

# Supplementary Material : Grid Cell Path Integration For Movement-Based Visual Object Recognition

Niels Leadholm<sup>1,2</sup>

niels.leadholm@seh.ox.ac.uk

Marcus Lewis<sup>1</sup>

mlewis@numenta.com

Subutai Ahmad<sup>1</sup>

sahmad@numenta.com

<sup>1</sup> Numenta, Inc.

Redwood City,  
California, USA

<sup>2</sup> Dept. of Experimental Psychology

University of Oxford  
Oxford, UK

## 1 Related Literature

**Grid Cells** During spatial navigation in an animal such as a rat, grid cells are notable for firing at regular intervals as space is traversed. These points of activity correspond to a triangular lattice with a particular phase, orientation and scale [27] (Figure 1a). Grid cells with the same orientation and scale, but different phases, form what are known as grid cell ‘modules’ [60]. As a rodent moves, any individual grid cell’s activity is ambiguous as a means of encoding the animal’s position. The joint activity of multiple grid cell modules, however, can uniquely encode a position. Importantly, this encoding scheme has a large representational capacity [60] (Figure 1a middle). Information about self-movement is used to update the current location representation by each grid cell’s firing corresponding to the positional change (Figure 1a bottom). This process, known as path-integration, means that after returning to the same position, the same grid cells will be active regardless of the path taken [27, 60]. Note that our work does not deal with how grid cells might actually implement path integration - rather we explore the significance of path integration in a neural population for developing useful object representations. The combined properties of a large capacity for unique spatial representations and path integration enable grid cells to act as a powerful substrate for encoding spatial information. For a more in depth discussion of grid cell computations as explored in this work, we direct readers to Lewis et al. [39].

**Cortical Models** Our work builds on previous models of the cortical architecture of the mammalian brain. Hawkins et al. [23] demonstrated that networks with a columnar architecture, where different layers correspond to sensory and location-based representations, can learn inputs such as objects composed of synthetic features. Neurons in these layers receive external sensory and self-movement information, while they share connections that enable learned associations between features and locations, as well as predictions during inference. Neural activity (including input features) are represented in the distributed activity of sparse binary vectors - a form of encoding where the dimensionality used is relatively high, but

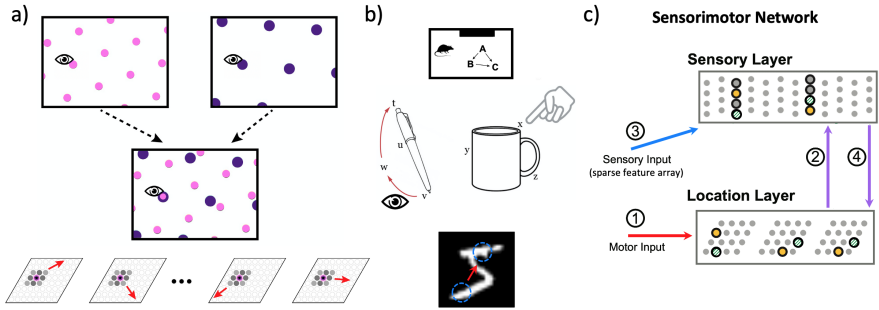


Figure 1: *Using grid cell representations for object recognition.* a) The combination of multiple grid cells of different scale and orientation can uniquely encode the location of a sensor (e.g. fovea). Here we use multiple grid cell modules with sparse activity (bottom, each indicated by a rhombus) to encode and update the sensor’s location with self-movement information. b) It is hypothesized that this process can be used for object recognition with active sensors, and we use sequences through a  $5 \times 5$  grid of feature patches extracted from MNIST images to test this. c) GridCellNet takes in motor input when the sensor moves (1) and updates its location representations. The current location representation is used to predict incoming sensory information (2), before this is received (3). Correctly predicted sensory information is then used to update the location representation (4). Locations are initially ambiguous, represented using a union of locations, and disambiguated over time via sensory input. Yellow and textured green dots in the location layer indicate two different objects which are compatible with the current sequence. Classification is successful once the representation is primarily consistent with a single object class. The two-layer network is based on cells with sparse binary activity, dendritic segments, and Hebbian-like learning, following Lewis et al. [49], and with figures reproduced/modified with permission from the authors.

where only a small subset of the available nodes are ever active at a time (taking values of 0 or 1). Such encoding has numerous appealing properties, including tolerance to noise [40], a large representational capacity, and the ability to encode notions of similarity between objects [27].

**Grid Cells for Object Recognition** The cortical models in Hawkins et al. [23] did not explicitly discuss how the brain might implement the encoding of location information. In Hawkins et al. [24], it was suggested that neurons akin to grid cells might exist outside of the entorhinal cortex in cortical columns throughout the brain, and could thereby support spatial encoding in sensory modalities such as touch and vision (Figure 1b). Recent experimental work has supported this possibility [40, 41]. In various sensory modalities, grid cells could then be used to encode feature locations in an object’s own reference frame. The idea that each of the columns throughout the brain would be learning object representations in a massively parallel process was dubbed the Thousand Brains Theory. This nomenclature was to contrast the theory to those that suggest a more strict hierarchy with object-like representations only existing at certain levels of processing [24]. While models developed from this theory were capable of rapidly learning objects and performing recognition, this was limited to synthetic data-sets [49].

In Bicanski and Burgess [5], a system that relied on grid cell computations to recognize

images was implemented, similar to our own approach. Importantly, however, they explored object recall (i.e. from the training data-set) rather than generalization to unseen data. As such, the implementation assumed that at any given time-point, only one learned representation would be consistent with the observed input, and so the system only represented one possible hypothesis. An additional difference from our work to Bicanski and Burgess [5] is that, in GridCellNet, the location representation in the grid cell layer needs to be inferred from sensory and self-movement information (Figure 1c). In Bicanski and Burgess [5], all objects share a fixed reference frame across learning and inference, and so with perfect path integration, the system always knows exactly where the current fixation is in the external environment. As we discuss separately, this raises issues for translation invariance. Finally, this model did not explicitly explore the benefits of such a system for the visual reconstruction of unsensed regions of an input.

**Perception Through Saccades in Humans** There is a long history of studying the representations that humans develop and combine over saccades, a process known as transsaccadic integration. Despite extensive study, the precise nature of these representations remains unclear [26]. At one extreme, it is suggested that humans view the world with very little if any history of previously represented features, as suggested by phenomena such as change blindness, where humans are remarkably insensitive to changes in the image during saccades [20]. At the other extreme is the suggestion that humans maintain a detailed, picture-like representation across saccades [44].

Experimental evidence supports a compromise of these views. On the one hand, humans cannot piece together percepts across saccades at a pictorial/pixel level, such as complex arrays of random dots [52]. Despite this, there is evidence that integration occurs at a more abstract, feature level. For example, humans are able to integrate sequentially presented points of light to make judgements about the shape of the triangle they form [25]. With regards to object recognition, humans can also recognise an object that is built up over progressive saccades, such as drawings of animals hidden among oriented lines [29]. Our work aligns with this evidence by demonstrating a model that builds up and integrates representations across space at a feature (rather than pixel) level.

It is also worth highlighting the central role that attention plays in human perception. In order to efficiently sample the environment, humans rely on signals such as bottom-up saliency to inform future fixations. There has been significant effort in the literature dedicated to modelling how the serial application of attention is driven [8, 69]. We note that we do not explicitly model attention in this work, such that fixation selection is described by a uniform distribution with inhibition of return, although it could be integrated to enable the system to more efficiently sample the input space.

**Multiple Views in Machine Vision** Object recognition in computer vision often focuses on processing static images from a single viewpoint. Systems that do integrate views sampled from multiple locations often make no assumptions about their spatial arrangement, simply aggregating these inputs in a spatially agnostic manner [61, 69]. Other systems that do take account of spatial information typically make use of an external, ground-truth reference frame. Such spatial information might take the form of indexed positions in the input image [70], a 2-dimensional coordinate [66, 45], or a combination of the 3-dimensional coordinate, pitch, and yaw of a camera in a room [0]. Furthermore, this spatial information is often used to solve machine vision tasks such as scene representation rather than object recognition [0]. Finally, in settings where translation invariance is explored, performance relies on training on tens of thousands of examples at different possible locations in the external environment [45, 60].

While computer vision techniques for object recognition typically focus on processing images in isolation, the field of robotics faces similar challenges (and potential benefits) to biological agents of embodiment and the ability to interact with the environment [2]. Similar to the work we present here, Browatzki et al. [10] presented a classification system in a robotic agent which framed visual object recognition as a problem of localization (what ‘environment’, i.e. object is the sensor viewing, and where is it viewing it from). In their case, recognition was investigated in a robot rotating a 3D object, and thus identifying its location on a view sphere. The object was inferred from its sensory observations and proprioceptive data via an iterative particle filter, not unlike our system. Pezzementi et al. [60] implemented a similar approach in a robotic system for touch perception of three-dimensional letters, and in particular demonstrated the benefit of such an approach for translation and rotation invariance. Notably, however, these approaches evaluated recognition on the same objects that were learned on (i.e. not investigating true generalization of object classes), and did not make use of grid cell representations or other biologically plausible elements when implementing the recognition algorithm.

Work in robotics has recently inspired new approaches in more general computer vision. Hoang et al. [27] presented a classifier where, similar to capsule networks [64] and our own work, recognition is predicated on the consistency of object features within an internal reference frame for the object. Unlike our work, they do not make use of grid cells, instead assuming an idealized 2D map for encoding feature locations, and they train and evaluate with five objects with highly diverse features that provide locally discriminative features (e.g. a teddy bear vs. a patterned flower vase), rather than assessing generalization to novel examples of similar classes.

