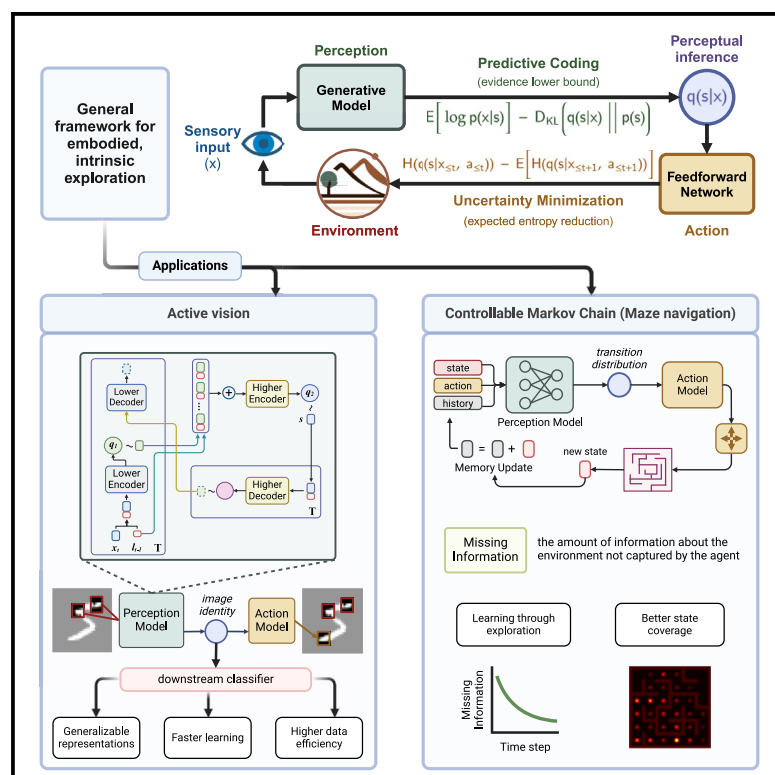# Active sensing with predictive coding and uncertainty minimization

Graphical abstract

Authors

Abdelrahman Sharafeldin, Nabil Imam, Hannah Choi

Correspondence

abdo.sharaf@gatech.edu (A.S.),
nimam6@gatech.edu (N.I.),
hannahch@gatech.edu (H.C.)

In brief

By integrating perception and action in an end-to-end differentiable architecture and utilizing ideas from theoretical neuroscience (e.g., predictive coding and uncertainty minimization), efficient learning through exploration in the absence of extrinsic reward can be achieved. Such integration, applied in real-time settings, leads to faster learning, more generalizable representations, meaningful action selection, and higher data efficiency.

Highlights

- Perception and action are usually decoupled in contemporary AI systems

- Contemporary systems lack intrinsic motivation and require external supervision or reward

- We propose an end-to-end perception-action architecture for intrinsically driven exploration

- Our approach leads to faster learning and higher data efficiency than alternative methods

CellPress

Article

Active sensing with predictive coding and uncertainty minimization

Abdelrahman Sharafeldin,[1,2,3,5,*] Nabil Imam,[1,2,4,*] and Hannah Choi[1,3,4,*]

[1]ML@GT, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[3]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA
[4]Senior author
[5]Lead contact
*Correspondence: abdo.sharaf@gatech.edu (A.S.), nimam6@gatech.edu (N.I.), hannahch@gatech.edu (H.C.)
https://doi.org/10.1016/j.patter.2024.100983

THE BIGGER PICTURE   While learning representations of the environment, biological organisms constantly take actions that alter their sensory experience. Perception depends on the active selection of this experience, a process known as active sensing. This perception-action coupling is often overlooked in conventional artificial intelligence (AI), which usually assumes passive perception or learns policies independently from an agent's representational objectives. Developing innovative approaches that couple perception and action could minimize uncertainty about sensory inputs, and embodied AI agents could focus on learning through exploration. This work explores the opportunities of integrating active sensing in the development of AI architectures.

SUMMARY

We present an end-to-end architecture for embodied exploration inspired by two biological computations: predictive coding and uncertainty minimization. The architecture can be applied to any exploration setting in a task-independent and intrinsically driven manner. We first demonstrate our approach in a maze navigation task and show that it can discover the underlying transition distributions and spatial features of the environment. Second, we apply our model to a more complex active vision task, whereby an agent actively samples its visual environment to gather information. We show that our model builds unsupervised representations through exploration that allow it to efficiently categorize visual scenes. We further show that using these representations for downstream classification leads to superior data efficiency and learning speed compared to other baselines while maintaining lower parameter complexity. Finally, the modular structure of our model facilitates interpretability, allowing us to probe its internal mechanisms and representations during exploration.

INTRODUCTION

Biological organisms interact with the world in cycles of perception and action. These two processes are intertwined and interact with one another to guide animal behavior.[1-4] Visual perception, for example, is not passive. Rather, we actively sample our visual field in search of information (Figure 1A), a process called active vision in neuroscience and psychology.[5-10] Similarly, an animal navigating a maze explores its environment and builds accurate representations of its structure, subsequently using them for various goal-directed tasks (Figure 1B). In contrast, most models of artificial intelligence (AI) treat perception and action as separate processes and aim to optimize performance with respect to task-specific objectives. For

example, visual recognition in machine learning often utilizes convolutional neural networks,[11,12] which passively receive entire images as input, to directly maximize classification accuracy on a given dataset. Another example is reinforcement learning (RL),[13] whereby actions are chosen primarily to maximize extrinsic reward without accounting for the agent's intrinsic motivations and priors. By leveraging insights from neuroscientific theories of perception and action, we can develop embodied AI models that actively explore their environment and interact with the physical world.[14]

In this work, we integrate two theories from systems neuroscience to develop a combined perception-action model for intrinsically driven active sensing. The perception component of our model is based on the theory of predictive coding.[15] According

to predictive coding, the brain maintains a generative model of the world,[16] which it uses to predict its sensory input. The goal of perception, therefore, is to infer the latent states of this generative model[17] so as to minimize prediction error. The action component of our model is based on the proposition that the brain minimizes uncertainty of inferred latent states during exploratory behavior.[4,18–21] Due to the intractability of the uncertainty reduction objective (or, equivalently, the information gain objective), most models that optimize it rely either on sample-inefficient RL methods or on restrictive assumptions that make it easier to evaluate. In our approach, we use a deep generative model based on predictive coding that allows us to optimize a Monte Carlo (MC) approximation to the information gain objective in a fully differentiable manner without assuming explicit knowledge of the true generative model of the environment. We show that this approximation, even when done in a greedy fashion, leads to a highly efficient exploration strategy.

Our model integrates perception and action within an end-to-end differentiable procedure and can be generally applied to any exploration setting in a task-independent manner without the need for extrinsic reward signals. To illustrate this, we evaluate our model on two sensorimotor tasks. First, we test the model on a simple maze navigation task with noisy transitions and show that it explores the environment more efficiently than both random exploration and visitation-count-based Boltzmann exploration. We show that our agent learns an exploration policy that enables its perception model to quickly discover the underlying transition distributions of the maze. Second, we apply our model to the more complicated task of active vision, illustrated in Figure 2. In this task, the model has a band-limited sensor, which it uses to perceive small patches of a hidden image through a limited number of fixations. We show that, despite its band-limited perception, the model is able to learn the spatial relationships between pixels of a given image, as demonstrated by its ability to generate full meaningful images by combining smaller generated patches at different locations. Furthermore, we show that although these representations are learned unsupervised, they enable a downstream classifier to quickly reach high test performance with fewer training data and lower parameter complexity. We compare these results to a feedforward network receiving full images as well as to other popular baselines from the RL literature including the recurrent attention model (RAM),[22] variational information-maximizing exploration (VIME),[23] and Plan2Explore.[24]

Our model selects actions that are purely intrinsically driven to minimize its uncertainty about the environment (Figures 1A and 1B). In that sense, the action component is blind to the task at hand and only has access to the perception model's internal states. This makes its ability to perform well on image classification quite remarkable. Importantly, the quality of the perceptual representations depends on the action selection strategy, highlighting the perception-action relationship captured by our framework. We quantify this effect by measuring the mutual information between image categories and learned representations under different action strategies. Furthermore, the modular structure of our model facilitates interpretability, allowing us to probe its mechanisms and representations during exploration and providing us with insights into the possible neural computations utilized in biological systems. For example, we show that

during active vision, the model learns representations that reflect the properties of the data and the structure of the task. Our approach demonstrates the promise of integrating neuroscientific theories of perception and action into embodied AI agents, and we hope that it will motivate more research in this area. An extended survey of related work is provided in Note S2.

RESULTS

Controllable Markov chains
Experimental setup
In the controllable Markov chain (CMC) setting, we test our model against two baselines, random exploration and visitation-count-based Boltzmann exploration, where the probability of a transition is inversely proportional to its visitation count. More precisely, during Boltzmann exploration, actions are sampled from the distribution

$$\pi(a_t) = \frac{\exp\left(-\frac{1}{\tau}\sum_{s'}\boldsymbol{h}_{s_t,a_t,s'}\right)}{\sum_a \exp\left(-\frac{1}{\tau}\sum_{s'}\boldsymbol{h}_{s_t,a,s'}\right)}, \quad \text{(Equation 1)}$$

where τ is a temperature parameter that is linearly annealed from 1.0 to 0.1 throughout the episode. For all agents, the perception model is used to learn the underlying distributions from observations. To quantify how well the model's learned distributions approximate the true distributions in the environment, we use the measure of missing information[20] (I_M),

$$I_M(\mathcal{P}\|\widehat{\mathcal{P}}) = \sum_{s \in S, a \in \mathcal{A}} D_{KL}(p(:|s,a)\|\widehat{p}(:|s,a)), \quad \text{(Equation 2)}$$

where p and $\widehat{p}$ are the true and learned distributions, respectively.

