Title: A non-spatial account of place and grid cells based on clustering models of concept learning

Authors: Robert M. Mok[1*], Bradley C. Love[1,2*]

Affiliations:

[1]Department of Experimental Psychology, University College London, 26 Bedford Way, London, WC1H 0AP, United Kingdom.

[2]The Alan Turing Institute, United Kingdom

*Correspondence to: robert.mok@ucl.ac.uk, b.love@ucl.ac.uk

ABSTRACT

One view is that conceptual knowledge is organized as a "cognitive map" in the brain, using the circuitry in the medial temporal lobe (MTL) that supports spatial navigation. In contrast, we find that a domain-general learning algorithm explains key findings in both spatial and conceptual domains. When the clustering model is applied to spatial navigation tasks, so called place and grid cells emerge because of the relatively uniform sampling of possible inputs in these tasks. The same mechanism applied to conceptual tasks, where the overall space can be higher-dimensional and sampling sparser, leads to representations more aligned with human conceptual knowledge. Although the types of memory supported by the MTL are superficially dissimilar, the information processing steps appear shared.

INTRODUCTION

Concepts organize experiences to enable generalization and inference. For example, a traveler encountering an unfamiliar bird species would reasonably infer the bird was born from an egg. One longstanding question is the basis for people's abstract conceptual knowledge. One intuitive idea is that concepts ground in a more basic and concrete substrate, such as sensory-motor experience[1]. For example, abstract concepts such as time may be represented in terms of experience of space[2]. Relatedly, conceptual knowledge may be organized as a "cognitive map" in the brain[3–5] relying on circuitry in the medial temporal lobe (MTL) that supports navigation[6].

This view is supported by recent studies that find the brain's responses to conceptual tasks parallel those previously found in spatial tasks. Place cells in the hippocampus[5] typically have single firing fields at circumscribed locations in an environment, and grid cells in the medial entorhinal cortex (mEC)[7–9] display multiple regularly-spaced firing fields arranged in a hexagonal pattern covering the environment. These spatially-tuned cells in the MTL are thought to implement a spatial cognitive map for navigation[10,11], and recent work suggests these cells also represent conceptual[4] and task spaces[12]. One key question is whether the same brain systems and computations support concept learning, memory, and spatial navigation.

One neglected possibility is that the relationship between spatial and conceptual representations has been framed backwards. Perhaps, rather than concepts grounding

in the machinery of navigation, spatial concepts are a limiting case of a single, more general, learning system. This general learning system would support learning concepts, which are typically clumpy in that they consist of clusters of interrelated features in a high-dimensional space[13]. For example, animals that fly tend to be small and have wings (see Fig. 1A). Not all possible combinations of features are relevant and represented. In contrast, many spatial tasks[7] and their conceptual analogs[4] typically involve a uniform and exhaustive sampling of all possible combinations within a low (two-) dimensional space corresponding to locations in an environment (see Fig. 1C-D).

We evaluate whether a domain general account is plausible by applying successful models of human concept learning to spatial contexts. In concept learning studies, these models learn to represent experience in terms of conceptual clusters, which are not uniformly distributed[14]. These clustering representations successfully capture patterns of activity in the MTL[15,16]. Furthermore, these models explain how individual episodes give rise to conceptual knowledge over the course of learning[17], consistent with both the hippocampus's importance in memory[18,19] and the presence of "concept" cells[20]. We evaluate whether the same mechanisms also offer a general understanding of place and grid cells, and their relationship to concepts.

To facilitate this evaluation, we simplified the clustering models to only include aspects necessary for this contribution. Clustering models that capture behavior on a trial-by-trial basis typically recruit a new cluster in response to a surprising error. These models also

learn attention weights that accentuate task-relevant stimulus dimensions and associate clusters with behavioral responses (e.g., respond “bird”). Without loss of generality, we simplified the models by pre-seeding with a fixed number of clusters and limiting learning to updating cluster positions. In particular, the cluster most similar to the current stimulus updated its position in representational space to be closer (more similar) to its newest member (see Online Methods for full details), much like cluster updating in Kohonen learning maps[21] and k-means clustering[22].

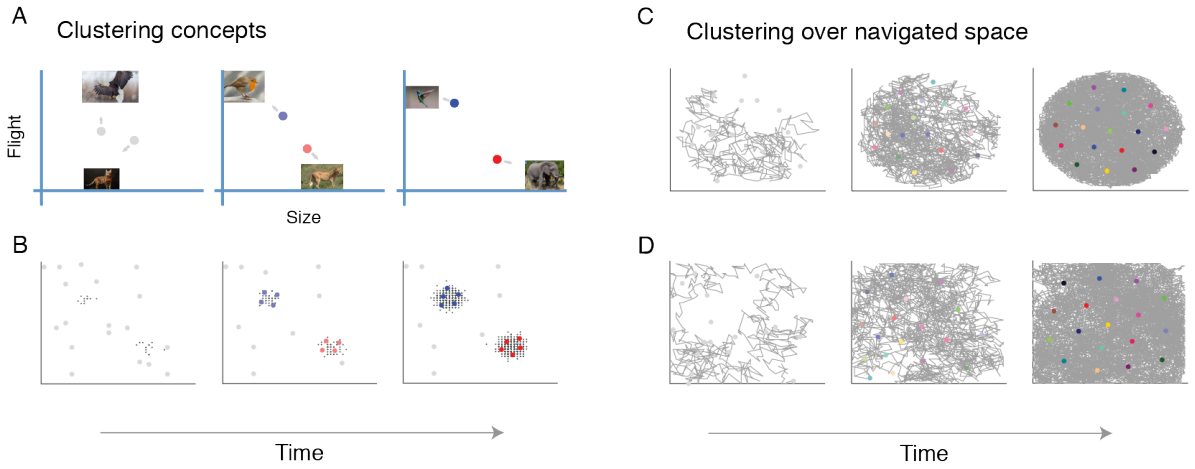

Fig. 1. Cluster learning applied to conceptual and spatial examples. (A) The most similar cluster moves (i.e., adjusts its tuning) toward its newest member and becomes associated with a response (blue for bird, red for mammal). (B) Out of a pool of many randomly tuned clusters, a subset comes to represent the two concepts over learning. (C-D) The same learning system applied to an agent locomoting in a circular or a square environment gives rise to a hexagonal cluster organization.

RESULTS

A common learning mechanism supports spatial and conceptual knowledge

As shown in Fig. 1A, the model when applied to categorizing animals as birds or mammals learns to segregate the items into two groupings. These clusters can be seen as concept cells, akin to place cells (Fig. 2A-B). Notice that the items (i.e., experiences) and the clusters only cover a select portion of the stimulus space. For example, no animal exists that is as massive as an elephant and can fly. Clustering solutions capture the structure of the environment, which enables generalization to novel cases.

In contrast, the same model applied to a rodent exploring a typical laboratory environment leads to clusters that uniformly cover the entire representational space in a hexagonal pattern (see Fig. 1B). In the spatial case, there is no salient structure present in the input to the model, which results in clusters covering the representational space, much like how a bunch of tennis balls dropped into a square box will self-organize into a grid-like lattice according to the mathematics of packing[23]. In the spatial case, the clusters function as place cells that code for (i.e., discriminate) locations.

In our account, grid-like responses arise from monitoring the match (inverse error) of the clustering solution (Fig. 2). In unsupervised learning, error or uncertainty is simply the inverse of how similar an item is to the best matching cluster. Notice that matching clusters in the spatial case should display a hexagonal pattern because of the hexagonal clustering pattern in representational space, resulting in canonical grid-like

receptive fields (see Fig. 2B). In the conceptual case, we predict that typical grid cell firing patterns should not be observed because the clusters (i.e., place cells) do not form a hexagon pattern (Fig. 2A) in representational space. One might object that our account is inconsistent with conceptual learning brain imaging studies that find grid-like response patterns[4]. However, these studies are consistent with the model because they follow the design principles of typical spatial studies -- all feature combinations within a 2-dimensional stimulus space are sampled, which would lead to a hexagonal clustering (Fig. 2B).

