No Free Lunch from Deep Learning in Neuroscience: A Case Study through Models of the Entorhinal-Hippocampal Circuit

Rylan Schaeffer [1 2] **Mikail Khona** [3 4] **Ila Rani Fiete** [2 4]

Abstract

Research in Neuroscience, as in many scientific disciplines, is undergoing a renaissance based on deep learning. Unique to Neuroscience, deep learning models can be used not only as a tool but interpreted as models of the brain. The central claims of recent deep learning-based models of brain circuits are that they make novel predictions about neural phenomena or shed light on the fundamental functions being optimized. We show, through the case-study of grid cells in the entorhinal-hippocampal circuit, that one often gets neither. We begin by reviewing the principles of grid cell mechanism and function obtained from analytical and first-principles modeling efforts, then rigorously examine the claims of deep learning models of grid cells. Using large-scale hyperparameter sweeps and theory-driven experimentation, we demonstrate that the results of such models may be more strongly driven by particular, non-fundamental, and post-hoc implementation choices than fundamental truths about neural circuits or the loss function(s) they might optimize. Finally, we discuss why these models cannot be expected to produce accurate models of the brain without the addition of substantial amounts of inductive bias, an informal No Free Lunch result for Neuroscience. In conclusion, caution and consideration, together with biological knowledge, are warranted in building and interpreting deep learning models in Neuroscience.

[1]Computer Science, Stanford University [2]Brain and Cognitive Sciences, Massachusetts Institute of Technology [3]Physics, Massachusetts Institute of Technology [4]McGovern Institute for Brain Research. Correspondence to: Rylan Schaeffer <rschaef@cs.stanford.edu>.

2nd AI4Science Workshop at the 39th International Conference on Machine Learning (ICML), 2022. Copyright 2022 by the author(s).

1. Introduction

Within the past decade, deep learning (DL) has leapt from obscurity to underpinning nearly every success story in machine learning, e.g., Mnih et al. (2015); Silver et al. (2016); Brown et al. (2020) and increasingly many advances in fundamental science research, e.g., Jumper et al. (2021); Davies et al. (2021); Bellemare et al. (2020). In neuroscience, deep learning is similarly gaining widespread adoption as a useful method for behavioral and neural data analysis (Sussillo et al., 2016; Saif-ur Rehman et al., 2019; Luxem et al., 2020; Pereira et al., 2019; Glaser et al., 2020; Livezey et al., 2019; Kim et al., 2021; Mathis et al., 2018).

But DL offers a unique contribution to neuroscience that goes beyond its role in other fields, in that neural networks can be viewed as models of the brain. The success of DL in matching or surpassing human performance means it is now possible to construct models of circuits that underlie human intelligence. As a recent review wrote, "Many researchers are excited by the possibility that deep neural networks may offer theories of perception, cognition and action for biological brains. This approach has the potential to radically reshape our approach to understanding neural systems" (Saxe et al., 2021).

Broadly, the essential claims of DL-based models of the brain are that 1) Because the models are trained on a specific optimization problem, if the resulting representations match what has been observed in the brain, then they reveal the optimization problem of the brain, or 2) That these models, when trained on sensibly motivated optimization problems, should make novel predictions about the brain's representations and emergent behavior.

However, given the nascent nature of such approaches and the excitement accompanying the claims, we should examine them carefully. In deep learning and deep reinforcement learning, performance improvements attributed to novel objective functions and algorithms have instead been shown to stem from seemingly minor and often-unstated implementation choices (Tucker et al., 2018; Henderson et al., 2019; Engstrom et al., 2020; Ilyas et al., 2020). Similar criticism has been raised in the context of Neuroscience (Hosseini et al., 2020; Schulz et al., 2020; Abrol et al., 2021). In this

paper, we add to this debate and ask whether Neuroscientists should similarly be cautious that DL-based models of neural circuits may tell us less about fundamental scientific truths and more about programmers' particular implementation choices, and ask whether these approaches are more post-hoc than predictive.

To explore these questions, we evaluate recent DL-based models of grid cells in the entorhinal-hippocampal circuit. The medial entorhinal cortex (MEC) and hippocampus (HPC) are part of the hippocampal formation, a system that displays beautiful and fascinating properties. In 1971, HPC was shown to contain **place cells**, neurons which fire if and only if the animal is at particular location(s) (O'Keefe & Dostrovsky, 1971). Later, MEC was shown to contain **grid cells**, neurons which fire if and only if the animal is at the vertices of a hexagonal lattice (Hafting et al., 2005). As a matter of terminology, we refer to cells that are periodically active at the vertices of either a square or hexagonal lattice as **lattice cells**, and refer to hexagonal lattice cells as **grid cells**. Over the past 40 years, the hippocampal formation has proved a rich vein for learning about how the brain organizes spatial and episodic memory, for experimentalists and theorists alike (Section 2), with many mysteries remaining. A recent series of papers (Cueva & Wei, 2018; Banino et al., 2018; Sorscher et al., 2019; Whittington et al., 2020; Nayebi et al., 2021)) have used DL to present a story that **path integration (PI)** (i.e., the task of estimating one's absolute spatial position in an environment by integrating one's velocity estimates) drives the formation of grid cells.

In this paper, we use code from prior publications to demonstrate that these results are due entirely to implementation details that tell us more about those choices than they do about MEC. Specifically, by leveraging theoretically-guided large-scale hyperparameter exploration and hypothesis-driven experimentation, we demonstrate that:

1. Networks trained on a path integration task almost always learn to optimally encode position, but almost never learn lattice cells (hexagonal or square) to do so.
2. The emergence of lattice cells depends wholly on a specifically chosen encoding of the supervised target, not on the task itself.
3. The grid periods and period ratios likewise depend on hyperparameters choices and are not set by the task.
4. The chosen encoding requires many other hyperparameter choices to produce lattice cells that are highly sensitive, in that small alterations result in loss of lattice cells.

Deep learning produces grid cells and their attendant properties only after making many specific design choices and searching hyperparameter space to obtain such representations, baking grid cells into the task-trained networks.

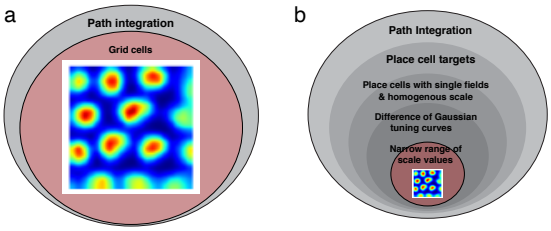

Figure 1. (a) Prior work (Cueva & Wei, 2018; Banino et al., 2018; Sorscher et al., 2019; Whittington et al., 2020; Nayebi et al., 2021) implicitly or explicitly suggests that the objective of path integration generically drives the formation of grid cells: in other words, that training ANNs on a path integration objective is sufficient to generate grid cells. (b) In this work, we show how only a very small select fraction of hyperparameter space yields grid cells, and therefore that ANN grid cell emergence results are post-hoc: they result from tuning hyperparameters to obtain grid cells.

Our main message is that it is highly improbable that DL models of path integration would have produced grid cells as a novel prediction simply from task-training, had grid cells not already been known to exist. Moreover, it is unclear what added intepretability or understanding these models contribute, beyond or even up to what has already been shown for these particular circuits. These results challenge the notion that deep networks offer a free lunch for Neuroscience in terms of discovering the brain's optimization problems or generating novel a priori predictions about single-neuron representations, and warn that caution is needed when building and interpreting such models.

We emphasize that our work would not have been possible without the open-source code released by previous publications, for which the authors should be commended; by making their code available, we have been able to present novel insights that we hope will contribute to a clearer understanding of the risks and rewards of using and interpreting DL models in Neuroscience.

2. Background and related work

Grid cells (Hafting et al., 2005) are specialized neurons found in the medial entorhinal cortex (MEC) of mammals that are tuned to represent the spatial location of the animal as it freely traverses 2D space. Each cell fires at every vertex of a triangular lattice that tiles the explored space, regardless of the speed and direction of movement through the space. As a population, grid cells exhibit several striking properties that provide support for a specialized circuit; here we list some of the properties most relevant to testing grid cell models: 1) The spatial period of the periodic grid response is independent of the spatial environment's size and shape

for familiar spaces. 2) The response is updated even in the dark, in the absence of external cues. 3) Grid cells form discrete modules (clusters), such that all cells within a module share a common period and orientation, with monotonically increasing grid scales along the dorso-ventral axis of the MEC (Stensola et al., 2012). Adjacent modules exhibit specific period ratios whose values are all close to 1.4. 4) The grid cells within each module exhibit stable cell-cell relationships across environments and during sleep (Yoon et al., 2013; Trettel et al., 2019; Gardner et al., 2019a; 2022), unlike place cells, which remap across environments and do not exhibit preserved structure during sleep.

The mechanism underlying grid cells is believed to be described by continuous attractor models (Burak & Fiete, 2009b; Fuhs & Touretzky, 2006; Burak & Fiete, 2006). They operate according to an elegant principle: Translation-invariant lateral connectivity within the grid cell network results in Turing instability and pattern formation. These models explain how velocity inputs can be converted into updated spatial estimates by grid cells and make several predictions that have been confirmed in experiments: 1) Stable cell-cell relationships regardless of environment (Yoon et al., 2013; Trettel et al., 2019; Gardner et al., 2019a; 2022); 2) A toroidal attractor manifold in neural state space across awake and sleep states (Gardner et al., 2022; 2019b).

