudrimoti, H.S., and McNaughton, B.L. (1995). A model of the neural basis of the rat's sense of direction. Adv. Neural Inf. Process. Syst. *7*, 173–180.

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

Solstad, T., Boccara, C.N., Kropff, E., Moser, M.-B., and Moser, E.I. (2008). Representation of geometric borders in the entorhinal cortex. Science *322*, 1865–1868.

Spalla, D., Dubreuil, A., Rosay, S., Monasson, R., and Treves, A. (2019). Can grid cell ensembles represent multiple spaces? Neural Comput. *31*, 2324–2347.

Stringer, S.M., Rolls, E.T., and Trappenberg, T.P. (2004). Self-organising continuous attractor networks with multiple activity packets, and the representation of space. Neural Netw. *17*, 5–27.

Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C.B., Carandini, M., and Harris, K.D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. Science *364*, 255.

Taxidis, J., Pnevmatikakis, E.A., Dorian, C.C., Mylavarapu, A.L., Arora, J.S., Samadian, K.D., Hoffberg, E.A., and Golshani, P. (2020). Differential Emergence and Stability of Sensory and Temporal Representations in Context-Specific Hippocampal Sequences. Neuron *108*, 984–998.e9.

Udell, M., Horn, C., Zadeh, R., and Boyd, S. (2016). Generalized Low Rank Models. arXiv, arXiv:1410.0342 https://arxiv.org/abs/1410.0342.

Vinck, M., Batista-Brito, R., Knoblich, U., and Cardin, J.A. (2015). Arousal and locomotion make distinct contributions to cortical activity patterns and visual encoding. Neuron *86*, 740–754.

Williams, A.H., Kim, T.H., Wang, F., Vyas, S., Ryu, S.I., Shenoy, K.V., Schnitzer, M., Kolda, T.G., and Ganguli, S. (2018). Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. Neuron *98*, 1099–1115.e8.

Wold, S. (1978). Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models. Technometrics *20*, 397–405.

Wood, E.R., Dudchenko, P.A., Robitsek, R.J., and Eichenbaum, H. (2000). Hippocampal neurons encode information about different types of memory episodes occurring in the same location. Neuron *27*, 623–633.

Yu, B.M., Cunningham, J.P., Santhanam, G., Ryu, S.I., Shenoy, K.V., and Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. J. Neurophysiol. *102*, 614–635.

Ziv, Y., Burns, L.D., Cocker, E.D., Hamel, E.O., Ghosh, K.K., Kitch, L.J., El Gamal, A., and Schnitzer, M.J. (2013). Long-term dynamics of CA1 hippocampal place codes. Nat. Neurosci. *16*, 264–266.

Neuron
Article

🏛 **CellPress**

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Raw and analyzed data	This paper	Mendeley Data: https://doi.org/10.17632/hntn6m2pgk.1
Experimental models: Organisms/strains		
Mouse: C57BL/6	The Jackson Laboratories	000664
Software and algorithms		
SciPy ecosystem of open-source Python libraries (numpy, matplotlib, scipy, etc.)	Jones et al., 2001; Hunter, 2007; Harris et al., 2020	https://www.scipy.org/
scikit-learn	Pedregosa et al., 2011	https://scikit-learn.org/
MATLAB	MathWorks	https://www.mathworks.com/products/matlab.html
Kilosort2		https://github.com/MouseLand/Kilosort
Other		
Phase 3B Neuropixels 1.0 silicon probes	Jun et al., 2017	https://www.neuropixels.org/probe
Original code	This paper	Zenodo: https://doi.org/10.5281/zenodo.5062491

RESOURCE AVAILABILITY

Lead contact
Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Lisa M. Giocomo (giocomo@stanford.edu).

Materials availability
This study did not generate new unique reagents.

Data and code availability
- All data required to reproduce the paper figures have been deposited at Mendeley and are publicly available as of the date of publication. DOIs are listed in the Key resources table.
- All original code has been deposited at Zenodo and is publicly available as of the date of publication. DOIs are listed in the Key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

EXPERIMENTAL MODEL AND SUBJECT DETAILS

Mice
All techniques were approved by the Institutional Animal Care and Use Committee at Stanford University School of Medicine. Recordings were made from 17 C57BL/6 mice aged 4 weeks to 3.5 months at the time of first surgery (15.7–35 g). All mice were female except litters 5 and 9 (6 mice), which were male. Mice were group housed with same-sex littermates, and in one case with the dam (2a, b, c with 3a), unless separation was required due to water restriction, aggression, or disturbance of prep site. Mice were housed in transparent cages on a 12-h light-dark cycle and experiments were performed during the light phase.

METHOD DETAILS

Training and handling
Mice were handled at least every 2 days following headbar implantation and given an in-cage running wheel. Starting 1 day after headbar implantation, same-sex mice were placed daily in a large (100x100cm), communal environment with enrichment objects including a running wheel, Lego tower, textured floor tape, and crushed chocolate cheerios for between 15 mins and 2 hours.

Mice were monitored for aggression and separated as needed. Mice were given free access to water until 3 days after headbar implantation, after which they were water restricted to 1mL of water per day and weighed daily to ensure a body weight of > 80% of their starting weight.

After > 1 day of water restriction, mice were acclimated to head fixation and trained to drink water from a custom lickport for 10-20 mins over 2 days. Mice were then trained to run on the virtual random forage task (described below) starting with a reward probability of 0.1/cm (essentially one reward per 50 cm), for gradually decreasing reward probability and gradually increasing session length as behavior improved. Mice were trained on the exact track(s) that they were recorded in, either cue poor (litters 1, 2, and 3), cue rich (litters 6, 7, 9 and 10), or both (litters 4 and 5). Mice trained on both tracks were exposed to each track in a counterbalanced fashion, initially alternating tracks over days and ultimately alternating the order of presentation as mice improved sufficiently to run 2 or more sessions per training day. Training continued at least until mice ran > 300 trials in 2 hours and demonstrated proficient licking from the lickport in the reward zone; training was sometimes extended in order to stagger recording periods (7 days to 7 weeks; mean ± SEM: 23 ± 3 days; 7 mice never learned the task).

In vivo survival surgeries

For all surgeries, anesthesia was induced with isoflurane (4%; maintained at 0.5%–1.5%) followed by injection of buprenorphrine (0.05–0.1 mg/kg). Animals were injected with baytril (10 mg/kg) and rimadyl (5 mg/kg) immediately following each surgery and for 3 days afterward. In the first surgery, animals were implanted with a custom-built metal headbar containing two holes for head fixation, as well as with a jewelers' screw with an attached gold pin, to be used as a ground. The craniotomy sites were exposed and marked during headbar implantation and the surface of the skull was coated in metabond. After completion of training, a second surgery was performed to make bilateral craniotomies ($\sim$200μm diameter) at 3.7–3.95mm posterior and 3.3–3.4mm lateral to bregma. A small plastic well was implanted around each craniotomy and affixed with metabond. Craniotomy sites were covered with a drop of sterile saline and with silicone elastomer (Kwik-sil, WPI) in between surgery and recordings.

