Discretization of continuous input spaces in the hippocampal autoencoder

Adrián F. Amil   **Ismael T. Freire**   **Paul F.M.J. Verschure**

Donders Institute for Brain, Cognition and Behaviour - Radboud University

Nijmegen, The Netherlands

`{adrian.fernandezamil, ismael.titofreire, paul.verschure}@donders.ru.nl`

Abstract

The hippocampus has been associated with both spatial cognition and episodic memory formation, but integrating these functions into a unified framework remains challenging. Here, we demonstrate that forming discrete memories of visual events in sparse autoencoder neurons can produce spatial tuning similar to hippocampal place cells. We then show that the resulting very high-dimensional code enables neurons to discretize and tile the underlying image space with minimal overlap. Additionally, we extend our results to the auditory domain, showing that neurons similarly tile the frequency space in an experience-dependent manner. Lastly, we show that reinforcement learning agents can effectively perform various visuo-spatial cognitive tasks using these sparse, very high-dimensional representations.

1   Introduction

Ever since the seminal work of O'Keefe and Dostrovsky, 1971 on place cells in rats, decades of animal research have established the hippocampus as a neural system that learns a cognitive map of the environment and uses it for spatial navigation (E. I. Moser, Kropff, and M.-B. Moser, 2008). Further experimental studies have also identified the hippocampus as a key structure in episodic memory formation (M.-B. Moser, Rowland, and E. I. Moser, 2015). Although several attempts have been made to unify these perspectives (Eichenbaum, 2017; Redish, 1999), a coherent framework that mechanistically relates them remains elusive. Moreover, numerous experiments reporting the instability of place cell activity and their modulation by non-spatial variables (Fenton and Muller, 1998; Jercog et al., 2019) raise an open question: what are these cells truly encoding?

Efforts to answer this question have shown that place cell-like activity can emerge under various conditions: when agents optimize a predictive coding objective (Chen et al., 2022; Gornet and Thomson, 2023; Levenstein et al., 2024; Ratzon, Derdikman, and Barak, 2023; Recanatesi et al., 2021; Uria et al., 2020), when networks optimize temporal stability and pairwise decorrelation in processing visual inputs (Wyss, König, and Verschure, 2006), or when building sparse, compressed representations of environmental states (Benna and Fusi, 2021; Ketz, Morkonda, and O'Reilly, 2013; Santos-Pata et al., 2021a,b). Notably, the approach where information compression leads to spatial tuning aligns with the earlier hippocampal autoencoder model (Gluck and Myers, 1993), and has been shown to replicate several distinct place cell phenomena following environmental manipulations (Santos-Pata et al., 2021a,b).

In this work, we further investigate the mechanisms behind episodic memory formation and the emergence of place cells in the hippocampal autoencoder model. We demonstrate that sparse autoencoders can create discontinuities in the neural population manifold, discretizing arbitrary input spaces into non-overlapping receptive fields, whereby subsets of similar inputs collapse onto single neurons. When applied to visual images, this process generates place cells operating on a very high-dimensional population code. Furthermore, we show that reinforcement learning agents can

effectively use such sparse and high-dimensional hippocampal-like representations to solve spatial cognitive tasks.

2 Model and Results

2.1 Hippocampal-like place cells emerge in sparse autoencoders

We studied the learning of spatial representations by training autoencoders (Figure 1a) with randomly sampled images from four different tasks in the Animal-AI environment (Beyret et al., 2019): Double T-maze, Cylinder, Object Permanence, and Thorndike. We trained two types of autoencoders. "Dense" autoencoders aimed solely to reconstruct the input images, thus preserving input information in their latent space Z. In contrast, "sparse" autoencoders had an additional objective beyond input reconstruction: to develop sparse activity patterns in the latent space Z. This was achieved using the following loss function:

$$\mathcal{L} = \frac{1}{m}\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 + \frac{\lambda}{mn}\|\mathbf{I}_n - \mathbf{Z}^\mathrm{T}\mathbf{Z}\|_\mathrm{F}, \tag{1}$$

where m denotes the batch size, n the number of neurons in Z, and the first term is the mean squared error (MSE) between inputs $\mathbf{X}$ and their reconstructions $\hat{\mathbf{X}}$, encouraging Z to preserve input information. The second term is an orthonormal activity regularization term, whose strength is controlled by λ, pushing the Gramian $\mathbf{Z}^\mathrm{T}\mathbf{Z}$ towards the identity matrix $\mathbf{I}_n$. Since $\mathbf{Z}^\mathrm{T}\mathbf{Z}$ captures the co-activation strengths between neurons in a training sample batch, the orthonormal activity regularization promotes pairwise decorrelation while ensuring equal contribution across neurons. We found this approach yields improved and more reliable results compared to the L1 activity regularization term typically used in sparse autoencoders, particularly in alleviating the dead ReLU problem (Lu et al., 2019). For dense autoencoders, λ was set to zero, leaving only the reconstruction error. We refer the reader to the Detailed methods section in the Appendix for a complete description of the environments, dataset generation, and parameters used in this study.

Training both types of autoencoders yielded significantly different internal representations of space in their latent space. Dense autoencoders developed many neurons that fired almost everywhere in space, with no defined place fields. In contrast, sparse autoencoders developed a majority of neurons with one or two localized place fields, similar to place cells in the hippocampus (Figure 1b, c). The spatial specificity of sparse autoencoder neurons was also reflected in significantly higher spatial information scores compared to dense autoencoder neurons (Figure 1d). These results demonstrate that single-unit spatial tuning emerges in sparse autoencoders but not in dense autoencoders, despite both types of networks containing the same amount of positional information at the population level, as shown by linear decoding analyses (Figure 1e).

2.2 Sparse autoencoders discretize and tile the image space with interpretable neurons

Identifying neurons with spatial selectivity similar to hippocampal place cells allowed us to investigate what these neurons encode. Given that their spatial selectivity must arise from some form of visual selectivity, we explored whether they also exhibit localized receptive fields in image space.

We created an image space by extracting semantically-relevant image embeddings of all samples using CLIP and further reducing the dimensionality to a 2D space with UMAP. We then searched for clusters in this 2D image space by running the DBSCAN algorithm on the points corresponding to images that maximally activated a particular neuron (see Figure 2a and Detailed methods in the Appendix). These clusters formed convex hulls (i.e., patches) that corresponded to the neuron's receptive fields in the image space. When pooled together, receptive fields across neurons partitioned and covered the entire image space (see example in Figure 2b). Furthermore, we computed an overlap metric to estimate the redundancy across the neurons' receptive fields. We observed that sparse autoencoder neurons tiled the image space in a minimally-overlapping manner, in contrast to dense autoencoder neurons, whose overlap tended to be significantly higher (Figure 2c).