The counterintuitive encoding of position by grid cells (a local, non-periodic variable such as location encoded by a non-local, periodic code) has led to theoretical examination of the properties of the population of grid cells (oftentimes called the grid code), with results showing that the grid code provides three properties not shared with many conventional (unimodal or grandmother cell-like) neural codes: 1) Equi-norm translation-invariant representations for path integration; 2) Exponentially large capacity of internally generated states as a function of neuron number (Fiete et al., 2008; Sreenivasan & Fiete, 2011; Mathis et al., 2012b); 3) Intrinsic error correcting code (Sreenivasan & Fiete, 2011).

3. Experimental Setup

Path integration (PI) is the task of using self-velocity estimates to track one's spatial position over time, a crucial component of spatial navigation. The prominent message of DL models of grid cells is that training recurrent networks to path integrate causes the networks to learn lattice cells (Cueva & Wei, 2018; Banino et al., 2018; Sorscher et al., 2019; Whittington et al., 2020; Nayebi et al., 2021).

Many of the previous papers use the following experimental setup. A 2-D bounded spatial environment is created, oftentimes a 2.2 m x 2.2 m open arena. Then, spatial trajectories (i.e. sequences of positions and velocities) are sampled. Networks receive as inputs the initial position and the sequence

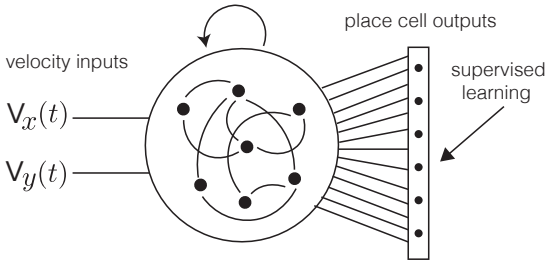

Figure 2. The setup of the path integration task: A single layer recurrent neural network with 4096 hidden units receives two-dimensional velocity inputs and is trained in a supervised manner to output (an encoding of) the two-dimensional position.

of velocities from a trajectory, and are trained to output the sequence of positions (Fig. 2) in a supervised manner. However, there are multiple ways of encoding the position, and as we shall show, this choice is critical. Two simple encodings of position are Cartesian (Cueva & Wei, 2018) or polar. Another common encoding scheme is via place cells (PCs), which works by sampling a large population of PCs with positions uniformly distributed in the environment, and then computing each place cell's activity at each position based on a particular function; two common functions are a Gaussian (Banino et al., 2018; Sorscher et al., 2019) and a Difference of Gaussians (Sorscher et al., 2019; Nayebi et al., 2021). If a PC encoding is used, the networks are trained to predict the high dimensional vector of PC activities. For all encodings, supervised learning is used to train the network via backpropagation through time.

Ratemaps are the primary method used to compare the hidden units of artificial networks to grid cells. They are computed using the following procedure: A trained network is evaluated on longer trajectories that roughly uniformly cover the 2-dimensional environment. As the network integrates instantaneous velocity inputs and outputs the positional encoding (typically a place cell code), each hidden unit's activity is binned against the true position in the 2-dimensional environment. For more details, see Appendix B.

4. Networks trained on path integration tasks learn to optimally encode position but rarely learn lattice (hexagonal or square) cells

As Sorscher et al. (2019) wrote in their section titled "Optimally encoding position yields diverse grid-like patterns," "Why do these diverse architectures, across diverse tasks (both navigation and autoencoding), all converge to a grid-like solution, and what governs the lattice structure of this solution?" We shall demonstrate that most networks do not converge to a grid-like solution, instead requiring very

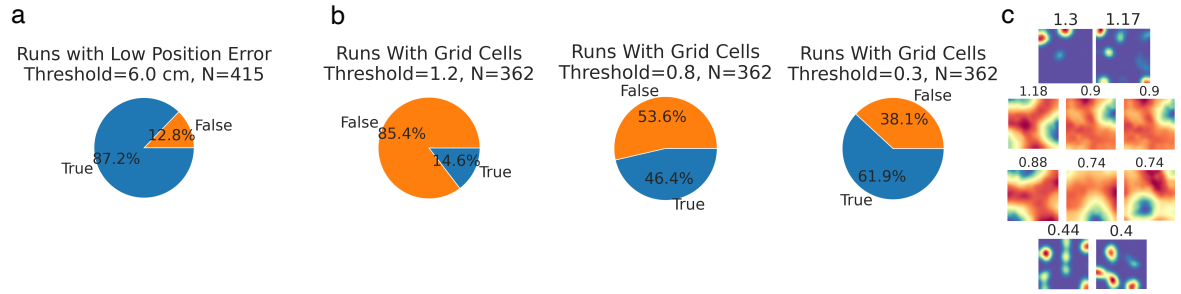

Figure 3. (a) Almost all configurations of loss function, target encoding and hyperparameter choices lead to networks that solve the path integration problem (i.e. achieve low position decoding error) but (b) very few networks learn grid-like cells. We use three different grid score thresholds: 0.3, 0.8 and 1.2. (c) We found that grid scores, though established in the field as the way to select grid cells, are a sub-optimal metric for identify grid cells because of the relatively high rate of false positives: They pick up on spurious features of ratemaps like triangular symmetry to produce a high grid score, as shown. Unlike previous approaches that used lax thresholds like 0.3, we find that a threshold of 1.2, though it does not solve all the problems of the grid score, is more appropriate; however, our results and conclusions remain qualitatively unchanged even with a lax threshold of 0.3.

specific hyperparameter choices, and what governs the structure (when it emerges) is determined entirely by what the programmer bakes in.

We ran large-scale hyperparameter sweeps across common implementation choices: 1) Network Architectures: RNN (Elman, 1990); LSTM (Hochreiter & Schmidhuber, 1997); GRU (Chung et al., 2014); UGRNN (Collins et al., 2017) 2) Activation: Linear; Sigmoid; Tanh; Rectified Linear 3) Optimizers: SGD, Adam (Kingma & Ba, 2017); RMSProp (Hinton et al.) 4) Supervised Targets: Cartesian spatial position; high-dimensional Place Cell (PC) population code with Gaussian tuning curves (Banino et al., 2018); high-dimensional PC population code with **Difference-of-Gaussians (DoG)** tuning curves (Sorscher et al., 2019; Nayebi et al., 2021) 5) Loss: mean squared error (MSE) on the agent's Cartesian spatial position (Kanitscheider & Fiete, 2016; Cueva & Wei, 2018); geodesic distance on the agent's polar spatial position (Kanitscheider & Fiete, 2017a); softmax cross entropy on a high-dimensional population of place cell (PC) units (Banino et al., 2018; Sorscher et al., 2019; Nayebi et al., 2021) 6) Miscellaneous: recurrent dropout, readout dropout, weight regularization, randomization seed.

When training networks on supervised place cell (PC) targets, we additionally swept: 1) The place cells' receptive field σ i.e. the standard deviation of the Gaussian tuning curve (often denoted $\sigma_E \stackrel{\text{def}}{=} \sigma$ in the literature); 2) Whether the PC population's fields have homogeneous or heterogeneous receptive fields; 3) In the case of PC targets with DoG tuning curves, the place cell's surround scale s i.e. the ratio between the inhibitory and excitatory Gaussian's standard deviations (often denoted $s \stackrel{\text{def}}{=} \sigma_I/\sigma_E$ in the literature); 4) The number of fields per place cell.

Evaluating the entire hyperparameter volume is computationally prohibitive, so we evaluated a subvolume we considered most consistent with previous approaches as well as most illustrative of previous approaches' limitations; complete details of every run sweep configurations are provided in Appendix C.

To evaluate whether a network learns to optimally encode its position, we measured the network's position decoding error using the same methods as prior works (Sorscher et al., 2019; Nayebi et al., 2021). Specifically, we computed position decoding error in a 2.2 m x 2.2 m environment using the networks' output Cartesian positions (if trained on Cartesian position targets) or using the networks' decoded positions based on their predicted PC population activity (if trained on PC targets). Any network was considered to have achieved optimal position encoding if its position decoding error fell below 6 cm; this threshold was chosen based on noise inherent in the position decoding algorithm.

In total, we trained 415 networks and found that almost every hyperparameter configuration succeeds in learning to path integrate (87.2%; Fig. 3a), but only few learn lattice cells at all (14.6% of the 87.2%, or 12.7% overall) representations (Fig. 3b). This is consistent with earlier work (Kanitscheider & Fiete, 2016; 2017a) demonstrating that networks can learn to path integrate and solve other navigational problems (e.g. estimating which of several environments correspond to the current location) without lattice cells emerging as a solution.