In vivo electrophysiological data collection

All recordings were performed at least 16-h after craniotomy surgery. Mice were head-fixed on the VR recording rig. Craniotomy site was exposed and rinsed with saline—occasionally dura was re-nicked or debris removed using a syringe tip. Recordings were performed using Phase 3B Neuropixels 1.0 silicon probes (Jun et al., 2017) with 384 active recording sites (out of 960 total) along the bottom $\sim$4 mm of a $\sim$10 mm shank (70 μm wide shank diameter, 24 μm thick, 25 μm electrode spacing), and reference and ground shorted together. The probe was positioned over the craniotomy site at 8–14° from vertical and targeted to $\sim$50–300 μm anterior of the transverse sinus using a micromanipulator. On consecutive recording days, probes were targeted medial or lateral of previous recording sites as permitted by the craniotomy and/or probe angle was varied to ensure that we were recording from a new population of cells each day (Figure S1). The reference electrode was then connected to a gold ground pin implanted in the skull. The probe was advanced slowly ($\sim$4–10 μm/s) into the brain until it encountered resistance or until activity quieted on channels near the probe tip, then retracted 100–500μm and allowed to sit for at least 30 mins prior to recording. While the probe was implanted, the craniotomy site was covered with sterile saline and silicon oil. Signals were sampled at 30 kHz with gain = 200 (7.63 μV/bit at 10 bit resolution) in the action potential band, digitized with a CMOS amplifier and multiplexer built into the electrode array, then written to disk using SpikeGLX software.

Virtual reality (VR) environment

The VR recording set-up was nearly identical to the set-up in Campbell et al. (2018). Head-fixed mice ran on a 15.2-cm-diameter foam roller (ethylene vinyl acetate) constrained to rotate about one axis. The cylinder's rotation was measured by a high-resolution quadrature encoder (Yumo, 1024 P/R) and processed by a microcontroller (Arduino UNO). The virtual environment was generated using commercial software (Unity 3D) and updated according to the motion signal. VR position traces were synchronized to recording traces on each frame of the virtual scene. The virtual scene was displayed on three 24-inch monitors surrounding the mouse. The gain of the linear transformation from ball rotation to translation along the virtual track was calibrated so that the virtual track was 4 m long. At the end of the track, the mouse was teleported seamlessly back to the start to begin the next trial, such that the track was seemingly infinite (all visual cues were repeated and visible into the distance as the mouse approached the track end).

Cue rich tracks consisted of 5 towers of different heights, widths, and patterns (black and white, neutral luminance), placed at 80 cm intervals starting at 0 cm and a black and white checkerboard on the floor for optic flow (see schematic, Figure 1B, top, and screenshot, Figure S1A). Cue poor tracks contained 2 towers of different patterns placed at 0 and 200 cm and a white to black horizontal sinusoidal pattern on the floor (see schematic, Figure 1B, bottom, and screenshot, Figure S1B). Both tracks had uniform gray walls and sky beyond the towers. For mice that experienced a single track, recording sessions consisted of 57–450 trials (mean ± SEM: 328 ± 14 trials). For mice that experienced both tracks, each track was presented in a block of 75–100 trials with $\sim$1 min of darkness in between tracks (Figure S8). Blocks alternated between cue rich and poor—each track was presented twice (barring rare cases when the mouse failed to complete the session) and which track was presented first was alternated on each day.

In a subset of cue rich, single-track sessions (N = 21 sessions in 5 mice) we appended 5 gain manipulation trials to the end of each session in which the mouse had to run twice as far to traverse the same distance along the virtual track, as in Campbell et al. (2018).

Random foraging task

In both cue rich and cue poor tracks, visually marked reward zones appeared at a probability of 0.01–0.001 per cm, titrated to mouse performance, within the middle 300 cm of the track and at least 50 cm apart. Reward zones were 50 cm long and track-width, were patterned with a black and white diamond checkerboard, and hovered slightly above the floor (Figure S1). Upon entering the reward zone, animals could request water by licking and breaking an infrared beam at the mouth of the lickport; if not requested, water was dispensed automatically at the center of the zone. For mice 1c, 4a, and 4b for some recording sessions there were between 1-5 probe trials every 10 trials in which water was only dispensed if requested in the reward zone (no automatic dispensation). Upon water dispensation (or next trial start for missed probe trials), the current reward zone disappeared and the next zone became visible. Water rewards (~1.5 μL) were delivered using a solenoid (Cole-Parmer) triggered from the virtual environment software, generating an audible click with water delivery. Licks were recorded as new breaks in the lickport infrared beam and were processed by a microcontroller (Arduino UNO).

Histology and probe localization

Before each implantation, probes were dipped in fixable lipophilic dye (1mM DiI, DiO, DiD, Thermo Fisher) 10 times at 10 s intervals. Within 7 days of the first probe insertion, mice were killed with an overdose of pentobarbital and transcardially perfused with phosphate-buffered saline (PBS) followed by 4% paraformaldehyde. Brains were extracted and stored in 4% paraformaldehyde for at least 24 h before transfer to 30% sucrose in PBS. Brains were then rapidly frozen, cut into 30-μm sagittal sections with a cryostat, mounted and stained with DAPI. Histological sections were examined and the location of the probe tip and entry into the dorsal MEC for each recording were determined based on the reference Allen Brain Atlas (Allen Institute for Brain Science, 2004; Figure S1). The location of each recording site along the line delineated by the probe tip and entry point was then determined based on each site's distance from the probe tip. Only cells within MEC, again based on the reference Allen Brain Atlas (Allen Institute for Brain Science, 2004), were included for analysis (Figure S1). In all cases, "depth" reported is the ventral distance from the dorsal boundary of MEC in the medial section where the probe entered MEC.