Additionally, we performed unit clamping experiments, setting neurons to their maximal recorded value while others were set to zero, and then decoded their activity back to images. The generated images showed a striking resemblance to the training images, making these neurons highly interpretable (Figure 2d). These results establish a solid relationship between episodic memory formation and spatial coding.
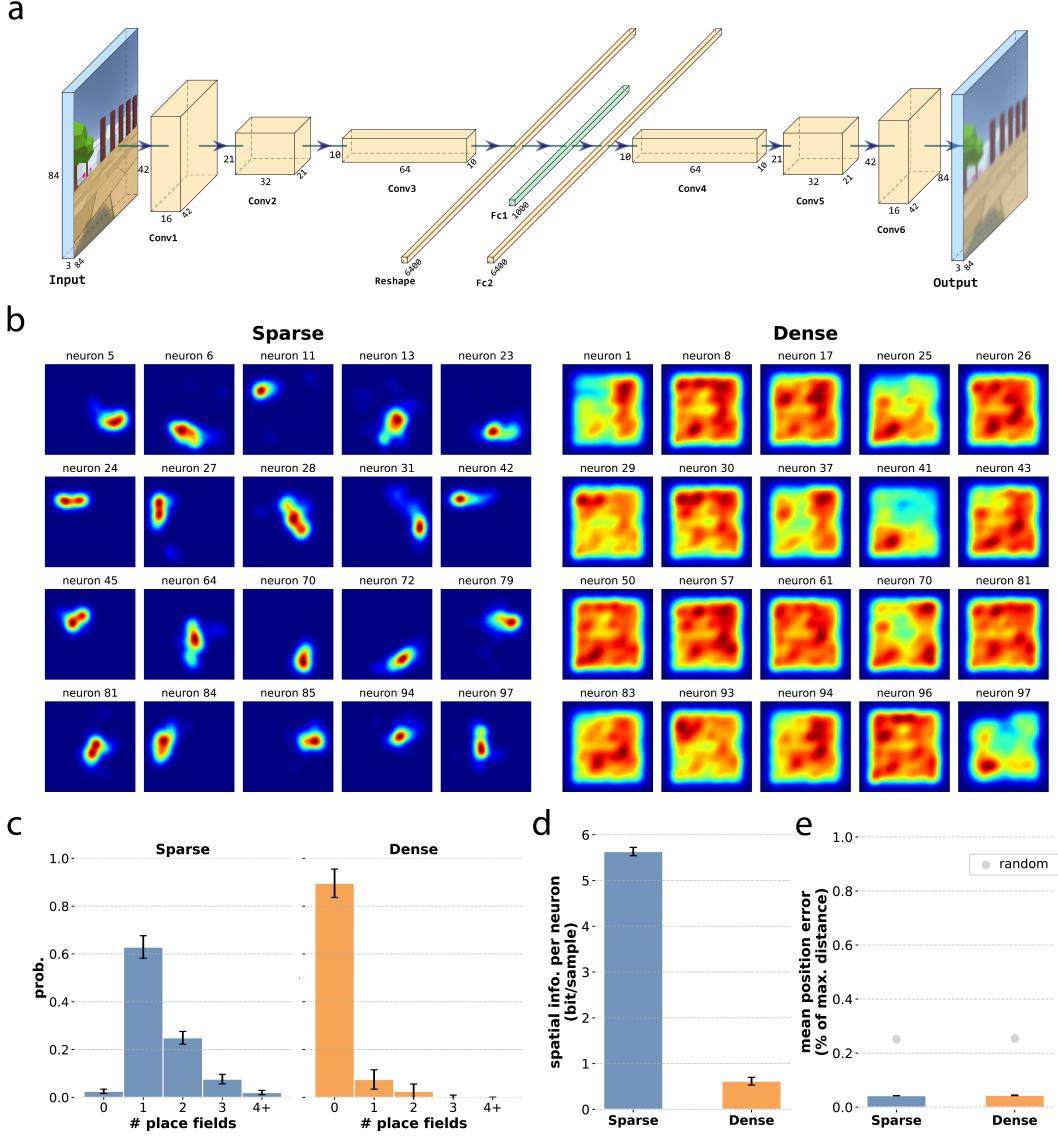
Figure 1: Hippocampal-like place cells emerge in sparse autoencoders. (a) Autoencoder architecture, featuring the hidden layer or latent space Z (denoted as Fc1) with 1000 neurons. (b) Representative examples of the neurons' spatial ratemaps for sparse and dense autoencoders. (c) Probability distribution of place field number across environments. (d) Average spatial information per neuron across environments. (e) Normalized average distance error of linear decoding of position with the ratemaps' population vectors, across environments. The grey dots represent the expected linear decoding errors after performing 1000 random permutations of the ratemaps' values.

2.3 Population structure in sparse autoencoders is grounded on mixed selectivity

Having linked the formation of episodic memories with the discretization of the image space in sparse autoencoders, we explored the population structure of the latent space representations. Inspired by Stringer et al., 2019 on the dimensionality of the population code in the mouse visual cortex, we examined the dimensionality in our autoencoders. Dimensionality was estimated by performing PCA on Z and computing the linear fit of the resulting eigenspectrum in log-log space, yielding a power-law exponent, α. High α values indicate low-dimensional codes, while low α values suggest high-dimensional codes. An $\alpha \approx 1$ indicates a high-dimensional but smooth (i.e., no discontinuities) underlying manifold, as seen in neural responses in the visual cortex (Stringer et al., 2019).
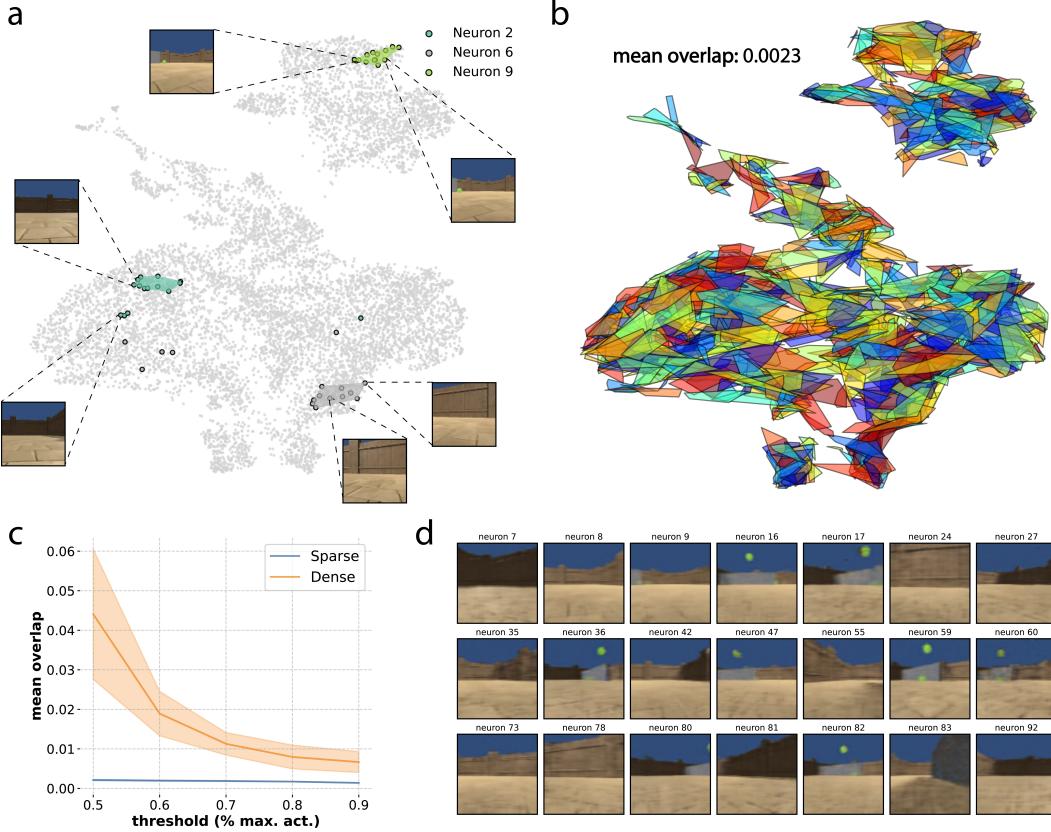
3

Figure 2: Sparse autoencoders discretize and tile the image space with interpretable neurons. (a) Images taken in one of the environments ('Cylinder'), encoded with CLIP and further reduced to two dimensions with UMAP. Points of different colors correspond to the images that maximally activate each example neuron (above the 50% threshold of the maximum neuron's recorded activity). Clusters of maximally activated images are extracted with DBSCAN, making up the convex hulls. (b) Convex hulls for all neurons in a sparse autoencoder trained with images from the 'Cylinder' environment. The overlap metric corresponds to the expected overlapping area (in %) of two randomly chosen hulls (see Detailed methods in the Appendix for further details). (c) Average overlap in 2D image space of sparse and dense autoencoders, across tasks and for a range of threshold values of maximal activation. (d) Example interpretable neurons in the sparse autoencoder. The corresponding neuron in latent space Z is set to its maximum recorded value across the dataset, while all other neurons are set to zero. Then, the enforced activity vector Z is deconvolved into an image by passing it through the decoder.

We found that dense autoencoders had dimensionality scores close to 1, similar to Stringer et al., 2019, whereas sparse autoencoders exhibited much higher dimensional representations (Figure 3a), aligning more with the efficient coding hypothesis (Barlow et al., 1961). The almost-flat eigenspectrum suggests that sparse autoencoders' population activity indeed encodes fine stimulus features. Moreover, the orthonormal activity regularization also disrupted the input-output similarity preservation typically seen in dense autoencoders (Figure 3b), further supporting the idea of a sharp discretization of the image space by sparse autoencoders.