5. Lattice cell emergence requires a highly specific choice of supervised target encoding

We next sought to characterize when lattice cells are learnt if the target output encodings are changed. We tested three different encodings of position in 2d space: a) Cartesian

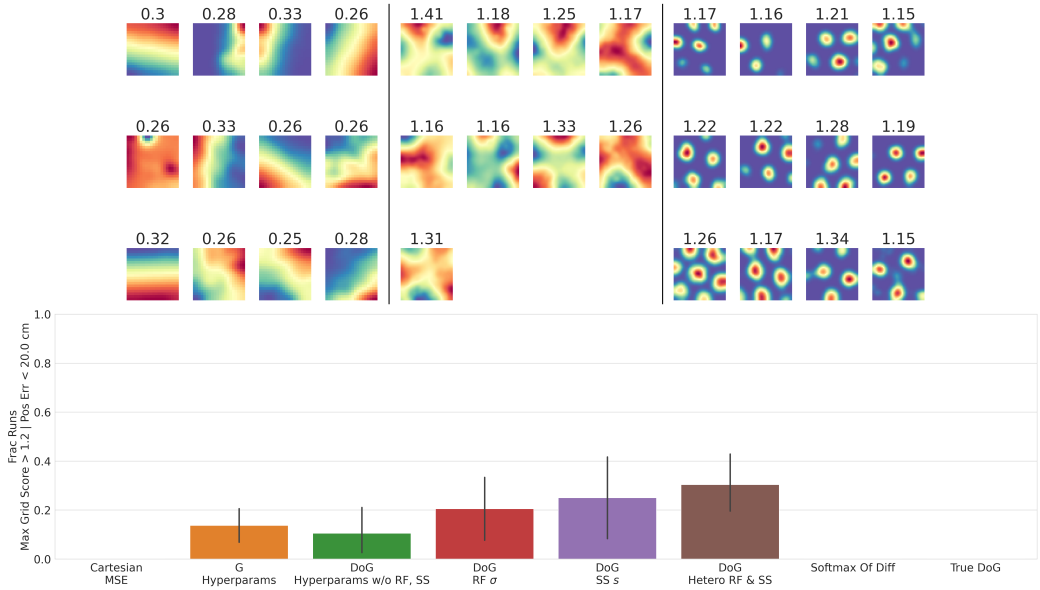

Figure 4. (a) Rate maps of highest-scoring units in deep networks trained on i) Cartesian readouts, ii) Gaussian readouts, iii) specifically selected (tuned) DoG readouts. i)-ii) do not learn any lattice cells. (b) Only networks trained on Difference-of-Gaussian (DoG) place cell tuning curves display lattice-like cells.

readouts b) Gaussian readouts c) a very specifically selected Difference-of-Gaussian (DoG) readout. We found that (cf. Fig. 4) grid cells do not emerge from Cartesian and Gaussian readout encodings, consistent with an earlier study done by some of the authors of the current paper (Kanitscheider & Fiete, 2017b). Only by making the positional readout encoded by a DoG shape of tuning curve sometimes resulted in lattices (Fig. 4ab).

5.1. Grid periods are parameter-dependent and multiple modules do not usually emerge

Next, a prominent feature of grid cells that is critical for their unambiguous encoding of position over large scales is the existence of a discrete set of grid periodicities, which tend to scale by a rough factor of 1.4 between adjacent scales (Stensola et al., 2012). We asked whether ANN models generate multiple periods and when they did so, what hyperparameters and other choices the formed periods depended on.

To ensure we would obtain at least some grid cells, we fixed the readouts to be DoGs, and swept over different scales of the place cell DoG. We found that almost all runs had a unimodal distribution of grid periods (Fig. 5a), meaning the networks learnt only one module of grid cells.

Further, we found that the period of the formed grid-like representation is completely determined by the scale of the externally imposed readout DoG (Fig. 5). The period the grid-like responses in every run increased monotonically with the width of the DoG readout (Fig. 5b). Since the models did not result in multiple modules in a single network, we used the somewhat discrete distribution of peaks of the single module formed when sweeping the DoG parameter more continuously to compute grid period ratios. These period ratios from adjacent peaks led to non-biological values (Fig. 5c).

5.2. Fourier analysis of Turing instability explains the preceding empirical results

Why do only Difference-of-Gaussians (DoG) place cell targets produce lattice cells? The reason is the same as for pattern-forming models of grid cells which showed that a difference of Gaussians recurrent interaction is sufficient to generate periodic activity patterns (Burak & Fiete, 2009b) in recurrent network models of grid cells, and can be understood through Fourier analyses (Burak & Fiete, 2009b; Khona et al., 2021). We restate the essence of the analyses here, and explain why it explains the current results. In an RNN with dynamics $\dot{r}(x) = -r(x) + g(W \star r)$, where x designates the neural index (in a continuum approximation for neurons), $W \star r$ designates the total (integrated) inputs from the network at the neuron at location x, and g is the neural non-linearity, let the recurrent weight interactions W be given by:

$$f(\Delta x) = \alpha_E \exp\left(-\frac{(\Delta x)^2}{2\sigma_E^2}\right) - \alpha_I \exp\left(-\frac{(\Delta x)^2}{2\sigma_I}\right) \quad (1)$$

where Δx refers to the difference of index between the neural pair linked by the weights. The Fourier transform of

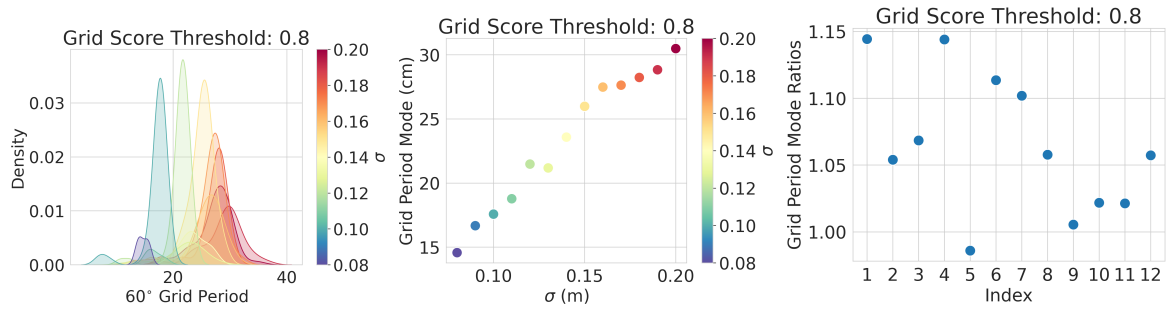

Figure 5. (a) Over a wide sweep of DoG target field widths σ, the distribution of grid periods is unimodal (each color is the distribution of perods obtained from one run), meaning multiple grid modules do not emerge, in contrast to biological grid cells. (b) The chosen target field with σ determines the grid period mode, meaning that hand-designed parameter choices, not an intrinsic emergent property, sets the grid period. (c) If we use grid period modes obtained from smoothly sweeping σ as a model for the formation of different modules, the period ratios are closer to 1 than to 1.4, the approximate experimental ratio values.

this interaction is given by

$$\tilde{f}(k) = \alpha_E \sigma_E \exp\left(-\frac{\sigma_E^2 k^2}{2}\right) - \alpha_I \sigma_I \exp\left(-\frac{\sigma_I^2 k^2}{2}\right) \tag{2}$$

Here α_E (α_I) denotes the strength and σ_E (σ_I) denotes the scale of excitation (inhibition). For linearized dynamics that approximate $\dot{r}(x) \sim -r(x) + f(\Delta x) \star r$ (i.e. g has been linearized), the solution will be periodic if the maxima of $\tilde{f}(k)$, given by $[k^*]^2 = \frac{2}{\sigma_E^2 - \sigma_I^2} \log\left(\alpha_E \sigma_E^3 / \alpha_I \sigma_I^3\right)$, contains sufficient power and if $k^* \neq 0$. Specifically, the condition for pattern formation is $\tilde{f}(k^*) > 1$ (Burak & Fiete, 2009a; Khona et al., 2021). In particular, the inhibitory surround contained in $f(\Delta x)$, with strength σ_I, is key to pattern formation; if $\sigma_I \to \infty$ or $\alpha_I \to 0$, the maximum is at the origin ($k^* = 0$), causing no pattern formation. Therefore, a Gaussian interaction cannot produce periodic patterns.

The theory in Sorscher et al. (2020); Dordek et al. (2016) similarly shows that the Fourier transform of the desired readout unit activity (in particular, its second-moment matrix) sets the activity of the RNN units. This readout matrix takes on the role of the recurrent interaction matrix of the continuous attractor grid cell model described above (Burak & Fiete, 2009b). The conditions for a pattern forming solution in supervised training-based RNN setup for obtaining grid cells are therefore the same as described above. That is, for Gaussian readout functions ($\alpha_I = 0$), the peak of their Fourier spectrum is at the origin, causing DC modes to dominate and no lattices to form regardless of the width σ_E. Thus, Gaussian readout positional encodings are predicted to not produce grid cells. Through a similar analysis, Carteisan readouts will also not produce grid cells. And only specifically tuned DoG readouts produce grid cells under the architectures considered here and in Sorscher et al. (2020). Our large-scale hyperparameter sweeps confirm this.

5.3. The importance of unexplored implementation details

A seemingly minor implementation detail that is not mentioned in the main texts or supplements (to the best of our knowledge) in several of the preceding papers (Sorscher et al., 2019; Nayebi et al., 2021; Sorscher et al., 2020) also proves critical to the emergence of grid cells. These papers refer to a Difference-of-Gaussian (DoG) supervised target function (Eqn. 1), but we discovered their code uses something different: a **Difference-of-Softmaxes (DoS)**. Specifically, the tuning curves are given by:

$$f(\Delta x_p) = \text{Softmax}\left(-\frac{(\Delta x_p)^2}{2\sigma_E^2}\right) - \text{Softmax}\left(-\frac{(\Delta x_p)^2}{2\sigma_I^2}\right) \tag{3}$$

where the softmaxes are taken over the place cell population (indexed here by p). This leads to an effective non-uniform averaging of the outputs. Everything previously labeled DoG should actually be labeled DoS; we keep the authors' terminology in the previous sections, but change the terminology to DoS moving forward to distinguish the previous "DoG" (actually DoS) from the true DoG.