**Few Shot and Continual Learning** Few shot learning is a large field, and prior work has addressed learning hand-written characters [19, 55, 70], or demonstrated the benefits of memory-like mechanisms in the few-shot setting [53]. Our intent is not to present the current work as a strong solution to the problem of few-shot learning. However, few-shot training captures our biological motivation of humans learning rapidly from arbitrary feature sequences. As such, we use the few-shot experimental setting to evaluate the performance of our system.

In continual learning, a classifier is evaluated on its ability to learn novel classes while retaining the ability to categorize older classes. Robustness in this setting remains a significant challenge for machine learning techniques such as deep learning that display catastrophic forgetting, the phenomenon where information required to solve a novel task obliterates previously learned representations [43]. Continual learning has received comparatively little study in recurrent architectures [15], having first been formally evaluated by Schak and Gepperth [57]. Such work, however, has confirmed that recurrent systems such as LSTMs suffer from significant catastrophic forgetting. Many biologically motivated approaches for continual learning have been identified for feed-forward networks, such as elastic weight consolidation [63]. Unfortunately, recent evidence has demonstrated that these can be challenging to apply in recurrent architectures on tasks requiring substantial working memory [18]. From the perspective of continual learning, the most similar approaches to GridCellNet are those that rely on external memory. Methods such as Gradient Episodic Memory [42, 60] and progressive memory banks [8] have been applied in recurrent neural networks. Unlike GridCellNet, these approaches rely on rehearsing on a stored bank of previous experiences [60], or require an explicit, external memory system that continuously expands [8]. As such, efficient and robust continual learning in sequential tasks remains an ongoing challenge.

## 2 Supplementary Methods

Code necessary to implement the models and results described is available at <https://github.com/numenta/htmpapers>.

### 2.1 Sparse Feature Extraction

For our pre-processing of the data-set, we trained a convolutional neural network (CNN) [8] in a supervised paradigm on a subset of the MNIST training data-set of handwritten digits (54,000 images) [8], tuning it using a hold-out cross-validation section (6,000 images). This encoder network has the architecture shown in Figure 2a. A k-Winner Take All (k-WTA) layer [9] follows the second max-pooling operation to enforce sparsity in the representation (Figure 2b). While the non-zero values in this layer take on real-number values (necessary for useful gradients during learning), we require binary feature vectors for input to the sensorimotor network. We therefore used the network after training to pass images through until the k-WTA layer, and then binarized this representation, providing us with a  $5 \times 5$  grid of features. Each grid location is a vector of dimension 128 which contains regional information (covering a  $16 \times 16$  pixel-patch) about the image representation in a sparse format. The sequential provision of these features forms the input to all of our downstream classifiers. Note therefore that none of our classifiers (GridCellNet or the classifiers we compare to) receive direct pixel inputs as their features, while there is some overlap in pixel space between the inputs to each of these feature representations.

**CNN Details** It is possible that binarization could lead to a significant loss of information. We verified that a linear classifier achieves accurate classification with these binarized features, with an evaluation-set accuracy of 99%+. We also verified that a decoder accurately reconstructs the input image from these features. In order to achieve optimal performance, it is useful for the feature vector to have a reasonably large dimension (128), number of non-zero elements (19 during training, 29 during evaluation), and high entropy (intuitively, how often does each input feature contribute to a representation across all examples). To optimize the entropy, we made use of two hyper-parameters. The k-WTA's duty cycle monitors how often a unit is contributing to representations. The boosting factor biases a unit's activity to target a given duty cycle (see Ahmad and Scheinkman [9] for details). Tuning the duty cycle and boosting factor ensures a greater number of neurons each contribute information at some point, and the representation becomes more distributed (Figure 2c).

In order of layers, the CNN architecture is composed of a convolution (kernel size 5, channels 64), max-pooling, convolution (kernel size 5, channels 128), max-pooling, k-WTA, and three fully connected layers (dimensions of 256, 128, and 10). k-WTA applied to the max-pooling layer is local (that is, the k-winners are determined across all channels at a given spatial location, rather than across the entire image space). This ensures that each extracted feature vector has the same sparsity. We used stochastic gradient descent with a learning rate of 0.01, momentum 0.5, batch size of 128, and 10 epochs of training.

### 2.2 Sensorimotor Network

**Network Architecture** The location layer consists of 40 grid cell modules, each a lattice of 50 by 50 cells. A grid cell module has a particular scale and orientation, while the active location corresponds to the current phase of activity in the module. In our model, grid cells can be either active or inactive, and activation is determined by either the current representation

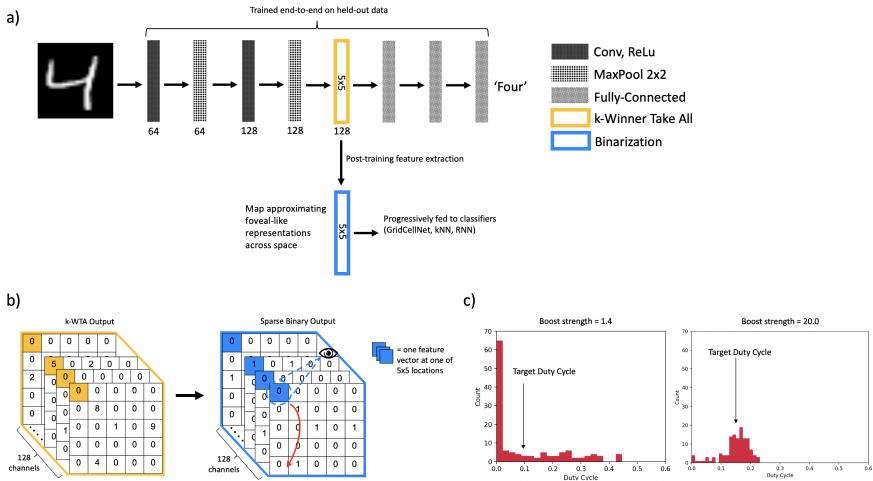


Figure 2: *The pre-processing convolutional neural network.* a) The encoder CNN is trained end-to-end to perform classification, with a k-WTA operation that constrains the mid-level representations to a specific level of sparsity. Numbers below operations show the channel dimension. b) An example of what the k-WTA and binarized representations might look like. Note that in all of our tasks, the classifiers are given a series of sparse feature vectors (each of dimension 128) from  $5 \times 5$  total locations. This forms a sequence of sensory inputs, here represented with the eye and its movement. The order with which the features are sampled across this  $5 \times 5$  space can either be fixed for all examples during both training and testing, or follow an arbitrary sequence. c) For optimal performance, the two parameters of k-WTA (target duty cycle and boost strength) are optimized so as to ensure most of the neurons achieve the target duty cycle. On the left is shown a typical value for the boosting factor used in past models, while on the right we show the result of using the larger boosting factor and target duty cycle that we arrived at through hyperparameter tuning.

in the sensory layer, or movement applied to the previous location representation; at time step  $t$  and for grid cell module  $i$ , these are denoted by the binary arrays  $A_{t,\text{sense}}^{\text{loc},i}$  and  $A_{t,\text{move}}^{\text{loc},i}$  respectively. We model the location phase that determines  $A_{t,\text{move}}^{\text{loc},i}$  using a square rather than the biologically motivated triangular lattice used in Lewis et al. [59], although this has no major consequence for the system.

The sensory layer is identical to that used in Hawkins et al. [24]. The input features are binary vectors of length 128 with 29 active values (i.e. approximately 77% sparsity). The sensory layer in turn consists of a corresponding 128 mini-columns, which receive the input features in a one-to-one fashion. Each mini-column in the sensory layer consists of multiple cells (here 32). This enables the mini-columns to use sparse activity to uniquely encode features associated with particular objects (i.e. location representations). The active cells in mini-column  $i$  at time-step  $t$  are denoted by the binary array  $A_{i,t}^{\text{in},i}$ .

**Stage 1: Using Movement to Update the Location Representation** If the location layer has active cells, then each module uses the current movement information to compute a new set of active cells. Each module will apply a translation to its 50 by 50 activation pattern, according to the movement information. The translation vector is different in each module.

and is determined by applying the following dilative rotation to the movement vector:

$$M_i = \frac{1}{s_i} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \quad (1)$$

where  $i$  denotes the particular grid cell module,  $\theta$  its orientation, and  $s$  its scale.

The translated 50 by 50 pattern will rarely align neatly with the original 50 by 50 cells, except in discrete environments. Typically each active cell in the pattern will land on the corner between four cells, so each active cell will activate up to four cells after the translation vector has been applied. During inference, this is indeed what happens, but during learning we allow the grid cell module to have more certainty about the current location. During learning, translating the active pattern will not increase the number of active cells; instead, the module's internal state includes a list of high-precision active phases, and the module applies the translation to those phases, rather than estimating those phases from the current set of active cells. This difference in the algorithm's behavior in inference and learning reflects the fact that binary representations will always lead to some spatial uncertainty during inference.

When inference begins for a new object, no location information is available - this will instead become available at stage 4, discussed below, and so inference proceeds to stage 2.