Borrowing concepts from sparse dictionary learning (Lewicki and Sejnowski, 2000), we considered the decoder weights to be the dictionary of kernels, and the sparse neuron activities to be the coefficients that use the dictionary to reconstruct the inputs. Dense autoencoders exhibited orthogonal kernels, while sparse autoencoders showed non-orthogonal kernels despite highly uncorrelated activity patterns (Figure 3c). This indicates that orthogonal activity does not imply orthogonal kernels, and that sparse autoencoder neurons learned similar feature combinations, indicative of mixed selectivity
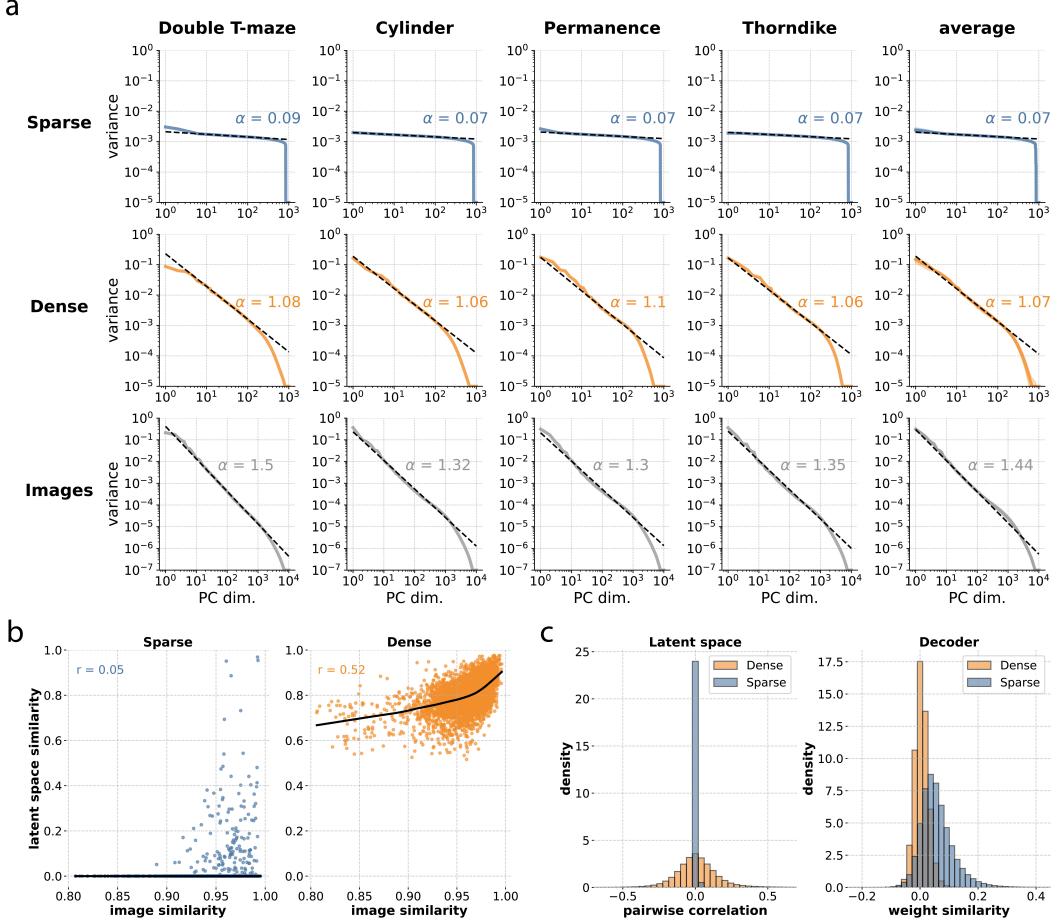
Figure 3: Population structure in sparse autoencoders is grounded on mixed selectivity. (a) Eigen-spectrum decay in latent space representations (first two rows) and images from the environments (third row). Parameter α corresponds to the power law exponent from linear fitting in log-log space. (b) Input-output similarity for sparse and dense autoencoders, with data pooled across environments. Correlation scores correspond to Spearman's rank coefficients, and fitting curves have been generated with a locally weighted scatter-plot smoother (LOWESS) for improved visualization. (c) Pairwise Pearson correlation scores between all neurons' activity in latent space, pooled across environments (left) and pairwise kernel similarity in the decoder weights (layer Fc2), representing the similarity density across "words" in the learned "dictionary" (right).

in neurons (Fusi, Miller, and Rigotti, 2016). These findings suggest that mixed selectivity and high dimensionality are closely linked to the formation of detailed episodic memories.

2.4 Zero-shot learning of place cells in sparse autoencoders

A typical observation in the hippocampus is that place cells can be identified within the first minutes of an animal being exposed to a new environment (Frank, Stanley, and Brown, 2004). Given that we have shown that sparse autoencoders exhibit very high dimensionality (Figure 3a) and single neurons tend to encode small clusters of samples (Figure 2), we investigated the extent to which neurons that learned to encode samples in one environment could generalize to encoding unseen environments and exhibit zero-shot place cells. Therefore, we trained sparse autoencoders in one environment and tested them across all others. Strikingly, neurons developed place fields with distributions very similar to those in their training environment (Figure 4a). Furthermore, the average spatial information across neurons and the mean decoding error of the rate maps were very similar, with no significant degradation compared to the training environment (Figure 4b). These results suggest that
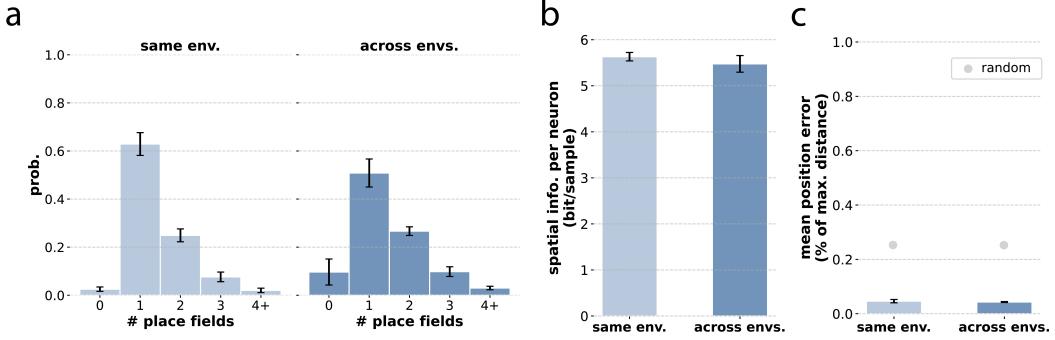
Figure 4: Zero-shot place cells in sparse autoencoders. (a) Probability distributions of place field number when testing a model within its training environment (light blue) or across unseen environments (dark blue). (b) Average spatial information per neuron, pooled across models and testing environments. (c) Normalized average distance error of linear decoding of position with the ratemaps' population vectors, across models and testing environments. The grey dots represent the expected linear decoding errors after performing 1000 random permutations of the ratemaps' values.

the network's circuitry learned to cluster and collapse similar samples onto single neurons in a more generic manner, beyond the specific details of the dataset.

2.5 Sparse autoencoders discretize and tile the input frequency space in an experience-dependent manner

If the hippocampus functions as a generic, modality-independent episodic memory system, our findings with the sparse autoencoder should generalize to other input modalities, such as sound. Indeed, "place cell"-like activity in the hippocampus has been reported for tasks involving "navigating" the sound frequency space, with neurons developing localized receptive fields around particular sound frequencies (Aronov, Nevers, and Tank, 2017). To investigate whether a similar effect could be observed within our framework, we trained autoencoders to compress and encode sound waves uniformly sampled from a linearly-varying frequency signal (Figure 5a).