To understand why, recall from Section 5.2 that for lattice cells to emerge, Fourier power (i.e. the eigenvalues of the place cell second-moment matrix $\Sigma = P^T P$) must peak on an annulus of a sufficiently large radius; if the radius is too small, lattice cells will not emerge. With DoS instead of DoG, the maximal eigenvalues of the second-moment matrix fall on an annulus with radius 3 (Fig. 7a), which is clearly separated from the origin, giving rise to lattice cells. But a DoG with the same parameters would produce an annulus radius of 1 (Fig. 7c), too small for lattice cells to emerge. We trained ideal grid-forming ReLU RNN networks on supervised PC targets with true DoG tuning curves, sweeping the receptive field σ and surround scale s. We found that using true Difference-of-Gaussian tuning curves did not result in grid cells (Fig. 7b).

In sum, the simple objective of only asking for accurate spatial representation through path integration is *not* sufficent for obtaining grid cells, and requires very specific additional choices.

6. Grid cells disappear with heterogeneously tuned readouts units

Experimentally recorded place cells differ significantly from overly simplified single-scale, single-field homogeneous Gaussian functions (and as we have noted above, are even further from being DoG functions). Many place cells have multiple fields (Rich et al., 2014; Eliav et al., 2021; Souza et al., 2018). This naturally leads to the question: Will readout target functions with heterogeneous scales or multiple fields per place cell still produce grid cells?

Because the Fourier analysis in the above argument is applicable when the population of place cells encode the readout in an approximately translationally-invariant manner (this happens when the place cell population is distributed isotropically in the room, and every place cell has a single peak of the same scale), we cannot predict the outcome of heterogeneous readouts.

To maximally favor the chances of the ANN approach to lead to the formation of lattice responses, we allow as before for the readouts to have DoS profiles. We tested what effect heterogeneous DoS scales have on the formation of grid cells. We trained ideal grid-forming networks (RNN, ReLU, $\sigma = 0.12$ cm, $s = 2.0$, 4 seeds) on heterogeneous DoS PC supervised targets with receptive field $\sigma \sim \text{Uniform}(0.06, 0.18)$ cm and surround scale $s \sim \text{Uniform}(1.5, 2.5)$; these distributions were chosen to ensure the expected values were ideal for grid cell formation. We found heterogeneity in the PC scales prevents the formation of grid cells (Fig. 6 left). Specifically, heterogeneity causes a clear decrease in the bulk of the networks' grid scores, as well as a decrease in the high grid score tail. Of the few remaining units (13 / 2048) with high grid scores, the distribution of grid periods remained unimodal (Fig. 6 right), showing that grids of multiple scales are not learnt. The networks trained on heterogeneous PC targets achieved equivalent position decoding error as networks trained on homogeneous PC targets, further demonstrating that optimal position encoding does not require lattice representations.

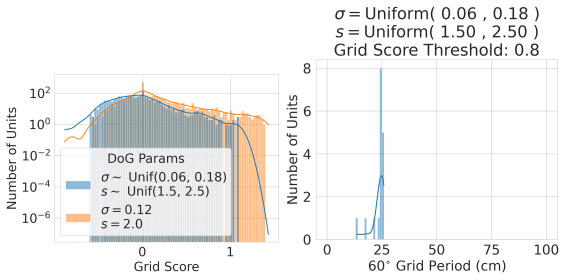

Figure 6. **Heterogeneous place cells prevent the formation of grid cells.** Even with Difference-of-Gaussian (DoG) place cell targets and non-negative network nonlinearities, adding heterogeneity to the DoG scale hyperparameters (receptive field σ and surround scale s) causes grid cells to disappear (left). The few remaining grid cells still exhibit a single period, rather than multiple periods (right). 3 runs per condition.

7. Why ANNs that path integrate achieve high predictivity of MEC data

We conclude by introducing a puzzle. A recent NeurIPS 2021 spotlight (Nayebi et al., 2021) claimed that networks trained on single-field single-scale Difference-of-Softmax targets, which we have seen produce very few lattice cells and produce only unimodally distributed grid period distributions explain variance in mouse MEC neural activity at nearly 100% of the level of variance explained by MEC activity from other mice. However, given our results demonstrating that these networks display properties inconsistent with biological grid cells (e.g., single grid periods, square lattices), and require supervised targets that are inconsistent with place cells, how are these networks able to predict mouse MEC neural activity so well?

The data preprocessing and analysis code is not public, so we are unable to investigate this question in detail. However, we offer a conjecture with preliminary supporting evidence. The analysis of Nayebi et al. (2021) linearly regressed rate maps from one agent (mouse or network) onto rate maps from another mouse, and used Pearson correlation as a measure of "neural predictivity." We conjectured that different combinations of architectures and activations achieve different neural predictivity scores because different architecture-activation combinations learn dynamical systems of different intrinsic dimensionalities that then provide richer or poorer bases for linear regressions.

To explore our conjecture, we trained the networks considered in Nayebi et al. (2021): 4 architectures (RNN, LSTM, GRU, UGRNN) and 4 activations (Linear, Sigmoid, Tanh, ReLU), adding 5 random seeds (0 through 4) per architecture-activation pair. For each trained network, we computed a standard linear measure of the intrinsic dimensionality (called participation ratio (Litwin-Kumar et al., 2017)) of the network's rate maps, and then plotted each network's participation ratio against the published neural predictivity score. We chose a linear measure of the networks' intrinsic dimensionalities since linear regression cannot fit nonlinear patterns in the data. We found a clear trend that networks with higher (lower) dimensional rate maps have higher (lower) neural predictivity scores (Fig. 8).

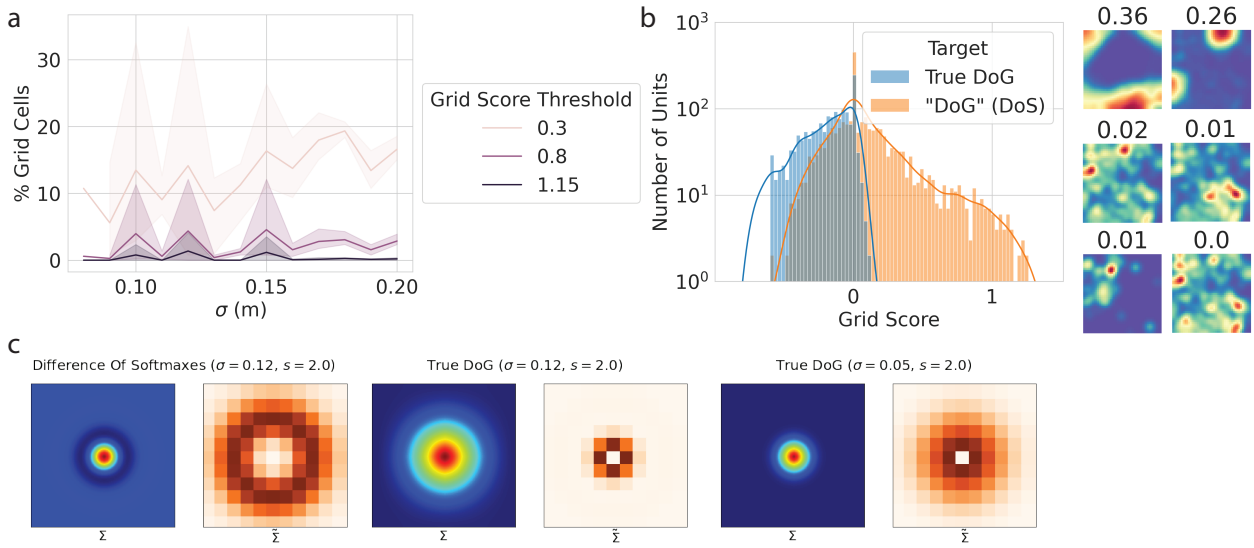

Figure 7. **Other changes to Difference-of-Gaussians affect the formation of grid cells:** (a) The existence of grid solutions even with DoG readout encodings is highly sensitive to parameters such as the target function receptive field σ, and small alterations result in the disappearance of grid cells regardless of grid score threshold. (b) (left) Comparison of grid scores of trained networks with true DoG shows that the DoG with normalization, as used by Sorscher et al. (2019); Nayebi et al. (2021); Sorscher et al. (2020), is critical for grid cell formation, (right) Ratemaps of the highest scoring cells from the true DoG networks. (c) Computing the Fourier transform (right of each pair) of the place cell second-moment matrix (left of each pair) explains why the particular choice of normalization affects representations.