We performed experiments in two CMC settings: Dense Worlds[20] and Mazes. In the Dense Worlds setting, there are 10 states and 4 actions. For each state-action combination, a transition distribution is drawn from a Dirichlet distribution with the concentration parameter $\alpha = 1$. Note that there is no particular interpretation associated with the states and actions in these environments; they are constructed primarily to test the perception model's ability to learn the true transition distributions. As such, this setting is not appropriate for testing our Bayesian action selection (BAS) model, since random action selection can perform well in these environments when run for a sufficient number of steps. Therefore, we conduct additional tests in the Mazes setting, where there are $N = n^2$ states corresponding to locations in an $n \times n$ maze, and 4 actions corresponding to the cardinal directions (up, down, right, and left). Each action produces a noisy translation, with more bias toward the cardinal direction associated with that action. All transitions that do not correspond to a one-step translation (i.e., a neighboring state) are assigned a probability of zero. The mazes are randomly generated and the probability distributions in $\mathcal{P}$ are drawn from a Dirichlet distribution with concentration parameters $\alpha = 0.25$ for states with non-zero probability. The experimental settings, as well as architecture and hyperparameter specifications, are included in section C of the supplemental experimental procedures.
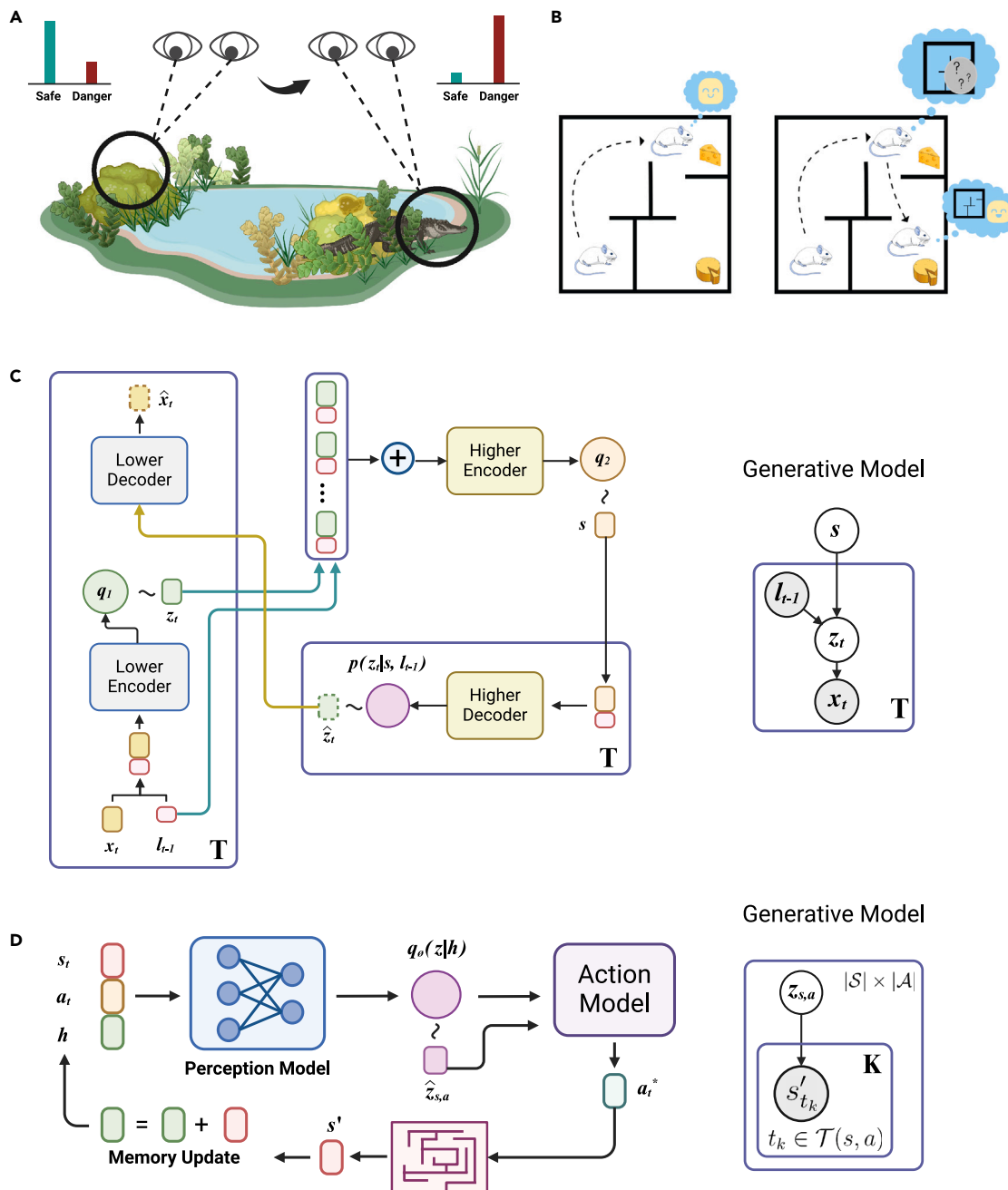
Figure 1. Models for active vision and exploration in CMCs

(A) We actively sample visual scenes to infer hidden states, in contrast to standard machine learning models, which assume passive perception.

(B) Biological systems have an intrinsic drive to actively explore the environment and build internal models of it; in contrast, traditional RL models are primarily guided by extrinsic reward.

(C) The generative model used in the active vision setting (right) and the architecture of the perception model for active vision (left). At every time step, the lower-level VAE receives an observation x_t along with the corresponding fixation location l_{t-1} and infers a lower-level representation z_t. After T time steps, the lower-level representations $z_{1:T}$ and the corresponding locations $l_{0:T-1}$ are concatenated, summed, and used as input to the higher-level VAE, which infers an abstract representation s that is then used to hierarchically output reconstructions of lower-level sensory states $\hat{z}_t$ and $\hat{x}_t$.

(D) The generative model for CMCs (right) and the architecture of the agent used in CMC exploration (left). Gray-shaded circles in the generative models represent observed variables, while unshaded circles represent latent variables.

Learning and exploration during maze navigation

To show that our perception model is able to learn the underlying transition distributions of CMCs, we run it for 2,000 time steps in the Dense Worlds environment. We then compute the differences between the distributions learned by the agent and the true environment distributions. To see the effect of
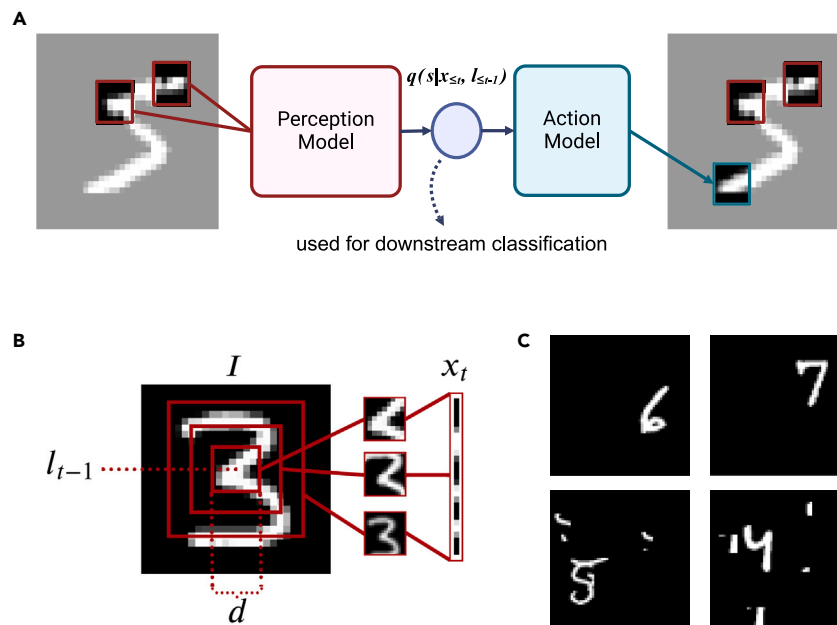
A

Figure 2. Setup for active vision

(A) The setup used in our active vision experiments. The perception model infers latent states based on observations selected by the action model. While this process is performed in an unsupervised manner, the inferred latent states can be used as input to downstream classifiers.

(B) Foveation setup for the band-limited sensor in the active vision task (described in the main text).

(C) Example images from the translated (top row) and cluttered (bottom row) MNIST datasets.

B

C

our exploration strategy, the same analysis is performed for a randomly exploring agent that uses the same perception model. Figure S2 shows that our model is indeed able to learn good estimates of the environment's transition distributions. These estimates are better when data are collected using BAS, which shows that our agent collects data that are particularly beneficial for the learning progress of the perception model.

Next, we test our model's ability to explore more complicated maze environments. We test three agents all using our perception model but employing different exploration strategies: BAS, random action selection, and Boltzmann exploration. The test environment was a 6×6 maze, and the agents were allowed to navigate for 3,000 time steps. We also performed the same tests in larger mazes and obtained similar results. Figure 3A shows that exploration using BAS leads to a significantly faster reduction in missing information compared to the other two baselines, despite BAS and Boltzmann achieving similar state-action space coverage performance. Figure 3A also shows that, while the BAS explorer covers the state-action space more quickly than the Boltzmann explorer, it does so more efficiently than the random explorer. To see this more clearly, we visualize the visitation frequencies of the BAS and random explorers as heatmaps that show how much time an agent spends in each state (Figures 3C and S4).