We observed the emergence of frequency-specific receptive fields in sparse autoencoders, but not in dense autoencoders (Figure 5b). These receptive fields tiled the entire frequency space in a linear manner. However, when sampling was biased towards certain frequencies, the neurons' receptive fields became denser and clustered around those frequencies (Figure 5d). Additionally, similar to our previous findings with visually interpretable neurons, we found that individual neurons encoded particular frequencies so that these representations were readily decodable via clamping (Figure 5c). These results demonstrate that sparse autoencoders can discretize arbitrary input spaces to support episodic memory formation.

2.6 DQN agents learn effectively with sparse, very high-dimensional representations

Representations used to build episodic memories are likely also employed for behavioral learning in the brain. Therefore, we investigated whether hippocampal-like representations emerging in sparse autoencoders would be suitable for reinforcement learning. To test this, we employed DQNs incorporating either sparse or dense autoencoders to solve a range of tasks in the Animal-AI environment, which inherently require spatial navigation skills (see Figure 6a, and Detailed methods in the Appendix for further details on the tasks, model, and training parameters).

Very high-dimensional representations (such as those based on efficient coding) are known to be highly sensitive to slight input perturbations and are thought to generalize poorly to new, unseen samples (Nassar et al., 2020). Thus, one might expect that DQNs would struggle with tasks requiring generalization to unseen samples during training in non-stationary environments. Contrary to this expectation, we found that DQNs using sparse autoencoders were not systematically worse than those using dense autoencoders (Figure 6b). Although sparse autoencoders seemed to perform worse than dense autoencoders in two of the four tasks tested here (Object Permanence and Thorndike), they
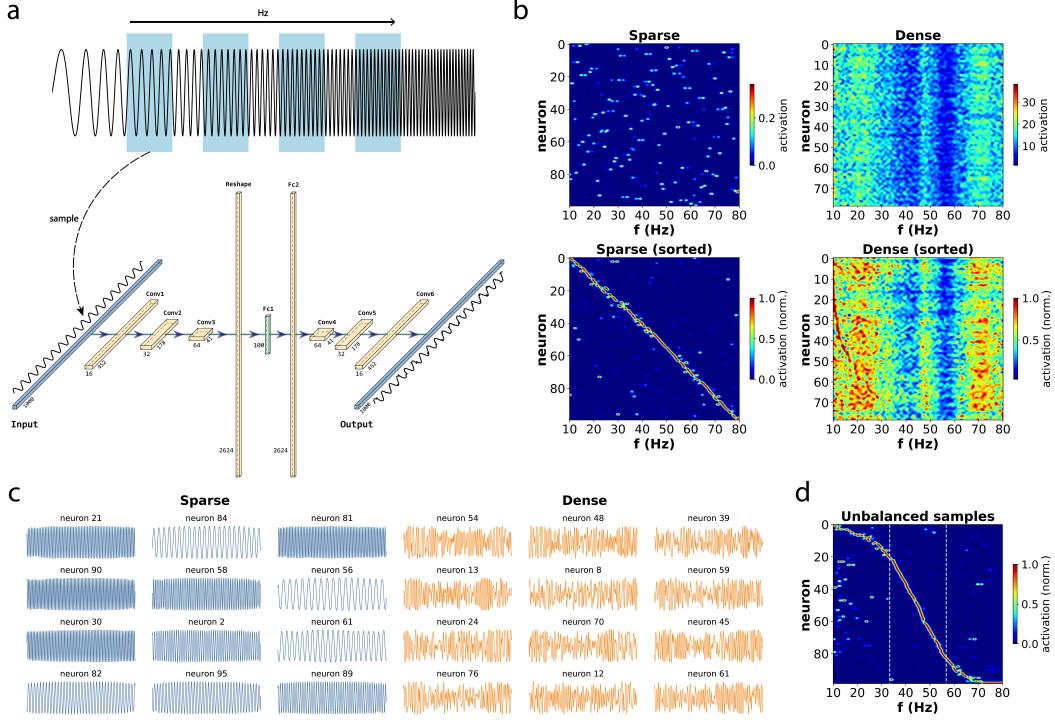
Figure 5: Sparse autoencoders discretize and tile the input frequency space in an experience-dependent manner. (a) Data samples are generated by applying a uniformly distributed sliding window to a linearly-varying frequency signal. The samples are fed into a convolutional autoencoder, analogous to the one used for vision (more details can be found in the Detailed Methods section of the Appendix). (b) Unsorted and sorted receptive fields by peak activity location for both sparse and dense autoencoders. Latent space activity Z responding to pure tone test inputs was convolved with a Gaussian kernel (sigma of 0.5 Hz), and then normalized by the maximum per neuron in the sorted plots. Lanczos interpolation was applied to all plots for improved visualization. (c) Decoded output signals after setting the corresponding neuron in latent space to its maximum recorded value across the dataset, while all other neurons were set to zero. (d) Sorted receptive fields in a sparse autoencoder trained with an unbalanced dataset. The data samples were generated with a sliding window that was not uniformly distributed in the frequency space, but whose density followed a Gaussian distribution centered at 45 Hz. Dashed vertical lines denote one standard deviation σ from the mean.

were superior in one of them (Cylinder) and had matched performance in the remaining one (Double T-maze). Therefore, while further testing is definitely needed to obtain a more reliable picture of their relative performance, these results suggest that, in practice, hippocampal-like representations can be suitable for reinforcement learning, despite their very high dimensionality.

3   Discussion

Optimization objectives underlying place cell emergence   We have demonstrated the distinct emergence of place cells in autoencoders with orthonormal activity regularization (Figure 1). Notably, sparse compression alone was sufficient to develop spatial tuning, without the need for a predictive objective (Chen et al., 2022; Gornet and Thomson, 2023; Levenstein et al., 2024; Ratzon, Derdikman, and Barak, 2023; Recanatesi et al., 2021; Uria et al., 2020). While predictive coding may explain other features of the hippocampus, such as place-cell theta sequences (Dragoi and Buzsáki, 2006), prediction does not appear to be necessary for the emergence of realistic place cells. We speculate that models optimized for next-input predictions likely learn compressed representations of the environment implicitly as part of the predictive objective. Furthermore, by training the autoencoders with randomly sampled and shuffled images, we have shown that neither temporal contiguity of samples nor any temporal correlations are required to develop place cells. This finding suggests
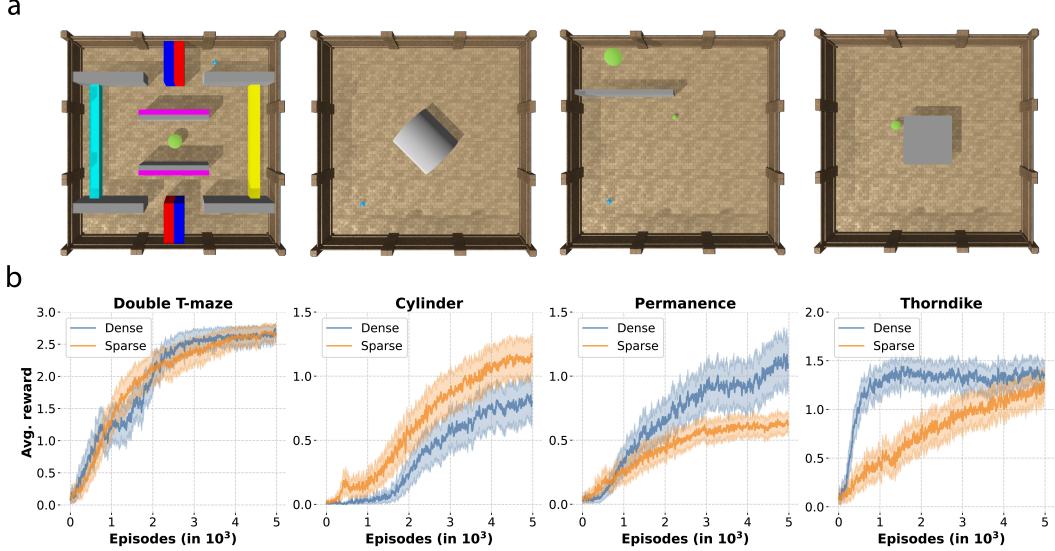
7

Figure 6: DQN agents learn effectively with sparse, very high-dimensional representations. (a) Overhead images captured from above the virtual arena of the four tasks (from left to right): Double T-maze, Cylinder, Object Permanence, and Thorndike. (b) Performance (average reward across episodes) of DQN agents using sparse and dense autoencoders across tasks.

that while predictive learning capturing temporal input correlations might correspond to experience-dependent theta sequences in the hippocampus, the formation of compressed state representations might correspond to time-independent learning processes at the gamma frequency scale (Lisman, 2005). Additionally, we have demonstrated that sparse autoencoders can learn localized receptive fields of the input space while breaking the relationship between input similarity and latent space similarity. This finding contrasts with previous research that emphasized the preservation of input-output similarity matching for learning localized receptive fields (Qin et al., 2023; Sengupta et al., 2018). Overall, it appears that sparse compression alone is sufficient to learn localized receptive fields, that in turn manifest as place cells when applied to the visual domain.