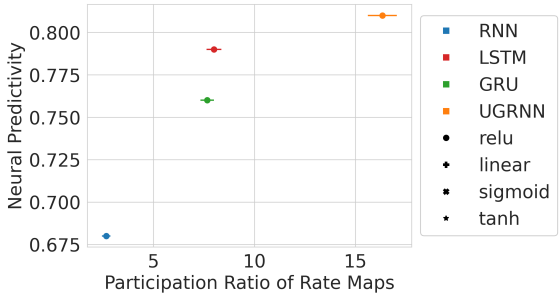

Figure 8. **Networks with higher (lower) dimensional rate maps display higher (lower) "neural predictivity" of mouse MEC rate maps.** This suggests that linear regression-based comparisons of artificial and biological networks may find "better" networks simply because those networks provide higher dimensional bases for the regressions. Each dot is an architecture-activation pair, with participation ratio (Litwin-Kumar et al., 2017) averaged over 5 runs. Neural predictivity scores from Fig. 2A of Nayebi et al. (2021).

We caution that this correlation between network dimensionality and neural predictivity is not strong evidence, but we are unable to investigate further without access to preprocessing and analysis code. If our conjecture is correct, it raises the concern that linear regression-based comparisons of biological networks and deep artificial neural networks, widely used and cited in vision, language and audition, may also lead to artefactual conclusions. DL-based models may not actually be close models of biological neural circuits, but rather may merely provide higher-dimensional bases than alternative models and thus trivially achieve higher correlation scores.

8. Discussion

For research that uses deep networks as models of the brain, there is a fundamental obstacle to making the claim that a given optimization problem is what the brain is solving: If we know the responses of a significant fraction of units from biological networks performing a certain task, we cannot infer the loss function that the brain is optimizing since in principle, numerous different loss functions can have the same minima (Fig. 9 top). In other words, there is typically a many-to-one mapping between loss functions and some point in state space where the functions have a minimum. Conversely, given a reasonable optimization problem that we select based on an organism's ecological niche, we cannot infer a single solution (and thus build truly predictive single-cell tuning models), since there exist several minima to that optimization problem (Fig. 9 bottom).

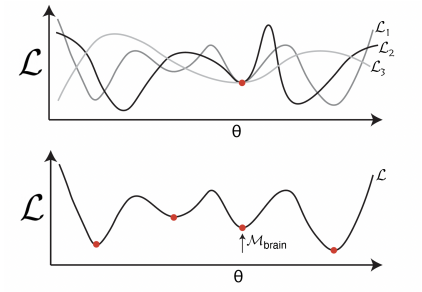

Figure 9. **Challenges in achieving the two central claims of recent DL models of neuroscience**: Top: Building a model that replicates observed neural responses does not guarantee that the loss function used is the brain's objective, as multiple objectives can share a solution. Bottom: Training a network on a plausible loss function or even the correct loss function need not yield the solution the brain has selected because the loss function may have multiple minima, of which the brain selects one based on its constraints, while an ANN selects another, based on the optimization technique used.

In other words, there is typically a one-to-many mapping from a loss function to its set of solutions.

To break this degeneracy of multiple minima and arrive at truly predictive models or a better understanding of the brain's optimization problems, we must acknowledge, understand, and model the host of specific implicit biases and constraints present in the biological system we are trying to model. It is untenable to expect and claim success without doing so. This is what we refer to as an informal neural 'no-free-lunch'.

What can we learn from DL models about brain circuits without considering and studying biological inductive biases? Population-wide low-dimensional latent representations and dynamics that arise as necessary for solving difficult problems are possibly robust enough and abstracted enough to be predictive of population dynamics in a neural circuit without the addition of detailed biological constraints. This can explain the success of the population-level analyses of the visual pathway (Kell et al., 2018; Bashivan et al., 2019), as well as the population-level low-dimensional dynamics of circuits solving inference tasks (Sussillo & Barak, 2013; Schaeffer et al., 2020; Voigts et al., 2022). In these cases, the emergent low-dimensional solutions are fundamental features of any system that solve the task, and by construction need not be specific to brains.

Coming back to grid cells, we have conclusively shown that they do not generically arise in networks trained to path-integrate.

This raises the question of what different additional architectural, hyperparameter, and constraint choices had to be made in previous papers (Banino et al., 2018; Sorscher et al., 2019; 2020) to obtain grid-like tuning, given that they used a path integration objective with Gaussian place cell targets. Indeed, we argue that the question of scientific interest is to explore and carefully characterize the conditions under which a particular tuning does and does not emerge. To this end, the question is what other ingredient(s) should be added to the task of path integration to always and robustly obtain grid cells? Theoretical work on grid cell representations (Fiete et al., 2008; Sreenivasan & Fiete, 2011; Mathis et al., 2012b) suggests that two factors: a very large coding range and the related property of robustness/intrinsic error correcting coding are two important features of the grid cell code, and that adding these properties to the loss are likely to be a more principled way to obtain grid-like fields rather than by hand-designing a specific and not biologically motivated DoG readout. Consistent with this, the very large amount of dropout required in (Banino et al., 2018; Cueva & Wei, 2018) suggests that the coding-theoretic insights on intrinsic error-correction properties of grid cells and their related large capacity may indeed be key ingredients to produce grid cells. There are a number of specific properties of the grid cell code elucidated by theoretical arguments outlined before, which we hypothesize form a sufficient set of biologically and computationally relevant properties for the emergence of grid cells: 1) non-negative activations; 2) equivariant population responses (i.e. the population response always lies on a hypersphere); 3) a path integrating (PI) code or in other words, translation invariant representations (Fuhs & Touretzky, 2006; Burak & Fiete, 2009b); 4) a high representational capacity (Burak & Fiete, 2008; Sreenivasan & Fiete, 2011; Mathis et al., 2012a); 5) intrinsic error correcting capabilities (Burak & Fiete, 2008; Sreenivasan & Fiete, 2011); and finally, 6) uniformly distributed and low spatial information per cell. In other words, the total spatial information of the grid code should be equally distributed across all modules and cells, and this distribution should be roughly equal.

References

Abrol, A., Fu, Z., Salman, M., Silva, R., Du, Y., Plis, S., and Calhoun, V. Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. *Nature Communications*, 12(1): 353, January 2021. ISSN 2041-1723. doi: 10.1038/s41467-020-20655-6. URL https://www.nature.com/articles/s41467-020-20655-6. Number: 1 Publisher: Nature Publishing Group.

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M. J., Degris, T., Modayil, J., Wayne, G., Soyer, H., Viola, F., Zhang, B., Goroshin, R., Rabinowitz, N., Pascanu, R., Beattie, C., Petersen, S., Sadik, A., Gaffney, S., King, H., Kavukcuoglu, K., Hassabis, D., Hadsell, R., and Kumaran, D. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0102-6. URL http://www.nature.com/articles/s41586-018-0102-6.

Bashivan, P., Kar, K., and DiCarlo, J. J. Neural population control via deep image synthesis. *Science*, 364 (6439):eaav9436, May 2019. doi: 10.1126/science.aav9436. URL https://www.science.org/doi/full/10.1126/science.aav9436. Publisher: American Association for the Advancement of Science.

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., and Wang, Z. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2939-8. URL https://www.nature.com/articles/s41586-020-2939-8. Number: 7836 Publisher: Nature Publishing Group.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv: 2005.14165.

Burak, Y. and Fiete, I. Do We Understand the Emergent Dynamics of Grid Cell Activity? *Journal of Neuroscience*, 26(37):9352–9354, September 2006. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2857-06.2006. URL https://www.jneurosci.org/content/26/37/9352. Publisher: Society for Neuroscience Section: Journal Club.

Burak, Y. and Fiete, I. R. Unpublished observations. 2008.

Burak, Y. and Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput Biol*, 5(2):e1000291, Feb 2009a. doi: 10.1371/journal.pcbi.1000291.

Burak, Y. and Fiete, I. R. Accurate Path Integration in Continuous Attractor Network Models of Grid Cells. *PLOS Computational Biology*, 5(2):e1000291, February 2009b. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000291. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000291. Publisher: Public Library of Science.

Campbell, M. G., Ocko, S. A., Mallory, C. S., Low, I. I. C., Ganguli, S., and Giocomo, L. M. Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. *Nature Neuroscience*, 21(8):1096–1106, August 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0189-y. URL https://www.nature.com/articles/s41593-018-0189-y. Number: 8 Publisher: Nature Publishing Group.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS Workshop Deep Learning and Representation Learning*, December 2014. URL http://arxiv.org/abs/1412.3555. Number: arXiv:1412.3555 arXiv:1412.3555 [cs].

Collins, J., Sohl-Dickstein, J., and Sussillo, D. Capacity and Trainability in Recurrent Neural Networks. *International Conference on Learning Representations*, March 2017. URL http://arxiv.org/abs/1611.09913. Number: arXiv:1611.09913 arXiv:1611.09913 [cs, stat].

Cueva, C. J. and Wei, X.-X. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *International Conference on Learning Representations*, pp. 19, 2018.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., Lackenby, M., Williamson, G., Hassabis, D., and Kohli, P. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887): 70–74, December 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-04086-x. URL https://www.nature.com/articles/s41586-021-04086-x. Number: 7887 Publisher: Nature Publishing Group.

Dordek, Y., Soudry, D., Meir, R., and Derdikman, D. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *eLife*, 5:e10094, March 2016. ISSN 2050-084X. doi: 10.7554/eLife.10094. URL https://doi.org/10.7554/eLife.10094. Publisher: eLife Sciences Publications, Ltd.

Eliav, T., Maimon, S. R., Aljadeff, J., Tsodyks, M., Ginosar, G., Las, L., and Ulanovsky, N. Multiscale representation of very large environments in the hippocampus of flying bats. *Science*, 372 (6545):eabg4020, May 2021. doi: 10.1126/science.abg4020. URL https://www.science.org/doi/10.1126/science.abg4020. Publisher: American Association for the Advancement of Science.