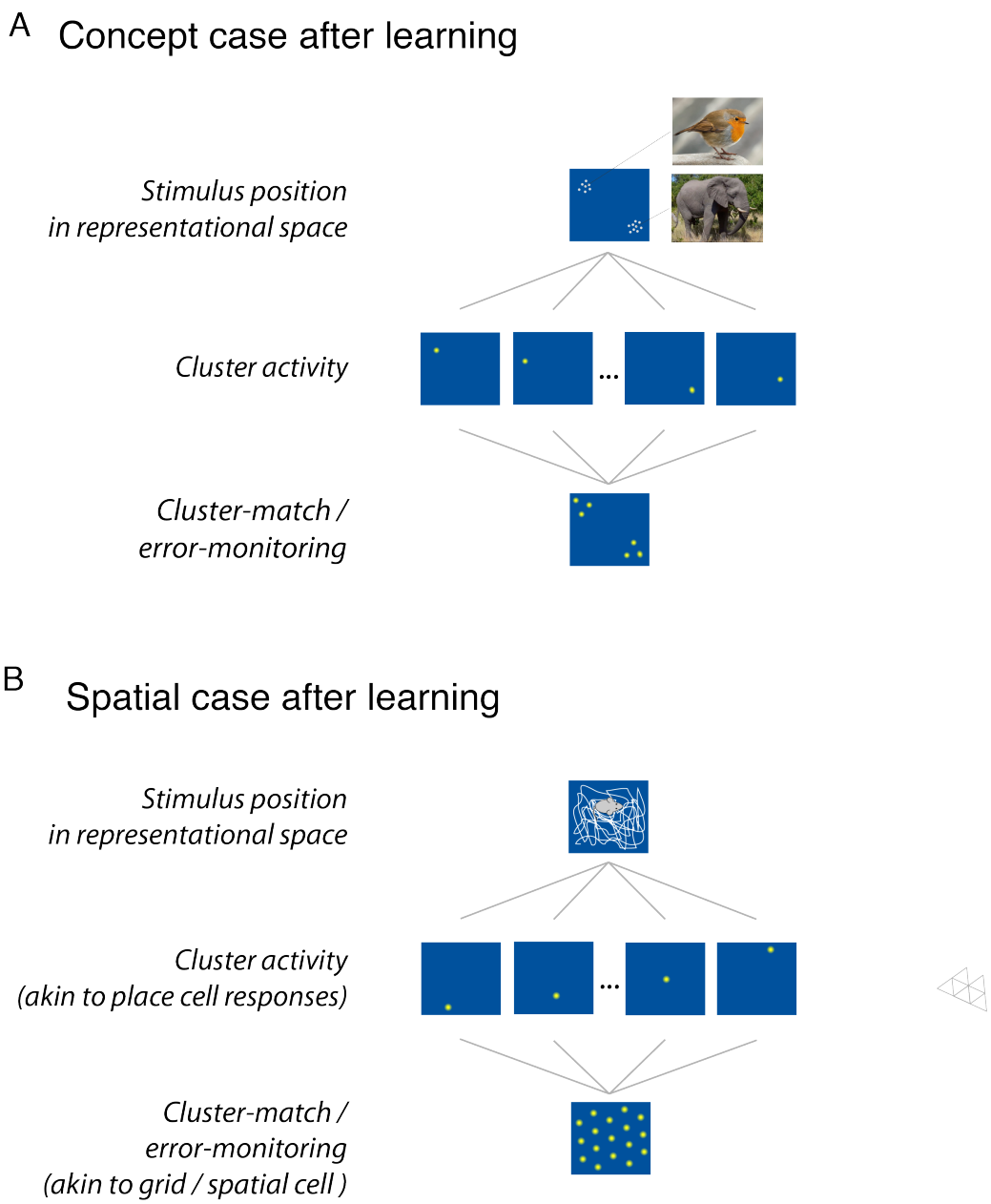

Fig. 2. Cluster representations after learning for conceptual and spatial tasks. (A) Clusters clump into two groups. Thus, novel bird and mammal stimuli will strongly activate one or the other grouping, which does not lead to a grid response across possible stimuli. (B) In contrast, for the spatial case, clusters form a hexagonal grid which leads to a grid-like response across possible stimuli when cluster activity is monitored.

Grid-like clustering solutions match grid and spatial cells in mEC

To relate our account to typical spatial studies, we simulated an agent moving through its environment as in a free-foraging rodent experiment. As expected, learning led to clusters forming a hexagonal pattern (see examples in Fig. 3A-B, left). To assess this quantitatively, we computed the spatial autocorrelograms of the cluster activation maps (Fig 3A-B, right) to obtain the grid score, which reflects the degree six-fold hexagonal symmetry in the cluster activation pattern across trials[7] (see Online Methods). We computed grid scores for each time bin during learning and found that grid scores tended to increase over learning in both the square (see Fig. 3C for examples and Fig. S1 for all conditions; mean slope: 0.0044, bootstrap CIs: [0.0040,0.0048]) and circular environment (mean slope: 0.0042, bootstrap CIs [0.0038,0.0046]; see Tables S1 and S2).

Following learning, we evaluated the gridness of the clustering solution (see examples in Fig. 3A-B and Fig. S2 and S3). A substantial proportion of simulations satisfied the criterion for grid-like organization, with 45.3% in the square and 38.6% in the circular environment, which closely match the proportions in empirical results (45% and 38%, respectively; see Supplementary Results)[24,25]. The average grid score in both the square environment (mean: 0.277, bootstrap CIs [0.273,0.0.280]) and circular environment (mean: 0.313, bootstrap CIs [0.309,0.318]) were greater than zero; see Tables S3 and S4).