Sparse and very high-dimensional codes   A slowly-decaying eigenspectrum, where fine details are over-weighted, represents codes that create discontinuities by disrupting the locality of the manifold structure supporting the input space distribution (Nassar et al., 2020). We demonstrated that this discontinuity can be induced by an orthonormal activity regularization objective, facilitating the creation of event memories by discretizing the image input space (Figure 2). Our results suggest that very high-dimensional codes underlie the formation of place cells (Figure 3), aligning with the efficient coding hypothesis (Barlow et al., 1961), which posits that the brain maximizes information by eliminating correlations in sensory inputs, leading to sparse coding (Olshausen and Field, 1996). Indeed, it has been shown that hippocampal neurons in rodents become sparser with prolonged exposure to the environment (Ratzon, Derdikman, and Barak, 2023). Moreover, the storage of social memories in mice has been linked to high-dimensional representations in the hippocampus (Boyle et al., 2024). Importantly, sparsity has been shown to control a generalization-discrimination trade-off (Barak, Rigotti, and Fusi, 2013), which could explain why the hippocampus relies on sparse representations, well-suited for progressive discrimination of similar environments and events. Our study shows that smooth place cell maps can coexist with and emerge from extremely high-dimensional sparse codes (Chettih et al., 2023). We therefore predict that the dimensionality of the population code along the sensory hierarchy should decrease to support the learning of invariant sensory representations (Froudarakis et al., 2020), and then increase sharply at the apex, in the hippocampus, to enable the formation of detailed memories based on the specific combination of such invariant representations. This role of the hippocampus aligns with our observation of mixed selectivity, i.e., neurons learning similar feature combinations (Figure 5c), which in turn has been proposed to enable high-dimensional representations important for higher cognition areas (Bernardi et al., 2020; Fusi, Miller, and Rigotti, 2016).

Circuit mechanisms underlying memory formation   The surprising observation of zero-shot learning of place cells (Figure 4) suggests that the sparse autoencoders learned to cluster similar samples onto single neurons in a generic manner. We hypothesize that the responsible circuits might correspond to known hippocampal processes, mainly pattern completion and pattern separation (Rolls, 2013). On the one hand, neurons are pushed to collapse across-sample variability by clustering samples based on similarity, an effect akin to pattern completion. On the other hand, sparsity also imposes sharp discontinuities between clusters in neuronal space, even when they might be close in input space, a process akin to pattern separation. The combination of both processes is reminiscent of the "locality-sensitive hashing" (LSH) algorithms used in the computer science field for fast image search (Kulis and Grauman, 2009). Future work will shed light on the learned circuit mechanisms behind such a LSH in sparse autoencoders, and their potential mapping to pattern separation and completion.

Place cell over-representation near reward areas   We have shown that the development of localized receptive fields can be generalized to other input modalities, such as sound. Crucially, we used this simplified framework to demonstrate that receptive field distribution tends to be modulated by the input sampling distribution (Figure 5d). Importantly, it has been observed that the density of place fields increases near reward areas (Mamad et al., 2017). This has led researchers to seek external reward signals in the hippocampus that would modulate the place cell map (Kaufman, Geiller, and Losonczy, 2020). However, based on our results, we propose an alternative explanation based on oversampling: animals tend to spend more time within rewarded areas due to consummatory behaviors, hence biasing sensory sampling and learning. Additionally, in line with this hypothesis, place cell trajectories leading to rewards or goals tend to be replayed more often than unrewarded past trajectories (Ambrose, Pfeiffer, and Foster, 2016), which would further reinforce the sampling bias.

Biological plausibility   Although it has been claimed that error backpropagation and gradient descent are mechanisms that could be implemented in the brain (Lillicrap et al., 2020), particularly in the hippocampus (Santos-Pata et al., 2021b), we believe that such strong assumptions are unnecessary to map our model and observations to the real hippocampus. The orthonormal activity regularization term used in our sparse autoencoders could be realized by combining strong lateral inhibition (promoting pairwise decorrelation) and homeostatic plasticity (ensuring that neurons maintain equalized firing rates over time). Additionally, the orthonormal term can be thought of as sparse whitening in ReLU-like neurons (i.e., pairwise decorrelation and variance normalization in the low-firing rate regime), a mechanism proposed to be realized in the brain by an overcomplete basis of inhibitory interneuron projections (Duong, Lipshutz, et al., 2023; Duong, Simoncelli, et al., 2023; Lipshutz, Pehlevan, and Chklovskii, 2022). Therefore, we contemplate several alternative mechanisms whereby the main objective driving our sparse autoencoders could be realized in brain circuits.

Limitations   While our reinforcement learning experiments suggest that DQNs can make use of very high-dimensional representations to solve complex tasks, the present study is limited in scope (Figure 6). We tested only a few tasks (four tasks within the Animal-AI testbed) and a single model (DQN). To gain a more comprehensive understanding of the suitability of hippocampal-like representations for behavioral and policy learning, further testing is required with a broader range of tasks and models, especially in non-stationary environments where unseen samples are the norm. Additionally, future research should explore how these sparse autoencoders could enhance reinforcement learning algorithms that rely on discrete representations, potentially enabling algorithms based on, e.g., the successor representations (Dayan, 1993), to extend beyond simplified grid worlds.

Code Availability

All the code necessary to reproduce the figures and results presented in this paper has been made publicly available at: `https://github.com/adriamilcar/HashingBrain`.

Acknowledgments and Disclosure of Funding

References

Ambrose, R Ellen, Brad E Pfeiffer, and David J Foster (2016). "Reverse replay of hippocampal place cells is uniquely modulated by changing reward". In: *Neuron* 91.5, pp. 1124–1136.

Aronov, Dmitriy, Rhino Nevers, and David W Tank (2017). "Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit". In: *Nature* 543.7647, pp. 719–722.

Barak, Omri, Mattia Rigotti, and Stefano Fusi (2013). "The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off". In: *Journal of Neuroscience* 33.9, pp. 3844–3856.

Barber, C Bradford, David P Dobkin, and Hannu Huhdanpaa (1996). "The quickhull algorithm for convex hulls". In: *ACM Transactions on Mathematical Software (TOMS)* 22.4, pp. 469–483.

Barlow, Horace B et al. (1961). "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1.01, pp. 217–233.

Benna, Marcus K and Stefano Fusi (2021). "Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence". In: *Proceedings of the National Academy of Sciences* 118.51, e2018422118.

Bernardi, Silvia et al. (2020). "The geometry of abstraction in the hippocampus and prefrontal cortex". In: *Cell* 183.4, pp. 954–967.

Beyret, Benjamin et al. (2019). "The animal-ai environment: Training and testing animal-like artificial cognition". In: *arXiv preprint arXiv:1909.07483*.

Boyle, Lara M et al. (2024). "Tuned geometries of hippocampal representations meet the computational demands of social memory". In: *Neuron*.

Chen, Yusi et al. (2022). "Predictive Sequence Learning in the Hippocampal Formation". In: *bioRxiv*, pp. 2022–05.

Chettih, Selmaan N et al. (2023). "Barcoding of episodic memories in the hippocampus of a food-caching bird". In: *bioRxiv*.

Dayan, Peter (1993). "Improving generalization for temporal difference learning: The successor representation". In: *Neural computation* 5.4, pp. 613–624.