Elman, J. L. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 1551-6709. doi: 10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO. *arXiv:2005.12729 [cs, stat]*, May 2020. URL http://arxiv.org/abs/2005.12729. arXiv: 2005.12729.

Fiete, I. R., Burak, Y., and Brookings, T. What grid cells convey about rat location. *J Neurosci*, 28(27):6858–71, Jul 2008. doi: 10.1523/JNEUROSCI.5684-07.2008.

Fuhs, M. C. and Touretzky, D. S. A spin glass model of path integration in rat medial entorhinal cortex. *J Neurosci*, 26(16):4266–4276, 2006. ISSN 1529-2401 (Electronic). doi: 10.1523/JNEUROSCI.4353-05.2006.

Gardner, R. J., Lu, L., Wernle, T., Moser, M.-B., and Moser, E. I. Correlation structure of grid cells is preserved during sleep. *Nat Neurosci*, 22(4):598–608, 04 2019a. doi: 10.1038/s41593-019-0360-0.

Gardner, R. J., Lu, L., Wernle, T., Moser, M.-B., and Moser, E. I. Correlation structure of grid cells is preserved during sleep. *Nature Neuroscience*, 22(4):598–608, April 2019b. ISSN 1546-1726. doi: 10.1038/s41593-019-0360-0. URL https://www.nature.com/articles/s41593-019-0360-0. Number: 4 Publisher: Nature Publishing Group.

Gardner, R. J., Hermansen, E., Pachitariu, M., Burak, Y., Baas, N. A., Dunn, B. A., Moser, M.-B., and Moser, E. I. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, February 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04268-7. URL https://www.nature.com/articles/s41586-021-04268-7. Number: 7895 Publisher: Nature Publishing Group.

Glaser, J. I., Benjamin, A. S., Chowdhury, R. H., Perich, M. G., Miller, L. E., and Kording, K. P. Machine Learning for Neural Decoding. *eNeuro*, 7(4), July 2020. ISSN 2373-2822. doi: 10.1523/ENEURO.0506-19.2020. URL https://www.eneuro.org/content/7/4/ENEURO.0506-19.2020. Publisher: Society for Neuroscience Section: Research Article: Methods/New Tools.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, August 2005. ISSN 1476-4687. doi: 10.1038/nature03721. URL https://www.nature.com/articles/nature03721. Number: 7052 Publisher: Nature Publishing Group.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep Reinforcement Learning that Matters. *arXiv:1709.06560 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1709.06560. arXiv: 1709.06560.

Hinton, G., Srivastava, N., and Swersky, K. Lecture 6e-RMSProp. URL http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Hosseini, M., Powell, M., Collins, J., Callahan-Flintoft, C., Jones, W., Bowman, H., and Wyble, B. I tried a bunch of things: The dangers of unexpected overfitting in classification of brain data. *Neuroscience & Biobehavioral Reviews*, 119:456–467, December 2020. ISSN 0149-7634. doi: 10.1016/j.neubiorev.2020.09.036. URL https://www.sciencedirect.com/science/article/pii/S0149763420305868.

Ilyas, A., Engstrom, L., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. A Closer Look at Deep Policy Gradients. *arXiv:1811.02553 [cs, stat]*, May 2020. URL http://arxiv.org/abs/1811.02553. arXiv: 1811.02553.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B.,

Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Number: 7873 Publisher: Nature Publishing Group.

Kanitscheider, I. and Fiete, I. Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. *arXiv:1609.09059 [q-bio]*, December 2016. URL http://arxiv.org/abs/1609.09059. arXiv: 1609.09059.

Kanitscheider, I. and Fiete, I. Emergence of dynamically reconfigurable hippocampal responses by learning to perform probabilistic spatial reasoning. preprint, Neuroscience, December 2017a. URL http://biorxiv.org/lookup/doi/10.1101/231159.

Kanitscheider, I. and Fiete, I. R. Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017b.

Kell, A. J. E., Yamins, D. L. K., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy. *Neuron*, 98(3):630–644.e16, May 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.03.044. URL https://www.sciencedirect.com/science/article/pii/S0896627318302502.

Khona, M., Chandra, S., and Fiete, I. Spontaneous emergence of topologically robust grid cell modules: A multiscale instability theory. *bioRxiv*, 2021.

Kim, T. D., Luo, T. Z., Pillow, J. W., and Brody, C. D. Inferring Latent Dynamics Underlying Neural Population Activity via Neural Differential Equations. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5551–5561. PMLR, July 2021. URL https://proceedings.mlr.press/v139/kim21h.html. ISSN: 2640-3498.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, January 2017. URL http://arxiv.org/abs/1412.6980. Number: arXiv:1412.6980 arXiv:1412.6980 [cs].

Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H., and Abbott, L. F. Optimal Degrees of Synaptic Connectivity. *Neuron*, 93(5):1153–1164.e7, March 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.01.030. URL https://www.sciencedirect.com/science/article/pii/S0896627317300545.

Livezey, J. A., Bouchard, K. E., and Chang, E. F. Deep learning as a tool for neural data analysis: Speech classification and cross-frequency coupling in human sensorimotor cortex. *PLOS Computational Biology*, 15(9):e1007091, September 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007091. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007091. Publisher: Public Library of Science.

Luxem, K., Fuhrmann, F., Kürsch, J., Remy, S., and Bauer, P. Identifying Behavioral Structure from Deep Variational Embeddings of Animal Motion. Technical report, bioRxiv, October 2020. URL https://www.biorxiv.org/content/10.1101/2020.05.14.095430v2. Section: New Results Type: article.

Mathis, A., Herz, A., and Stemmler, M. Optimal population codes for space: grid cells outperform place cells. *Neural Comp.*, 24:2280–2317, 2012a.

Mathis, A., Herz, A. V. M., and Stemmler, M. Optimal Population Codes for Space: Grid Cells Outperform Place Cells. *Neural Computation*, 24(9):2280–2317, September 2012b. ISSN 0899-7667. doi: 10.1162/NECO_a_00319. URL https://doi.org/10.1162/NECO_a_00319.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, September 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0209-y. URL https://www.nature.com/articles/s41593-018-0209-y. Number: 9 Publisher: Nature Publishing Group.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14236. URL http://www.nature.com/articles/nature14236.

Nayebi, A., Attinger, A., Campbell, M. G., Hardcastle, K., Low, I. I., Mallory, C. S., Mel, G. C.,

Sorscher, B., Williams, A. H., Ganguli, S., Giocomo, L. M., and Yamins, D. L. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. preprint, Neuroscience, November 2021. URL http://biorxiv.org/lookup/doi/10.1101/2021.10.30.466617.

O'Keefe, J. and Dostrovsky, J. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34:171–175, 1971. ISSN 1872-6240. doi: 10.1016/0006-8993(71)90358-1. Place: Netherlands Publisher: Elsevier Science.

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M., and Shaevitz, J. W. Fast animal pose estimation using deep neural networks. *Nature Methods*, 16(1):117–125, January 2019. ISSN 1548-7105. doi: 10.1038/s41592-018-0234-5. URL https://www.nature.com/articles/s41592-018-0234-5. Number: 1 Publisher: Nature Publishing Group.

Rich, P. D., Liaw, H.-P., and Lee, A. K. Large environments reveal the statistical structure governing hippocampal representations. *Science*, 345(6198):814–817, August 2014. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1255635. URL https://www.sciencemag.org/lookup/doi/10.1126/science.1255635.

Saif-ur Rehman, M., Lienkämper, R., Parpaley, Y., Wellmer, J., Liu, C., Lee, B., Kellis, S., Andersen, R., Iossifidis, I., Glasmachers, T., and Klaes, C. SpikeDeeptector: a deep-learning based method for detection of neural spiking activity. *Journal of Neural Engineering*, 16(5):056003, July 2019. ISSN 1741-2552. doi: 10.1088/1741-2552/ab1e63. URL https://doi.org/10.1088/1741-2552/ab1e63. Publisher: IOP Publishing.

Saxe, A., Nelli, S., and Summerfield, C. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, January 2021. ISSN 1471-0048. doi: 10.1038/s41583-020-00395-8. URL https://www.nature.com/articles/s41583-020-00395-8. Number: 1 Publisher: Nature Publishing Group.

Schaeffer, R., Khona, M., Meshulam, L., International, B. L., and Fiete, I. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. *Advances in Neural Information Processing Systems*, 33: 4584–4596, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/30f0641c041f03d94e95a76b9d8bd58f-Abstract.html.

Schulz, M.-A., Yeo, B. T. T., Vogelstein, J. T., Mourao-Miranada, J., Kather, J. N., Kording, K., Richards, B., and Bzdok, D. Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets. *Nature Communications*, 11 (1):4238, August 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18037-z. URL https://www.nature.com/articles/s41467-020-18037-z. Number: 1 Publisher: Nature Publishing Group.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529 (7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961. URL https://www.nature.com/articles/nature16961. Number: 7587 Publisher: Nature Publishing Group.

Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B., and Moser, E. I. Representation of Geometric Borders in the Entorhinal Cortex. *Science*, 322 (5909):1865–1868, December 2008. doi: 10.1126/science.1166466. URL https://www.science.org/doi/10.1126/science.1166466. Publisher: American Association for the Advancement of Science.