Active vision
Experimental setup

In the active vision task, the model explores a hidden image through a sequence of fixations. Each fixation yields a sample of the image at the fixation location. The size of this sample corresponds to the size of the model's fovea and is always less than the size of the input image. Furthermore, "foveated" samples can be extracted by a foveation process[22] as illustrated in Figure 2B. Specifically, let l_{t-1} denote the location of the fixation that generates the sample x_t from the input image I. We use

normalized coordinates so $l_{t-1} \in [-1,1] \times [-1,1]$, with $(-1,-1)$ corresponding to the top left corner of the image. Let d denote both the height and width of the model's fovea, and let N_{fov} denote the number of foveated patches (the number of red squares in Figure 2B). We first extract N_{fov} patches of increasing size, all centered at l_{t-1}. We then downsample all patches so they all have the same size $d \times d$. The patches are then flattened and concatenated to generate x_t, which is the input to the model. Note that this is the same setup used by Mnih et al.[22]

We test our active vision model on multiple image datasets, including the Modified National Institute of Standards and Technology (MNIST) dataset,[25] fashion MNIST,[26] and grayscale CIFAR-10.[27] First, we test the model's ability to produce meaningful images by generating and combining small patches at different locations. This ability reflects an implicit understanding of the spatial relationships between different locations on an image of a given object. Second, despite the model being trained with unsupervised objectives, we test its representations on a downstream image classification task, whereby only a separate decision network is trained with the supervised classification loss. The setup for this task is illustrated in Figure 2A.

We also test the model's ability to build translation and clutter invariant representations using two variants of the MNIST dataset: translated MNIST and cluttered MNIST.[22] The translated MNIST dataset consists of 60×60 images with a handwritten digit placed at a random location in the image. The cluttered MNIST dataset consists of 60×60 images created by first placing a full digit at a random location and then randomly scattering multiple cropped parts of other digits over the canvas. Image examples from both datasets are shown in Figure 2C.

The complete specification of hyperparameters and settings for the active vision experiments is included in section D of the supplemental experimental procedures. For the image classification tasks, we evaluate four variants of our model to investigate the role of each component. Below is a description of each variant.

BAS + perception. This is our full proposed model. Fixation locations are selected using BAS and the internal representation s of the perception model is used as input to the decision network.

Random + perception. Here, fixation locations are selected randomly. The input to the decision network is still the abstract state s inferred by the perception model.
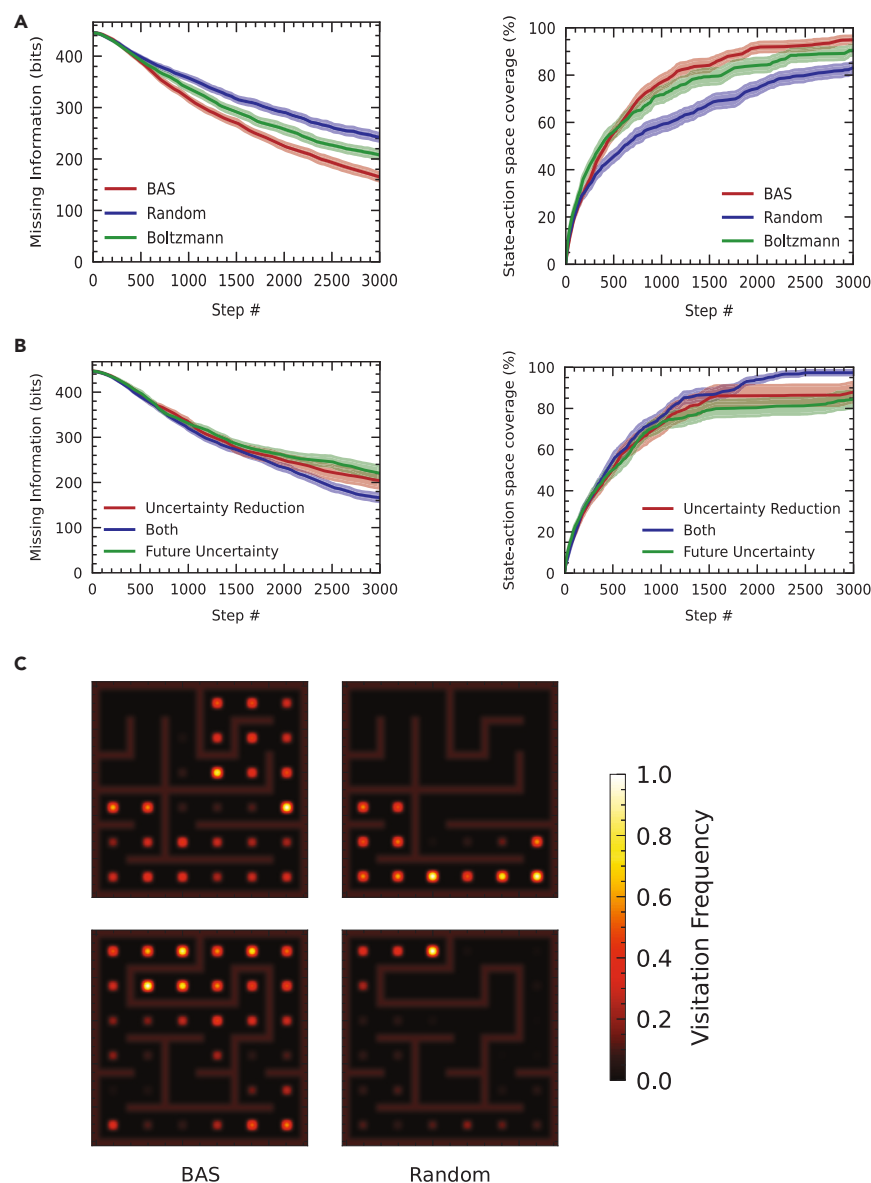
Figure 3. Learning and exploration in CMCs
(A) Missing information and percent state-action space coverage in a 6×6 maze for all three agents described in the main text. Exploration with BAS (our model) achieves faster learning of the underlying transition distributions while visiting more positions in the maze. Shaded error bars represent SEM (n = 20 random seeds).
(B) Results of the ablation analysis of the two terms in the BAS objective for CMCs. Exploration with the full objective combining the uncertainty reduction and the expected future uncertainty terms is more effective than using either term alone. Shaded error bars represent SEM (n = 20 random seeds).
(C) Example visitation frequency maps for a 6×6 maze explored by BAS versus a random exploration strategy. Both agents were allowed to run in the environment for 1,000 time steps. Visitation frequencies are normalized by the maximum visitation frequency in each case. Positions visited more often have colors closer to yellow or white, while less visited states correspond to darker shades of red. More examples are shown in Figure S4.

that action selection relies on this model, the optimality of our actions depends on the quality of our generative models. In this section, we investigate how good our trained model is at generating new patches of images and inferring the underlying states from sequences of random fixations.

Figure 4A shows examples of trials in which a random sequence of patches is given to the perception model. At the end of the sequence, the model infers the abstract state s that might underlie the given observations. From its estimate of s, it computes reconstructions of each observed patch. Additionally, we can generate unobserved patches from this inferred estimate by querying the decoder networks at different loca-

BAS + RNN. Here, the fixation locations are selected with BAS. However, instead of using the internal representation s as input to the decision network, we use a separate recurrent neural network (RNN) that integrates the collected observations $x_{1:T}$ and passes its hidden state vector to the decision network. To ensure a fair comparison between this variant and the perception-based variants, the size of this RNN is always the same as the dimensionality of the abstract latent variable s. Note that, in this case, the perception model is still involved in the action selection process, since BAS determines action scores based on the entropy of the latent state distribution.
Random + RNN. This is the same as BAS + RNN, but we replace BAS with a random selection of fixation locations.

Probing the generative model
Predictive coding posits that the brain learns and maintains a generative model of the world that allows it to make good predictions about the environment. Additionally, since we assume

tions in space. As seen in Figure 4A, when the generated patches at the nine central locations are put together, we get a meaningful image that corresponds to what the model imagines the underlying digit is. This is interesting, since the entire image at once is never observed by the model, nor is it used in the training losses. A similar effect is observed when testing the model on CIFAR-10 images (Figure S6), although the generated images tend to capture global statistics rather than local details due to the simplicity of our architecture, which utilizes only feedforward networks. These results demonstrate that the model successfully learns the spatial relationships between patches corresponding to individual image categories in a completely unsupervised manner, which explains its superior performance during classification later on.