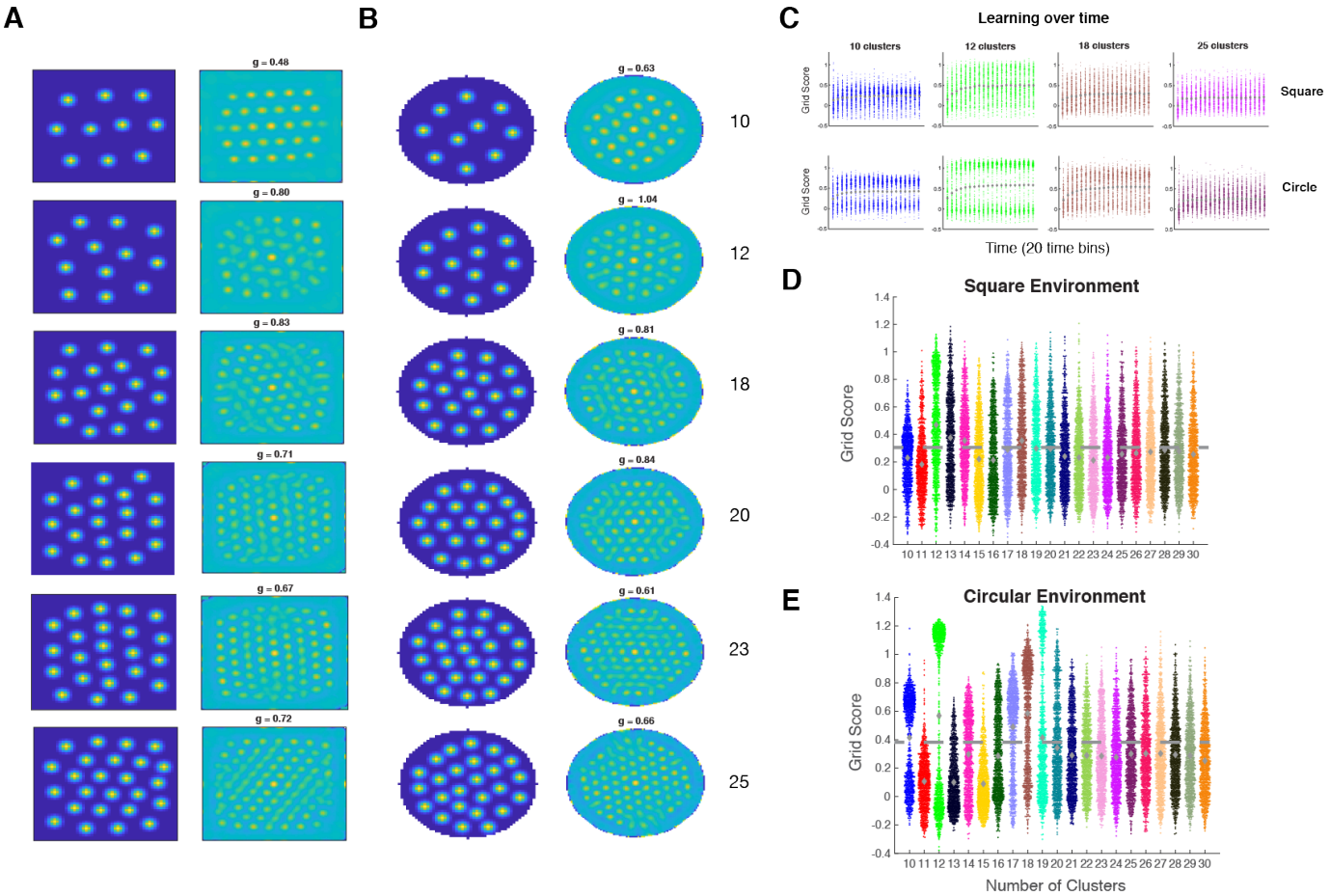

Fig. 3. Clustering leads to activation maps similar to grid and non-grid spatial cell activity maps in medial entorhinal cortex. (A-B) Examples of activation maps with grid patterns in a square environment (A-left) and their corresponding spatial autocorrelograms (A-right), and activation maps in a circular environment (B-left), and spatial autocorrelograms (B-right). (C) Examples showing grid scores increasing over learning in the square (top) and circle (bottom). (D-E) Univariate scatterplots showing grid scores for simulations in the square (D) and circle (E). Dashed line represents the most conservative threshold for a grid 'cell'.

Grid-like structure in cluster representations are shaped by environmental geometry

According to the clustering account, grid-like responses should only arise under very specific conditions in which the environment and how it is sampled is fairly uniform. The imposition of any structure, including changes to the overall geometry of the environment, should affect the clustering in a manner that makes it less grid-like.

Related, Krupic et al.[26] identified grid cells in rodent mEC in a square box, then placed the animals in a trapezoid environment. They found that activity maps of grid cells became less grid-like in the trapezoid and that the decline was greatest for responses elicited on the narrow side of the trapezoid. To simulate this experiment, the model was first trained in a square and then transferred to a trapezoid environment (see Fig. 4A-D for an example and Fig. S4 for more examples). As in the empirical studies, the model's overall grid scores declined in the trapezoid environment (Fig. 4E; trapezoid mean grid score: 0.058, bootstrap CIs [0.054,0.061]; Fig. 4F; square minus trapezoid mean: 0.219, bootstrap CIs [0.214,0.224]) and the grid scores were higher on the wide than on the narrow side of the trapezoid (Fig 4G; wide minus narrow mean: 0.133, bootstrap CIs [0.127,0.139]; see Tables S5-S7).

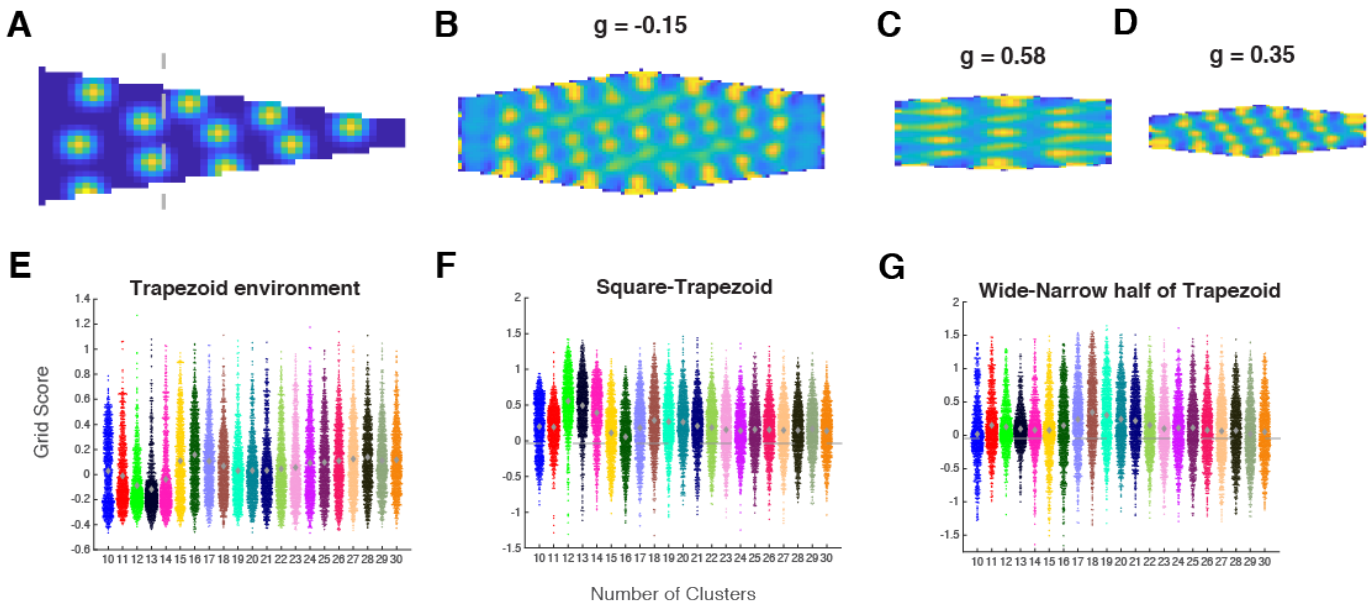

Fig. 4. The clustering model captures declines in grid responses in trapezoid environments. (A-D) Example of distortion in a trapezoid environment. (A) Example of an activity map in a trapezoid environment with 18 clusters. The dotted line demarcates the wide and narrow halves of the trapezoid (B) Spatial autocorrelogram of the trapezoid. (C) Spatial autocorrelograms of the wide (left) and (D) narrow (right) portion of the trapezoid in (A). (E) Univariate scatterplot showing grid scores for simulations in the trapezoid enclosure after learning in a square enclosure. (F) The difference (positive) between grid scores in the square and trapezoid. (G) The difference (positive) between grid scores for the wide and narrow halves of the trapezoid.

DISCUSSION

Previous work has explained a wide array of learning and memory phenomena in terms of clustering computations supported by the MTL[17]. Here, this same basic account was

shown to account for basic spatial navigation phenomenon, including place and grid cell-like response patterns.

Rather than spatial mechanisms providing a scaffolding for more abstract conceptual knowledge[6], the current results suggest that key findings in the spatial literature naturally arise as limiting cases of a more general concept learning mechanism. Whereas concepts can be clumpy, structured, and high dimensional, typical spatial tasks involve exhaustive and uniform sampling of simple two-dimensional environments, which leads to degenerate clustering solutions that pack clusters into a hexagon lattice, giving rise to so-called grid cells (Fig. 3). The clustering account correctly predicted how deviations from these unstructured learning environments should reduce grid-like cell responses (Fig. 4).

Building this integrative bridge between the concept, memory, and spatial literatures allows for findings from one domain to inform the other. For example, task goals and attentional mechanisms in the concept literature have been found to shape hippocampal representations [15,16]. Analogous tasks can readily be constructed to evaluate whether spatial cells support broader information processing functions (cf. [12]) and how general learning algorithms shape their response properties (cf. [27]). Likewise, the concept literature emphasizes the hippocampus's interactions with other brain areas, such as medial prefrontal cortex, to assist in encoding task relevant information[15]. When richer spatial tasks are considered, there is a ready set of candidate mechanisms and neural

systems that may offer domain general explanations that link across brain, behavior, and computation.