Dragoi, George and György Buzsáki (2006). "Temporal encoding of place sequences by hippocampal cell assemblies". In: *Neuron* 50.1, pp. 145–157.

Duong, Lyndon R, David Lipshutz, et al. (2023). "Statistical whitening of neural populations with gain-modulating interneurons". In: *arXiv preprint arXiv:2301.11955*.

Duong, Lyndon R, Eero P Simoncelli, et al. (2023). "Adaptive whitening with fast gain modulation and slow synaptic plasticity". In: *arXiv preprint arXiv:2308.13633*.

Eichenbaum, Howard (2017). "On the integration of space, time, and memory". In: *Neuron* 95.5, pp. 1007–1018.

Ester, Martin et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34, pp. 226–231.

Fenton, André A and Robert U Muller (1998). "Place cell discharge is extremely variable during individual passes of the rat through the firing field". In: *Proceedings of the National Academy of Sciences* 95.6, pp. 3182–3187.

Frank, Loren M, Garrett B Stanley, and Emery N Brown (2004). "Hippocampal plasticity across multiple days of exposure to novel environments". In: *Journal of Neuroscience* 24.35, pp. 7681–7689.

Froudarakis, Emmanouil et al. (2020). "Object manifold geometry across the mouse cortical visual hierarchy". In: *BioRxiv*, pp. 2020–08.

Fusi, Stefano, Earl K Miller, and Mattia Rigotti (2016). "Why neurons mix: high dimensionality for higher cognition". In: *Current opinion in neurobiology* 37, pp. 66–74.

Gillies, Sean (2013). "The shapely user manual". In: *URL https://pypi. org/project/Shapely*.

Gluck, Mark A and Catherine E Myers (1993). "Hippocampal mediation of stimulus representation: A computational theory". In: *Hippocampus* 3.4, pp. 491–516.

Gornet, James A and Matt Thomson (2023). "Automated construction of cognitive maps with predictive coding". In: *bioRxiv*, pp. 2023–09.

Jercog, Pablo E et al. (2019). "Heading direction with respect to a reference point modulates place-cell activity". In: *Nature communications* 10.1, p. 2333.

Kaufman, Alexandra Mansell, Tristan Geiller, and Attila Losonczy (2020). "A role for the locus coeruleus in hippocampal CA1 place cell reorganization during spatial reward learning". In: *Neuron* 105.6, pp. 1018–1026.

Ketz, Nicholas, Srinimisha G Morkonda, and Randall C O'Reilly (2013). "Theta coordinated error-driven learning in the hippocampus". In: *PLoS computational biology* 9.6, e1003067.

Kulis, Brian and Kristen Grauman (2009). "Kernelized locality-sensitive hashing for scalable image search". In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 2130–2137.

Levenstein, Daniel et al. (2024). "Sequential predictive learning is a unifying theory for hippocampal representation and replay". In: *bioRxiv*, pp. 2024–04.

Lewicki, Michael S and Terrence J Sejnowski (2000). "Learning overcomplete representations". In: *Neural computation* 12.2, pp. 337–365.

Lillicrap, Timothy P et al. (2020). "Backpropagation and the brain". In: *Nature Reviews Neuroscience* 21.6, pp. 335–346.

Lipshutz, David, Cengiz Pehlevan, and Dmitri B Chklovskii (2022). "Interneurons accelerate learning dynamics in recurrent neural networks for statistical adaptation". In: *arXiv preprint arXiv:2209.10634*.

Lisman, John (2005). "The theta/gamma discrete phase code occuring during the hippocampal phase precession may be a more general brain coding scheme". In: *Hippocampus* 15.7, pp. 913–922.

Lu, Lu et al. (2019). "Dying relu and initialization: Theory and numerical examples". In: *arXiv preprint arXiv:1903.06733*.

Mamad, Omar et al. (2017). "Place field assembly distribution encodes preferred locations". In: *PLoS biology* 15.9, e2002365.

McInnes, Leland, John Healy, and James Melville (2018). "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426*.

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *nature* 518.7540, pp. 529–533.

Moser, Edvard I, Emilio Kropff, and May-Britt Moser (2008). "Place cells, grid cells, and the brain's spatial representation system". In: *Annu. Rev. Neurosci.* 31, pp. 69–89.

Moser, May-Britt, David C Rowland, and Edvard I Moser (2015). "Place cells, grid cells, and memory". In: *Cold Spring Harbor perspectives in biology* 7.2, a021808.

Nassar, Josue et al. (2020). "On 1/n neural representation and robustness". In: *Advances in Neural Information Processing Systems* 33, pp. 6211–6222.

O'Keefe, John and Jonathan Dostrovsky (1971). "The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat." In: *Brain research*.

Olshausen, Bruno A and David J Field (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583, pp. 607–609.

Qin, Shanshan et al. (2023). "Coordinated drift of receptive fields in Hebbian/anti-Hebbian network models during noisy representation learning". In: *Nature Neuroscience* 26.2, pp. 339–349.

Radford, Alec et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR, pp. 8748–8763.

Ratzon, Aviv, Dori Derdikman, and Omri Barak (2023). "Representational drift as a result of implicit regularization". In: *bioRxiv*, pp. 2023–05.

Recanatesi, Stefano et al. (2021). "Predictive learning as a network mechanism for extracting low-dimensional latent space representations". In: *Nature communications* 12.1, p. 1417.

Redish, A David (1999). *Beyond the cognitive map: from place cells to episodic memory*. MIT press.

Rolls, Edmund T (2013). "The mechanisms for pattern completion and pattern separation in the hippocampus". In: *Frontiers in systems neuroscience* 7, p. 74.

Santos-Pata, Diogo et al. (2021a). "Entorhinal mismatch: A model of self-supervised learning in the hippocampus". In: *Iscience* 24.4.

– (2021b). "Epistemic autonomy: self-supervised learning in the mammalian hippocampus". In: *Trends in Cognitive Sciences* 25.7, pp. 582–595.

Sengupta, Anirvan et al. (2018). "Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks". In: *Advances in neural information processing systems* 31.

Skaggs, William, Bruce Mcnaughton, and Katalin Gothard (1992). "An information-theoretic approach to deciphering the hippocampal code". In: *Advances in neural information processing systems* 5.

Stringer, Carsen et al. (2019). "High-dimensional geometry of population responses in visual cortex". In: *Nature* 571.7765, pp. 361–365.

Uria, Benigno et al. (2020). "The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains". In: *BioRxiv*, pp. 2020–11.

Wyss, Reto, Peter König, and Paul FM J Verschure (2006). "A model of the ventral visual system based on temporal stability and local memory". In: *PLoS biology* 4.5, e120.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: There is no claim in the abstract that is not strongly supported by the results in the Model and results section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are noted in the Discussion section, especially regarding the limited number of experiments in the reinforcement learning tasks.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All mathematical assumptions of the models are clearly stated in the Detailed methods section in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the algorithms, metrics, and parameters used for all models an analyses are clearly stated in the Details methods section in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: A section of Code availability was created to share our open-source code that contains all the necessary code, data, and instructions to reproduce the results and figures.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the procedures, hyperparameters, optimizers etc. are clearly stated in the Detailed methods section in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All the plots where data is averaged or pooled across conditions include error bars or shaded areas around the means for clear visualization of significant differences.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: The experiments performed in this paper only require standard computers with standard GPUs. Therefore, we considered it irrelevant to report such information given the small scale nature of our study.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our research has been done without violating any aspect in the NeurIPS code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The nature of our experiments and results are quite theoretical and more related to the understanding of the computations in the brain than to any practical, deployable application. Therefore, we considered it irrelevant to add a discussion about positive or negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The type of data (images from an open, virtual environment) and models (simple autoencoders) used in this paper do not posses any risk we are aware of.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: This paper does not use any particular existing asset that requires special consideration besides citation of openly available, standard python libraries. The code and data was generated by us.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No particular assets are being released with this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: There are no experiments with human subjects included in this paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There are no experiments with human subjects included in this paper.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Appendix

A.1 Detailed methods

A.1.1 Model's architecture and training

Visual autoencoder (Vis-AE) To compress the images from the Animal-AI enviroment, we employed a convolutional autoencoder (ConvAE) that maps the $3 \times 84 \times 84$ to a latent space with 1000 neurons, effectively compressing the images by a factor of ~ 21. The details on the architecture can be found in Table 1.