**Stage 2: Predicting the Sensory Input with the Location Representation** The cells in the mini-columns have dendritic segments which receive activity from the location layer. Let  $D_c$  be the set of dendritic segments of cell  $c$ , with segments indexed by  $d$ . If a dendritic segment is active (that is, a cell in the sensory layer is predicted by the activity of the location layer), then it is in a predictive state. Let the binary vector  $\pi_t^{\text{in}}$  denote the sensory cells that have at least one active dendritic segment, and  $\theta^{\text{in}}$  a dendritic threshold, then:

$$\pi_t^{\text{in},c} = \begin{cases} 1, \exists_d [D_{c,d}^{\text{in}} \cdot A_{t,\text{move}}^{\text{loc}} \geq \theta^{\text{in}}] \\ 0, \text{otherwise} \end{cases} \quad (2)$$

**Stage 3: Determining Activity in the Sensory Layer** If a given sensory layer cell that is predicted also receives activity from the input feature (that is, it is in a column receiving sensory input and was therefore correctly predicted), it will be active and inhibit any other cells in the mini-column that are not predicted. Note that multiple cells in any given mini-column can be active if they are predicted by the current location representation. If no cells in a mini-column are predicted but it receives sensory input, then all cells in the mini-column will become active.

**Stage 4: Using the Sensory Representation to Update the Location** After the sensory representation has been determined, the location layer receives inputs from the sensory layer. In particular, the sensory features help to recall location information, and supplement the location representation arrived at by path integration. Similar to equation 2, this is determined by the overlap between the active cells in the sensory layer, and the learned weights. In this case however, an active dendritic segment is sufficient for a location cell to now be active, such that:

$$\pi_t^{\text{loc},i,c} = \begin{cases} 1, \exists_d [D_{c,d}^{\text{loc},i} \cdot A_t^{\text{in}} \geq \theta^{\text{loc}}] \\ 0, \text{otherwise} \end{cases} \quad (3)$$

$$A_{t,\text{sense}}^{\text{loc},i} = \begin{cases} \pi_t^{\text{loc},i}, & \|\pi_t^{\text{loc},i}\| > 0 \\ A_{t,\text{move}}^{\text{loc},i}, & \text{otherwise} \end{cases} \quad (4)$$



At this stage, the next movement is received, and the four stages are repeated for the next time-step. Classification using the location representations as input features is discussed separately in the main text, Section 2.4.

**Learning in the Sensorimotor Network** Learning takes place via the reciprocal strengthening of connections between the active representation in the sensory layer, and the current location representation. The aim is to associate a given feature with a given location in that object’s reference frame. When a new object is learned, the location representation at the first sensation is randomly initialized, such that each object operates in a different location space. This enables multiple objects to be jointly represented during inference, as the probability of overlap between different objects’ location spaces is low. As further sensations are performed during learning, the location representation is updated using movement information as described in Stage 1 above. For the sensory layer, as the feature-location association has yet to be learned, a random cell in each mini-column receiving an input will be selected to be active. Each active cell in the sensory and location layer will then form reciprocal connections on one of their dendritic segments ( $d'$ ), according to the following:

$$D_{c,d'}^{\text{loc}} := D_{c,d'}^{\text{loc}} | A_{t,\text{learn}}^{\text{in}} \quad (5)$$

$$D_{c,d'}^{\text{in}} := D_{c,d'}^{\text{in}} | A_{t,\text{sense}}^{\text{loc}} \quad (6)$$

Here " $|$ " is used to indicate the bitwise OR operator; that is, if a synapse already exists between two cells, then it is unaffected by the learning rule.

Following the above, the network can rapidly learn objects by visiting each feature once, performing a single set of weight updates for each feature. Note that due to the path-integration performed by the grid cells, both learning and inference can take place using an arbitrary order through the features of the object - there need be no correspondence between the order taken at learning vs. that used at testing. Note also that the learning process could in principle be implemented on hardware in a parallelized form, although for biological plausibility, this is run as a serial process.

There are several hyperparameters of the model that might be tuned to optimize performance, such as increasing the grid cell module size to increase the capacity of the model (see Lewis et al. [69] for a quantitative exploration of this). The main parameter we tuned was the dendritic threshold  $\theta^{\text{loc}}$ . If this was too high, grid cells were too stringent in which sensory features were present to become active; if it was too low, grid cells were too easily activated by spurious sensory features. Details on how we tuned hyper-parameters are provided in Section 2.5.

## 2.3 Recall Method

As well as evaluating generalization using the approach outlined above, we also evaluate the ability of the system to recall examples from the training data-set, as this is a common evaluation method in the literature for related systems. To do so, we use the object recognition method employed in Lewis et al. [69]. In particular, as sensations are progressively provided to the network, recall of a particular object occurs when the location representation converges to that of a single learned object. If that location is a subset of the learned location representations for the true (target) object at the corresponding position of the sensor, then recall is successful. This may fail if the representation is a subset of any other learned location representations, in which case the process has converged to an incorrect object or



position. An alternative failure case is that the representation never converges to a subset of a learned representation. Formally, the recall system’s probabilities for a particular object instance  $y$  during recall correspond to:

$$p(y) = \begin{cases} 1, (A_t^{\text{loc}} \subseteq L_y^m) \\ 0, \text{ otherwise} \end{cases} \quad (7)$$

where  $L_y^m$  is the learned location representation for any particular position  $m$ . The above recall is only considered successful if  $p(y) = 1$ ,  $y$  is the target example object, and  $m$  is the actual position of the sensor on the object at the current time-point. Note that the position of the sensor  $m$  in the external reference frame is only used to evaluate the correctness of the system from an experimenter’s point of view, and this is not information privy to GridCellNet itself.

Intuitively, recall is considered to have taken place when the representation has converged to that of a single representation. Thus, there is no additional read-out node; instead, the uniqueness of the location representation represents the object’s identity. As such, this process can take place within the same system (and indeed at the same time) as the classification algorithm outlined separately. For example, the system might successfully classify an input, but never converge to a single representation, in which case no single learned example ever appeared to align well with the test object. Alternatively, it might classify an object, and subsequently converge to a single representation. In this case, the system both recognises the class of the object, as well as the similarity of the particular test object to a previously learned example. Note that for our main result we use a slightly higher dendritic threshold  $\theta^{\text{loc}}$  when evaluating recall than for generalization (20 for recall vs 16 for generalization). We discuss the significance of these processes operating in parallel and the different dendritic thresholds separately.

## 2.4 Comparison Classifiers and Decoder Network

We compare GridCellNet to both a recurrent-neural network (RNN) and a k-Nearest Neighbors (k-NN) classifier [14]. For our RNN, we use a long short-term memory (LSTM) classifier [28]. This network receives an input sequence of length 25, corresponding to the 25 locations in the image feature space. Each feature vector in this sequence is a sparse feature vector extracted from our CNN, as for GridCellNet. In addition, sensor location or movement information is concatenated to the feature, outlined below.

The RNN has a single hidden layer of dimension 128. Using additional layers did not appear helpful. We used weight decay [28] of 0.001 and optimized with Adam [14]. Learning rates were selected via a grid-search for the best performance on each classification task independently, where each task is distinguished by both the number of training examples, and whether evaluation was using a fixed or arbitrary input sequence.

**Providing Movement and Location Information to the RNN** In order to provide a fair comparison (and in particular the possibility of the LSTM developing an internal reference frame), we provide the LSTM with movement information similar to that provided to GridCellNet. Specifically, at each step in the input sequence, two values are concatenated to the input feature vector. These are the Euclidean magnitude of the just-performed movement, and the angle (measured in radians in standard position). These are derived from the previous and current location of the sensed feature in the  $5 \times 5$  input space. This movement information is always provided to the LSTM when the input follows an arbitrary sequence.

In cross-validation experiments, we also explored providing movement as the discrete  $x$  and  $y$ -displacements in the  $5 \times 5$  grid, but we found the separate magnitude and angle form to provide the better classification accuracy. In any setting where the input follows a fixed sequence, the explicit location information (as the 0 to 24 index in the  $5 \times 5$  space) is instead provided, as this appeared to provide better performance.

**Decoder Network** In order to enable visualization of the current feature representations in GridCellNet, we trained a multi-layer perceptron (decoder network) on the sparse binary features from a subset of the MNIST training data-set (54,000 images). This decoder has a single hidden layer of dimension 512, with input  $128 * 5 * 5$  and output  $28 \times 28$ . For the decoder we used Adam with a learning rate of 0.001, a batch size of 64, and 10 epochs of training.

**Other Details** We also compared classification in our network to a k-NN classifier. This receives the same input as the RNN and GridCellNet, but as a single extended array, rather than sequentially, and without the additional movement/location information. The number of neighbours for the k-NN classifier and the dendritic threshold  $\theta^{\text{loc}}$  /  $\gamma$ -threshold were, like the LSTM learning rates, selected via a grid-search for each classification task. Additional details of the selected hyper-parameters and how the data-set was divided are provided in Section 2.5.

## 2.5 Hyper-parameter Selection

In Figure 3, we provide a break-down of how different splits of the data are used for different steps in feature extraction, hyper-parameter tuning, and learning. In Table 1, we list the hyper-parameters arrived at for each classifier, specific to each learning task.

For the continual learning tasks, where fixed input sequences of 20 examples-per-class were used, the same LSTM and GridCellNet hyper-parameters were used as for the equivalent standard (i.e. not continual) learning setting.

## 3 Supplementary Experiments

### 3.1 Supplementary Comments : Translation Invariance and Inference Given Arbitrary Sequences

In this section, we elaborate on the connection between arbitrary input sequences and translation invariance. As shown in Figure 4a, a particular sequence of eye movements will be followed the first time an object is seen. The next time this object is seen, however, the eyes are unlikely to sample it at the exact same location they did the first time it was seen, nor are they likely to follow the same sequence after this. Such an approach would require a priori knowledge about what the object is before it has been inferred, or that bottom-up saliency signals could support such a perfect alignment of sensation within the object. Experimental evidence supports such an approach being implausible. While humans occasionally demonstrate repeated patterns in the sequences they take when viewing complex objects or scenes [19], evidence does not support regimented, fixed adherence to such sampling being beneficial for recognition [18].