Offline spike sorting

Electrophysiological recordings were common-average referenced to the median across channels and high-pass filtered above 150 Hz. Automatic spike sorting was then performed using Kilosort2, a high-throughput spike sorting algorithm that identifies clusters in neural data and is designed to track small amounts of neural drift over time (open source software by Marius Pachitariu, Nick Steinmetz, and Jennifer Colonell; https://github.com/MouseLand/Kilosort; see also Kilosort1; Pachitariu et al., 2016). After automatic spike-sorting, all clusters with peak-to-peak amplitude over noise ratio < 3 (with noise defined as the standard deviation of voltage traces in a 10ms window preceding detected spike times), total number of spikes < 100, and repeated refractory period violations (0-1 ms autocorrelegram bin > 20% of maximum autocorrelation) were excluded. All remaining clusters were manually examined and labeled as "good" (i.e., stable and likely belonging to a single, well-isolated neural unit), "MUA" (i.e., likely to represent multi-unit activity), or "noise." Only well-isolated "good" units from within MEC (barring Figures S1I and S1J, which were non-MEC units) with greater than 400 spikes were included for analysis in this paper (Figure 1G). Sessions with fewer than 10 cells meeting these criteria were excluded.

Behavioral data preprocessing

On each frame of the virtual reality scene, the virtual position and time stamps were recorded and a synchronizing TTL pulse was sent from an Arduino UNO to the electrophysiological recording equipment. These pulses were recorded in SpikeGLX using an auxiliary National Instruments data acquisition card (NI PXIe-6341 with NI BNC-2110). The location of each reward zone, time of each lick, and time of each reward dispensation were also recorded. Thus all time stamps and behavioral factors were synchronized to the neurophysiological data. Time stamps were adjusted to start at 0 and all behavioral data was interpolated to convert the variable VR frame rate to a constant frame rate of 50Hz. As the track was effectively circular and 400 cm long, recorded positions less than 0 or greater than 400 cm were converted to the appropriate position on the circular track (e.g., a recorded position of 404 cm would be converted to 4 cm and a recorded position of −4 cm would be converted to 396 cm). Trial transitions were identified as time points where the difference in position across time bins was less than −100 cm (i.e., a transition from ~400 cm to ~0 cm) and a trial number was accordingly assigned to each time point.

Running speed for each time point was computed by calculating the difference in position between that time point and the previous, divided by the framerate (speed at the first time point was assigned to be equal to that at the second time point). Speeds greater than 150 cm/s or less than −5 cm/s were removed. Speed was then interpolated to fill removed time points and smoothed with a Gaussian filter (standard deviation 0.2 time bins). For all analyses except lick and reward zone analyses (Figures 1C–1F), stationary time bins (speed < 2 cm/s) were excluded.

Running speed across maps

Because the k-means cluster labels are arbitrarily assigned, we designated the map with the slower overall average running speed as map 1 for consistency across analyses. To assess possible differences in running speed across maps, we computed the fractional difference in running speed across the k-means identified maps in all 2-map sessions. To control for any systematic changes in

running speed over time in the session, we calculated the average running speed for each block of adjacent trials from the same map and compared it to the average running speed of the subsequent block of trials from the other map. We compared the resulting distribution of fractional difference in running speed to a shuffle control in which running speed was randomly shuffled across time bins (re-designating map 1 according to whichever shuffled map had a slower overall average running speed; 1000 shuffles per session) (Figure S8J).

Population analysis and clustering model

The 1D track was divided into 5 cm position bins (total of 80 bins). On each traversal of the track, the empirical firing rate of each neuron—i.e., number of spikes divided by time elapsed—was computed for every position bin. We then smoothed the firing rate traces with a Gaussian filter (standard deviation 5 cm). For each session this resulted in a 3-dimensional array of raw firing rates, with dimensions corresponding to trials, positions, and neurons.

Because these raw firing rates varied widely across neurons, we rescaled them so that the peak firing rate was commensurate across cells. Similar normalization steps or variance-stabilizing transformations have been used in previous population analyses of neural data (Churchland et al., 2012; Williams et al., 2018; Yu et al., 2009), to prevent neurons with high firing rates from washing out low firing rate neurons. Here, we normalized firing rates by first clipping the maximum firing rate of each neuron at its 90th percentile (to exclude large outliers), and then re-scaling each neuron's firing rate to range between zero and one. That is, if $\tilde{X}_{ijk}$ denotes the clipped firing rate on trial i, position bin j, and neuron k, then the normalized firing rate was computed as:

$$X_{ijk} = \frac{\tilde{X}_{ijk} - \min\left(\tilde{X}_{::k}\right)}{\max\left(\tilde{X}_{::k}\right) - \min\left(\tilde{X}_{::k}\right)} \qquad \text{(Equation 1)}$$

The $\max(\cdot)$ and $\min(\cdot)$ operations (as well as the 90th percentile clipping operation) are applied on a neuron-by-neuron basis, pooling observations across all trials and timebins. For double-track sessions (which contained 2 blocks of cue rich trials and 2 blocks of cue poor trials, in alternating order) we added an additional per-neuron correction factor to account for drift in firing rates: the mean normalized firing rate for each neuron (across all trials and position bins) was subtracted within each block of trials, and the result was renormalized to values between zero and one, as above.

On each trial, MEC's representation of position is summarized by a matrix, denoted $X_{i::}$ for trial i, with rows and columns respectively corresponding to position bins and neurons. A simple measure of similarity between two trials, indexed by i and i', is given by the Pearson correlation between the vectors $vec(X_{i::})$ and $vec(X_{i'::})$. Network-wide trial-by-trial similarity matrices (as in Figures 2, 3, S1, S2, S4, and S5) were found by computing this correlation across all pairs of trials.

One can view X_{ijk} as a $I \times J \times K$ array, or tensor, of normalized firing rates. As before the index variables i, j, and k, respectively represent trials, position bins, and neurons. Now consider the following low-rank matrix factorization model (Singh and Gordon, 2008; Udell et al., 2016) of these data:

$$\hat{X}_{ijk} = \sum_{r=1}^{R} U_i^{(r)} V_{jk}^{(r)} \qquad \text{(Equation 2)}$$

Where $R < \min(I, J, K)$ denotes the number of model components, or the model rank. Equation 2 represents an approximate factorization of the $I \times JK$ matricization or tensor unfolding of the data array (Kolda and Bader, 2009; Seely et al., 2016). We will see that k-means clustering arises as a special case of this model, and in this special case R represents the number of clusters (i.e., the number of spatial maps).

The parameters $U_i^{(r)}$ and $V_{jk}^{(r)}$ are optimized according to a least-squares criterion:

$$\text{minimize} \sum_{ijk} \left(\sum_{r=1}^{R} U_i^{(r)} V_{jk}^{(r)} - X_{ijk} \right)^2 \qquad \text{(Equation 3)}$$

It is well-known that a rank-R truncated singular value decomposition (SVD) provides a solution to this optimization problem (Eckart and Young, 1936). Further, the solution provided by truncated SVD is closely related to Principal Components Analysis (PCA)—indeed, these two methods are identical for the case of mean-centered data (see, e.g., Shlens, 2005). Since the normalized firing rate array X_{ijk} in Equation 3 is not mean-centered, we refer to this initial model as "uncentered PCA." We use the uncentered coefficient-of-determination (uncentered R^2) as a normalized measure of model performance associated with Equation 3.