REFERENCES

1. Barsalou, L. W. Grounded Cognition. *Annu. Rev. Psychol.* **59,** 617–645 (2008).
2. Boroditsky, L. Metaphoric structuring: Understanding time through spatial metaphors. *Cognition* **75,** 1–28 (2000).
3. Behrens, T. E. J. *et al.* What is a cognitive map? Organising knowledge for flexible behaviour. *bioRxiv* (2018). doi:https://doi.org/10.1101/365593
4. Constantinescu, A. O., O'Reilly, J. X. & Behrens, T. E. J. Organizing conceptual knowledge in humans with a gridlike code. *Science* **352,** 1464–1468 (2016).
5. O'Keefe, J. & Nadel, L. *The Hippocampus as a Cognitive Map*. (Oxford University Press, 1978).
6. Buzsáki, G. & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature Neuroscience* **16,** 130–138 (2013).
7. Hafting, T., Fyhn, M., Molden, S., Moser, M. B. & Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **436,** 801–806 (2005).
8. Doeller, C. F., Barry, C. & Burgess, N. Evidence for grid cells in a human memory network. *Nature* **463,** 657–661 (2010).
9. Horner, A. J., Bisby, J. A., Zotow, E., Bush, D. & Burgess, N. Grid-like processing of imagined navigation. *Curr. Biol.* **26,** 842–847 (2016).
10. Tolman, E. C. Cognitive maps in rats and men. *Psychol. Rev.* **55,** 189–208 (1948).
11. Epstein, R. A., Patai, E. Z., Julian, J. B. & Spiers, H. J. The cognitive map in humans: Spatial navigation and beyond. *Nature Neuroscience* **20,** 1504–1513

(2017).

12. Aronov, D., Nevers, R. & Tank, D. W. Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit. *Nature* **543,** 719–722 (2017).

13. Rosch, E. & Mervis, C. B. Family resemblances: Studies in the internal structure of categories. *Cogn. Psychol.* **7,** 573–605 (1975).

14. Love, B. C., Medin, D. L. & Gureckis, T. M. SUSTAIN: A Network Model of Category Learning. *Psychol. Rev.* **111,** 309–332 (2004).

15. Mack, M. L., Love, B. C. & Preston, A. R. Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proc. Natl. Acad. Sci.* **113,** 13203–13208 (2016).

16. Davis, T., Love, B. C. & Preston, A. R. Learning the exception to the rule: Model-based fMRI reveals specialized representations for surprising category members. *Cereb. Cortex* **22,** 260–273 (2012).

17. Mack, M. L., Love, B. C. & Preston, A. R. Building concepts one episode at a time: The hippocampus and concept formation. *Neurosci. Lett.* (2017). doi:10.1016/j.neulet.2017.07.061

18. Nadel, L. & Moscovitch, M. Memory consolidation, retrograde amnesia and the hippocampal complex. *Current Opinion in Neurobiology* **7,** 217–227 (1997).

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

20. Quian Quiroga, R. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience* **13,** 587–597 (2012).

21. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **43,** 59–69 (1982).

22. Lloyd, S. P. Least Squares Quantization in PCM. *IEEE Trans. Inf. Theory* **28,** 129–137 (1982).

23. Hales, T. *et al.* A formal proof of the Kepler conjecture. *Forum Math. Pi* **5,** 1–29 (2017).

24. Krupic, J., Burgess, N. & O'Keefe, J. Neural representations of location composed of spatially periodic bands. *Science* **337,** 853–857 (2012).

25. Perez-Escobar, J. A., Kornienko, O., Latuske, P., Kohler, L. & Allen, K. Visual landmarks sharpen grid cell metric and confer context specificity to neurons of the medial entorhinal cortex. *Elife* **5,** (2016).

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

27. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20,** 1643–1653 (2017).

Acknowledgments: The authors thank all members of the LoveLab for valuable input. We thank Roddy Grieves for his advice on analysis of grid measures.

Funding: This work was funded by the National Institutes of Health [grant number 1P01HD080679]; the Leverhulme Trust [grant number RPG-2014-075]; and Wellcome Trust Senior Investigator Award [WT106931MA] to Bradley C. Love.

Author contributions: R.M.M.: conceptualization, formal analysis, methodology, software, visualization, writing – original draft, writing – review & editing. B.C.L.: conceptualization, formal analysis, funding acquisition, methodology, resources, supervision, visualization, writing – original draft, writing – review & editing.

Competing interests: Authors declare no competing interests.

Data and materials availability: All code for the simulations, simulated data, plotting scripts are available at https://osf.io/2dz3x/.

ONLINE METHODS

A simulation run comprised of a learning period with a million trials (training phase) where clusters updated their positions in relation to the agent's position as it explored the environment. After learning, we quantitatively assessed the regularity of the cluster position arrangements (test phase). We ran 1000 simulation runs for each condition (number of clusters).

Simulation procedure and model specifications

At the beginning of the learning phase of each simulation run, we set the number of clusters, number of learning trials, the environment (square, circle), the learning rate, and the learning update batch size. The number of clusters were set (ranging from 10 to 30) and were initiated at random locations in the environment. The shape of the environment was defined by a set of points that could be visited by the agent. The square environment was 50 by 50, where each point was a location specified by a value on the x- and y-axis. The circular environment was defined by drawing a circle in Matlab with a radius of 50, and selecting the points within the bounds of the circle. The starting position and movement trajectory of the agent was then determined as a random walk over one million trials. The agent started at a random position and steps in the horizontal and vertical axes were computed separately. On each trial, the agent could go up, down, or stay on the vertical axis, and left, right, or stay on the horizontal axis. The step was sampled from [-4, -2, -1, -1, 0, 1, 1, 2, 4], where negative values are steps to the left, positive steps are steps to the right, and zero means stay. Movement on the

vertical dimension was determined in the same way, but negative values were upward steps and positive values were downwards steps. If the generated step brought the agent out of the environment, the step was cancelled and a new step was generated as above.

We considered a simple winner-take-all network in which only the cluster at position $\boldsymbol{pos_i}$ closest to stimulus $\boldsymbol{x}$ (agent's location) had a non-zero activation. Bold type is reserved for vectors. The distance between $\boldsymbol{pos_i}$ and $\boldsymbol{x}$ is defined as:

$$\boldsymbol{dist_i} = \|\boldsymbol{pos_i} - \boldsymbol{x}\| \tag{1}$$

In the Kohonen learning rule, cluster i updates its position $\boldsymbol{pos_i}$ to move toward stimulus $\boldsymbol{x}$ according to:

$$\Delta\boldsymbol{pos_i} = \eta_t \cdot (\boldsymbol{x} - \boldsymbol{pos_i}) \tag{2}$$

where η_t is the learning rate at time t. In the present simulations, we used batch updating to increase numerical stability in which 200 updates were performed simultaneously. The learning rate η for batch time t followed an annealing schedule:

$$\eta_t = \frac{\eta_0}{1 + \rho \cdot t} \tag{3}$$

where η_0 is the initial learning rate set to 0.25 and ρ is the annealing rate set to 0.02.

Assessing regularity of cluster positions

To assess whether cluster positions formed a regular hexagonal structure with learning in a comparable manner way to grid cells found in the medial entorhinal cortex (mEC), we followed the method of Hafting et al.[7] and Perez-Escobar et al.[25].

In Hafting et al.[7], rodents traversed circular and square environments whilst they recorded electrophysiological signals from mEC neurons. They found cells that displayed multiple firing fields and resembled a grid of regularly tessellating triangles spanning the recorded environment. To assess this regularity quantitatively, they computed the spatial autocorrelogram of the firing rate map. If the fields were arranged in a regular grid, the center peak of the autocorrelogram should be surrounded by six equidistance peaks, forming a regular hexagon. The spatial autocorrelogram was computed as follows. With $\lambda_1(x, y)$ denoting the cluster activation at location (x, y), the autocorrelation with spatial lags of τ_x and τ_y was estimated as:

$$r(\tau_x, \tau_y) = \frac{n \sum \lambda_1(x,y)\lambda_2(x-\tau_x, y-\tau_y) - \sum \lambda_1(x,y) \sum \lambda_2(x-\tau_x, y-\tau_y)}{\sqrt{n \sum \lambda_1(x,y)^2 - (\sum \lambda_1(x,y))^2} \times \sqrt{n \sum \lambda_2(x-\tau_x, y-\tau_y)^2 - (\sum \lambda_2(x-\tau_x, y-\tau_y))^2}} \quad (4)$$

where $r(\tau_x, \tau_y)$ is the autocorrelation between bins offset of τ_x and τ_y, $\lambda_1(x, y)$ and $\lambda_2(x, y)$ are equivalent for an autocorrelation indicates the average firing rate of the cell in each location (x, y), and n is the number of spatial bins over which the estimation was made.