Sorscher, B., Mel, G. C., Ganguli, S., and Ocko, S. A. A unified theory for the origin of grid cells through the lens of pattern formation. *Advances in Neural Information Processing Systems*, pp. 18, 2019.

Sorscher, B., Mel, G. C., Ocko, S. A., Giocomo, L., and Ganguli, S. A unified theory for the computational and mechanistic origins of grid cells. Technical report, bioRxiv, December 2020. URL https://www.biorxiv.org/content/10.1101/2020.12.29.424583v1. Section: New Results Type: article.

Souza, B. C., Pavão, R., Belchior, H., and Tort, A. B. L. On Information Metrics for Spatial Coding. *Neuroscience*, 375:62–73, April 2018. ISSN 0306-4522. doi: 10.1016/j.neuroscience.2018.01.066. URL https://www.sciencedirect.com/science/article/pii/S0306452218301088.

Sreenivasan, S. and Fiete, I. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nat Neurosci*, 14(10):1330–7, Sep 2011. doi: 10.1038/nn.2901.

Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., and Moser, E. I. The entorhinal grid

map is discretized. *Nature*, 492(7427):72–78, December 2012. ISSN 1476-4687. doi: 10.1038/nature11649. URL https://www.nature.com/articles/nature11649. Number: 7427 Publisher: Nature Publishing Group.

Sussillo, D. and Barak, O. Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649, March 2013. ISSN 0899-7667, 1530-888X. doi: 10.1162/NECO_a_00409. URL http://www.mitpressjournals.org/doi/10.1162/NECO_a_00409.

Sussillo, D., Jozefowicz, R., Abbott, L. F., and Pandarinath, C. LFADS - Latent Factor Analysis via Dynamical Systems. Technical Report arXiv:1608.06315, arXiv, August 2016. URL http://arxiv.org/abs/1608.06315. arXiv:1608.06315 [cs, q-bio, stat] type: article.

Trettel, S., Trimper, J., Hwaun, E., Fiete, I., and Colgin, L. Grid cell co-activity patterns during sleep reflect spatial overlap of grid fields during active behaviors. *Nat Neurosci*, 22(4):609–617, 04 2019. doi: 10.1038/s41593-019-0359-6.

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R. E., Ghahramani, Z., and Levine, S. The Mirage of Action-Dependent Baselines in Reinforcement Learning. *arXiv:1802.10031 [cs, stat]*, November 2018. URL http://arxiv.org/abs/1802.10031. arXiv: 1802.10031.

Voigts, J., Kanitscheider, I., Miller, N., Toloza, E., Newman, J., Fiete, I., and Harnett, M. Spatial reasoning via recurrent neural dynamics in mouse retrosplenial cortex. *bioRxiv*, 2022.

Whittington, J. C. R., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T. E. J. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183(5):1249–1263.e23, November 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.10.024. URL https://www.sciencedirect.com/science/article/pii/S009286742031388X.

Yoon, K., Buice, M., Barry, C.and Hayman, R., Burgess, N., and Fiete, I. Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat Neurosci*, 16(8):1077–84, Aug 2013. doi: 10.1038/nn.3450.

A. Selecting the Grid Score Threshold

What qualifies as a grid cell? The most commonly used method of quantifying grid cells is via grid score, which functions by binning neural activity into rate maps using spatial position, applying an adaptive smoother, then taking a circular sample of the autocorrelation centered on the central peak and comparing it to rotated versions of the same circular sample. Experimentalists have used thresholds of 0.3 (Solstad et al., 2008) and 0.349 (Campbell et al., 2018) on biological neurons, whereas computationalists have used 0.3 (Nayebi et al., 2021) and 0.37 (Banino et al., 2018; Sorscher et al., 2019) on artificial neurons. For artificial neurons, we believe that these thresholds are far too low. This is because biological neurons are noisy and undersampled, whereas artificial neurons are noiseless and oversampled, and the grid score decreases with missing/noisy rate maps. Consequently, as shown in Fig. 10, it is common for artificial neurons to achieve grid scores above 0.35 without being grid-like. Even artificial neurons with grid scores above 0.8 are arguably not grid cells. After internal disagreement, we compromised at a grid score threshold of 0.8; one author would like to note that they believe artificial neurons should be held to the highest standard possible and argued for a threshold of 1.15.

B. Number of Bins for Computing Rate Maps

The first step in computing grid scores is determining the number of bins to use to compute rate maps. However, establishing the number of bins used by previous publications was challenging. The original experimental work used 5 cm by 5 cm bins (Hafting et al., 2005) in a 2.2 m by 2.2 m arena, meaning the number of bins should be 44 x 44. However, the deep learning papers (Banino et al., 2018; Sorscher et al., 2019; Nayebi et al., 2021) revealed discrepancies between text and code.

Banino et al. (2018)'s text notes using 32 x 32 bins of 6.875 cm x 6.875 cm ("Spatial (ratemaps) and directional activity maps were calculated for individual units as follows. Each point in the trajectory was assigned to a specific spatial and directional bin according to its location and the direction in which it faced. Spatial bins were defined as a 32 × 32 square grid spanning each environment"), but their code [1] used 20 x 20 bins of 11 cm x 11 cm. Sorscher et al. (2019) noted using 2 cm x 2 cm bins ("Grid score was evaluated as in Banino et al. (2018). A spatial ratemap was computed for each neuron by binning the agent's position into 2 cm x 2 cm bins, and computing the average firing rate within each bin.") but their code [2] similarly used 20 x 20 bins.

[1] https://github.com/deepmind/grid-cells/blob/master/train.py#L201

[2] https://github.com/ganguli-lab/grid-pattern-formation/blob/master/

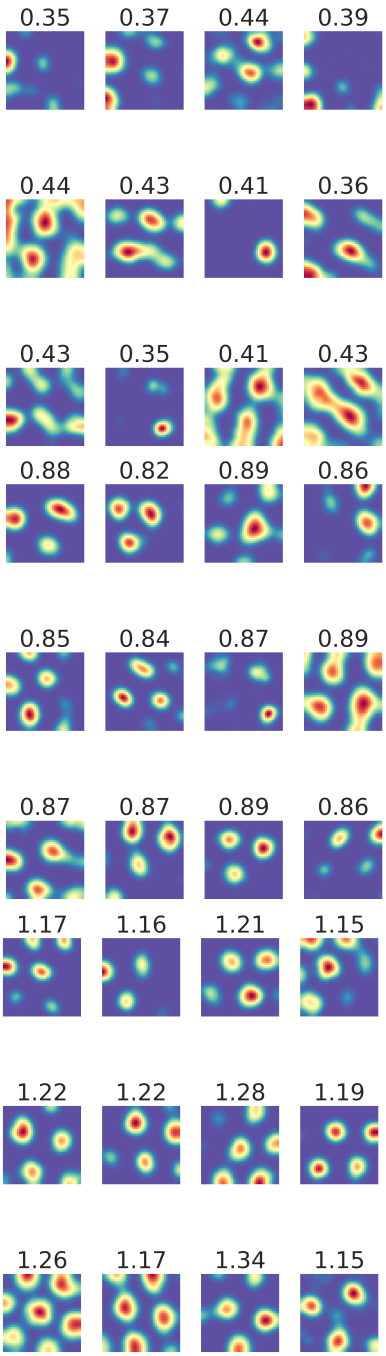

Figure 10. **Grid scores are an imperfect way to characterize grid cells.** Example rate maps from three grid score ranges: $[0.35, 0.45)$, $[0.8, 0.9)$, $[1.15, \infty)$. We considered three grid score thresholds: 0.3 (used by experimentalists on biological data), 0.8 (decent probability of finding grid cells), 1.15 (high probability of finding grid cells). Only the grid cell with score 1.26 looks like a grid cell. Grid scores for each rate map are shown above each rate map.

Nayebi et al. (2021) noted using 5 cm x 5 cm bins ("Nayebi: "We bin the positions in each environment using 5 cm bins, following prior work [Hardcastle et al., 2017, Butler et al., 2019, Low et al., 2020]. Thus, the 100cm^2 environment used 400 (20 × 20) bins, the 150cm^2 environment used 900 (30 × 30) bins, and the 400cm 1D track used 80 bins.") but their code similarly used 20 x 20 bins. The code suggests that Nayebi et al. (2021) used the grid scorer of Sorscher et al. (2019), who in turn used the grid scorer of Banino et al. (2018).

Consequently, we tested what effect the number of bins has on the distribution of grid scores. We found that the number of bins appears to have little to no effect (Fig. 11), so we used 44 x 44 bins since this yields bins of size 5 cm x 5 cm, matching the norm in the experimental physiology literature, to which models are compared.