Interaction between the perception and action models
One of the benefits of our framework is its modularity (perception and action components), which allows us to examine how each
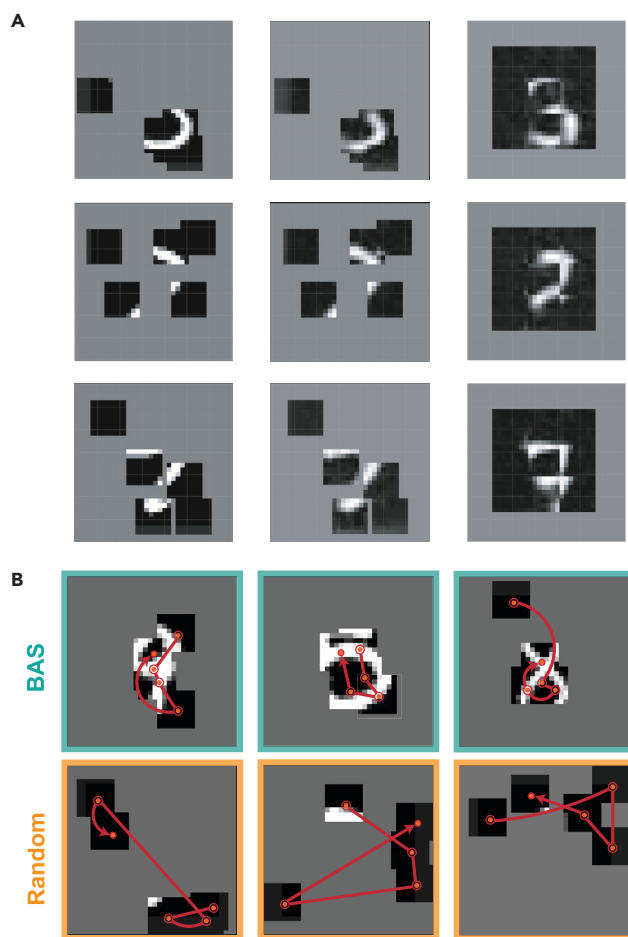
Figure 4. Probing the generative model and its effect on action selection

(A) Original (observed) patches of input images (left column) and their reconstructions (middle column). After the model infers an abstract representation, it generates an imagined digit at the unobserved locations (right column). Note that the model never observes the full image of any digit in the dataset during training, yet it is able to capture the spatial relationships between patches associated with a digit from many partial observations of its images in the dataset. Similar analyses were performed for the fashion MNIST and the grayscale CIFAR-10 datasets (Figures S5 and S6).

(B) Fixation sequences generated using BAS versus random action selection. After the initial random fixation, the BAS strategy always fixates near the center of the image, reflecting an understanding of the underlying structure of the data and its most informative locations.

component affects the other. First, we look at how the representations learned by the perception model affect what actions are selected. In the centered MNIST dataset, the most informative location about the category of the image is the center. This is reflected in the representations of the perception model, which is able to produce meaningful digits by generating and combining individual patches. Therefore, a strategy that minimizes uncertainty would ideally choose to fixate at the center most of the time. Figure 4B shows that this is exactly the case. When we compare fixation sequences selected by BAS against those selected by a random strategy, the BAS strategy almost always chooses the center as its second fixation location after the initial

random fixation. This shows that the statistical regularities in the environment are reflected in the behavior of the action model.

Second, we study how selecting actions with BAS affects the latent representations developed through perception. We trained two perception models on data collected with random fixations and data collected with BAS. We then presented both models with data collected using BAS and examined the resulting representations in their latent spaces. We use principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE)[28] to visualize these representations in 3D space, as shown in Figures 5B and S3. We also quantify the quality of the representations with respect to image categories by estimating the mutual information[29,30] between the three principal component (PC) features and image labels, as shown in Figure 5C.

Performance on downstream image classification

We demonstrate our model's ability to perform image recognition using the unsupervised representations it learns through free visual exploration. Figure 5A shows performance on the centered, translated, and cluttered MNIST datasets. Performance on fashion MNIST is reported in Figure S5. On centered MNIST, our BAS strategy yields better performance than a random action selection strategy. Furthermore, when the internal states of the perception model are used as input to the decision network, the classification accuracy is higher compared to using a separate RNN that integrates previous observations, indicating that the learned representations are more informative about the data.

On both translated and cluttered MNIST, the performance generally gets worse. However, we can still see that our BAS strategy outperforms a random exploration strategy, while using the perception representations leads to slightly higher asymptotic performance than using RNN states. However, the representations of the perception model do not seem to offer more benefit than a regular RNN in terms of learning speed. One reason for this might be the absence of a statistical regularity in the locations of digits. Therefore, encoding abstract representations in an individual state s may not be sufficient, since other hidden states, such as digit location, affect the generative process. Nevertheless, the model is able to learn informative representations as evidenced by the effectiveness of BAS in selecting fixation locations. Note that, in the case of translated and cluttered MNIST, the model observes foveated samples that cover a larger area of the image at each location, albeit with lower resolution toward the periphery. Therefore, it is possible for the model to accumulate observations of all parts of the digit during the fixation process. This may explain why RNN-based methods learn more quickly on translated and cluttered MNIST compared to centered MNIST.

Faster training and generalization with Bayesian action selection

An important feature of our approach is that it can be trained in a completely unsupervised manner to explore visual scenes and build generative representations of them. We asked whether this feature can help improve the computational efficiency and training speed of a separate downstream classifier whose parameters are trained with the supervised classification loss.

To test this, we look at the learning speed of a downstream classifier trained with full images (Full Images + FF) versus one
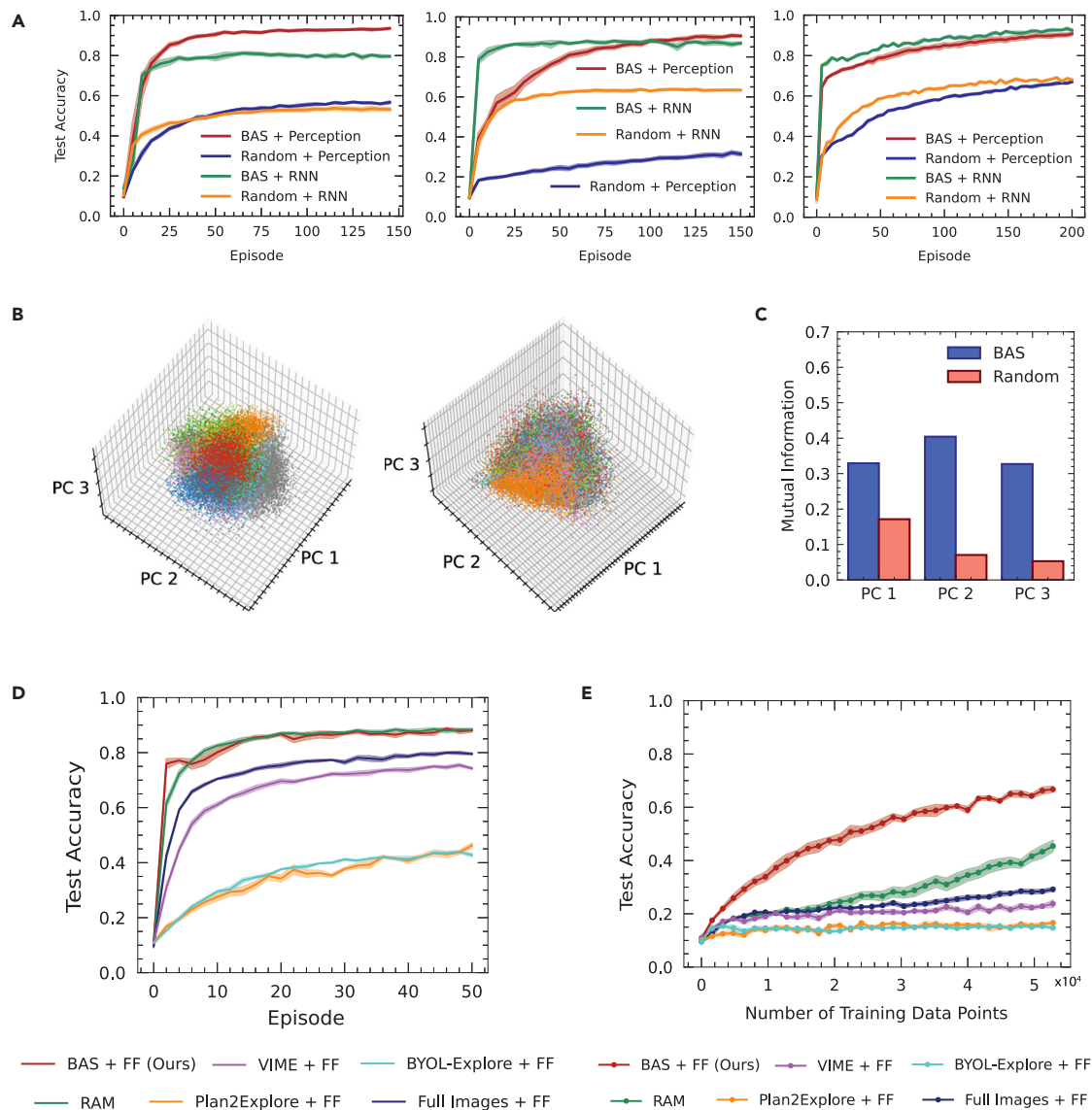
Figure 5. Performance on downstream image classification

(A) Performance of the four model variants described in the main text on the centered (left), translated (middle), and cluttered (right) MNIST datasets. Shaded error bars indicate SEM (n = 5 random seeds).

(B) PCA projections of the latent representations learned by a BAS-trained perception model (left) and a randomly trained perception model (right). Each point in the PC space corresponds to the projection of the inferred latent state s for a given input image. Points are colored based on the class of their corresponding input images. Figure S3 shows the results of performing the same analysis but with t-SNE[28] projections instead.

(C) Estimated mutual information between the latent representations' PC projections and class labels for the perception model trained with BAS and that trained with random exploration.

(D) Comparison of our model with different baselines in terms of their learning speed on the classification task with translated MNIST. Shaded error bars represent the SEM (n = 5 random seeds).

(E) Data efficiency and generalization experiments. We compare the same methods in (D) in terms of their test performance during the first episode of supervised training, when the classification networks see the data for the first time. Each point on the plot represents the test score after observing only x training data points for the first time. Shaded error bars represent the SEM (n = 5 random seeds).

trained on concatenated patches collected using BAS (BAS + FF). We also compare our approach to four baselines from the RL literature: the RAM,[22] VIME,[23] Plan2Explore,[24] and BYOL-Explore.[31] RAM is a popular method from the machine learning literature that is known to achieve high performance on the same task we consider here, while VIME, Plan2Explore, and

BYOL-Explore are popular intrinsically motivated exploration methods in RL that use uncertainty to guide exploration. These experiments were conducted on the translated MNIST data, since the dimensionality of its observation space is higher, so it serves as a good test of learning speed and data efficiency. In all test cases, the decision network used for classification had

two hidden layers with a consistent number of units across all conditions. However, since our BAS strategy selects a few locations to observe on the full image, the total number of parameters trained with the supervised loss was approximately 50% less for the model trained with BAS-collected data than for the model trained with full images. Figure 5D shows that, in addition to having lower parameter complexity, our method learns much faster than all other baselines and achieves higher asymptotic performance than VIME, Plan2Explore, and Full Images + FF.