To quantify the degree of this regularity, a 'grid score' is commonly used[25] by computing the correlation between the center region of the spatial autocorrelogram (a masked region including the six surrounding peaks but excluding the centre peak) and a 60° and 120° rotated version (to assess the six-fold hexagonal symmetry) minus the correlation between the spatial autocorrelograms and a 30°, 90°, and 150° rotated version (where there should be a low correlation):

$$\frac{(r_{60^\circ} + r_{120^\circ})}{2} - \frac{(r_{30^\circ} + r_{90^\circ} + r_{150^\circ})}{3} \quad (5)$$

To assess the regularity of the cluster positions in a given environment in the current study and compare our results with empirical findings, we followed the method described above. We first computed activation maps to emulate firing rate maps in empirical neuronal recordings, and computed the spatial autocorrelogram to obtain the grid score.

Assessing change in gridness with learning and gridness after learning

To characterize how cluster positions changed over time in the learning phase, activation maps were computed over trials during learning in a set of 200 simulation runs. Trials were binned into 20 equally spaced time bins with 50,000 trials in each time bin. We assumed that the activation strength of the winning cluster was a Gaussian function of distance from the agent:

$$act_i = \frac{1}{\sqrt{2\pi^2}} e^{-\frac{1}{2} dist_i{}^2} \quad (6)$$

where act_i is cluster i's activation strength. To compute activation maps for each time bin, activations were computed at each location and normalized by the number of visits by the agent (as done in empirical studies) to create a normalized activation map. The maps were smoothed (Gaussian kernel, SD=1), spatial autocorrelograms were computed, and grid scores were computed for each time bin. As the clusters moved continuously over time (not defined by the time bins), activation maps changed over each time bin.

To test whether gridness increased over time, we used a linear model to estimate the slope (beta value) of the grid score of activation maps over each time bin (20 bins) for each simulation run during the learning phase. For each condition (number of clusters), we estimated the slope for 200 simulation runs, giving 200 beta values. We computed the mean and bootstrapped 95% confidence intervals (CIs) over all conditions and simulation runs to test if the grid score increased over time. We also computed the mean and bootstrapped 95% confidence intervals (CIs) over the 200 beta values for each condition.

To assess gridness at the end of learning, a new movement trajectory was generated with 100,000 trials and cluster positions were fixed. Grid score after learning was assessed for all 1000 simulation runs. The activations and normalized activation map were computed over all test trials, the activation map was smoothed (Gaussian kernel, SD=1) and the spatial autocorrelogram of the activation map was computed following

Hafting et al.[7], except firing rates were replaced with normalized cluster activation values at each location. Grid scores were then computed based on the spatial autocorrelograms using equation (5). We computed the mean grid scores and bootstrap 95% CIs over all conditions and simulation runs. We also computed the mean and bootstrap 95% CIs over each condition.

Classification and percentage of grid cell-like activation maps after learning

To assess whether activation maps showed a regular hexagonal pattern that would be classified as a 'grid cell' according to criteria set in empirical studies, and to compare the percentage of grid-like activation maps from our clustering model to the percentage of grid cells found in the mEC, we used a shuffling procedure to find the statistical threshold of the grid score that passes the criterion for a 'grid cell' described in Wills et al.[28].

The procedure was performed on spatial autocorrelograms of the activation maps produced on the test phase, where cluster positions were fixed. Since cluster activations were generated in relation to the agent's location during movement, they were temporally correlated. Therefore, to break the location-activation relationship, the vector of activations were randomly shuffled in time, and we ensured that each location was at least 20 trials from its original position. The activation map was smoothed (Gaussian kernel, SD=1) then the grid score was computed. For each condition, this shuffling procedure was performed 500 times on each simulation run (on a subset of 200

simulations). The threshold was defined as the 95th percentile of the 500 shuffled grid scores, giving 200 threshold values (from each simulation run) per condition (number of clusters). The highest threshold value (most conservative) was used as the threshold for each condition. In the figure in the main text (Fig. 3D and 3E), the thresholds plotted are the highest (most conservative) thresholds across all conditions in that particular environment.

For each condition, we computed the percentage of activation maps that exceeded the shuffled grid score threshold. We computed the percentage of 'grid cells' for each condition (number of clusters) separately and then computed the mean percentage across conditions.

Gridness in trapezoid environments

To simulate the effect of asymmetric boundaries in a trapezoid enclosure on gridness[26], we took cluster positions from simulations after learning in square environments, and ran an additional learning phase for 250,000 trials. In this new learning phase, the shape of the environment was now a trapezoid (the agent could only move to those locations), and the annealed learning rate schedule continued (starting at 0.0025, reducing to 0.002 at the end). The trapezoid dimensions were 5 x 24 x 50 pixels, closely matching the proportions in *(25)* (0.2 x 0.9 x 1.9 meters; multiplied by (50/1.9) equals to 5.26, 23.7, and 50).

In order to test whether the asymmetric boundaries of the trapezoid affected gridness, the trapezoid was split into two halves and we computed the grid score for the spatial autocorrelogram on the left (wide) and right (narrow) side of the shape. Due to discretization, we split it as close to equal as possible. The wide half extended from the leftmost pixels to the 17th pixel (338 pixels), and the narrow side extended from the 18th pixel to the 50th pixel (339 pixels).

Due to the asymmetrical shape of the trapezoid environment, the procedure for generating a movement trajectory above leads to a slightly biased sampling of the wide part of the trapezoid, and less exploration of the middle and top parts of the shape. To deal with this, we made a slight change to the possible steps after generating a step that brings the agent out of the environment, described below. For each trial, the step was generated as before. If the generated step was out of the environment, the step was cancelled, and the next step was determined as follows. If the step generated would have brought the agent out of the bottom of the trapezoid, the next step was sampled from [0,0,1,1] (stay or up). If the step brings the agent out to the top, the next step was sampled from [-1,-1,0,0] (down or stay). When the step takes the agent out of the left of the trapezoid, then the next step to be sampled on the horizontal axis were [0,1,1,2,4], towards the inner portion of the environment. If the step took the agent out of the right side of the trapezoid, the next step was generated as before, from [-4,-2,-1,-1,0,1,1,2,4]. This is because when the agent is out of the trapezoid on the horizontal (left-right) axis, the agent could still be in the middle of the shape on the vertical axis,

since the shape becomes more narrow as it reaches the right. Finally, when it lands exactly in the middle of the horizontal axis, but is out of the shape (on the horizontal axis), the next step to be sampled from on the vertical axis is [-1,0,1].

SUPPLEMENTARY RESULTS

Comparable grid-like cell proportions with empirical data

Our simulations produce activations maps that emulate spatial cells in the mEC with multi-peaks spatial fields, so the appropriate test for the proportion of grid cells is the number of grid cells relative to the total number of spatial cells with multiple peaks. Most studies report percentage of grid cells in relation to all cell types (including head-direction cells, border cells, etc.), but only a few reported and quantified the number of non-grid spatial cells, or if they are multi-peaked or not.

Krupic et al.[24] tested rodents in a square environment and reported the percentage of non-grid spatial cells relative to grid cells, and also the non-grid spatial cells that were periodic using a Fourier analysis. In the mEC population they found 26% grid cells, and 44% non-grid spatial cells with multiple peaks, which means they found (26/(26+44))x100 = 37% grid cells relative to non-grid spatial cells. Using the Fourier analysis method on only spatial and head-direction cells (ignoring other cells in the population), they found 35% grid cells and 43% non-grid periodic spatial cells, and 2% conjunctive grid cells. This amounts to (35/(35+43))x100 = 45%, or (37/(37+43))x100 = 46% grid cells with respect to non-grid periodic grid cells, matching to our 45.3% value in the square environment. Perez-Escobar et al.[25] tested rodents in a circular environment and also report the number of non-grid spatial cells, finding 139 grid cells and 226 non-grid spatial cells, meaning they found (139/(139+226))*100 = 38% grid cells, matching our 38.6% value in the circular environment.

REFERENCES

28. Wills, T. J., Cacucci, F., Burgess, N. & O'Keefe, J. Development of the hippocampal cognitive map in preweanling rats. *Science* **328,** 1573–1576 (2010).