The k-means clustering model incorporates an additional constraint into the uncentered PCA model. Specifically, k-means seeks to minimize Equation 3,

$$\text{subject to } \sum_r U_i^{(r)} = 1 \quad \text{for all } i \in \{1, ..., I\}, \quad \text{and}$$

$$U_i^{(r)} \in \{0, 1\} \quad \text{for all } i, r \in \{1, ..., I\} \times \{1, ..., R\}. \qquad \text{(Equation 4)}$$

Neuron
Article

CellPress

Thus, if we view $U_i^{(r)}$ as the elements of an $R \times I$ matrix, the rows of this matrix are constrained to be standard Cartesian basis vectors of R-dimensional space ("one-hot vectors"). Each of these vectors specifies the cluster assignment label for every trial (see Figure 3A for a schematic illustration for the $R = 2$ case). Further, we can interpret $V_{jk}^{(r)}$ as elements of an $R \times J \times K$ array. For a fixed cluster index r, the remaining elements form a $J \times K$ matrix, called a "slice" of the original array (Kolda and Bader, 2009). Each slice corresponds to a cluster centroid, which we may interpret as a spatial map—the columns are J-dimensional vectors holding the spatial tuning curves for every neuron, so different slices correspond to different sets of spatial tuning curves (i.e., different spatial maps).

This connection between k-means clustering and other matrix factorization models is well-known and expanded upon in detail by (Singh and Gordon, 2008; Udell et al., 2016). We exploit this connection to assess the k-means model, which is more constrained than uncentered PCA (i.e., truncated SVD) in that each row of $U_i^{(r)}$ is constrained to be a one-hot vector as opposed to an arbitrary vector. Intuitively, this allows us to interpret each trial as belonging to one of R types, as opposed to a linear combination of them. The fact that the more restrictive k-means model performs nearly as well as uncentered PCA gives credence to the multiple-map interpretation. To compare these two models we used a randomized cross-validation procedure in which 10% of the data, representing the validation set, were censored in a speckled holdout pattern (Williams et al., 2018; Wold, 1978). Ten randomized replicates were performed for all models. For the case of $R = 2$ components, we often observed similar performance (measured by the uncentered R^2 averaged over validation sets) between uncentered PCA and k-means for all sessions (Figure 3C).

Further, we compared the test performance of k-means on "shuffled" datasets (Figures 3C, right, and 3D). Firing rates from a behavioral session were shuffled by applying a random rotation (i.e., an orthogonal linear transformation) to X_{ijk} across trials. That is, we sample a random rotation matrix $Q_{ii'}$ and define

$$X_{ijk}^* = \sum_{i'=1}^{I} Q_{ii'} X_{i'jk}$$

as the new shuffled dataset, which is substituted into the objective function defined in Equation 3. This form of shuffling preserves many features of the data, including the overall norm of the data and correlations between neurons and position bins. However, it destroys the sparsity pattern on $U_i^{(r)}$ which is imposed by the k-means model. This procedure is similar in spirit to methods proposed by Elsayed and Cunningham (2017).

Sessions that were well-approximated by the k-means model with $R = 2$ clusters were classified as "two-map" sessions (Figure 3D). We required that the performance gap (measured by the uncentered R^2 averaged over validation sets) between k-means and uncentered PCA and be less than 70% relative to the shuffle control. Further, we required an uncentered R^2 of at least 0.63 for the k-means model with $R = 2$ maps. Sessions not meeting these criteria sometimes displayed more than two maps (see Figure S5), long periods of unstable coding (see Figures S1 and S2), or little to no remapping at all (Figure S2). For all k-means analyses, we ran the clustering model at least 100 times on all neural data from each session to account for model variability, keeping the model with the best fit to the data.

Manifold alignment analysis

We used standard Procrustes analysis methods (Gower and Dijksterhuis, 2004) to assess the degree to which the two ring manifolds, representing spatial maps, were aligned in neural firing rate space. Recall that the k-means centroids provide an estimate of each spatial map—in this case, we restrict our focus to $R = 2$ maps, so the two maps are given by $V_{jk}^{(1)}$ and $V_{jk}^{(2)}$. Geometrically, these maps are represented as J points embedded in a K-dimensional space (recall that J denotes the number of position bins and K is the number of simultaneously recorded neurons). Procrustes Analysis begins by centering each of these manifolds at the origin and rescaling them to have unit norm. Let $\overline{V}_{jk}^{(1)}$ and $\overline{V}_{jk}^{(2)}$ denote the maps after these preprocessing steps have been applied, i.e.,

$$\overline{V}_{jk}^{(1)} = \frac{V_{jk}^{(1)} - \frac{1}{J}\sum_j V_{jk}^{(1)}}{\left(\sum_{jk}\left(V_{jk}^{(1)} - \frac{1}{J}\sum_j V_{jk}^{(1)}\right)^2\right)^{1/2}}$$

$$\overline{V}_{jk}^{(2)} = \frac{V_{jk}^{(2)} - \frac{1}{J}\sum_j V_{jk}^{(2)}}{\left(\sum_{jk}\left(V_{jk}^{(2)} - \frac{1}{J}\sum_j V_{jk}^{(2)}\right)^2\right)^{1/2}}$$

Since position bin j in map 1 and position bin j in map 2 correspond to the same location on the track, we consider the root-mean-squared-error (RMSE) between these centered and rescaled maps as the empirically "observed" alignment score (reported in Figure 5J):

$$\left(\frac{1}{JK}\sum_{jk}\left(\overline{V}_{jk}^{(1)} - \overline{V}_{jk}^2\right)^2\right)^{1/2}$$

The central step of Procrustes analysis is to find the optimal rotation matrix that aligns these two point clouds. That is, we wish to find the matrix Q_{kk}^* that solves the following optimization problem:

$$\text{minimize} \sum_{jk} \left(\overline{V}_{jk}^{(1)} - \sum_{k'} Q_{kk'} \overline{V}_{jk'}^{(2)} \right)^2$$

$$\text{subject to} \quad \sum_{k} Q_{\ell k} Q_{kp} = \delta_{\ell p} \quad \text{for all } \ell, p \in \{1, \ldots, K\} \times \{1, \ldots, K\}$$

This optimization problem admits a closed form solution that is expressed in terms of the singular value decomposition of the $K \times K$ matrix $\sum_j \overline{V}_{jk}^{(1)} \overline{V}_{jk'}^{(2)}$ (Schönemann, 1966). See Gower and Dijksterhuis (2004) for further background material. In Figure 7J, we report the RMSE compared to the optimal rotation (misalignment score = 0) and a random rotation matrix ("shuffle").