We also considered a comparison of the same methods described above in terms of their data efficiency. Specifically, we ask the following question: during the first supervised training episode, how many training examples does a model need to observe to reach a given performance on the test set? This question addresses issues of few-shot learning and fast generalization. Since our BAS strategy utilizes the perception model's abstract representations of the task, we hypothesized that it would lead to a higher test performance with fewer training examples. This is exactly what we find through our analysis, as shown in Figure 5E. From these results, we see that our model is able to learn significantly faster from fewer training examples, highlighting the generalizability and effectiveness of the model's abstract representations in guiding action selection. A description of the experiments and hyperparameters used in these analyses is included in section D of the supplemental experimental procedures.

DISCUSSION

We developed a biologically inspired model of active sensing by combining two theories from neuroscience: predictive coding for perception and uncertainty minimization for action. Although these two theories have been utilized previously, our model incorporates them in a unique, scalable, and end-to-end framework, enabling flexible intrinsically driven exploration for embodied AI. Furthermore, the proposed model provides an approximate method for learning policies that optimize information gain in a differentiable manner, utilizing a deep generative model. We test this model in two sensorimotor tasks that integrate perception and action: (1) learning transition dynamics through pure exploration in discrete environments, and (2) learning unsupervised visual representations in a continuous setting.

An important aspect of our approach is its generality; it can be applied to any perception-action setting, while only requiring the specification of a generative model that relates internal representations to sensory observations. By parameterizing this generative model with neural networks, we are able to compute the uncertainty with respect to perceptual states in a differentiable manner, allowing the action and perception models to interact in an end-to-end fashion. We have shown how to instantiate this framework in the discrete setting of CMCs as well as the continuous setting of active vision. We note that the perception model in each case is a variational autoencoder (VAE), despite the different nature of each task and, thus, the different generative model. In both cases, the evidence lower bound (ELBO) objective is used to learn a probabilistic relationship between latent states and observed variables. However, the action selection model is different due to the discrete nature of the CMC problem versus its continuous counterpart in the active vision

task. In CMCs, actions are evaluated directly, while in active vision, a neural network is trained to output continuous actions. Nonetheless, in each case, action selection still aims to maximize uncertainty reduction, complying with our general active exploration framework.

Our approach emphasizes the relationship between perception, action, and learning during exploratory behavior. The importance of this relationship shows up in the CMC and active vision settings. Specifically, we see that, in both cases, the quality of the representations learned by the perception model directly depends on the exploration strategy used to collect information from the environment. For example, in CMCs, we see that our BAS exploration strategy leads to faster and more efficient learning of the underlying transition distributions compared to Boltzmann and random exploration (Figure 3A). This is despite BAS and Boltzmann performing somewhat similarly in terms of state-action space coverage, which shows that the uncertainty reduction objective leads to collecting observations that are particularly useful for the perception model to learn accurate representations of its environment. In active vision, we see a similar effect; when the perception model is trained on data collected with BAS (as opposed to random) exploration, the representations are well clustered in the latent space and contain more information about image categories (Figures 5B and 5C). These perception-action dependencies are captured more efficiently in our approach as a result of using a generative model with respect to which uncertainty can be measured. In contrast, RL exploration methods (e.g., Houthooft et al.,[23] Shyam et al.,[32] Stadie et al.,[33] Pathak et al.[34]) either develop approximate methods for information-based action selection or learn only the dynamics of a given Markov decision process (MDP) instance solely to aid in exploration. As a result, these methods are mainly concerned with efficient policies for action (exploration) without much emphasis on perception, i.e., learning useful representations of the environment through exploration.

Our final objective for action selection in CMCs consists of two terms: uncertainty reduction and expected future uncertainty (see experimental procedures). The uncertainty reduction term is a one-step approximation to the long-term expected information gain (EIG) objective used in the context of optimal experimental design.[35,36] From this perspective, eye saccades are viewed as experiments performed by an agent to gain information about the world.[10] In our approach, we add an expected future uncertainty term that acts as a heuristic that drives the model toward more uncertain states, thus providing a better approximation of the long-term information gain without requiring multi-step planning over long horizons. To test the effect of these two terms on the learning and exploration in CMCs, we performed an ablation analysis, which showed that using either term alone is less effective than using both terms together (Figure 3B). These results show that our combined objective works better for exploration than the one-step EIG. Another important benefit of our approach is the integration of action selection with a differentiable perception model, which allows for end-to-end learning as opposed to the traditional multi-step planning and RL approaches used for optimizing the EIG.

While the perception component of our method is based on the theory of predictive coding, its implementation relies on amortized variational inference. This is different from the

standard implementation of predictive coding in the neuroscience literature,[15,37,38] which relies on maximum a posteriori estimation using gradient-based optimization of the log-likelihood objective. Despite the apparent differences, methods such as iterative amortized inference[39] and hybrid predictive coding[40] provide a connection between the two implementations. These methods combine an inference model with iterative updates on the approximate posterior to compute better hidden state estimates. We include an in-depth exposition of the relationship between predictive coding and VAEs in Note S1. For a more comprehensive review, see Marino[41] and Salvatori et al.[38]

Limitations of the study

In discrete-action settings such as CMCs, a potential limitation of our approach is that action scoring, which relies on enumeration, may not scale well to larger environments with bigger state-action spaces. Although directly evaluating actions is more accurate, we can still select actions with sufficient accuracy while improving scalability by using a neural network trained to minimize uncertainty, similar to the active vision model. Additionally, one potential concern (especially for the active vision setting) relates to how well this approach can be applied to more complex real-world datasets. In this work, we opted to keep our models as simple as possible (using only feedforward networks) to illustrate the advantages of our approach. It is, however, possible to implement this framework with more complex architectures, utilizing more advanced types of networks such as convolutional neural networks, which may allow for applications in more complicated real-world problems. Such applications will be interesting to investigate as a future extension of this work.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Materials availability
No new unique reagents were generated by the study.

Data and code availability
The code and data to reproduce the results of this study are available as a GitHub repository: https://github.com/AbdoSharaf98/active-sensing-paper. The code has also been deposited to Zenodo: https://doi.org/10.5281/zenodo.10837552.[42]

General approach for active exploration
Our approach consists of two components: a perception model, which is based on predictive coding,[15] and an action model, which selects actions that reduce the perception model's uncertainty about inferred states. For perception, we rely on a generative model of the world that can be learned through experience. Perceiving an observation corresponds to inverting this model to infer the hidden states of the world that gave rise to that observation. Therefore, the first step when applying our approach to a given problem is to specify a reasonable generative model; in biological systems, this corresponds to niche-specific priors. The second step is to specify a method for learning and inference in this model. To this end, we use VAEs[43] to perform amortized variational inference on a generative model parameterized by neural networks. In essence, given a simple generative model in which a latent variable s and a set of actions $a_{\leq t}$ give rise to a set of observed variables $x_{\leq t}$ at times up to time t, the goal of variational inference[44] is to find an approximate posterior $q(s|x_{\leq t}, a_{\leq t})$ which maximizes the objective

$$\mathbb{E}_{s \sim q(\cdot|x_{\leq t}, a_{\leq t})}[\log p(x_{\leq t}|s, a_{\leq t})] - D_{KL}(q(s|x_{\leq t}, a_{\leq t})\|p(s)), \quad \text{(Equation 3)}$$

where D_{KL} is the KL divergence. This objective is the ELBO. Amortized learning can be done by using neural networks to parameterize the distributions in Equation 3 and optimizing the objective with gradient descent. Inference in this case corresponds to a simple forward pass through a neural network.

The second component of our approach is the action model, which relies on uncertainty reduction measured using Shannon entropy. As such, an application of our approach requires a method of computing the entropy of the posterior distribution $q(s|x_{\leq t}, a_{\leq t})$ inferred by the perception model. Actions are then selected to maximize the reduction in uncertainty as represented by the following score function:

$$\text{Score}(a_{t+1}) = H(q(s|x_{\leq t}, a_{\leq t})) - \mathbb{E}_{x_{t+1} \sim p(\cdot|a_{t+1}, x_{\leq t})}[H(q(s|x_{\leq t+1}, a_{\leq t+1}))], \quad \text{(Equation 4)}$$

where H denotes the Shannon entropy. The first term in Equation 4 represents the agent's current uncertainty about s. The second term represents the expected uncertainty if action a_{t+1} is executed. The expectation in the second term is taken with respect to fictitious future states drawn from the agent's current transition distribution $p(x_{t+1}|x_{\leq t}, a_{t+1})$.

The score function in Equation 4 is equivalent to the one-step EIG,[35] which has been used in prior works on modeling eye movements.[9,10,18] The key contribution of this work is in the integration of this objective, formulated in terms of the Shannon entropy, with a deep generative model that allows us to learn an estimate of the score function in an amortized and differentiable manner. The benefit of this approach is 2-fold. First, while this score function can be evaluated directly in discrete-action spaces, it becomes intractable in continuous state and action spaces. By formulating this function in terms of entropy and integrating it with a deep generative model, we can compute an MC approximation to the expectation in Equation 4 in a differentiable manner without having to lose information by discretizing the action space. Second, this approach allows us to amortize action selection by training a neural network end-to-end to output actions that minimize the uncertainty of the perception model, which leads to faster learning and higher sample efficiency compared to traditional RL approaches for optimizing information gain.[23,32–34] We refer to our action selection strategy as BAS.