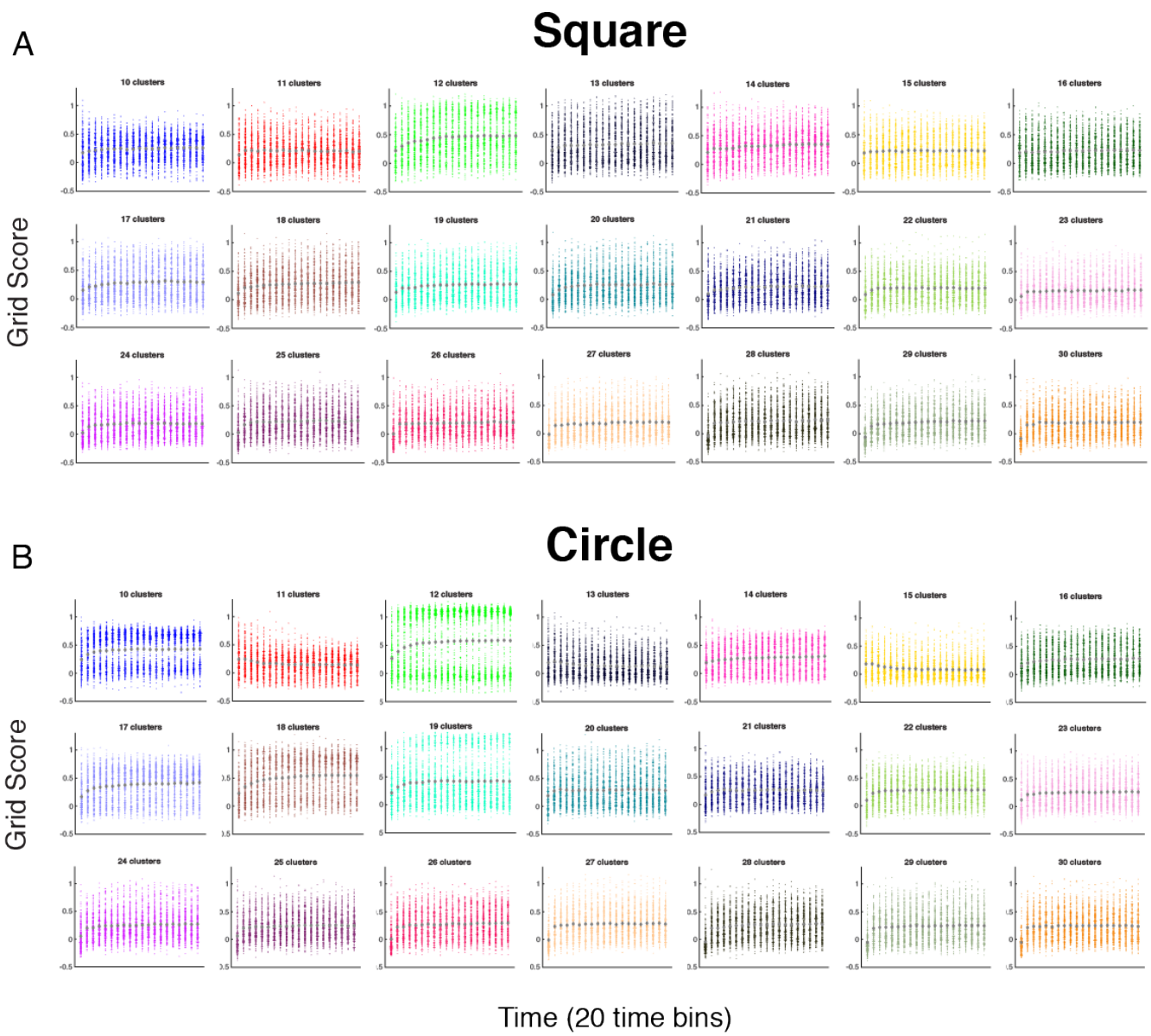

Fig. S1.

Univariate scatterplots showing grid scores increasing over learning in the square (A) and circle (B) for all conditions.

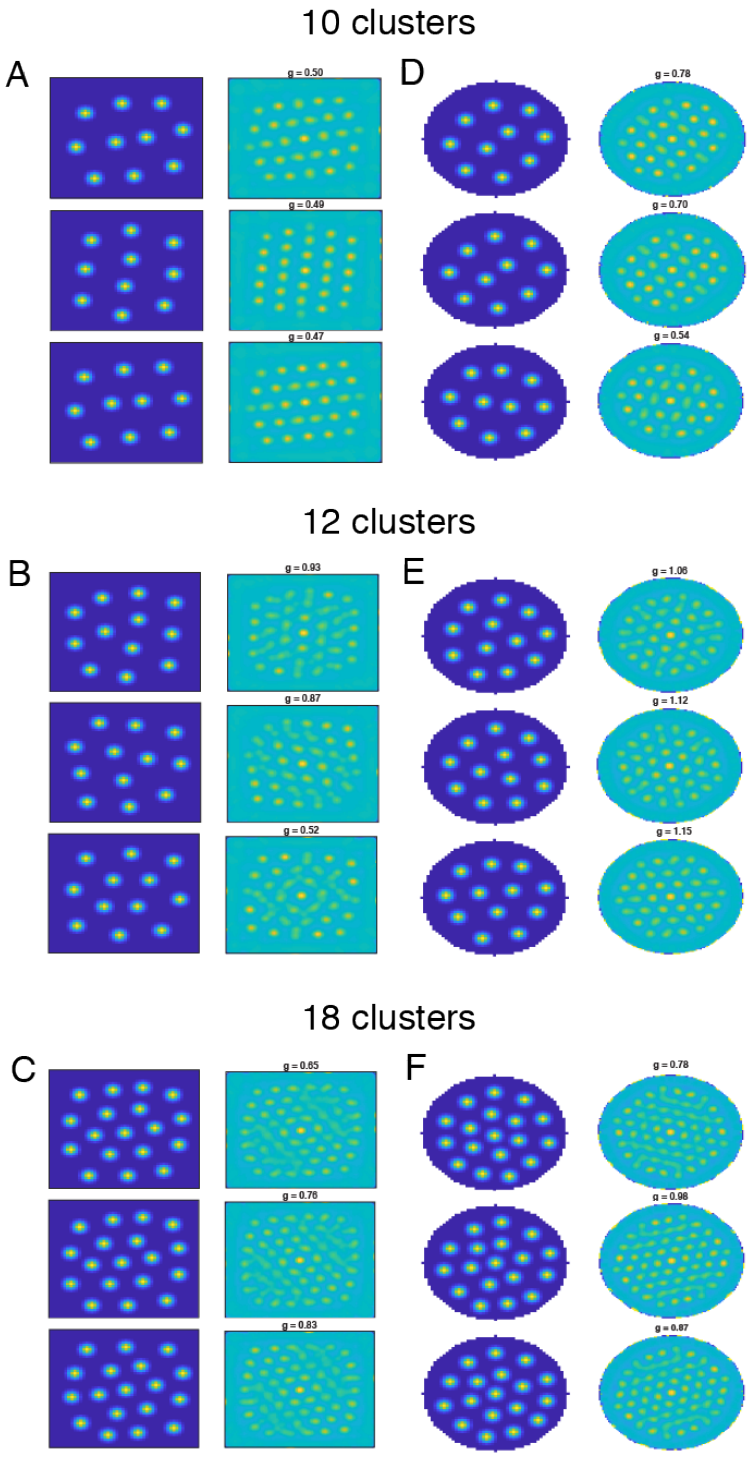

Fig. S2.

Examples of activation maps with grid patterns (left) and their corresponding spatial autocorrelograms (right) in square (A-C) and circular (D-F) environments with 10, 12, and 18 clusters.