Manifold entanglement calculation

We quantified the entanglement of a manifold as the maximum ratio of intrinsic distance (i.e., distance along the manifold) to extrinsic distance (i.e., Euclidean distance in K-dimensional space) between any two points on the manifold. Concretely, the extrinsic distance between two points corresponding to position bins j and j' was computed as:

$$d_{jj'} \left(\sum_k \left(V_{jk} - V_{j'k} \right)^2 \right)^{1/2}$$

The intrinsic distance, $D_{jj'}$, was the sum of extrinsic distances along the shortest path from j to j' (see diagram in Figure 7D). Depending on whether one travels clockwise or counterclockwise along the ring, there are two paths connecting any pair of points—the intrinsic distance is given by whichever path is shorter. The triangle inequality implies that $D_{jj'} \geq d_{jj'}$ for every pair of points along the manifold. When the ratio $D_{jj'}/d_{jj'}$ is large, this implies that the pair of points (j, j') on the manifold are close together in neural firing rate space despite encoding very different positions on the VR track—speaking loosely, we say these points are "entangled." Conversely, when $D_{jj'}/d_{jj'}$ is small, the two points are far apart in firing rate space in proportion to their encoded positions on the VR track. Thus, a measure of manifold entanglement is given by $\max[D_{jj'}/d_{jj'}]$, which measures the worst-case entanglement over the full manifold (the $\max[\cdot]$ operator is understood to be taken over all pairs of position bins, j and j'). Noting that this raw entanglement score is upper bounded by $B = \max[D_{jj'}]/\min[d_{jj'}]$ and lower bounded by one, we can normalize the tangling metric to range between zero (no entanglement) and one (high entanglement) as follows:

$$\frac{\max[D_{jj'}/d_{jj'}] - 1}{B - 1}$$

We report this normalized entanglement metric in our results.

Distance to cluster calculations

After fitting the k-means model and obtaining two centroids, $V_{jk}^{(1)}$ and $V_{jk}^{(2)}$, we can quantify how close network activity is to each centroid on a trial-by-trial or neuron-by-neuron basis. In each case we project the activity onto a one-dimensional space where −1 corresponds to one centroid and +1 corresponds to the other centroid. That is, for each trial i, we compute

$$P_i = \frac{\sum_{jk} \left(2X_{ijk} - \left(V_{jk}^{(1)} + V_{jk}^{(2)} \right) \right) \left(V_{jk}^{(1)} + V_{jk}^{(2)} \right)}{\sum_{jk} \left(V_{jk}^{(1)} + V_{jk}^{(2)} \right)^2}$$

Note that $P_i = 1$ when $X_{ijk} = V_{jk}^{(1)}$ and $P_i = -1$ when $X_{ijk} = V_{jk}^{(2)}$. Further, if the network activity on trial i is at the midpoint, then $X_{ijk} = (V_{jk}^{(1)} + V_{jk}^{(2)})/2$ and so $P_i = 0$.

We can compute an analogous statistic for each combination of trial i and position bin j:

$$P_{ij} = \frac{\sum_k \left(2X_{ijk} - \left(V_{jk}^{(1)} + V_{jk}^{(2)} \right) \right) \left(V_{jk}^{(1)} - V_{jk}^{(2)} \right)}{\sum_k \left(V_{jk}^{(1)} - V_{jk}^{(2)} \right)^2}$$

Likewise, we can compute for each combination of trial i and neuron k:

$$P_{ik} = \frac{\sum_j \left(2X_{ijk} - \left(V_{jk}^{(1)} + V_{jk}^{(2)} \right) \right) \left(V_{jk}^{(1)} - V_{jk}^{(2)} \right)}{\sum_k \left(V_{jk}^{(1)} - V_{jk}^{(2)} \right)^2}$$

Note that P_i, P_{ij}, and P_{ik} refer to three different quantities, which are only distinguished on the basis of their indices. This concise, somewhat informal, notation is common in tensor algebra, but is restricted to the present section to prevent potential confusion.

In Figure 3, we use P_{ik} to identify neurons that consistently remap. Let c_i denote the cluster label of each trial such that $c_i = 1$ if trial i is in map 1 and $c_i = -1$ if trial i is in map 2. Then $\log(1 + \exp(-c_i P_{ik}))$ provides a measure of distance between the spatial firing pattern of neuron k to the spatial map (i.e., cluster centroid) on trial i. Specifically, this calculation corresponds to a logistic loss function in the context of classification models. Averaging this distance over trials summarizes the remapping strength—intuitively, an average distance close to zero implies that the neuron "agrees with" the rest of the population on each trial, while a large average distance implies that the neuron is inconsistent (e.g., because the neuron exhibits high levels of noise on each trial). We classified neurons as "consistent remappers" when the average distance was less than 1.

In Figures 8 and S8, we used P_{ij} to assess the relationship between running speed and the distance of neural coding to the midpoint between clusters. In Figure 8D, we plotted $|P_{ij}|$, i.e., the distance to midpoint, in 10 running speed bins for an example session—the first 9 bins were evenly spaced between 0 cm/s and 20 cm/s below the maximum speed; the last bin included all top speeds above this final threshold (this was done to account for rare bursts of high speeds). Similarly, in Figure 8E we plotted $|P_{ij}|$ in 10 running speed bins, normalized within each session. In Figure S8C-E, we use histograms to visualize P_{ij} for all trials and position bins. To account for arbitrary map assignment, we randomly flipped the sign of P_{ij} for each session in Figure S8E. Likewise, the white-to-black heatmaps in Figures 8C, S8H, and S8I visualize P_{ij} for a subset of trials. In all cases, P_{ij} was z-scored to normalize across sessions.

In Figure S8 we performed a similar analysis examining the effect of track position (and therefore landmarks) or reward on the distance of neural coding to the midpoint between clusters. In Figure S8M we examined the average distance to the midpoint within each 5cm position bin along the track, split by cue rich and cue poor session types (to assess the effect of landmarks). We compared these distances to a shuffle control in which neural activity was shifted by a random distance up to 400cm. Similarly, in Figure S8L, we examined the average distance to midpoint for 5cm position bins for 100cm centered on each random reward zone (25cm approaching, 50cm within the zone, and 25cm exiting). We compared these distances to a shuffle control created by the same procedure.