Exploration in controllable Markov chains
As a proof of concept, we first demonstrate our model in the setting of discrete state and action spaces. Specifically, we develop an instance of the general framework described above for CMCs.[20,45] A CMC is essentially an MDP but without the specification of a reward function. It is formally defined as a 3-tuple (S, A, P), where: S is a set of finite states, for example the set of possible locations in a maze; A is a finite set of allowable actions, e.g., movement directions; and P is a three-dimensional kernel of transition probabilities $p : S \times A \to D(S)$, where $D(S)$ is the set of probability distributions on S. That is, P is a $|S| \times |A| \times |S|$ matrix containing the probabilities $P_{s,a,s'} = p(s'|s,a)$, where s is the current state, a is the action taken, and s' is the resulting next state.

The goal of an agent in this setting is to efficiently explore the environment and learn an estimate, $\widehat{P}$, of the underlying transition probability matrix P. This setting models the embodiment of the agent because, at any given time, the agent's interaction with the world is restricted by its current state.

Perception
We begin by specifying a generative model for this task. This generative model constitutes the perception component and is learned from observations collected by the agent. Whenever the agent visits a state s and takes action a, the observation consists of the resulting state s'. The agent's goal is to infer the distribution $\widehat{p}(: |s,a)$ for each state-action combination that best explains all observations collected when that combination was visited. Let $z_{s,a} = \widehat{p}(: |s,a)$ denote the model's estimated distribution of the next state after executing action a in state s. Then, the generative model contains $|S| \times |A|$ latent variables, $\{z_{s,a}\}_{s \in S, a \in A}$, each corresponding to a state-action combination. If we let $T(s,a)$ be the set of times at which state s was visited and action a was taken, then each latent variable $z_{s,a}$ gives rise to $K = |T(s,a)|$ observations (or next states), denoted by $\{s'_{t_k}\}_{t_k \in T(s,a)}$. This is illustrated in Figure 1D.

Let $H = \{(s_t, a_t, s'_t)\}_{t=1}^{T}$ denote the full history of experiences collected by the agent, and $H_{s,a} = \{(s, a, s'_t)\}_{t \in T(s,a)}$ be the subset of H containing

experiences in state s when action a was taken. Because the latent variables are independent, the variational posterior in this model can be expressed as

$$q(\{z_{s,a}\}|\mathcal{H}) = \prod_{s \in S, a \in \mathcal{A}} q(z_{s,a}|\mathcal{H}_{s,a}).$$ (Equation 5)

To simplify notation, we will drop the subscript (s,a), but it should be clear that, in what follows, the latent variables z and the histories $\mathcal{H}$ depend on the specific state-action combination. For a single latent variable z, the corresponding ELBO can be expressed as

$$\log p(\mathcal{H}) \geq \mathbb{E}_{z \sim q(\cdot|\mathcal{H})}[\log p(z, \mathcal{H}) - \log q(z|\mathcal{H})],$$ (Equation 6)

$$= \mathbb{E}_{z \sim q(\cdot|\mathcal{H})}[\log p(\mathcal{H}|z)] - D_{KL}(q(z|\mathcal{H})\|p(z)).$$ (Equation 7)

The first term in Equation 7 corresponds to the likelihood of the observations given the inferred transition distribution z, while the second term controls the deviation of that distribution from the model's own prior $p(z)$. We assume that the prior for all latent variables is a Dirichlet distribution with concentration parameters $\alpha = 1$ for all states in S. The posterior $q(z|\mathcal{H})$ is also assumed to be a Dirichlet distribution with a concentration parameter α that is the output of a simple feedforward neural network with parameters φ.

To compute the quantities in Equation 7, we need a representation of s, a, and $\mathcal{H}$ that facilitates updating the agent's history with new observations. To achieve this, we use one-hot vectors to represent the states and actions. We represent the complete history of the agent as an $|S| \times |\mathcal{A}| \times |S|$ matrix whose entries represent number of visits to state s' from state s when action a was taken. That is, we represent the history $\mathcal{H}_{s,a}$ as the vector $\boldsymbol{h}_{s,a} \in \mathbb{N}_0^{|S|}$ with entries

$$\boldsymbol{h}_{s,a}(i) = \sum_{t \in \mathcal{T}(s,a)} \delta(s_t', S(i)) \quad \text{for } i = 1, 2, \ldots, |S|.$$ (Equation 8)

This representation makes updating the history given new observations a simple summation operation. That is, when the agent takes action a in state s and receives a new observation s', the history can simply be updated as $\boldsymbol{h}_{s,a} \leftarrow \boldsymbol{h}_{s,a} + s'$. Now, the log likelihood in Equation 7 can be computed as

$$\log p(\mathcal{H}|z) = \log \prod_{i=1}^{|S|} z(i)^{\boldsymbol{h}_{s,a}(i)} = \sum_{i=1}^{|S|} \boldsymbol{h}_{s,a}(i)\log z(i).$$ (Equation 9)

Amortized Inference in this model is performed by the perception network φ, which receives as input the current state s, action a, and history vector $\boldsymbol{h}_{s,a}$ and outputs a concentration parameter α that parameterizes the Dirichlet posterior distribution $q_\varphi(z|\mathcal{H})$. The inferred distribution z can be obtained by drawing a reparameterized sample from $q_\varphi(z|\mathcal{H})$.[43] Note that the full ELBO for this model should be the sum of the terms in Equation 7 over all state-action combinations. However, since a single transition only changes the history for the current state s and action a, the inference network will be affected only by the corresponding term in the full ELBO. Therefore, optimizing the entire ELBO at every step is equivalent to taking a gradient step with respect to the single ELBO in Equation 7 for the relevant s and a. The derivation of the ELBO objective in Equation 7 and the full training algorithm for this model are included in section A of the supplemental experimental procedures.

Action

Our framework relies on uncertainty minimization for action selection. In this setting, our model selects actions that lead to the greatest reduction in its uncertainty about the inferred transition distributions $z_{s,a}$. This can be done by planning over long horizons[46] or by using value iteration techniques.[20] However, to keep our model simple and more computationally efficient, we employ a greedy approach with a simple heuristic that guides the model toward states with greater uncertainty. Despite this, our model is still able to achieve faster and more efficient exploration compared to random or visitation-count-based action selection. At every time step, the agent evaluates each action based on the following uncertainty reduction score

$$\begin{aligned} \text{Uncertainty Reduction}(a) = {} & H(q_\varphi(z_{s,a}|\boldsymbol{h}_{s,a})) \\ & - \mathbb{E}_{s' \sim z_{s,a}|\boldsymbol{h}_{s,a}}[H(q_\varphi(z_{s,a}|\boldsymbol{h}_{s,a} + s'))], \end{aligned}$$ (Equation 10)

which is an adaptation of Equation 4 to this specific application. Overall, Equation 10 represents the expected reduction in uncertainty for action a over a single step and a single transition distribution. To guide the agent toward future uncertain states, we add the following heuristic that describes the expected future uncertainty about the transition distributions in the next state:

$$\text{Expected Future Uncertainty}(a) = \mathbb{E}_{s' \sim z_{s,a}|\boldsymbol{h}_{s,a}}\left[\sum_{a' \in \mathcal{A}} H(q_\varphi(z_{s',a'}|\boldsymbol{h}(s', a')))\right].$$ (Equation 11)

The final score maximized by the agent is the sum of the uncertainty reduction in Equation 10 and expected future uncertainty in Equation 11.

Active vision model

Perception

The active vision perception model is based on a simple hierarchical two-level generative model that reflects the structure of the task and can be interpreted as a dynamical VAE.[47] This model is shown in Figure 1C. Let l denote the image presented on a given trial. The higher level of the perception model encodes a single abstract representation s, which may reflect high-level properties of the class to which l belongs. For example, s can represent the number of lines and circles and the spatial correlations specific to drawing a certain digit. The lower level contains individual units whose activations are entirely driven by sensory input. From a biological perspective, these units correspond to neurons in the primary visual cortex whose receptive fields overlap with the spatial span of the fovea. At each time step t and for a given sensory input x_t, we denote the activities of these neurons by z_t. From a generative perspective, z_t can contain information about the lower-level properties of image l at a given location l_{t-1}. These properties, for example, can include stroke width, style, and so forth.

Using the chain rule and the fact that the $\{x_{1:T}\}$ and s are conditionally independent given $\{z_{1:T}\}$, the full variational posterior can then be factorized as a product of two variational posteriors,

$$q(z_{1:T}, s|x_{1:T}, l_{0:T-1}) = q_1(z_{1:T}|x_{1:T}, l_{0:T-1}) \times q_2(s|z_{1:T}, l_{0:T-1}).$$ (Equation 12)

For notational simplicity, we will omit the conditioning in the variational posteriors, e.g., use $q(z_{1:T}, s)$ to refer to $q(z_{1:T}, s|x_{1:T}, l_{0:T-1})$. The corresponding ELBO, derived in section B of the supplemental experimental procedures, is found to be

$$\mathcal{L}_{ELBO} = \sum_{t=1}^{T} \mathbb{E}_q[\log p(x_t|z_t)] - \mathbb{E}_q\left[\frac{p(s)}{q_2(s|z_{1:T}, l_{0:T-1})}\right] - \sum_{t=1}^{T} \mathbb{E}_q\left[\frac{p(z_t|s, l_{t-1})}{q_1(z_t|x_t, l_{t-1})}\right].$$ (Equation 13)

Note that the loss function does not depend on the entire image except at the locations sampled and viewed by the model. This maintains consistency with the natural setting, where the agent's perception does not encompass the entire image and so it cannot (as a whole) be used for training. Throughout our experiments, we assume the prior over s to be a standard Gaussian. We also assume the likelihood distributions, $p(x_t|z_t)$ and $p(z_t|s, l_{t-1})$, as well as the variational posteriors, q_1 and q_2, to be Gaussian with means and variances parameterized by feedforward neural networks. The full architecture of the model with these networks is shown in Figure 1C and is described below.