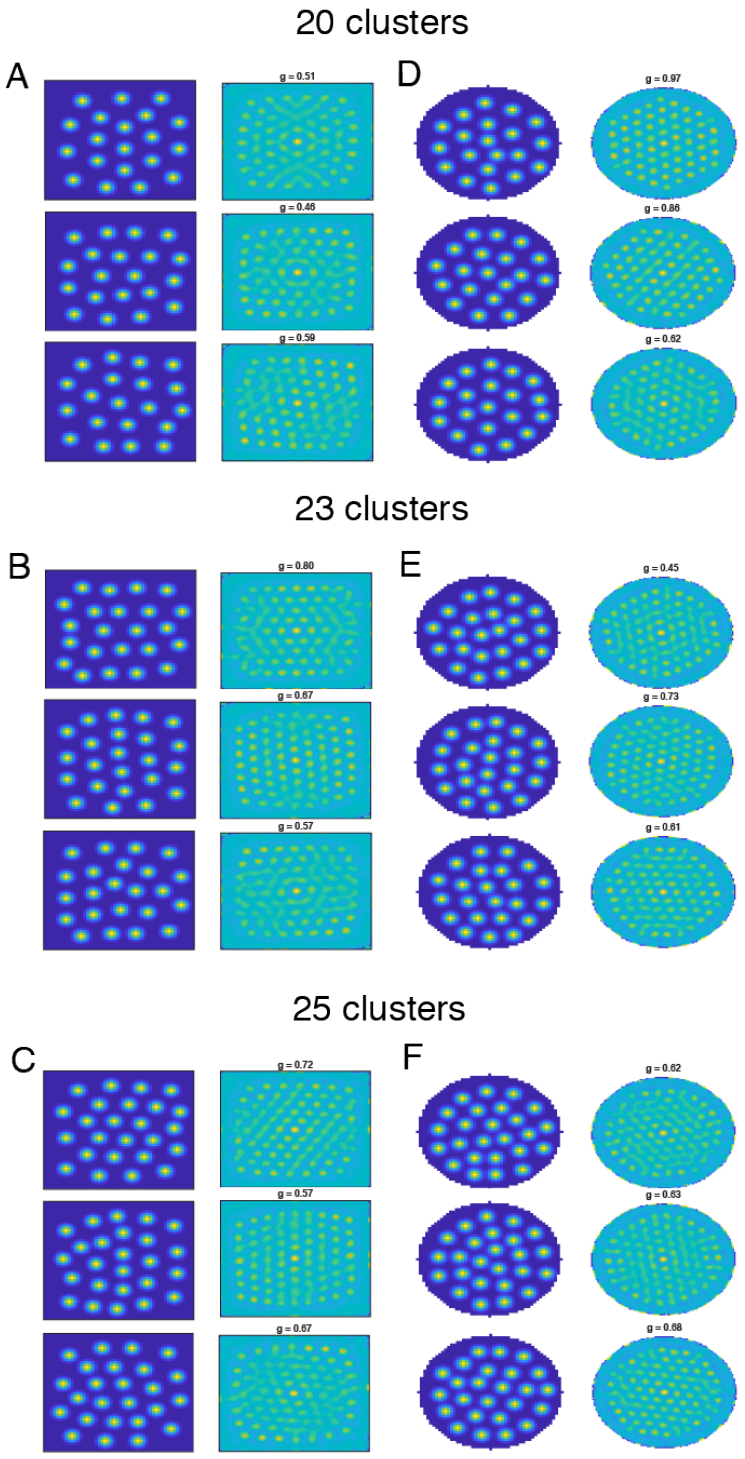

Fig. S3.

Examples of activation maps with grid patterns (left) and their corresponding spatial autocorrelograms (right) in square (A-C) and circular (D-F) environments with 20, 23, and 25 clusters.

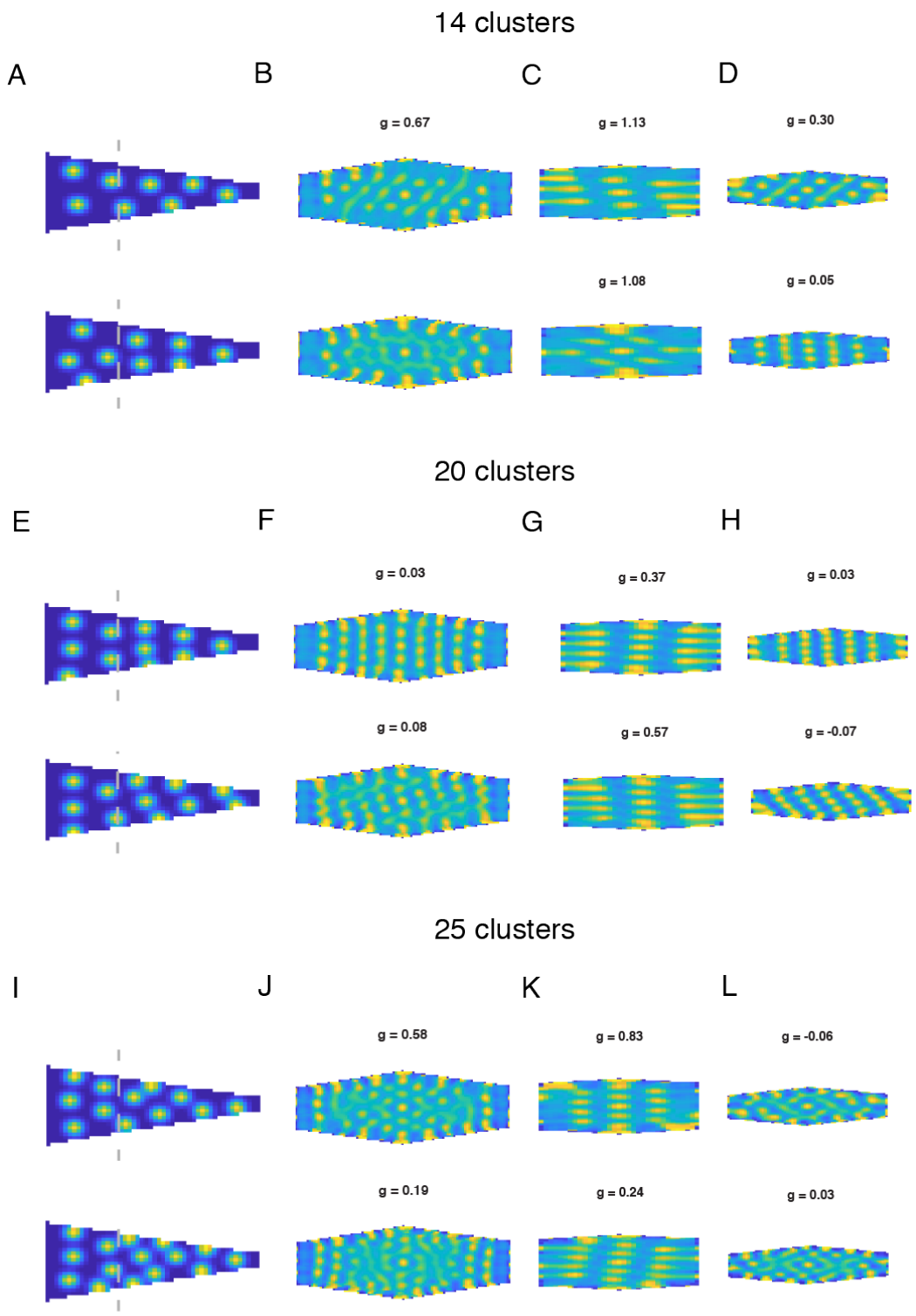

Fig. S4.

Examples of activation maps in the trapezoid environment and their corresponding spatial autocorrelograms with (A-D) 14, (E-H) 20, and (I-L) 25 clusters. (A) Activation

map with 14 clusters, (B) spatial autocorrelogram of the full trapezoid, (C) spatial autocorrelogram of the wide (left) half of the trapezoid, and (D) spatial autocorrelogram of the narrow (right) half of the trapezoid. Same conventions in (E-H) and (I-L).

	Mean	bootstrap CIs
10 clusters	0.0032	[0.0017, 0.0050]
11 clusters	0.0003	[-0.0011, 0.0021]
12 clusters	0.0097	[0.0075, 0.0123]
13 clusters	0.0041	[0.0020, 0.0060]
14 clusters	0.0058	[0.0037, 0.0083]
15 clusters	0.0011	[-0.0006, 0.0031]
16 clusters	0.0021	[-0.0001, 0.0035]
17 clusters	0.0053	[0.0038, 0.0071]
18 clusters	0.0060	[0.0041, 0.0080]
19 clusters	0.0048	[0.0028, 0.0070]
20 clusters	0.0051	[0.0035, 0.0073]
21 clusters	0.0044	[0.0029, 0.0067]
22 clusters	0.0020	[0.0005, 0.0038]
23 clusters	0.0027	[0.0010, 0.0043]
24 clusters	0.0035	[0.0024, 0.0051]
25 clusters	0.0046	[0.0030, 0.0064]
26 clusters	0.0045	[0.0032, 0.0059]
27 clusters	0.0056	[0.0040, 0.0073]
28 clusters	0.0059	[0.0045, 0.0077]
29 clusters	0.0072	[0.0058, 0.0088]
30 clusters	0.0052	[0.0038, 0.0065]

Table S1.