Position decoding analysis

We fit linear models to predict the animal's position from the spiking activity of all MEC neurons, and call the optimized model a "decoder" following common terminology and practice (Kriegeskorte and Douglas, 2019) (Figure 6). Let $y_t \in [0, 2\pi)$ denote the position on the circular track at time t, and let s_{nt} denote the number of spikes fired by neuron n at time t after smoothing with a Gaussian filter (standard deviation = 200 ms). Due to the nature of the VR environment, y_t is a circular variable—i.e., it should be interpreted as an angle on the unit circle. In the statistics literature, a regression that predicts a circular variable from linear covariates is known as a *circular-linear regression* model. Several approaches to circular-linear regression have been developed (Fisher and Lee, 1992; Pewsey and García-Portugués, 2020; Sikaroudi and Park, 2019). Here, we used a *spherically projected multivariate linear model* (Presnell et al., 1998). Two regression coefficients, $\beta_n^{(1)}$ and $\beta_n^{(2)}$, are optimized for each neuron using the expectation maximization routine described by Presnell et al. (1998). After fitting the model to a set of training data, the model estimate for a given set of inputs is given by

$$\widehat{y}_t = \text{atan2}\left(\sum_{n=1}^{N} \beta_n^{(1)} s_{nt}, \ \sum_{n=1}^{N} \beta_n^{(2)} s_{nt} \right)$$

where $\text{atan2}(\cdot, \cdot)$ corresponds to the "2-argument arctangent" function. The "model score" referenced in Figure 6 is the average of $\cos(y_t - \widehat{y}_t)$ over time bins in the testing set. Thus, a decoder which randomly guesses angles over the unit circle would have an expected score of zero, while a perfect decoder would have a score of one. Note that training data was downsampled to match spike number, position bins, running speed, and number of observations across training sets (map 1, map 2, and both maps) for each session. In Figures 6D–6F each point represents the average model score across all 10 possible test sets. In Figure S6D, we performed the same analysis with synthetic ablations of putative grid cells, putative non-grid spatial cells, and putative non-spatial cells (we ablated an equal number of cells in each case, equal to the minimum number of cells across groups in each session) to test whether model performance depended on any particular functional cell type.

In Figures 6D and 6G, we compared model performance to performance on a shuffle control. For each map, we created a shuffled version of that map (i.e., the "synthetic map") in which the neuron labels were permuted and the tuning curves of all neurons were shifted by a random amount (less than or equal to half the track length). For Figure 6D, we then fit the models on a training set comprised of each map and its corresponding synthetic map and tested them on unseen data from the same pair of maps. For Figure 6G, we fit the models on each true map and tested them on the shuffled version of the other map (e.g., fit on the true map 1 and test on the shuffled map 2).

Spike waveform analysis

Waveforms were extracted from the Kilosort2 data output using our modification of Jennifer Colonell's fork of the Allen Institute for Brain Science ecephys library (open source from GitHub: https://github.com/jenniferColonell/ecephys_spike_sorting/). To explicitly compare waveforms across remap events and across the session (as in Figure S3), we identified stable blocks of 10 or more trials from each k-means-identified map for all 2-map sessions. For 19/28 2-map sessions, we selected the 10 trials abutting up to 3 remap events near the beginning, middle, and end of the session (e.g., Figures S3B–S3D, left panels, colors indicate selected trials). For the

remaining sessions, there were not adjacent trial blocks of 10 or more sessions in each map, and so we selected a matched number of blocks of 10 trials from each map, up to 3 pairs, from throughout the session. We then extracted the waveforms for up to 1000 spikes randomly chosen from the selected trial blocks (for cells with fewer than 1000 spikes in either the pre- or post-remap trial block, we extracted an equal number of spikes from each block, equal to the minimum number of spikes across the two blocks). We then computed the average and standard deviation of the waveforms within each trial block using our modification of the ecephys library (e.g., Figures S3B-S3D, right panels).

To determine waveform similarity across all 2-map sessions, we computed the correlation between waveforms from different maps for each cell (Figure S3A). For both analyses, we first identified the "best channel" for each cell, by identifying the channel closest to the Kilosort2 depth output for that cell. We then selected just the 20 channels centered on the best channel for each cell (or off-center from the best channel, if it was within 10 channels of the probe tip) and compared only the waveforms on these 20 channels. We then calculated the Pearson correlation between the vectorized average waveforms for each pre- and post-remap pair and computed the average correlation across all pairs (up to three pairs per cell)(Figure S3A).

Tetrode recordings and analysis

The tetrode data included in Figure S3 were collected for a previous publication (n = 296 cells from 112 sessions in 19 mice) (Campbell et al., 2018). Because VR gain manipulations as performed for that study can induce remapping of spatial representations, all data examined for this figure were from "baseline trials" in which no VR manipulation occurred. However, because these mice experienced frequent gain manipulations and were handled by a different experimenter (MGC), we compared these tetrode data to Neuropixels data collected by the same experimenter from mice who also experienced frequent gain manipulations, to account for potential lasting effects of frequent manipulations or effects of handling differences (n = 3,075 cells from 89 sessions in 20 mice) (Campbell et al., 2020).

For each cell (Neuropixels or tetrodes), we examined a single block of 20 baseline trials in which no VR manipulation occurred. For each cell, we computed firing rate maps in single trials and computed a cross correlation matrix over the 20 trial block, taking the peak cross correlation over lags from −20 cm to +20 cm to allow for small shifts. We focused our analysis on cells that were "spatially stable" within the first 6 trials (defined as having mean trial-trial peak cross correlation > 0.5 in the first 6 trials), and asked how the rate maps changed in the following 14 trials. To quantify this, we computed the peak cross correlation between each of these 14 trials and each of the 6 "baseline" trials, and averaged over baseline trials. When the pattern remaps, this cross correlation should be low; when it is stable, it should be high. We performed a statistical comparison of the distribution of cross correlations for cells recorded with tetrodes to the distribution for cells recorded with Neuropixels probes. Note that the tetrode recordings were performed on a VR track with more salient visual landmarks than that used for Neuropixels recordings, including a clearly delineated trial structure with teleportations between each trial.

Spatial information calculations

Following the procedures in Skaggs et al. (1996), we calculated spatial information content in bits per second over 2 cm position bins for each neuron as follows:

$$\sum_i p_i \lambda_i \log_2 \left(\frac{\lambda_i}{\lambda} \right)$$

Where p_i is the probability that the animal is in position bin i (occupancy time in position bin i divided by total session time), λ_i is the average firing rate of the neuron in position bin i, and λ is the overall average firing rate for the neuron. Firing rates were computed empirically (number of spikes in position bin i divided by occupancy time). For Figure 4, spatial information was calculated separately for each map by first separating trials by their k-means-defined cluster label. Cells were defined as significantly spatial if their spatial information score was > 95th percentile of a null distribution comprising 1,000 shuffle controls. Shuffles were computed separately for each map in each session, and were implemented by shifting all spikes for all cells by a random time interval, up to a maximum of 10 s, to disrupt the spike/position relationship without changing inter-spike intervals or correlations across cells. "Spatial cells" in Figure 4 and as referred to in the corresponding results section, were significantly spatial in at least one map.