As a result of the above reality, translation invariance therefore represents a real challenge for sequential classifiers. In particular, as the stimulus can move in the real world, and the starting position of where classification begins can translate, the system must be able to

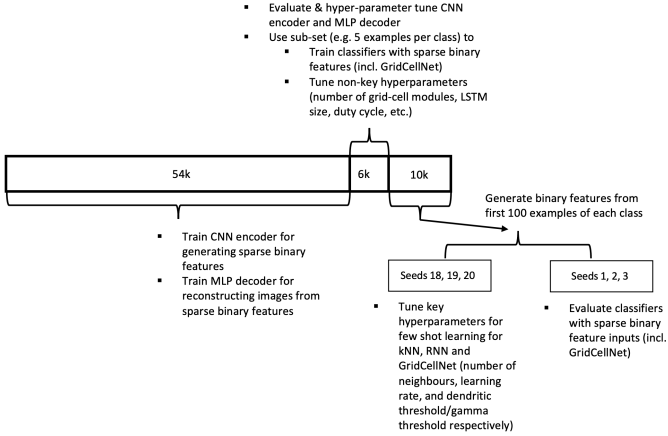


Figure 3: *Division of the data-set for training and evaluation.* As the down-stream classifiers use features derived from a network classified end-to-end, and we are interested in few shot learning, we divide the data set such that the training of the various systems uses different sub-sets of the data. This also allows us to perform hyper-parameter tuning on hold-out data, with the exception of the key hyperparameters which are deliberately selected to enhance few-shot learning on the final evaluation data for all classifiers. Note that although source images are therefore re-used during this step of hyper-parameter tuning and final evaluation, the feature vectors generated by the CNN encoder will vary with the random seed used, and therefore between the settings in which hyper-parameter tuning and final evaluation are performed.

adapt. If the classifier requires a fixed spatiotemporal input sequence regardless of where it begins on the object, it will, by definition, generally fail to sample the object (Figure 4b). Instead, it must be capable of beginning anywhere on the object, and subsequently follow whatever trajectory samples the input (Figure 4c). In our main results, we discussed that two requirements of translation invariance are therefore i) this ability to integrate features from arbitrary starting positions and sequence inputs and ii) the use of an internal reference frame for classification. A classifier can be envisioned that can handle arbitrary sequence inputs, but whose classification process relies on an external reference frame to encode the spatial relations of features. While this satisfies (i), this system will fail if that object is moved in the environment, exposing it to out-of-distribution spatial coordinates. Finally, for complete translation invariance, a third requirement is iii) feature inputs that are invariant given small translations, and *equivariant* across larger translations.

To satisfy requirement (iii), we assume that the pre-processed feature map we generate with the CNN is approximately translation equivariant, i.e. that each high-level sparse feature is the product of approximately the same transformation, simply shifted in space. This is a reasonable assumption in shallow CNNs due to weight sharing [17], although as we discuss later, there are more natural ways to satisfy this requirement. Furthermore, the representation should be invariant to small translations within the receptive field. As we cannot guarantee requirement (iii), we do not evaluate the classifiers on images translated in pixel space. As noted separately however, GridCellNet satisfies conditions (i) and (ii), and

Table 1: Choice of Hyper-parameters for Classifiers Given Fixed or Arbitrary Input Sequences

# of training examples per class	Fixed	Arbitrary
GridCellNet	$\theta^{\text{loc}}$ -threshold / $\gamma$ -threshold	
1	12 / 0.7	12 / 0.7
5	12 / 0.5	12 / 0.5
10	14 / 0.5	14 / 0.5
20	16 / 0.3	16 / 0.3
LSTM (1 epoch)	Learning Rate	
1	0.005	0.01
5	0.005	0.01
10	0.005	0.01
20	0.01	0.005
LSTM (50 epochs)	Learning Rate	
1	0.002	0.002
5	0.02	0.002
10	0.005	0.002
20	0.005	0.005
k-NN	# of Neighbours	
1	1	1
5	1	1
10	1	7
20	1	9

therefore represents a significant step towards a sequential classifier that can perform online translation invariance to arbitrary locations.

For a fair comparison in these experiments, we include an LSTM that receives self-movement information, such that it could in principle also develop an internal reference frame. As we demonstrate however, only GridCellNet has the inductive biases to perform object recognition given arbitrary input sequences given so few training examples.

We also include a k-NN model in our results. We note that the k-NN actually performs better than the LSTM with 50 epochs of training, despite the LSTM being provided with location information. This appears to be due to the challenge the LSTM faces of learning longer range dependencies given so few training examples, in spite of being provided with information about where the current feature is located in the sequence. The ability of the LSTM to solve this appears sensitive to adjustments in hyper-parameters including the learning rate.

It is worth noting that given sufficient training time and examples, the LSTM’s performance steadily improves. Deep learning architectures are undeniably powerful; indeed we used them to perform the initial feature extraction step for all of our classifiers. It is therefore likely that, given enough training examples, the LSTM’s performance would match GridCellNet. Moreover, given appropriate training conditions, recurrent architectures can learn to develop grid cell-like units capable of path integration [68]. The purpose of GridCellNet is to demonstrate the benefit of the inductive bias that path integration together with only simple, Hebbian-like learning rules can provide. The contrast in performance in the

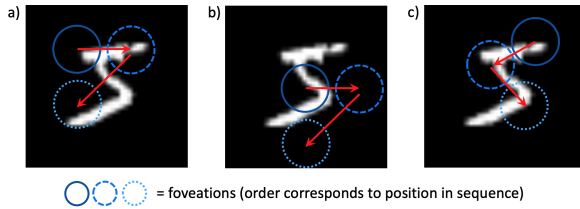


Figure 4: *Translation Invariance in Sequential Classifiers.* a) A sequence of hypothetical foveations during learning. b) During inference, the initial sensation may begin on a different part of the object (translation). Classifiers with a rigid input sequence requirement won’t adequately sample the image. c) Regardless of where inference begins on the object, the system should sample the object and correctly classify it (i.e. translation invariance).

few-shot setting supports the proposed architecture as a principled and biologically plausible mechanism by which humans might rapidly learn under the challenging settings explored here, and a useful approach for designing machine learning models.

### 3.2 Supplementary Comments : Predictive Representations

We limit visualization to situations where the network converges to a single representation (e.g. a single example of a ‘9’, as opposed to several compatible ‘9’s), as the decoder in its current form is unable to make sense of the input if the representation at inference includes a union of object representations. This isn’t to say that GridCellNet’s predictive representations before this point are not also meaningful, they are simply more difficult to visualize without bespoke training of the decoder. When GridCellNet is trained on 5 examples per class, single-object convergence occurs for around 25% of all objects. On a small subset of these (<5% of the single convergence objects, or around 1% of all objects), GridCellNet fails to predict a feature for every region of the image. Note that this is not the same regime as in Section 3.3 (Recall of Training Examples Despite Noise); in that setting, the training data was used at evaluation time, and a higher dendritic threshold was used. In contrast, we evaluate predictive representations in the same regime as generalization to unseen examples of objects, and using a dendritic threshold that supports such generalization.

Note that at every time-step, the model can make a prediction about any arbitrary location. It would therefore be possible, in principle, to query every location *before* single-convergence. As noted above, however, our decoder network would not be able to produce meaningful visualizations from these representations. Also note that GridCellNet does not make predictions at the pixel-level. Rather these are at the abstract, feature-level, but the decoder enables us to visualize these directly.

### 3.3 Recall of Training Examples Despite Noise

In the Section 3.1 of our main submission, we demonstrated that GridCellNet can generalize to novel examples of MNIST that it has not seen in the training data. The following section demonstrates that generalization occurs without the loss of recall ability. In particular, GridCellNet can also operate in a regime focused on recalling a specific learned example, rather than classification. This has clear relevance in the natural world, where occasionally

it is necessary to recognise whether one has seen a particular instance of an object before. Also note that while the following results replicate the broad functionality seen in previous work, this is performed using an internal object reference frame. Prior biologically plausible approaches assumed a fixed reference frame across all objects [5], rather than inferring locations from the combination of sensory inputs and self-movement.

Similar to Bicanski and Burgess [5], where 33 faces, 33 objects, and 33 scenes were used, we evaluate performance after learning 100 objects (10 of each MNIST class), and in the context of noise. In their primary simulation, noise was applied at learning in the form of pixel blurring, rather than at inference time (although they also explored conditions such as occlusion, which we do not consider here). We were concerned that any noise robustness in GridCellNet might in fact be attributable to the pre-processing stage, and so we applied feature-level noise in the form of randomly flipped bits (changing a 0 to 1 or vice-versa) at inference. For each feature vector in the  $5 \times 5$  input space,  $n$  randomly chosen bits are flipped from their original values (Figure 5a). We demonstrate the ability of GridCellNet to correctly recall the precise object that it observes as a function of the amount of noise.