Mean slopes bootstrap confidence intervals (CIs) for learning over time in the square environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.0051	[0.0035, 0.0066]
11 clusters	-0.0044	[-0.0060, -0.0032]
12 clusters	0.0104	[0.0081, 0.0131]
13 clusters	-0.0038	[-0.0047, -0.0024]
14 clusters	0.0044	[0.0031, 0.0059]
15 clusters	-0.0048	[-0.0060, -0.0033]
16 clusters	0.0042	[0.0029, 0.0059]
17 clusters	0.0081	[0.0065, 0.0096]
18 clusters	0.0113	[0.0091, 0.0135]
19 clusters	0.0056	[0.0033, 0.0084]
20 clusters	0.0019	[0.0003, 0.0039]
21 clusters	0.0017	[0.0003, 0.0033]
22 clusters	0.0042	[0.0026, 0.0057]
23 clusters	0.0038	[0.0023, 0.0052]
24 clusters	0.0057	[0.0040, 0.0073]
25 clusters	0.0043	[0.0031, 0.0057]
26 clusters	0.0067	[0.0053, 0.0081]
27 clusters	0.0057	[0.0045, 0.0069]
28 clusters	0.0075	[0.0059, 0.0091]
29 clusters	0.0060	[0.0046, 0.0073]
30 clusters	0.0050	[0.0032, 0.0061]

Table S2.

Mean slopes bootstrap CIs for learning over time in the circular environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.2291	[0.2162, 0.2414]
11 clusters	0.1821	[0.1679, 0.1976]
12 clusters	0.4685	[0.4478, 0.4905]
13 clusters	0.3755	[0.3541, 0.3943]
14 clusters	0.3554	[0.3407, 0.3702]
15 clusters	0.2205	[0.2037, 0.2373]
16 clusters	0.2130	[0.1952, 0.2301]
17 clusters	0.2891	[0.2703, 0.3068]
18 clusters	0.3544	[0.3370, 0.3734]
19 clusters	0.2979	[0.2789, 0.3149]
20 clusters	0.2941	[0.2760, 0.3102]
21 clusters	0.2398	[0.2245, 0.2551]
22 clusters	0.2315	[0.2167, 0.2483]
23 clusters	0.2118	[0.1967, 0.2288]
24 clusters	0.2315	[0.2166, 0.2473]
25 clusters	0.2568	[0.2398, 0.2738]
26 clusters	0.2644	[0.2492, 0.2834]
27 clusters	0.2729	[0.2559, 0.2898]
28 clusters	0.2862	[0.2702, 0.3040]
29 clusters	0.2785	[0.2621, 0.2933]
30 clusters	0.2549	[0.2390, 0.2692]

Table S3.

Mean grid scores and bootstrap CIs in the square environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.4154	[0.3961, 0.4339]
11 clusters	0.1071	[0.0961, 0.1184]
12 clusters	0.5691	[0.5320, 0.6081]
13 clusters	0.1003	[0.0893, 0.1133]
14 clusters	0.2975	[0.2803, 0.3147]
15 clusters	0.0896	[0.0776, 0.1022]
16 clusters	0.2859	[0.2693, 0.3050]
17 clusters	0.4905	[0.4699, 0.5086]
18 clusters	0.5854	[0.5625, 0.6063]
19 clusters	0.4141	[0.3901, 0.4424]
20 clusters	0.3430	[0.3238, 0.3623]
21 clusters	0.2894	[0.2720, 0.3060]
22 clusters	0.2893	[0.2737, 0.3060]
23 clusters	0.2854	[0.2686, 0.3008]
24 clusters	0.2767	[0.2624, 0.2933]
25 clusters	0.3013	[0.2851, 0.3171]
26 clusters	0.3036	[0.2876, 0.3203]
27 clusters	0.3029	[0.2887, 0.3197]
28 clusters	0.2944	[0.2792, 0.3102]
29 clusters	0.2873	[0.2715, 0.3022]
30 clusters	0.2519	[0.2369, 0.2674]

Table S4.

Mean grid scores and bootstrap CIs in the circular environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.0293	[0.0118, 0.0470]
11 clusters	-0.0097	[-0.0263, 0.0054]
12 clusters	-0.0871	[-0.0997, -0.0745]
13 clusters	-0.1177	[-0.1302, -0.1035]
14 clusters	-0.0369	[-0.0550, -0.0204]
15 clusters	0.1096	[0.0921, 0.1282]
16 clusters	0.1581	[0.1414, 0.1774]
17 clusters	0.1074	[0.0912, 0.1233]
18 clusters	0.0678	[0.0525, 0.0837]
19 clusters	0.0309	[0.0161, 0.0466]
20 clusters	0.0322	[0.0181, 0.0468]
21 clusters	0.0332	[0.0179, 0.0505]
22 clusters	0.0459	[0.0279, 0.0633]
23 clusters	0.0568	[0.0385, 0.0751]
24 clusters	0.0916	[0.0723, 0.1094]
25 clusters	0.0981	[0.0793, 0.1143]
26 clusters	0.1095	[0.0902, 0.1272]
27 clusters	0.1250	[0.1092, 0.1415]
28 clusters	0.1392	[0.1232, 0.1538]
29 clusters	0.1120	[0.0973, 0.1275]
30 clusters	0.1172	[0.1012, 0.1330]

Table S5.

Mean grid scores and bootstrap CIs in the trapezoid environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.1998	[0.1754, 0.2234]
11 clusters	0.1918	[0.1749, 0.2096]
12 clusters	0.5556	[0.5300, 0.5807]
13 clusters	0.4931	[0.4671, 0.5162]
14 clusters	0.3924	[0.3687, 0.4185]
15 clusters	0.1110	[0.0869, 0.1365]
16 clusters	0.0550	[0.0301, 0.0760]
17 clusters	0.1818	[0.1551, 0.2073]
18 clusters	0.2865	[0.2599, 0.3132]
19 clusters	0.2670	[0.2427, 0.2900]
20 clusters	0.2619	[0.2386, 0.2869]
21 clusters	0.2066	[0.1851, 0.2283]
22 clusters	0.1856	[0.1628, 0.2068]
23 clusters	0.1549	[0.1309, 0.1779]
24 clusters	0.1398	[0.1183, 0.1615]
25 clusters	0.1587	[0.1360, 0.1832]
26 clusters	0.1549	[0.1335, 0.1775]
27 clusters	0.1479	[0.1231, 0.1680]
28 clusters	0.1470	[0.1241, 0.1693]
29 clusters	0.1666	[0.1448, 0.1874]
30 clusters	0.1377	[0.1170, 0.1594]

Table S6.

Mean grid difference scores and bootstrap CIs between the square and trapezoid environment for each condition.

	Mean	bootstrap CIs
10 clusters	0.0179	[-0.0128, 0.0444]
11 clusters	0.1503	[0.1263, 0.1765]
12 clusters	0.1270	[0.1063, 0.1492]
13 clusters	0.0965	[0.0786, 0.1151]
14 clusters	0.0784	[0.0540, 0.1013]
15 clusters	0.0767	[0.0491, 0.1089]
16 clusters	0.1451	[0.1141, 0.1784]
17 clusters	0.2387	[0.2069, 0.2691]
18 clusters	0.3434	[0.3106, 0.3761]
19 clusters	0.3020	[0.2723, 0.3361]
20 clusters	0.2424	[0.2131, 0.2700]
21 clusters	0.2161	[0.1875, 0.2423]
22 clusters	0.1515	[0.1237, 0.1768]
23 clusters	0.0999	[0.0746, 0.1278]
24 clusters	0.1108	[0.0848, 0.1388]
25 clusters	0.1122	[0.0834, 0.1392]
26 clusters	0.0789	[0.0508, 0.1055]
27 clusters	0.0635	[0.0359, 0.0940]
28 clusters	0.0714	[0.0421, 0.1013]
29 clusters	0.0197	[-0.0096, 0.0495]
30 clusters	0.0499	[0.0214, 0.0754]

Table S7.

Mean grid difference scores and bootstrap CIs between the wide and narrow portion of the trapezoid environment for each condition.