Rate remapping versus global remapping

In order to identify the extent to which each cell remapped across trials, we first divided trials by their 2-factor k-means cluster label. We then computed the trial-averaged firing rate in 2 cm position bins for each map, smoothing with a 1D Gaussian filter (standard deviation 2 cm). We quantified the degree of rate remapping in each neuron by the percent change in the peak firing rate (i.e., largest firing rate in any spatial bin) across the two maps. As a measure of global remapping, we calculated an alignment score between the normalized firing rate vectors in activity space (i.e., spatial dissimilarity). To do so, we computed the cosine similarity (vector dot product after normalization) between the spatial profiles of the within-map averaged firing rates. We then subtracted this value from 1, such that dissimilarity score of 0 would indicate an identical spatial firing pattern and a score of 1 would indicate orthogonal spatial representations. These results are reported in Figure 4.

Neuron
Article

CellPress

Functional cell type analysis

Using the gain manipulation described above (and as in Campbell et al., 2018), we identified putative grid cells and putative border cells. For this analysis, we first removed cells with a mean firing rate $\geq$ 15Hz (putative high-firing interneurons). We further selected only cells that were spatially stable in both the 10 trials preceding gain manipulation ("normal trials") and in the 5 gain manipulation trials ("gain manipulation trials")(average trial-trial correlation greater than 0.25). We then computed the spatial tuning curves for the last 10 normal trials and 5 gain manipulation trials and calculated the Pearson correlation between these tuning curves for each cell. Using the results from Campbell et al. (2018) as a guide, we designated all cells with a normal-gain manipulation correlation greater than 0.29 as putative border cells and all cells with a normal-gain manipulation correlation less than 0.1 as putative grid cells (Figure S6A). This method is a rough classification of each cell type, which likely includes some false positives and may falsely reject some true grid and border cells. We thus first designate these cells as putative grid and putative border cells, but subsequently refer to them as grid and border cells for simplicity and readability.

In Figures S6F–S6K, we investigated the periodic firing properties of putative grid and border cells. We first split trials by their k-means identified map. For each trial, we computed the firing rate in 5 cm position bins and concatenated trials of each type to obtain a continuous vector of firing rate by absolute distance traveled for each condition. We then computed the spatial autocorrelation of this signal, up to a maximum distance of 800 cm, normalized such that the maximum autocorrelation at 0 cm = 1. We identified peaks in this signal with a minimum prominence of 0.05 and compared each peak to a null distribution to determine whether it was higher than could be expected by chance. The null model was given by a first-order autoregressive process, i.e., an AR(1) model, with no drift term. The decay parameter of the model was chosen to be 0.55 to roughly match the autocorrelation of cells with small spatial fields. The autocorrelation function of the AR(1) null model admits a closed-form expression, as covered in standard references on time series analysis (Chatfield, 1984). To evaluate whether any peak in the empirical autocorrelation function was significant with respect to this null distribution, we computed Bonferroni-corrected 95% confidence intervals around the observed autocorrelation function. In Figure S6F-H, we show the autocorrelation for all putative grid and border cells for 3 example sessions, normalized by the maximum/minimum autocorrelation from all but the first 100 position bins.

Bistable ring attractor model

Attractor networks are a class of circuit models that describe how a recurrently connected population of neurons can sustain internal representations in the absence of persistent sensory input (Amari, 1977; Samsonovich and McNaughton, 1997; Seung, 1996). These model networks are often formulated as a system of differential equations:

$$\tau \frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x} + \phi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{u}) \tag{S.1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ denotes a vector of firing rates for a network containing N neurons, $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ is the "connectivity matrix" holding synaptic weights, $\phi(\cdot)$ is an elementwise nonlinear activation function, $\boldsymbol{u} \in \mathbb{R}^N$ is a constant input to the network, and $\tau > 0$ is the time constant of the system. We assume that $\phi(\boldsymbol{x}) = 1/(1 + \exp(-\boldsymbol{x}))$, though other choices of activation function are possible.

Values of $\boldsymbol{x}$ for which $d\boldsymbol{x}/dt = 0$ are called *fixed points*. Of particular interest are *attractive fixed points*. If $\boldsymbol{x}(t)$ denotes the vector of firing rates at time t, and if $\boldsymbol{x}_*$ is a fixed point, then $\boldsymbol{x}_*$ is called an attractive fixed point if $\boldsymbol{x}(t)$ approaches $\boldsymbol{x}_*$ in the limit of $t \to \infty$, when the system is initialized sufficiently close to $\boldsymbol{x}_*$ (i.e., when $\boldsymbol{x}(0) = \boldsymbol{x}_* + \delta$ for any vector $\delta \in \mathbb{R}^N$ with sufficiently small norm). *Attractor manifolds* can be thought of as continuous sets of attractive fixed points. While attractive fixed points are limited to representing a discrete set of states, attractor manifolds can represent continuous quantities, such as the orientation of a visual stimulus (Ben-Yishai et al., 1995), motor neuron drive (Seung, 1996), or the animal's heading direction (Skaggs et al., 1995).

In our experiments, mice traversed a 1-dimensional virtual hallway that seamlessly looped back to the starting position after 400 cm. This can be thought of as a VR analog to a circular maze environment. If there is stable position coding and it is encoded on a single attractor manifold, then the topology of the environment implies that the attractor manifold should form a 1-dimensional ring (a "ring attractor"). Further, the velocity of the animal on the track is calibrated to the velocity in neural firing rate space along the ring attractor so that every physical location on the track is one-to-one matched to a position along the ring. This ring attractor model is similar to well-known models of interactions among comodular grid cells (Burak and Fiete, 2009; Fuhs and Touretzky, 2006; Guanella et al., 2007). However, while the mathematical details we describe below are similar to this past work, there is an important conceptual difference. Namely, our model is meant to include the full MEC circuit, under the assumption that it stores a stable estimate of the animal's position. In this respect, our model is most similar to spatial attractor models originally developed and studied in the context of hippocampal circuits (Samsonovich and McNaughton, 1997). The model we describe is a coarse description of MEC— e.g., it does not capture the diversity of functionally defined cell types involved in navigation. However, it allows us to demonstrate a simple possibility for how aligned ring manifold structures (like those found in our experimental data) might arise in a circuit of recurrently connected neurons.