Figure 5b shows that in our main condition with a higher dendritic threshold ( $\theta^{\text{loc}} = 20$ ), GridCellNet remains robust up to around 30-35 flipped bits, or around 1/4 of all bits. This is also slightly more than the number of on-bits in the feature vector without noise (29). Such robustness is consistent with the known advantages of sparse representations for robustness [14]. We also show that, using the same dendritic threshold that was used for *classification* with 20 training objects per class ( $\theta^{\text{loc}} = 16$ ), recall is generally not as high-performing. This is especially the case in the low-noise setting. Interestingly, there is actually an improvement in recall accuracy as a small amount of noise is added, particularly for this lower dendritic threshold. This appears to be a result of noiseless feature vectors occasionally being compatible with multiple learned objects, particularly with an easily surpassed matching threshold. In this case, the network never settles on any one particular example that is close to the input. The introduction of some noise appears to eliminate spurious matches to learned representations, as the features that truly relate to the learned representation are more robust to noise. Importantly, in either the low or high threshold case, there is a regime in which recall accuracy approaches 100%. We discuss the significance of different dendritic thresholds and improvements with noise later.

While it is challenging to make direct comparisons to related work due to differences in approaches (e.g. Bicanski and Burgess [5] manually selected 9 salient feature locations for each object, rather than sampling the entire input space as we do), it is worth noting that successful recall is observed in spite of remarkable inter-stimulus similarity. In particular, many previous studies showing successful recall used stimuli with often high inter and intra-class diversity, such as objects and scenes as varied as teddy bears vs flower pots [27], or a Greek statue vs an image of a brain [5]. Such variety significantly simplifies the task of recalling a particular learned object, whereas any particular MNIST digit can be very similar to another within its own class, or even to that of another class.

Finally, while the total number of possible sensations (25 features) is much higher than in Bicanski and Burgess [5] (9 features), we highlight that recall normally occurs long before all of these features are sampled. For example, the mean number of sensations required for recall is  $4.2 \pm 0.1$  (mean model performance across three random seeds and 95% confidence interval) when there are 0 flipped bits and  $\theta^{\text{loc}} = 20$ . In a noisy regime, such as 20 flipped bits, we actually observe a slight decrease in the mean number of sensations, to  $2.8 \pm 0.1$ . Thus we observe that GridCellNet operates both robustly, as well as efficiently, in the recall regime.

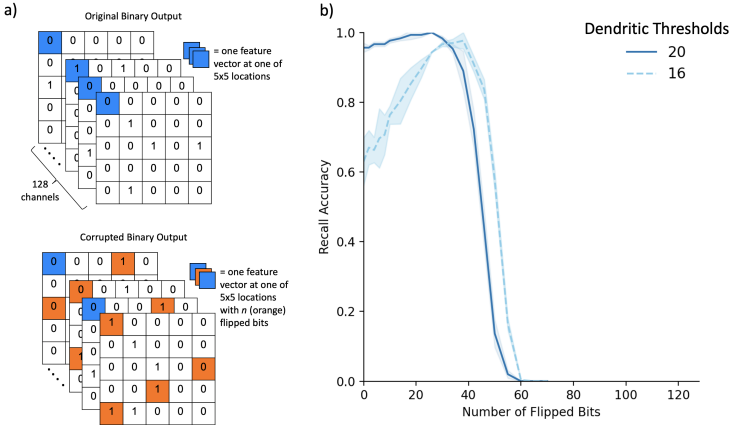


Figure 5: *Recall of Training Examples as a Function of Noise*. Following previous work, the ability of the system to recall objects learned during training is evaluated. a) To determine the robustness of this process to corrupted inputs, noise is applied at the feature level in terms of flipped bits.  $n$  random bits are flipped in *each* of the  $5 \times 5$  input features individually. b) Accuracy of recall shown as a function of flipped bits, with two different dendritic threshold values ( $\theta^{\text{loc}}$ ). The shaded region shows the 95% confidence interval of the mean across three random seeds.

### 3.4 The Use of Self-Movement Information

We have argued that GridCellNet is able to perform well in the setting of arbitrary sequence inputs because of its use of path integration. By updating internal location representations with self-movement and sensory inputs, the mutual consistency between these two sources of information enable it to accurately identify an input stimulus, even when the sequence of inputs follows an arbitrary sequence across space. It is possible however that GridCellNet is simply an effective ‘bag-of-features’ classifier - that classification would be successful given any random order of sensory inputs, with no alignment between their location on the object and the movement of sensors. More-over, it is notable that the LSTM begins to approach the performance of GridCellNet on arbitrary sequence inputs as the number of training examples increases. While this is impressive, the LSTM may also be simply integrating features without any consideration of their spatial arrangement.

To evaluate this, we perform a form of ablation where the GridCellNet and LSTM classifiers are provided with sensory and self-movement information as usual during learning, but then false motor information during inference. This motor information is derived from an alternative, fabricated sequence of movements over the object, and therefore does not align with the sequence of sensory inputs (Figure 6a). For a classifier that relies on sensorimotor alignment to predict the next sensation and appropriately update its internal representation, such an ablation should significantly effect classification accuracy. For a system that simply keys off the sensory features it receives, with no concern for their spatial relations, such a change should have minimal effect. We perform the evaluation in the setting of arbitrary sequence inputs with 10 learning examples per class.

The results in Figure 6b show that GridCellNet’s classification accuracy suffers consid-



erably in the context of false-movement information, while the LSTM’s does not. This result is consistent with the proposal that GridCellNet has the inductive bias to make use of self-movement information, and does not simply key off a random assortment of the features that it receives.

These results also support the conclusion that the LSTM uses the capabilities of back-propagation of error to learn a relatively effective mapping between randomly ordered sensory features and the output class, with no concern for their spatial arrangement. While this more simplistic approach might seemingly resolve the task at hand given sufficient training data, it would be inconsistent with how humans generally recognize objects (i.e. relying on global shape and the structured arrangements of features [9, 10]), and as observed in Figure 2 in our main submission, appears inadequate for few-shot learning with less than a dozen training examples. Furthermore, this makes the classifier vulnerable to incorrectly arranged feature inputs that a human would not classify as a particular object.

We once again note that we do not argue that RNNs are incapable of learning path integration, as given an appropriate learning setting, grid cell-like representations have been observed to emerge [68]. Rather, our aim is to show that at the outset, GridCellNet has this capability, and that this enables it to process the input sequence in such a way that accounts for the consistent spatial arrangement of features.

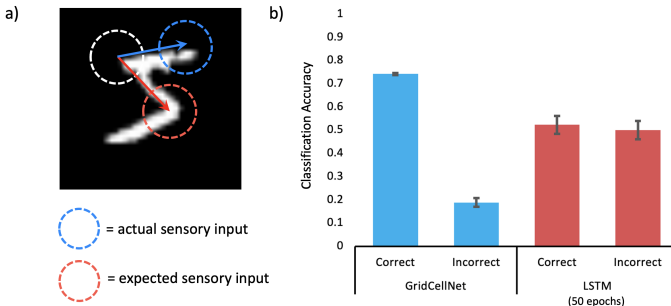


Figure 6: *Classifier Use of Self-Movement Information.* a) We aim to measure whether the classifiers are truly integrating self-movement information with sensory information, or simply keying off randomly arranged sensations without any consideration for spatial arrangement. In this condition, the motor information provided (red arrow) to the classifiers does not correspond to the actual movement required for the provided sensory input (blue circle). b) Accuracy when given the correct self-movement information, or fabricated (incorrect) self-movement information during evaluation. The error bars show the 95% confidence interval of the mean across three random seeds.

### 3.5 Rapid Inference With Partial Input Sequences

In principle, GridCellNet can successfully classify an object before it has received a complete sequence of all 25 features. Specifically, classification occurs as soon as GridCellNet’s location representation drives a particular class node’s activation above a relative threshold. Such early inference is obviously desirable as a means for increasing the efficiency of any agent relying on the classification process to interact with the world.

To assess the efficiency of GridCellNet’s inference process, we simply determined the cumulative accuracy as a function of the number of sensations used (total possible of 25). Note that, as detailed in the methods section, GridCellNet accumulates information for the first several sensations, but only begins attempting classification on the fifth sensation in order to avoid spurious classification. In Figure 7, we show that GridCellNet classifies most of the examples given to it after observing only a fraction of the total input sequence; indeed, the majority of the successful classifications occur before half of the total number of possible sensations.

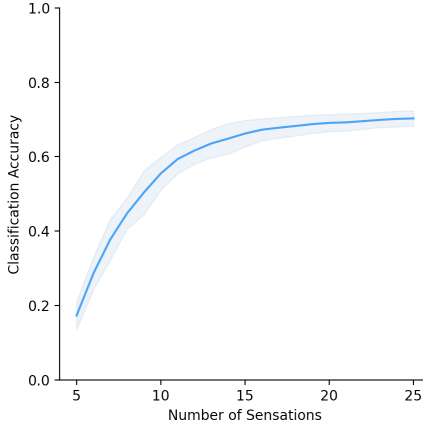


Figure 7: *Performance of GridCellNet as a Function of Sensations.* Accuracy as a function of the number of sensations. GridCellNet was trained with 5 training examples per class, using arbitrary input sequences at training and test time. NB that GridCellNet waits for 5 sensations before it begins making predictions so as to avoid false hits in early inference. The shaded regions show the 95% confidence interval of the mean across three random seeds.

### 3.6 Continual Learning

GridCellNet is not designed to specifically address the challenging issue of continual learning. Due to its use of Hebbian learning rules however, it benefits from natural robustness in this setting, which is here explored in comparison to an LSTM trained with back-propagation of error.

Continual learning is evaluated in two main learning splits (summarised in Figure 8a). The classifiers are either trained on the first 5 MNIST digits and then the next 5 (i), or the first 9 and then the final digit (ii). Accuracy is always evaluated across all classes, and so the possible performance of an ideal classifier is also shown. Note that this is different from the ‘split-MNIST’ evaluation sometimes used for continual learning [17], as it requires an optimal classifier to accumulate information to eventually be able to recognise any of the classes seen over the course of learning. The classifier heads are not modified beyond standard learning rules. Additionally, weights are never artificially fixed.