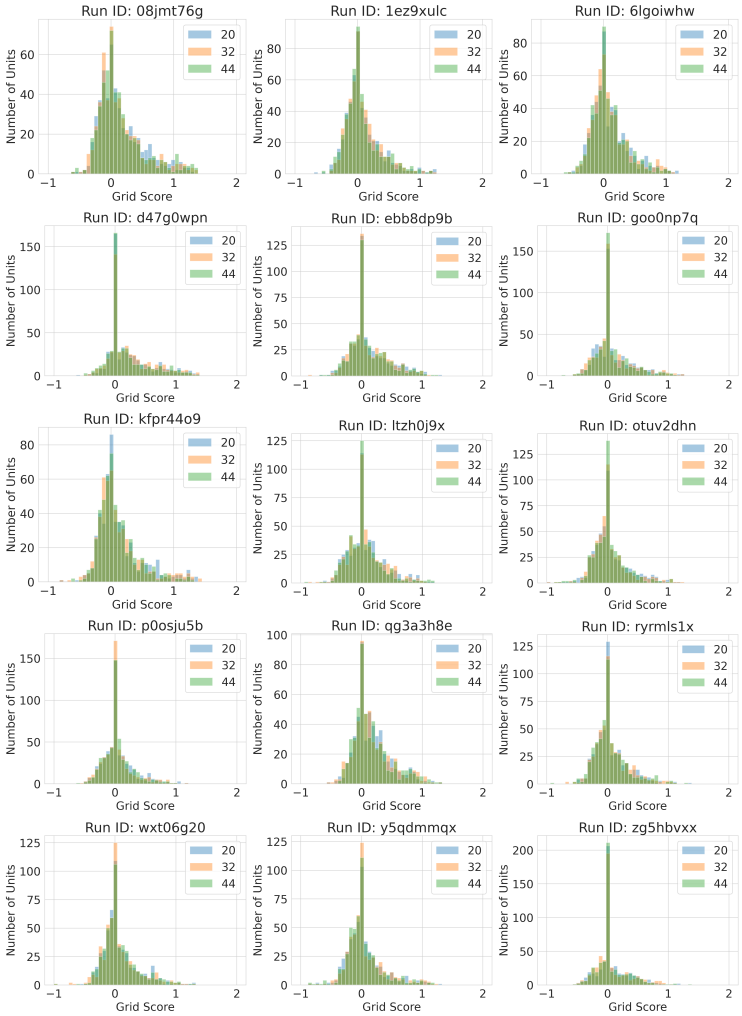

Figure 11. Grid score distributions do not differ as a function of number of bins: 400 (20 x 20; blue), 1024 (32 x 32; orange), 1936 (44 x 44; green).

C. Sweep Configurations

C.1. Cartesian Position with Mean Squared Error

```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 2
  activation:
    values:
      - relu
      - tanh
      - sigmoid
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
```

visualize.py#L136

```
      - glorot_normal
      - orthogonal
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 20
  n_grad_steps_per_epoch:
    values:
      - 10000
  optimizer:
    values:
      - adam
  place_cell_rf:
    values:
      - 0
  place_field_loss:
    values:
      - mse
  place_field_normalization:
    values:
      - none
  place_field_values:
    values:
      - cartesian
  readout_dropout:
    values:
      - 0
  rnn_type:
    values:
      - RNN
      - LSTM
      - UGRNN
      - GRU
  seed:
    values:
      - 0
      - 1
  sequence_length:
    values:
      - 20
  surround_scale:
    values:
      - 1
  weight_decay:
    values:
      - 0
      - 0.0001
```


C.2. Gaussian Place Cells with Cross Entropy Loss


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - linear
      - relu
      - tanh
      - sigmoid
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
      - glorot_normal
      - orthogonal
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 10
  n_grad_steps_per_epoch:
    values:
      - 10000
  n_place_fields_per_cell:
    values:
      - 1
  optimizer:
    values:
      - adam
```

```
  place_cell_rf:
    values:
      - 0.12
      - 0.2
  place_field_loss:
    values:
      - crossentropy
  place_field_normalization:
    values:
      - global
  place_field_values:
    values:
      - gaussian
  readout_dropout:
    values:
      - 0
  rnn_type:
    values:
      - RNN
      - LSTM
      - UGRNN
      - GRU
  seed:
    values:
      - 0
      - 1
      - 2
  sequence_length:
    values:
      - 20
  surround_scale:
    values:
      - 1
  weight_decay:
    values:
      - 0
      - 0.0001
```


C.3. True Difference of Gaussian Place Cells with Cross Entropy Loss


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - relu
      - tanh
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 20
  n_grad_steps_per_epoch:
    values:
      - 10000
  n_place_fields_per_cell:
    values:
      - 1
  optimizer:
    values:
      - adam
  place_cell_rf:
    values:
      - 0.1
      - 0.12
      - 0.2
      - 0.3
  place_field_loss:
    values:
      - crossentropy
  place_field_normalization:
    values:
      - global
  place_field_values:
    values:
      - true_difference_of_gaussians
  readout_dropout:
    values:
      - 0
  rnn_type:
```

```
      values:
        - RNN
    seed:
      values:
        - 0
        - 1
        - 2
    sequence_length:
      values:
        - 20
    surround_scale:
      values:
        - 1.5
        - 2
        - 2.5
    weight_decay:
      values:
        - 0.0001
```


C.4. "Difference of Gaussian" (Difference of Softmax) Place Cells with Cross Entropy Loss, Sweeping Receptive Field


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - relu
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 20
  n_grad_steps_per_epoch:
    values:
      - 10000
  optimizer:
    values:
      - adam
  place_cell_rf:
    values:
      - 0.04
      - 0.05
      - 0.06
      - 0.07
      - 0.08
      - 0.09
      - 0.1
      - 0.11
      - 0.12
      - 0.13
      - 0.14
      - 0.15
      - 0.16
      - 0.17
      - 0.18
      - 0.19
      - 0.2
  place_field_loss:
    values:
      - crossentropy
  place_field_normalization:
    values:
      - global
  place_field_values:
    values:
      - difference_of_gaussians
  readout_dropout:
    values:
      - 0
  rnn_type:
    values:
      - RNN
  seed:
    values:
      - 0
      - 1
      - 2
  sequence_length:
    values:
```

```
      - 20
  surround_scale:
    values:
      - 2
  weight_decay:
    values:
      - 0.0001
```


C.5. "Difference of Gaussian" (Difference of Softmax) Place Cells with Cross Entropy Loss, Sweeping Surround Scale


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - relu
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 10
  n_grad_steps_per_epoch:
    values:
      - 10000
  n_place_fields_per_cell:
    values:
      - 1
  optimizer:
    values:
      - adam
  place_cell_rf:
    values:
      - 0.12
      - 0.2
  place_field_loss:
    values:
      - crossentropy
  place_field_normalization:
    values:
      - global
  place_field_values:
    values:
      - difference_of_gaussians
  readout_dropout:
    values:
      - 0
  rnn_type:
    values:
      - RNN
  seed:
    values:
      - 0
      - 1
  sequence_length:
    values:
      - 20
  surround_scale:
    values:
      - 1.5
      - 1.75
      - 2
      - 2.25
      - 2.5
      - 3
  weight_decay:
    values:
      - 0.0001
```


C.6. "Difference of Gaussian" (Difference of Softmax) Place Cells with Cross Entropy Loss, Heterogeneous Scales


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
```

```
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - relu
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
      - 2.2
  initializer:
    values:
      - glorot_uniform
  is_periodic:
    values:
      - false
  learning_rate:
    values:
      - 0.0001
  n_epochs:
    values:
      - 20
  n_grad_steps_per_epoch:
    values:
      - 10000
  n_place_fields_per_cell:
    values:
      - 1
  optimizer:
    values:
      - adam
  place_cell_rf:
    values:
      - 0.12
      - Uniform( 0.10 , 0.14 )
      - Uniform( 0.08 , 0.16 )
      - Uniform( 0.06 , 0.18 )
  place_field_loss:
    values:
      - crossentropy
  place_field_normalization:
    values:
      - global
  place_field_values:
    values:
      - difference_of_gaussians
  readout_dropout:
    values:
      - 0
  rnn_type:
    values:
      - RNN
  seed:
    values:
      - 0
      - 1
      - 2
      - 4
  sequence_length:
    values:
      - 20
  surround_scale:
    values:
      - 2
      - Uniform( 1.90 , 2.10 )
      - Uniform( 1.75 , 2.25 )
      - Uniform( 1.50 , 2.50 )
  weight_decay:
    values:
      - 0.0001
```


C.7. Softmax of Difference of Squared Distances Place Cells with Cross Entropy Loss


```
method: grid
metric:
  goal: minimize
  name: pos_decoding_err
parameters:
  Ng:
    values:
      - 4096
  Np:
    values:
      - 512
  activation:
    values:
      - relu
  batch_size:
    values:
      - 200
  bin_side_in_m:
    values:
      - 0.05
  box_height_in_m:
    values:
      - 2.2
  box_width_in_m:
    values:
```

```
    - 2.2
initializer:
  values:
    - glorot_uniform
is_periodic:
  values:
    - false
learning_rate:
  values:
    - 0.0001
n_epochs:
  values:
    - 10
n_grad_steps_per_epoch:
  values:
    - 10000
n_place_fields_per_cell:
  values:
    - 1
optimizer:
  values:
    - adam
place_cell_rf:
  values:
    - 0.09
    - 0.12
    - 0.2
place_field_loss:
  values:
    - crossentropy
place_field_normalization:
  values:
    - global
place_field_values:
  values:
    - softmax_of_differences
readout_dropout:
  values:
    - 0
rnn_type:
  values:
    - RNN
seed:
  values:
    - 0
    - 1
    - 2
sequence_length:
  values:
    - 20
surround_scale:
  values:
    - 1.5
    - 2
    - 2.5
weight_decay:
  values:
    - 0.0001
```