Network architecture. The perception architecture consists of two VAEs, one for each posterior in Equation 12. The encoders and decoders for both VAEs are simple feedforward networks. The following is a description of each model component.

Lower-level VAE. At each time step t, the model receives an observation x_t which, together with the corresponding location, l_{t-1}, is passed through an encoder network to infer the posterior over sensory representations $q_1(z_t)$. Let $f_{enc,\mu}^{(1)}(x_t, l_{t-1})$ and $f_{enc,\sigma}^{(1)}(x_t, l_{t-1})$ denote the outputs of the lower-level encoder network at time t. In our experiments, we assume $q_1(z_t)$ is an isotropic Gaussian $\mathcal{N}(z_t|\mu_t^z, \sigma_t^z I)$, where

$$\mu_t^z = f_{enc,\mu}^{(1)}(x_t, l_{t-1}),$$ (Equation 14)

$$\sigma_t^z = \exp\left(\frac{1}{2}f_{enc,\sigma}^{(1)}(x_t, l_{t-1})\right).$$ (Equation 15)

Algorithm 1. Bayesian action selection in active vision

Input: observations $x_{1:t}$, locations $l_{0:t-1}$, perception model F, action network ψ_t, number of MC samples K

$q_t(s) = F.Encode(x_{1:t}, l_{0:t-1})$

$l_t = \psi_t(\mathbb{E}[q_t(s)])$

$p(x_{t+1}) = F.Decode(q_t(s), l_t)$

Draw K samples from $p(x_{t+1})$

for $k = 1$ **to** K **do**

$\quad q_{t+1}^{(k)}(s) = F.Encode(x_{1:t}, x_{t+1}^{(k)}, l_{0:t-1})$

end for

$\tilde{V}(l_t) = H(q_t(s)) - \frac{1}{K}\sum_{k=1}^{K} H(q_{t+1}^{(k)}(s))$

Update action network parameters using gradient descent on $\tilde{V}$: $\psi_{t+1} = \psi_t + \mu \nabla_{\psi_t} \tilde{V}(l_t)$

return: selected action l_t and updated action network ψ_{t+1}

The lower-level VAE decoder network takes sensory representations z_t and outputs the likelihood distribution $p(x_t|z_t)$. We assume this distribution is Gaussian $\mathcal{N}(x_t|\hat{x}_t, I)$, where $\hat{x}_t$ is the output of the decoder.

Higher-level VAE. At the end of a fixation sequence, the higher-level encoder network receives the sum of the past sensory representations and uses it to infer the posterior over abstract representations $q_2(s)$. Similar to $q_1(z_t)$, we assume $q_2(s)$ is an isotropic Gaussian $\mathcal{N}(s|\mu^s, \sigma^s I)$ parameterized by the output of the higher-level encoder $f_{enc}^{(2)}$. The decoder network at this level receives an abstract representation s and a query location l_{t-1}, and predicts a distribution over the corresponding lower-level representations $p(z_t|s, l_{t-1})$.

Generative mechanism. We can generate new data from the model as follows. First, we sample an abstract representation s from a standard Gaussian distribution. We then pick a query location l' from the interval $[-1,1] \times [-1,1]$. Next, we pass s and l' through the higher-level decoder $f_{dec}^{(2)}$, which outputs a distribution $p(z')$. We sample z' from this distribution and pass it through the lower-level decoder $f_{dec}^{(1)}$, which outputs an observation x' that is the same size as the model's retina.

Action

Adapting the uncertainty reduction objective for action selection in active vision gives the following value function for a given action (or fixation location) l_t:

$$V(l_t|x_{1:t}, l_{0:t-1}) := H(s|x_{1:t}, l_{0:t-1}) - \mathbb{E}_{p(x_{t+1}|s, l_t)}[H(s|x_{1:t+1}, l_{0:t})], \quad \text{(Equation 16)}$$

where $H(\cdot)$ denotes the Shannon entropy. Intuitively, $V(l_t)$ quantifies how much information the agent expects to gain as a result of observing the input image at location l_t. Therefore, a good estimate of information gain depends on how accurate the agent's generative model is. The objective in Equation 16 is intractable because it requires evaluating the posterior over s for all possible observations x_{t+1} in a continuous space. So, instead, we compute an approximation of it using an MC sampling approach. First, we independently draw K samples from the agent's likelihood model $p(x_{t+1}|s, l_t)$. The approximate expected entropy can then be computed as the average over the K samples, yielding the approximate value

$$\tilde{V}(l_t|x_{1:t}, l_{0:t-1}) \approx H(s|x_{1:t}, l_{0:t-1}) - \frac{1}{K}\sum_{k=1}^{K} H\left(s^{(k)}\Big|x_{1:t}, x_{t+1}^{(k)}, l_{0:t}\right), \quad \text{(Equation 17)}$$

where $x_{t+1}^{(k)}$ denotes the kth sample and $s^{(k)}$ denotes the corresponding updated posterior estimate. To make our model end-to-end differentiable, we use a neural network to select fixation locations that maximize the value in Equation 17. This action network can be trained with gradient descent because all the terms in Equation 17 are computed from the neural networks in the perception model, which receive fixation locations as part of their input. Therefore, it is possible to compute the gradients of Equation 17 with respect to the output of the action network. For all experiments, the action network we use is a simple two-layer feedforward network which receives as input the current estimate of the state s and outputs the mean of a Gaussian distribution over fixation locations. The standard deviation of this distribution is a fixed hyperparameter that we specify. The agent chooses a fixation location by sampling from the output distribution of the action network. Algorithm 1 describes our differ-

entiable approach to selecting continuous actions with uncertainty reduction in active vision tasks.

AUTHOR CONTRIBUTIONS

Conceptualization, A.S., N.I., and H.C.; methodology, A.S., N.I., and H.C.; software, A.S.; formal analysis, A.S.; investigation, A.S.; resources, N.I. and H.C.; writing – original draft, A.S.; writing – review & editing, A.S., N.I., and H.C.; visualization, A.S.; supervision, N.I. and H.C.; project administration, N.I. and H.C.; funding acquisition, N.I. and H.C.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Guillery, R.W., and Sherman, S.M. (2011). Branched thalamic afferents: What are the messages that they relay to the cortex? Brain Res. Rev. *66*, 205–219.

2. Guillery, R.W. (2005). Anatomical pathways that link perception and action. Progress in Brain Research *149*, 235–256.

3. Linson, A., Clark, A., Ramamoorthy, S., and Friston, K. (2018). The active inference approach to ecological perception: General information dynamics for natural and artificial embodied cognition. Front. Robot. AI *5*, 21.

4. Friston, K. (2010). The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. *11*, 127–138.

5. Yarbus, A.L. (1967). Eye Movements and Vision (Plenum Press).

6. Hayhoe, M., and Ballard, D. (2005). Eye movements in natural behavior. Trends Cognit. Sci. *9*, 188–194.

7. Krajbich, I., Armel, C., and Rangel, A. (2010). Visual fixations and the computation and comparison of value in simple choice. Nat. Neurosci. *13*, 1292–1298.

8. Land, M.F., and Tatler, B.W. (2009). Looking and Acting: Vision and Eye Movements in Natural Behaviour (Oxford University Press).

9. Yang, S.C.-H., Lengyel, M., and Wolpert, D.M. (2016). Active sensing in the categorization of visual patterns. Elife 5, e12215.

10. Friston, K., Adams, R.A., Perrinet, L., and Breakspear, M. (2012). Perceptions as hypotheses: saccades as experiments. Front. Psychol. 3, 151.

11. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Commun. ACM. 60, 84–90. https://doi.org/10.1145/3065386.

12. He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. Preprint at Arxiv. https://doi.org/10.48550/arXiv.1512.03385.

13. Sutton, R.S., and Barto, A.G. (2018). Reinforcement Learning: An Introduction, Second Edition (The MIT Press).

14. Zador, A., Escola, S., Richards, B., Ölveczky, B., Bengio, Y., Boahen, K., Botvinick, M., Chklovskii, D., Churchland, A., Clopath, C., et al. (2023). Catalyzing next-generation artificial intelligence through neuroAI. Nat. Commun. 14, 1597.

15. Rao, R.P., and Ballard, D.H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. Nat. Neurosci. 2, 79–87.

16. Rao, R.P.N., Olshausen, B.A., and Lewicki, M.S. (2002). Probabilistic Models of the Brain: Perception and Neural Function (MIT Press).

17. Olshausen, B.A. (2014). Perception as an inference problem. In The Cognitive Neurosciences, V.M. Gazzaniga and R. Mangun, eds. (MIT Press), pp. 295–304.

18. Butko, N.J., and Movellan, J.R. (2010). Infomax control of eye movements. IEEE Transactions on Autonomous Mental Development 2, 91–107.

19. Butko, N.J., and Movellan, J.R. (2008). I-Pomdp: An Infomax Model of Eye Movement. In 2008 7th IEEE International Conference on Development and Learning, pp. 139–144.