Table 1: Architecture of the Visual Autoencoder (Vis-AE)

Layer	Type	Act. Func.	Filters/Units	Kernel Size	Stride/Padding
Input	Input	-	$3 \times 84 \times 84$	-	-
Conv1	2D Conv.	ReLU	16	4×4	2/1
Conv2	2D Conv.	ReLU	32	4×4	2/1
Conv3	2D Conv.	ReLU	64	4×4	2/1
Reshape	Reshape	-	6400	-	-
Fc1 (Z)	Linear	ReLU	1000	-	-
Fc2	Linear	ReLU	6400	-	-
Conv4	Trans. 2D Conv.	ReLU	32	4×4	2/1 (out. pad. 1)
Conv5	Trans. 2D Conv.	ReLU	16	4×4	2/1
Conv6	Trans. 2D Conv.	Sigmoid	3	4×4	2/1
Output	Output	-	$3 \times 84 \times 84$	-	-

Audio autoencoder (Aud-AE) To compress the frequency signals from the synthetic sound dataset, we employed another ConvAE that maps one-second time series data (with a resolution of 10^{-3} s) to a latent space with 100 neurons, effectively compressing the signals by a factor of 10. The details on the architecture can be found in Table 2.

Table 2: Architecture of the Audio Autoencoder (Aud-AE)

Layer	Type	Act. Func.	Filters/Units	Kernel Size	Stride/Padding
Input	Input	-	1×1000	-	-
Conv1	1D Conv.	ReLU	16	100	2/1
Conv2	1D Conv.	ReLU	32	100	2/1
Conv3	1D Conv.	ReLU	64	100	2/1
Reshape	Reshape	-	2624	-	-
Fc1 (Z)	Linear	ReLU	100	-	-
Fc2	Linear	ReLU	2624	-	-
Conv4	Trans. 1D Conv.	ReLU	32	100	2/1
Conv5	Trans. 1D Conv.	ReLU	16	100	2/1
Conv6	Trans. 1D Conv.	Tanh	1	100	2/1
Output	Output	-	1×1000	-	-

Loss function The loss function to be minimized in both autoencoders (Vis-AE and Aud-AE) includes a mean squared error (MSE) term as a reconstruction error to force the latent space to preserve the input information, and a orthonormal activity regularization term that promotes sparse representations in the latent space Z:

$$\mathcal{L} = \frac{1}{m}\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 + \frac{\lambda}{mn}\|\mathbf{I}_n - \mathbf{Z}^{\mathrm{T}}\mathbf{Z}\|_{\mathrm{F}}, \qquad (2)$$

where $\mathbf{X}$ is the input, $\hat{\mathbf{X}}$ is the output (i.e., the reconstructed input), λ is the regularization coefficient (10^3 by default), $\mathbf{I}_n$ is the identity matrix with shape $n \times n$, $\mathbf{Z}$ is the middle layer's activity matrix of shape $m \times n$, m being the batch size, and n the number of hidden units. Therefore, the Gramian $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$ (with shape $n \times n$) captures the pairwise co-activation strengths between neurons in latent space.

The symbols $\|\cdot\|_2$ and $\|\cdot\|_\mathrm{F}$ denote the squared L2-norm and the Frobenius norm, respectively. The orthonormal activity regularization term promotes pairwise neuron decorrelation while achieving an equalized contribution across neurons, alleviating the dying ReLU problem. For comparisons with the dense AE, λ was simply set to 0. We have found this orthonormal activity regularization to give improved and more reliable results than the L1 activity regulation term used in standard sparse autoencoders, especially in preventing the dead ReLU problem.

Data generation and training of Vis-AE For the visual experiments, datasets were generated by sampling a total of 10000 images in each of four Animal-AI environments: "doubleTmaze", "permanence", "cylinder", and "thorndike", at random locations within the arena (excluding the 10% of the space closest to each wall) and with random angles, following uniform distributions. Training was conducted using batches of 256 images for 10000 epochs, with independent training runs per environment. The Adam optimizer with no weight decay was used to train the network and the learning rate was set at 10^{-4}. In addition, the regularization strength λ was set to 10^3 for the sparse autoencoder, and 0 for the dense autoencoder. All weights were initialized using Xavier initialization except for Z (i.e., Fc1), whose weights were initialized following a random asymmetric initialization to minimize the dying ReLU problem (Lu et al., 2019). An early stop of 0.0005 in reconstruction loss was used to compare sparse and dense autoencoders with similar reconstruction capabilities.

Data generation and training of Aud-AE For the synthetic audio experiments, training consisted of batches of 256 one-second audio slices of varying frequencies. A sliding window of 1 second (with 1 ms shift) was applied to a linearly-varying frequency signal of total time 100 seconds, moving from 10 to 80 Hz, hence resulting in a total of 99001 samples. The sampling frequency was set at (10^4 Hz, so that the kernel size (1000) matched to one full cycle at the lowest input frequency (10 Hz). Training was conducted for 1000 epochs using the Adam optimizer with a learning rate of 10^{-4}, with no weight decay. Here, the regularization strength λ was set to 10^4 for the sparse autoencoder and 0 for the dense autoencoder. The weights were initialized as with the visual-AE, with Xavier initialization in all layers except for the the latent space Z that followed a random asymmetric initialization. An early stop of 0.002 in reconstruction loss was used to have a fair comparison between sparse and dense autoencoders.

A.1.2 Spatial tuning

Firing ratemaps To generate ratemaps from latent space activity, we first created a grid of 60×60 bins (or 30×30 for computing spatial information scores) for each neuron. For each bin in the grid, we summed the neuron's activity values for images sampled within that bin, generating an activity map in space. Then, an occupancy map was generated to account for the variability in the number of images sampled at each spatial bin (sampling density), which was used to normalize the values in the activity map. Finally, Gaussian smoothing was applied to each neuron's normalized activity map, using a standard deviation of 3 bins. The resulting maps were normalized to their corresponding maximum values, yielding smooth ratemaps representing spatially-distributed neural activity.

Place field identification To identify and quantify place fields in each neuron's ratemap, we first binarized the ratemap by setting pixels with activity below 20% of the maximum activity to zero (inactive bins) and those above to one (active bins). Clusters were identified by grouping adjacent active bins, forming a cluster if a group of active bins was completely surrounded by inactive bins. Clusters not meeting the size criteria for place cells (between 3% and 50% of the total number of bins, 3600) were discarded. The remaining clusters were considered place fields.

Spatial information Spatial information (SI) scores measure the amount of information a neuron's firing rate (ν) conveys about the agent's position ($\mathbf{r}$). For each neuron, we first normalized its ratemap (using a 30×30 bin grid) by the overall mean activity $\bar{\nu}$. Then, we computed an occupancy map that was normalized by the total number of samples to reflect the proportion of "time" spent in each bin of the ratemap, denoted as $p(\mathbf{r})$. Finally, we applied the formula introduced in Skaggs, Mcnaughton, and Gothard, 1992 to compute the SI scores:

$$\mathrm{SI} = \sum_{\mathbf{r} \in \mathbf{R}} \frac{\nu(\mathbf{r})}{\bar{\nu}} \log_2 \left(\frac{\nu(\mathbf{r})}{\bar{\nu}} \right) p(\mathbf{r}). \tag{3}$$

The average SI across all neurons in the latent space Z provides an estimate of the degree of spatial tuning that the model has developed.

Spatial position decoding   The spatial decoding error measures the expected error of a linear decoder using latent space activations Z to predict the spatial position $\mathbf{r}$. We fit a linear regression model with Z as the independent variables and $\mathbf{r}$ as the dependent variables, predicting positions as $\hat{\mathbf{R}} = Z\mathbf{W}$. Then, we compute the mean squared error (MSE) between the predicted positions $\hat{\mathbf{R}}$ and the actual ones $\mathbf{R}$:

$$\text{MSE} = \frac{1}{n_{\text{samples}}} \|\mathbf{R} - \hat{\mathbf{R}}\|_2^2. \tag{4}$$

Finally, the average spatial decoding error (MSE) is re-scaled by dividing it by the maximum distance in the environment, that is, the diagonal of the arena, computed as $d = s\sqrt{2}$, with s being the side length.