Prior work has found that multiple attractor network connectivities can be linearly combined to create a network with multiple attractor manifolds that operate independently (Romani and Tsodyks, 2010; Roudi and Treves, 2008; Samsonovich and McNaughton, 1997; Stringer et al., 2004). That is, to store M attractor manifolds we can set $\boldsymbol{W} = \sum_{m=1}^{M} \boldsymbol{W}^{(m)}$, where each $\boldsymbol{W}^{(m)} \in \mathbb{R}^{N \times N}$ holds the connectivity pattern of a single attractor structure. If $M \ll N$, and if the maps are sufficiently decorrelated (e.g., if the neuron

indices are randomly permuted for each $\boldsymbol{W}^{(m)}$), then the attractors can operate independently (Samsonovich and McNaughton, 1997).

To create a bistable pair of ring attractor, we linearly combined a classic ring attractor connectivity pattern (Figure S7A, left), with a "winner-take-all" connectivity pattern (Figure S7A, right). Intuitively, the winner-take-all pattern creates two sub-populations of neurons that mutually exclude each other from firing. Because the ring attractor structure is present within both sub-populations, two separated 1D ring attractors are formed, which we can interpret as two spatial maps.

Concretely, we instantiate the model by defining a fine grid over ring angles, $\theta_i = 2\pi i/N$ for $i \in \{1, \ldots, N\}$. Additionally, for each neuron we define a sub-population indicator variable $v_i = \mathbf{1}[i \leq N/2] - \mathbf{1}[i > N/2]$; where $\mathbf{1}[c]$ is a binary indicator function which evaluates to one if c is true and zero if c is false. Then, let $\sigma(\cdot)$ denote a random permutation of the N neurons, and define the connectivity of the network to be $\mathbf{W} = \mathbf{W}^{(1)} + \mathbf{W}^{(2)}$, where

$$\left[\boldsymbol{W}^{(1)}\right]_{ij} = J_1 \cos(\theta_i - \theta_j) - J_0 \tag{S.2}$$

$$\left[\boldsymbol{W}^{(2)}\right]_{ij} = J_2 v_{\sigma(i)} v_{\sigma(j)} \tag{S.3}$$

The connectivity encoded in $\boldsymbol{W}^{(1)}$ implements the ring attractor, while $W^{(2)}$ implements the winner take all connectivity. The three scalar hyperparameters, $J_0 > 0$, $J_1 > 0$, $J_2 > 0$, respectively determine the strength of a global inhibition term, the strength of connections modulated by the ring, and the strength of the winner-take-all connectivity. For now, we assume the network has no input, $u = 0$.

Figure S7B shows W_{ij} and $W_{\sigma(i)\sigma(j)}$ for a small network with N = 150 neurons; this visualization demonstrates that simply re-ordering the neurons is sufficient to reveal the two connectivity patterns embedded in the same network. We numerically simulated a network with N = 10,000 neurons to explore whether this model produced the expected attractor structures. For these simulations we set $J_0 = J_1 = 0.01$ and $J_2 = 0.004$. The steady-state of eq. (S.1) is solely determined by the initial state, $\boldsymbol{x}(0)$, which we set equal to

$$[\boldsymbol{x}(0)]_i = \phi\left(\cos(\theta_i - \psi) + \varepsilon v_{\sigma(i)}\right) \tag{S.4}$$

where $\psi \in [0, 2\pi)$ is an angle specifying the tuning of initial activity along the ring, and ε scales the competitive winner-take-all connectivity pattern. When $\varepsilon > 0$ the network activity reliably converges to position ψ in map 1; conversely, when $\varepsilon < 0$, the network converges to position ψ on map 2. (We label map 1 and map 2 in this way as a convention for the model.) By varying ψ and ε, and numerically simulating the firing rate dynamics to steady-state, we recover the expected pair of ring manifolds (Figure S7C).

In contrast to the experimental data, when both maps are simultaneously embedded into the same 3D space by PCA, the ring manifolds do not appear aligned (Figure S7C). We then introduced a population of "shared neurons" that participated in both spatial maps by defining

$$p_i = \begin{cases} 0 & \text{with probability } \mu \\ 1 & \text{with probability } 1 - \mu \end{cases} \quad \text{for all } i \in \{1, \ldots, N\} \tag{S.5}$$

where $\mu \in [0, 1]$ corresponds to the proportion of "shared neurons." Then, we modified equation (S.3) to be:

$$\left[\boldsymbol{W}^{(2)}\right]_{ij} = J_2 v_{\sigma(i)} v_{\sigma(j)} p_i p_j \tag{S.6}$$

and set the network input to $u_i = J_4(1 - p_i)$ for each neuron. Intuitively, J_4 scales additional excitatory input to the "shared neurons" that lack the mutual excitation from the winner-take-all connectivity. We set $J_4 = 4$ in our numerical simulations. Figure S7D displays the connectivity of a circuit with shared neural population (compare with Figure S7B).

When the proportion of shared neurons, μ, is appropriately tuned, the ring manifolds appear geometrically aligned in PCA embeddings (Figure S7E). As done for the experimental data in Figure 4, we used Procrustes analysis to quantify manifold misalignment scores, which range between zero (perfect alignment) and one (RMSE misalignment equal to a shuffle control). Increasing μ monotonically improved the ring alignment in simulations (Figure S7F).

Finally, to numerically simulate remapping events and demonstrate that each ring attractor is locally stable and capable of operating independently, we introduced a noise term (formally, Brownian motion) into the dynamics equation (S.1). We numerically integrated stochastic instantiations of the system by the Euler–Maruyama method. Inspired by the experimental findings in Figures 7K and 7L, we use a mix of isotropically distributed noise with occasional perturbations preferentially oriented in the dimension separating the manifolds. These directed perturbations occasionally push the network activity from one ring attractor to the other, resulting in a remap event (Videos S1 and S2). Overall, this preliminary model captures two key features of the experimental data—geometric alignment of two ring manifolds, and a reversible remapping mechanism.

CellPress

QUANTIFICATION AND STATISTICAL ANALYSIS

Statistics

All data were analyzed in Python, using the scipy stats library to compute statistics, except for data in Figures S3E–S3G, which were analyzed in MATLAB. Unless otherwise noted, all tests are two-sided, correlation coefficients represent Pearson's correlation, and values are presented as mean ± standard error of the mean (SEM). Non-parametric tests were used to assess significance, specifically Wilcoxon signed-rank tests for paired data, Wilcoxon rank-sum tests for unpaired data, and Kruskal-Wallis H-tests for comparisons of > 2 values. Data collection and analysis were not performed blind to the conditions of the experiments. No statistical methods were used to predetermine sample sizes, but our sample sizes are consistent with previous similar studies.