Before examining the results, it is worth clarifying how learning proceeds in GridCellNet. For a back-propagation trained network to perform well, training examples are generally interleaved and shuffled for each epoch. In GridCellNet, the default form of learning

is carried out in a continuous fashion, each class following the other and with only a single pass over each object, with no returning to previously learned objects or requirement for multiple classes to be observed in an alternating fashion. As a result, GridCellNet is in fact naturally capable of a continual learning regime that would be considered even more challenging than the 5-5 or 9-1 split we simulate (Figure 8a.iii). While we show results for a 5-5 and 9-1 split, we do not show the results of such a ‘1-1-1...-1’ split. In particular, an LSTM classifier would be essentially bounded to 10% accuracy, classifying everything as the most recently learned class. Despite such a setting being obviously catastrophic for the LSTM, it is worth noting that this is a plausible situation for an agent in the natural world to encounter. Imagine, for example, discovering a cluster of trees filled with a new kind of edible fruit, with no other fruit varieties nearby; all learning must then proceed on that single class. Such concentrations are common in the natural world [12].

The results in Figure 8b) demonstrate that GridCellNet is perfectly suited to gradually accumulate new types of objects that it encounters, with the accuracy across all the possible classes consistently improving as new classes are introduced in learning. In contrast, the LSTM demonstrates the well-known property of catastrophic forgetting. As soon as a new task is presented, knowledge required to solve the new classification problem rapidly overwrites that used for previous tasks. The results also reinforce how GridCellNet’s use of a single pass over each object (rather than multiple epochs) enables very rapid learning. Unlike the LSTM, GridCellNet does not need to repeatedly re-visit examples it has seen before.

Note that these results are in the context of non-arbitrary (i.e. fixed) input sequences, as well as 20-training examples per class. Thus we compare the performance of the systems in a setting where the LSTM has a chance of performing better than GridCellNet, even though this is not a particularly naturalistic setting (i.e. requiring more than a dozen examples per class, and spatiotemporally fixed sensory input sequences). Despite this advantage (note in particular the strong performance of the LSTM when trained on the first 9 classes in the 9-1 split), the LSTM is in a sense a victim of its own strength, and rapidly fails when attempting to extend its learning to a novel class. If we were to evaluate performance in a more natural setting with arbitrary sequences and e.g. 5-shot learning, GridCellNet’s advantage would be even more stark. Finally, we note that our evaluation is not a strict continual learning setting, in that the pre-processed sensory features were extracted across all 10 classes, although this benefit is shared by both GridCellNet and the LSTM.

## 4 Supplementary Discussion

**Grid Cells for Visual Object Recognition** This work was partly motivated by the observation that humans solve our opening task effortlessly when performing saccades, and that grid cells might enable a biologically plausible solution. It has been proposed that humans might perform object recognition in a variety of sensory modalities by making use of grid cell computations [24], including in vision [5]. As we note in our results, the ability to achieve the same classification performance on an arbitrary sequence input represents a form of out-of-distribution generalization, as the probability of a particular sequence trajectory re-occurring is astronomically low. This leaves a classifier with the option of either ignoring spatial relations entirely, or implementing a method to handle arbitrary sequences. Our work demonstrates that grid-cell computations represent a credible approach to the latter when implemented in a machine learning model.

In Bicanski and Burgess [5], the authors used features extracted from multiple locations

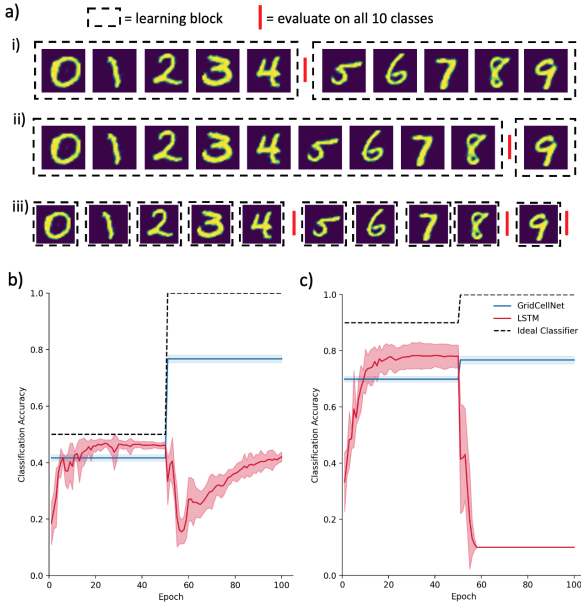


Figure 8: *Robustness in a Continual Learning Setting.* a) The training and evaluation splits used for the LSTM are shown in (i) and (ii), and for GridCellNet in (iii). For the LSTM, training examples are interleaved within a particular learning block. b) Results of the 5-5 split (left) and 9-1 split (right). As evaluation accuracy is always assessed across all 10 classes (even before they have been observed during training), the hypothetical performance of an ideal classifier is also included. The shaded regions show the 95% confidence interval of the mean across three random seeds.

of an image to perform a task with some similarities to our own, and this important work was the first biologically plausible model to demonstrate such capabilities. As in the work presented here, the features were sequentially fed to a classifier that integrated these into a learned representation. When subsequently challenged to recall which of a handful of memorized images were presented, the system successfully did so, even under settings such as partial occlusion. Importantly however, their focus was on visual recognition *memory*, and images used during training were the same as those used at evaluation time. There was therefore no need for the system to generalize to unseen examples, recognising the commonality between different instances of a class. Consistent with this, the system’s working hypothesis was always constrained to a single object. Our system is designed to represent multiple compatible hypotheses at any given time-point. These multiple representations then serve as features for GridCellNet’s final classification decision, supporting generalization. Our work therefore represents the first demonstration that grid cell-like computations can be leveraged to enable generalization on a visual task to unseen examples of an object class.

**Empowering Agents with Flexibility** Providing a system with the flexibility to perform well with novel sampling sequences has obvious appeal. Together with performing classification without traversing the entire sequence space, an agent employing such object recognition could sample the most informative regions in a principled manner, and thereby operate

much more efficiently. This can be contrasted to an agent that is constrained to sample every point, always following the same sequence, such as a raster scan across the image. A future area of investigation will be pairing GridCellNet with a reinforcement learning agent that can learn to optimally control the movement of its sensor.

**Translation Invariance** An issue relevant to both neuroscience and machine learning is that of translation invariance as it applies to classifiers that rely on sequential inputs (see Figure 4 for an intuitive demonstration). This form of translation invariance is different to the concept explored in massively parallel processing, such as invariant recognition of an object presented to a novel part of the retina [8], and yet it is still vital for a system that can operate robustly in the real world [66]. In particular, translation invariance to novel locations remains challenging for state-of-the-art classifiers, including architectures such as capsule networks [47]. This supports the notion that massively parallel processing may ultimately be limited in its ability to cope with out-of-distribution translations. More fundamentally, operating in the natural world will always require some degree of integrating separately sampled inputs. These realities support the importance of translation invariant sequential classifiers. GridCellNet uses an internal reference frame to encode the spatial relations of features and perform classification. As such, it is agnostic to the location of an object in the external environment. By demonstrating that GridCellNet can handle sensory sequences with arbitrary starting points and trajectories, we provided evidence that its classification process brings the possibility of online, arbitrary translation invariance closer.

Bicanski and Burgess [5] argued that translation invariance would be possible in their sequential, grid cell based model following additional changes. As the model was presented, however, it assumed that all of the objects were aligned in the same grid cell reference frame across training and testing. Such an assumption requires perfect path integration, and for objects in the environment to always be correctly aligned with the grid cell’s responses. As a method for the grid cells to re-align based on sensory inputs was not presented, translation invariance does not automatically follow. While we do not simulate noisy path integration, GridCellNet uses sensory inputs and iterative refinements with self-movement to infer positions in an internal reference frame, addressing these requirements. Research in the robotics literature that used an internal reference frame to achieve translation invariance [50] represented an important proof of principle, but this work was limited to a tactile (as opposed to a visual) task, and it did not explore generalization to novel instances of a class. GridCellNet addresses these limitations, serving an important step towards translation invariance for sequential classifiers.

An important limitation in our approach is that translation invariance in GridCellNet assumes that the pre-processing/foveal-response map we generate is translation equivariant above sufficient movements, and invariant to small movements. In other words, a large movement of an image in the input space should cause a corresponding shift of the representation at the feature-map level. That equivariance will hold in biology is intuitive (the foveal circuitry does not change depending on where the fovea is fixated in the external environment), which is reassuring for the plausibility of the classification system we propose. From a machine-learning perspective, our pre-processing method was justified on the basis that translation equivariance approximately holds in shallow CNNs [42], and that it is a convenient way to extract abstract features from the pixel-level input. However, in order to deploy GridCellNet to natural translated images, the pre-processing would need to be streamlined to operate in a patch-like manner end-to-end (i.e. like a biological fovea), and with a degree of translation invariance to feature movement within the receptive field. Exploring architectures that support this this will be an area for future work.

Another limitation of our work is that we do not address other forms of invariance, such as rotation or scale invariance, the latter of which was explored in Bicanski and Burgess [5]. They showed that, assuming an estimate of stimulus size is available, grid-cell representations can be appropriately updated by saccades of varying length, enabling scale invariance. Implementing a model that naturally estimates stimulus size prior to recognition, as well as addresses the question of rotation invariance, represents an interesting area for future research.