A.1.3   Interpretability

Visualizing and quantifying the network's tiling of the image space   We employed the CLIP neural network (Radford et al., 2021) to encode images (resized from $3 \times 84 \times 84$ to $3 \times 224 \times 224$) into 512-dimensional vectors. These vectors were subsequently reduced to a two-dimensional representation using UMAP (McInnes, Healy, and Melville, 2018), with 10 neighbors and a minimum distance of 0.1, enabling the visualization of the high-dimensional image space.

Neurons in the hidden layer of the autoencoder that exhibited strong activation in response to specific images—those triggering activations exceeding a certain % of their maximum activation across the dataset—were mapped to points in the 2D image space. We then identify clusters of points using the DBSCAN algorithm (Ester et al., 1996), with radius ϵ of 1 and minimum samples of 4. These parameters are very dataset-dependent and were thus selected and validated via extensive visual inspection to ensure reliable cluster identification. Convex hulls were constructed around these clusters using the Quickhull algorithm (Barber, Dobkin, and Huhdanpaa, 1996) to delineate their spatial boundaries. This allowed us to identify the regions of the input space that each neuron encodes in their activations, i.e., their receptive fields.

Let $\{H_i\}$ denote the set of convex hulls corresponding to each neuron's activated image space. The average overlap metric, $\overline{O}$, was calculated as follows:

$$\overline{O} = \frac{1}{\binom{k}{2}} \sum_{i<j} \frac{\text{Area}(H_i \cap H_j)}{\text{Area}(H_i \cup H_j)}, \tag{5}$$

where $\text{Area}(H_i \cap H_j)$ represents the area of intersection between hulls H_i and H_j, $\text{Area}(H_i \cup H_j)$ is the area of the union of hulls H_i and H_j, and k is the total number of hulls. The hull calculations were performed using the Shapely Python library (Gillies, 2013). The metric $\overline{O}$ thus represents the average proportion of overlap relative to the union for each pair of hulls and ranges from 0 (no overlap) to 1 (complete overlap), thereby providing a quantitative measure of the redundancy in the neurons' receptive fields across the image space.

Neuron clamping and decoding   To test whether neurons in Z were directly interpretable based on their single-neuron activity (therefore obviating population codes), we conducted clamping experiments. This involved setting the activation of a specific neuron i in Z to its maximum activation value observed across the dataset $\mathcal{X}$, while setting the activations of all other neurons to zero. This is represented as $z_i' = (0, \dots, 0, x_{\text{max}}, 0, \dots, 0)$ where $x_{\text{max}} = \max(\{z_i | z = f(x), x \in \mathcal{X}\})$ and $f(x)$ represents the encoding function mapping $\mathcal{X}$ to z. Then, z_i' is processed by the decoder $g(z_i')$ (with $g(x)$ representing the decoding function mapping z to $\hat{\mathcal{X}}$) to yield an output signal (image of audio wave, depending on the AE).

Population code dimensionality   The dimensionality of the population code was estimated by computing the power-law exponent α of the latent space activity Z (Stringer et al., 2019). We performed PCA on Z and computed the linear fit of the resulting eigenspectrum in log-log space over the range of the first 10 to 100 principal components. Since the exponent α provides an estimate of how fast the population activity eigenspectrum decays as new dimensions are added, high α values are indicative of low-dimensional codes, whereas low α values indicate high-dimensional codes.

A.1.4 Reinforcement learning experiments

Animal-AI Testbed The Animal-AI testbed is a comprehensive platform designed for evaluating the cognitive and learning capabilities of AI agents in a variety of tasks that simulate real-world challenges (Beyret et al., 2019). This testbed provides diverse environments where agents must use visual cues and navigate complex structures to achieve specific goals. The visual inputs from these environments are standardized to a resolution of 84 by 84 pixels, and agents can perform actions defined by a 2-dimensional vector of integers: the first component goes from 0 to 2 and corresponds to not moving, moving forward, or moving backwards, respectively; and the second component also goes from 0 to 2 and corresponds to not rotate, rotate left, or rotate right, respectively. To encourage efficient behavior, a standard frameskip of 4 is applied, and the reward value decreases by 0.001 at each step. Episodes terminate either when the agent obtains the reward or after 1000 frames.

We evaluate our reinforcement learning agents using four distinct benchmarks within the Animal-AI testbed: the Double T-maze, Object Permanence, Cylinder, and Thorndike tasks. Each of these tasks presents unique challenges that require the agent to apply different strategies and cognitive abilities.

- **Double T-maze**. Each episode starts with the agent positioned randomly at one of the corners of the maze, and the objective is to navigate to the center to obtain the reward. The center contains the only positive reward (+3) available in the environment. Due to the high and opaque maze walls, the agent cannot directly see the reward and must explore the maze to find it.

- **Object Permanence**. At the beginning of each episode, the agent observes a large reward (+3) falling behind a wall until it is completely occluded. The agent must then navigate to the hidden reward, avoiding a small and visible reward (+1) along the way.

- **Cylinder**. This task involves an opaque cylinder with a medium-sized reward hidden inside. The agent begins outside the cylinder and must navigate into the cylinder to obtain the reward (+2).

- **Thorndike**. The task tests the agent's ability to escape from a closed box to reach a reward located outside the box. The box is semi-transparent, allowing the agent to see the reward from inside. The only exit is blocked by a movable obstacle that the agent must push to escape. A medium reward (+2) outside the box is the sole positive reward available.

Model To evaluate the performance of our sparse autoencoders in reinforcement learning scenarios, we used a standard Deep Q-Network (DQN) architecture (Mnih et al., 2015) with modifications to the input layer. Instead of feeding raw pixel data from the Animal-AI environments, we used the compressed representations of 1000 units generated by the Visual Autoencoder (Vis-AE).

The loss function optimized by the Deep Q-Network (DQN) is the Mean Squared Error (MSE) between the predicted Q-values and the target Q-values, calculated using the Bellman equation:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s',d)\sim\text{ReplayBuffer}} \left[\left(r + \gamma \cdot (1-d) \cdot \max_{a'} Q_{\text{target}}(s',a';\theta^-) - Q_{\text{main}}(s,a;\theta) \right)^2 \right] \quad (6)$$

where Q_{main} is the main Q-network with parameters θ, Q_{target} is the target Q-network with parameters θ^-, s is the current state, a is the action taken, r is the reward received, s' is the next state, d is a boolean indicating whether s' is a terminal state, and γ is the discount factor. This loss function aims to minimize the difference between the Q-value predicted by the main network and the target Q-value, which is computed based on the reward and the maximum Q-value of the next state predicted by the target network. The training of the DQN was performed by using the RMSprop optimizer. The target network was periodically updated with the weights of the main DQN to stabilize training. The DQN was trained with the following hyperparameters: a learning rate of 0.00025, a discount factor (γ) of 0.99, an update frequency of 4 steps, and a target network update frequency of 2500 steps. The ϵ for the epsilon-greedy policy started at 1 and decayed linearly to 0.1 over 25000 steps. The replay buffer size was set to 25000, with a batch size of 32 for experience replay. The details on the architecture can be found in Table 3.

Training and performance metrics Each reported experiment tested two DQN agents, Sparse and Dense, which differ only in their use of different Vis-AE models (sparse and dense autoencoders, respectively) to obtain compressed representations from the environment observations as input. The

Table 3: Architecture of the Deep Q-Network (DQN)

Layer	Type	Act. Func.	Units
Input	Input	-	1000
Fc1	Linear	ReLU	100
Fc2	Linear	ReLU	50
Fc3	Linear	ReLU	25
Fc4	Linear	ReLU	9
Output	Output	-	9

two agents were evaluated across the four Animal-AI tasks described earlier. Each model run lasted 5000 episodes, and to ensure statistical reliability, each model played each task between 20 and 27 times. The reported average performance metric was calculated using a sliding window of 20 episodes.