**Recall** While classifying objects is a useful ability, so too can be recalling a particular learned instance of an item. Humans appear to be quite capable at this task for a diverse range of objects [9]. Such recall has also been the primary evaluation in previous systems relying on path-integration supported recognition [8, 11, 27, 39, 50]. As such, we also demonstrated GridCellNet’s ability to perform this task, including showing that recall was robust even at considerable levels of noise at the feature-level. This finding is consistent with GridCellNet’s use of high-dimensional, sparse representations, which can be remarkably robust to noise [11]. Unlike other biologically plausible models [8], GridCellNet relies entirely on an internal reference frame to perform this recall, eliminating an important issue of neuro-plausibility, and satisfying a requirement for translation invariance.

An additional benefit of GridCellNet is that recall can operate alongside the computations that support recognition of a general object category. While we observed more robust memory recall using a higher dendritic threshold than that that used for classification, the common value of  $\theta^{\text{loc}} = 16$  appeared compatible with some degree of both recall and classification. On the one hand, it is possible that the brain might dynamically alter the sensitivity of the dendritic threshold to suit various purposes. In particular, a lower threshold supports classification, but can be too accommodating when attempting to recall a specific example. The possibility of neurons implementing such dynamic threshold adjustments is supported by both detailed computational models and experimental recordings of neurons [50].

Alternatively, we observed that the addition of small amounts of noise could actually improve recall, seemingly by ensuring the system did not spuriously believe a partly matching feature was similar to a learned representation. Unlike such spurious matches, the true feature that aligns with the learned representation should be more robust to this added noise, and as such recall can successfully converge to the correct learned object. This benefit of noise is interesting as we observed it rescuing recall to nearly 100% in the case of a fixed dendritic threshold that could also be used for classification. The implication is that even if a neuron cannot rapidly alter its threshold for dendritic spikes, intentionally adding noise to the feature-level representations could be used to dynamically switch the system from a classification state to a memory recall state.

**Predictive Representations** One appealing aspect of GridCellNet is its predictive nature. While this is crucial to how it performs inference, we demonstrated that this has the additional advantage that the system can predict unsensed regions of the input. These predictions take place even before inference, but for the sake of being able to visualize these directly, we demonstrated examples of these predictions when GridCellNet’s representation had converged to that of a single object.

As we noted in our results, GridCellNet makes its predictions at the feature level, rather than at a pixel-level. The separately trained decoder then enabled us to visualize these feature-level predictions as images. This is significant, as experimental evidence suggests humans can predict visual representations across saccades at an abstract feature level [25, 29], but not at a pixel-like level [52]. As such, GridCellNet’s capabilities align well with the psychological literature on such abilities in humans.

It is worth noting that learning in GridCellNet can be viewed as rapid binding between sensory features and their location in an object’s reference frame. The presented model therefore represents a reasonable approach to arbitrary feature binding [63] using grid-cells [24, 63], albeit here demonstrated in visual space [9]. The ability to rapidly form associative representations and use these for a variety of downstream tasks, such as predicting unsensed (e.g. occluded) features, supports the utility of this approach. It is worth emphasizing the particular strength of path integration in this context, which allows the map used for binding features to be traversed using pathways that have never been experienced. Such binding with grid cells would likely be complimentary to other forms of visual feature binding requiring more hard-coded, learned associations [24, 80, 63, 65].

**Continual, Few-shot, and Rapid Learning** We demonstrated that the proposed architecture naturally displays several desirable properties. As a consequence of the use of Hebbian learning rules and sparse representations, GridCellNet is robust in a continual learning setting, enabling it to learn to recognise a novel object without obliterating its previous classification abilities. Notably, this can take place even when the novel class is shown in isolation, as shown in our 9-1 split results. In other words, learning does not require that multiple new classes are provided in a batch. This is appealing for its realism, as it seems odd to assume that in the real world one would always encounter novel objects in batches with other novel objects, rather than in isolation.

All of our evaluations were also conducted in the setting of few-shot learning, where GridCellNet excels. We emphasize that while the accuracy achieved by GridCellNet is not as high as some other approaches to few-shot learning [35, 70], our intention is not to propose the model as a strong solution to that general setting. Rather our purpose is to show that in the context of few-shot learning, the use of grid cell representations can provide robustness to unpredictable input sequences, which should have downstream benefits for embodied agents.

Finally, the weight updates used also support extremely rapid learning in GridCellNet from an algorithmic perspective - given a novel object, the system need only perform a single pass and set of weight updates. Unlike the multiple epochs typical of back-propagation of error, GridCellNet does not need to repeatedly revisit a given example to perform learning.

**Alternative Architectures for Sequential Inputs** While we compare our architecture to an LSTM, it is possible that transformer networks [66] would perform better in this setting. They currently represent the state of the art in many sequence based tasks, including visual tasks [43], and explicitly encode positional information. This positional information is usually absolute and fixed to an external reference frame, such as the dimensions of an unrolled image. This brings with it the usual issues for translation invariance, although more recent work has explored using relative positional information in order to address this. Interestingly, it was empirically found that transformer networks using relative positional information perform less well on classification tasks [67]. It is also worth noting that, while not explored here, grid cells can in principle encode 3D structure [64]. Transformer networks already suffer from efficiency issues with long sequences, for which the introduction of a third dimension in the input representation would be problematic. The performance of transformer networks on our task and the generalization of GridCellNet to 3D objects will therefore be topics for future investigations.

Graph neural networks [68] may also be a viable basis for the tasks we outline here. However, we are not aware of work that has been done implementing graph-neural networks with an a priori representation for the 2D structure of image space together with Hebbian-learning based object recognition. It is likely that such features will be important to achieve the same performance we see across the variety of tasks explored (i.e. rapid learning and



robustness to continual learning).

**Additional Data Sets** The results presented here are based on a challenging formulation of MNIST [28] where local feature representations are sequentially fed to a classifier in a few-shot learning setting. We acknowledge that demonstrating the architecture on additional data-sets would be valuable. We would like to highlight however that MNIST is not trivial in the setting we explore. This is exemplified by the results of the baseline models, as well as the fact that previous, related models to our own have been restricted to synthetic or non-generalization settings. This is also consistent with other domains in object recognition where MNIST remains sufficiently difficult so as to be unsolved, including adversarial examples [58, 62] and robustness to general noise corruptions and out of distribution translations [47].

**General Limitations** Despite the above promising results and avenues for future research, we must highlight some general limitations of the current work. Although we attempted to ameliorate this by using separate data sub-sets, our method of feature extraction with a CNN on a held-out sample of MNIST digits is counter-intuitive given our setting of sequentially sampling feature patches, as well as our explorations of few-shot and continual learning. Importantly however, all the classifiers we compare to share this privileged feature access. As noted above, this approach also constrains the deployment of GridCellNet to real-world settings, and as such, alternative feature extraction methods in terms of both architecture (e.g. a patch-wise auto-encoder) and training methods (e.g. contrastive learning [64]) will be explored.

It is worth noting that while our results demonstrate multiple benefits of the proposed GridCellNet architecture, they also reinforce some of the strengths of neural networks that use continuous activation values and learn via back-propagation of error. In particular, while the LSTM does not have the inductive bias of GridCellNet to perform path integration in a few shot setting, it makes rich use of the features available, and scales very well as more learned examples are encountered. In regards to the former, the results from our false-motor information ablation demonstrate that GridCellNet does not extract as much information as is available from the features for predicting the target class. In terms of performance with more training data, the ceiling for accuracy appears to be higher for the LSTM as the number of learning examples grows, and indeed our own preliminary results suggest that with thousands of training examples, GridCellNet’s memorized representations begin to interfere with one-another. It is likely that the learning and inference mechanisms proposed for GridCellNet could work well in tandem with more standard deep-learning approaches (as indeed are used at the pre-processing stage) to benefit from both of these paradigms.

## Acknowledgements

Work conducted by NL was in part supported by the Biotechnology and Biological Sciences Research Council (BBSRC). We thank Jeff Hawkins, Lucas Souza, and Karan Grewal for helpful discussions and comments.

## References

- [1] Subutai Ahmad and Luiz Scheinkman. How Can We Be So Dense? The Robustness of Highly Sparse Representations. *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*, 2019.
- [2] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 2018. ISSN 10959203. doi: 10.1126/science.aar6170.
- [3] Nabiha Asghar, Lili Mou, Kira A. Selby, Kevin D. Pantasdo, Pascal Poupart, and Xin Jiang. Progressive memory banks for incremental domain adaptation, 2020. ISSN 23318422.
- [4] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology*, 2018. ISSN 15537358. doi: 10.1371/journal.pcbi.1006613.
- [5] Andrej Bicanski and Neil Burgess. A Computational Model of Visual Recognition Memory via Grid Cells. *Current Biology*, 2019. ISSN 09609822. doi: 10.1016/j.cub.2019.01.077.
- [6] Ryan Blything, Valerio Biscione, Ivan I. Vankov, Casimir J.H. Ludwig, and Jeffrey S. Bowers. The human visual system and CNNs can both support robust online translation tolerance following extreme displacements. *Journal of Vision*, 21(2), 2021. ISSN 15347362. doi: 10.1167/jov.21.2.9.
- [7] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6), 2017. ISSN 15523098. doi: 10.1109/TRO.2017.2721939.
- [8] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 2013. ISSN 01628828. doi: 10.1109/TPAMI.2012.89.
- [9] Timothy F. Brady, Talia Konkle, George A. Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences of the United States of America*, 105(38), 2008. ISSN 00278424. doi: 10.1073/pnas.0803390105.

- [10] Wieland Brendel and Matthias Bethge. Approximating CNNs with Bag-of-Local-Features Models Works Surprisingly Well on ImageNet. In *International Conference on Learning Representations*, 2019.
- [11] Bjorn Browatzki, Vadim Tikhanoﬀ, Giorgio Metta, Heinrich H. Bulthoﬀ, and Christian Wallraven. Active in-hand object recognition on a humanoid robot. *IEEE Transactions on Robotics*, 30(5), 2014. ISSN 15523098. doi: 10.1109/TRO.2014.2328779.
- [12] Richard Condit. Spatial patterns in the distribution of tropical tree species. *Science*, 288(5470), 2000. ISSN 00368075. doi: 10.1126/science.288.5470.1414.
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [14] Akihiro Eguchi, James B. Isbister, Nasir Ahmad, and Simon Stringer. The emergence of polychronization and feature binding in a spiking neural network model of the primate ventral visual system. *Psychological Review*, 2018. ISSN 0033295X. doi: 10.1037/rev0000103.
- [15] Benjamin Ehret, Christian Henning, Maria Cervera, Alexander Meulemans, Johannes von Oswald, and Benjamin Grewe. Continual Learning in Recurrent Neural Networks. *International Conference on Learning Representations*, 2021.
- [16] Ila R. Fiete, Yoram Burak, and Ted Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27), 2008. ISSN 02706474. doi: 10.1523/JNEUROSCI.5684-07.2008.
- [17] Evelyn Fix and J. L. Hodges. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3), 1989. ISSN 03067734. doi: 10.2307/1403797.
- [18] Tom Foulsham and Alan Kingstone. Fixation-dependent memory for natural scenes: An experimental test of scanpath theory. *Journal of Experimental Psychology: General*, 142(1), 2013. ISSN 00963445. doi: 10.1037/a0028227.
- [19] Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, Alex Lavin, and D. Scott Phoenix. A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science*, 358(6368), 2017. ISSN 10959203. doi: 10.1126/science.aag2612.
- [20] John Grimes. On the Failure to Detect Changes in Scenes across Saccades. *Perception*, 1996. doi: 10.1093/acprof.
- [21] Torkel Hafting, Marianne Fyhn, Sturla Molden, May Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 2005. ISSN 00280836. doi: 10.1038/nature03721.
- [22] Jeff Hawkins and Subutai Ahmad. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Frontiers in neural circuits*, 10, 2016. ISSN 16625110. doi: 10.3389/fncir.2016.00023.

- [23] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Frontiers in Neural Circuits*, 11(October):1–18, 2017. ISSN 1662-5110. doi: 10.3389/fncir.2017.00081. URL <http://journal.frontiersin.org/article/10.3389/fncir.2017.00081/full>.
- [24] Jeff Hawkins, Marcus Lewis, Mirko Klukas, Scott Purdy, and Subutai Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, 2019. ISSN 16625110. doi: 10.3389/fncir.2018.00121.
- [25] M. Hayhoe, J. Lachter, and J. Feldman. Integration of form across saccadic eye movements. *Perception*, 20(3), 1991. ISSN 03010066. doi: 10.1068/p200393.
- [26] Emily Higgins and Keith Rayner. Transsaccadic processing: stability, integration, and the potential role of remapping, 2014. ISSN 1943393X.
- [27] Kevin Hoang, Alexandre Pitti, Jean Francois Goudou, Jean Yves Dufour, and Philippe Gaussier. Active vision: On the relevance of a bio-inspired approach for object detection. *Bioinspiration and Biomimetics*, 15(2), 2020. ISSN 17483190. doi: 10.1088/1748-3190/ab504c.
- [28] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8), 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.
- [29] Linus Holm, Johan Eriksson, and Linus Andersson. Looking as if you know: Systematic object inspection precedes object recognition. *Journal of Vision*, 8(4), 2008. ISSN 15347362. doi: 10.1167/8.4.14.
- [30] Bernhard Hommel and Lorenza S. Colzato. When an object is more than a binding of its features: Evidence for two mechanisms of visual feature integration. *Visual Cognition*, 17(1-2), 2009. ISSN 13506285. doi: 10.1080/13506280802349787.
- [31] Monika Jadi, Alon Polsky, Jackie Schiller, and Bartlett W. Mel. Location-dependent effects of inhibition on local spiking in pyramidal neuron dendrites. *PLoS Computational Biology*, 8(6), 2012. ISSN 1553734X. doi: 10.1371/journal.pcbi.1002550.
- [32] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [33] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13), 2017. ISSN 10916490. doi: 10.1073/pnas.1611835114.
- [34] Mirko Klukas, Marcus Lewis, and Ila Fiete. Efficient and flexible representation of higher-dimensional cognitive variables with grid cells. *PLoS Computational Biology*, 16(4), 2020. ISSN 15537358. doi: 10.1371/journal.pcbi.1007796.

- [35] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, (2012):1–101, 2016. ISSN 14691825. doi: 10.1017/S0140525X16001837.
- [36] Hugo Larochelle and Geoffrey Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.
- [37] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Ha. LeNet. In *Proceedings of the IEEE*, number November, pages 1–46, 1998. ISBN 0018-9219. doi: 10.1109/5.726791.
- [38] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.
- [39] Marcus Lewis, Scott Purdy, Subutai Ahmad, and Jeff Hawkins. Locations in the neo-cortex: A theory of sensorimotor object recognition using cortical grid cells. *Frontiers in Neural Circuits*, 2019. ISSN 16625110. doi: 10.3389/fncir.2019.00022.
- [40] Xiaoyang Long and Sheng Jia Zhang. A novel somatosensory spatial navigation system outside the hippocampal formation, 2021. ISSN 17487838.
- [41] Xiaoyang Long, Bin Deng, Jing Cai, Zhe Chen, and Sheng-Jia Zhang. A compact spatial map in V2 visual cortex. *bioRxiv*, 2021.
- [42] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.
- [43] Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C), 1989. ISSN 00797421. doi: 10.1016/S0079-7421(08)60536-8.
- [44] G. W. McConkie and K. Rayner. Identifying the span of the effective stimulus in reading: Literature review and theories of reading. In *Theoretical models and processes of reading*. 1976.
- [45] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, volume 3, 2014.
- [46] Edvard I. Moser, Emilio Kropff, and May Britt Moser. Place cells, grid cells, and the brain’s spatial representation system, 2008. ISSN 0147006X.
- [47] Norman Mu and Justin Gilmer. MNIST-C: A Robustness Benchmark for Computer Vision. *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

- [48] Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, 2004. ISBN 1581138385. doi: 10.1145/1015330.1015435.
- [49] David Noton and Lawrence Stark. Scanpaths in eye movements during pattern perception. *Science*, 171(3968), 1971. ISSN 00368075. doi: 10.1126/science.171.3968.308.
- [50] Zachary Pezzementi, Caitlin Reyda, and Gregory D. Hager. Object mapping, recognition, and localization from tactile geometry. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5980363.
- [51] Marc’Aurelio Ranzato. On Learning Where To Look. *arXiv preprint arXiv:1405.5488*, 2014.
- [52] Keith Rayner and Alexander Pollatsek. Is visual information integrated across saccades? *Perception & Psychophysics*, 34(1), 1983. ISSN 00315117. doi: 10.3758/BF03205894.
- [53] M. Riesenhuber and T. Poggio. Are cortical models really bound by the ‘binding problem’?, 1999. ISSN 08966273.
- [54] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017.
- [55] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot Learning with Memory-Augmented Neural Networks. *NeurIPS 2016 Deep Learning Symposium*, arXiv:1605.06065, 2016.
- [56] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 2009. ISSN 10459227. doi: 10.1109/TNN.2008.2005605.
- [57] Monika Schak and Alexander Gepperth. A Study on Catastrophic Forgetting in Deep LSTM Networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11728 LNCS, 2019. doi: 10.1007/978-3-030-30484-3{\\_}56.
- [58] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.
- [59] Marco Seeland and Patrick Mäder. Multi-view classification with convolutional neural networks, 2021. ISSN 19326203.
- [60] Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. Towards training recurrent neural networks for lifelong learning, 2020. ISSN 23318422.
- [61] Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May Britt Moser, and Edvard I. Moser. The entorhinal grid map is discretized, 2012. ISSN 00280836.
- [62] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jorn Henrik Jacobsen. Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations. In *37th International Conference on Machine Learning, ICML 2020*, volume PartF168147-13, 2020.

- [63] Anne Treisman. Feature binding, attention and object perception. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 1998. ISSN 09628436. doi: 10.1098/rstb.1998.0284.
- [64] Aaron Van Den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. ISSN 23318422.
- [65] Rufin VanRullen. Binding hardwired versus on-demand feature conjunctions. *Visual Cognition*, 17(1-2), 2009. ISSN 13506285. doi: 10.1080/13506280802196451.
- [66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.
- [67] Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. On Position Embeddings in BERT. *ICLR*, 2021.
- [68] James C.R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E.J. Behrens. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183, 11 2020. ISSN 00928674. doi: 10.1016/j.cell.2020.10.024.
- [69] Calden Wloka, Iuliia Kotseruba, and John K. Tsotsos. Active Fixation Control to Predict Saccade Sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00336.
- [70] Alex Wong and Alan Yuille. One shot learning via compositions of meaningful patches. *Proceedings of the IEEE International Conference on Computer Vision (2015)*, 2015.
- [71] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *34th International Conference on Machine Learning, ICML 2017*, volume 8, 2017.
- [72] Richard Zhang. Making convolutional networks shift-invariant again. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, 2019.