20. Little, D.Y., and Sommer, F.T. (2013). Learning and exploration in action-perception loops. Front. Neural Circ. 7, 37.

21. Cassey, T.C., Evens, D.R., Bogacz, R., Marshall, J.A.R., and Ludwig, C.J.H. (2013). Adaptive sampling of information in perceptual decision-making. PLoS One 8, e78993.

22. Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). Recurrent models of visual attention. In Proceedings of the 27th International Conference on Neural Information Processing Systems, 2, pp. 2204–2212. https://dl.acm.org/doi/10.5555/2969033.2969073.

23. Houthooft, R., Chen, X., Duan, Y., Schulman, J., Turck, F.D., and Abbeel, P. (2016). Curiosity-driven exploration in deep reinforcement learning via Bayesian neural networks. Preprint at Arxiv. https://doi.org/10.48550/arXiv.1605.09674.

24. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. (2020). Planning to explore via self-supervised world models. In International conference on machine learnin, pp. 8583–8592.

25. Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. 29, 141–142.

26. Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. Preprint at Arxiv. https://doi.org/10.48550/arXiv.1708.07747.

27. Krizhevsky, A. (2009). Learning multiple layers of features from tiny images (University of Toronto). Technical Report. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

28. Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.

29. Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. Phys. Rev. E 69, 066138.

30. Ross, B.C. (2014). Mutual information between discrete and continuous data sets. PLoS One 9, e87357.

31. Guo, Z., Thakoor, S., Pîslar, M., Avila Pires, B., Altché, F., Tallec, C., Saade, A., Calandriello, D., Grill, J.-B., Tang, Y., et al. (2022). BYOL-explore: Exploration by bootstrapped prediction. Adv. Neural Inf. Process. Syst. 35, 31855–31870.

32. Shyam, P., Jaśkowski, W., and Gomez, F. (2019). Model-based active exploration. In Proceedings of the 36th International conference on machine learning, 97, pp. 5779–5788.

33. Stadie, B.C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. Preprint at Arxiv. https://doi.org/10.48550/arXiv.1507.00814.

34. Pathak, D., Agrawal, P., Efros, A.A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In Proceedings of the 34th International conference on machine learning, 97, pp. 2778–2787.

35. Emery, A.F., and Nenarokomov, A.V. (1998). Optimal experiment design. Meas. Sci. Technol. 9, 864–876.

36. Tsilifis, P., Ghanem, R.G., and Hajali, P. (2017). Efficient Bayesian experimentation using an expected information gain lower bound. SIAM/ASA J. Uncertain. Quantification 5, 30–62.

37. Millidge, B., Seth, A., and Buckley, C.L. (2021). Predictive coding: a theoretical and experimental review. Preprint at arxiv. https://doi.org/10.48550/arXiv.2107.12979.

38. Salvatori, T., Mali, A., Buckley, C.L., Lukasiewicz, T., Rao, R.P., Friston, K., and Ororbia, A. (2023). Brain-inspired computational intelligence via predictive coding. Preprint at Arxiv. https://doi.org/10.48550/arXiv.2308.07870.

39. Marino, J., Yue, Y., and Mandt, S. (2018). Iterative amortized inference. In Proceedings of the 35th International Conference on Machine Learning, 80, pp. 3403–3412.

40. Tscshantz, A., Millidge, B., Seth, A.K., and Buckley, C.L. (2023). Hybrid predictive coding: Inferring, fast and slow. PLoS Comput. Biol. 19, e1011280.

41. Marino, J. (2022). Predictive Coding, Variational Autoencoders, and Biological Connections. Neural Comput. 34, 1–44.

42. Sharaf, A. (2024). Abdosharaf98/active-sensing-paper: v1.0.0 (v1.0.0) (Zenodo). https://doi.org/10.5281/zenodo.10837552.

43. Kingma, D.P., and Welling, M. (2013). Auto-encoding variational Bayes. Preprint at Arxiv. https://doi.org/10.48550/arXiv.1312.6114.

44. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K. (1998). An Introduction to Variational Methods for Graphical Models. In Learning in Graphical Models (Springer), pp. 105–161. https://doi.org/10.1007/978-94-011-5014-9_5.

45. Gimbert, H. (2007). Pure stationary optimal strategies in Markov decision processes. In STACS 2007: 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007. Proceedings 24 (Springer), pp. 200–211.

46. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: a process theory. Neural Comput. 29, 1–49.

47. Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., and Alameda-Pineda, X. (2020). Dynamical variational autoencoders: A comprehensive review. Preprint at Arxiv. https://doi.org/10.1561/2200000089.

Supplemental information

Active sensing with predictive coding

and uncertainty minimization

Abdelrahman Sharafeldin, Nabil Imam, and Hannah Choi

Supplemental Notes

Note S1: Relationship between Predictive Coding and Variational Autoencoders

According to the predictive coding framework, the brain maintains a generative model of the world which approximates a mapping between observed sensory input and hidden states of the environment. This is illustrated in Figure S1. Perception, therefore, corresponds to inverting this model to infer hidden states, while learning corresponds to updating the parameters of this model based on prediction errors. Here, we outline the relationship between hierarchical predictive coding as presented in Rao and Ballard (1999)[1] and the framework of Variational Autoencoders in machine learning[2]. A similar outline of this relationship is given in Jiang and Rao (2021)[3] and Marino (2022)[4] (with more details on the connections between theory and biology).

To simplify the discussion, we assume the generative model consists of two hierarchical layers (an input layer and a sensory layer) as shown in Figure S1. In reality, the sensory areas in the brain contain many more hierarchical levels, and this discussion can be easily extended to multi-layer hierarchical generative models.

The goal of inference is to find the best estimate $s^\star$ of the true hidden state $\hat{s}$ given an observation x. In the predictive coding framework, this is done by maximizing the posterior distribution $p(s|x)$, which is known as maximum a posteriori (MAP) estimation. Equivalently, we can maximize $\log p(s|x)$ since the log is a monotonic function in p. This problem can be formulated as follows

$$s^\star = \arg\max_s \log p(s|x) \tag{1}$$

$$= \arg\max_s \log \frac{p(x|s)p(s)}{p(x)} \tag{2}$$

$$= \arg\max_s [\log p(x|s) + \log p(s)] \tag{3}$$

To perform this optimization, we adopt some assumptions about the likelihood distribution $p(x|s)$ and the prior on the hidden state $p(s)$. In their original implementation, Rao and Ballard assume the following parameterizations

$$p(x|s) = \mathcal{N}(f(\boldsymbol{W}s); \sigma_x^2 \boldsymbol{I}) \tag{4}$$

$$p(s) = \mathcal{N}(\mu_s; \sigma_s^2 \boldsymbol{I}) \tag{5}$$

Generative Model (Brain)

$p(s, x)$

$p(s|x)$

$\hat{p}(\hat{s}, x)$

Generative Process (Environment)

Figure S1: Simple two-layer hierarchical generative model. This model is parameterized by θ and approximates the true distribution of the generative process giving rise to observations x. The predictive coding framework postulates that neural activities encode hidden state estimates, e.g. s. Thus, perception corresponds to inverting the generative model and inferring those neural activities (red arrow).

where $\boldsymbol{W}$ is a weight matrix, $f(\cdot)$ is a non-linearity, and $\mathcal{N}(\mu, \sigma\boldsymbol{I})$ is an isotropic gaussuian with mean μ and covariance $\sigma\boldsymbol{I}$. Without loss of generality, we can simplify this further by assuming the prior $p(s)$ is a standard gaussian, i.e. $\mu_s = \boldsymbol{0}$ and $\sigma_s^2 = 1$. Substituting this into equation 3, we get

$$s^\star = \arg\max_s \log \mathcal{N}(f(\boldsymbol{W}s); \sigma_x^2 \boldsymbol{I}) + \log \mathcal{N}(0; \boldsymbol{I}) \tag{6}$$

$$= \arg\min_s \frac{1}{\sigma_x^2} \|x - f(\boldsymbol{W}s)\|_2^2 + \|s\|_2^2 \tag{7}$$

Equation 7 is the predictive coding objective for the simple generative model in Figure S1. The first term in the objective is a reconstruction loss (or prediction error) and the second term is a regularization term which ensures that the inferred state $s^\star$ is consistent with the prior over s. To learn the parameters $\boldsymbol{W}$, the same objective is minimized with respect to $\boldsymbol{W}$ while fixing the inferred state s.

The predictive coding formulation described above attempts to find a point estimate $s^\star$ which maximizes $p(s|x)$. An alternative is to find the full posterior distribution

$$p(s|x) = \frac{p(x|s)p(s)}{p(x)} = \frac{p(x|s)p(s)}{\int_s p(x, s)ds} \tag{8}$$

This is intractable to do exactly since it requires evaluating an integral over the continuous space of hidden states. Variational inference[5] allows us to approximate the posterior $p(s|x)$ with some *variational* posterior $q(s)$. Given a
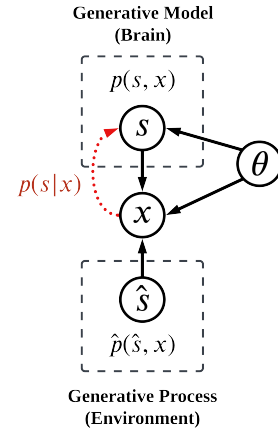
family of distributions $\mathcal{Q}$ defined over the space of hidden states s, we aim to find the distribution $q(s) \in \mathcal{Q}$ which minimizes the objective

$$D_{KL}\Big(q(s)||p(s|x)\Big) = \mathbb{E}_{s \sim q(s)}\left[\log \frac{q(s)}{p(s|x)}\right] \tag{9}$$

$$= \mathbb{E}_{s \sim q(s)}\left[\log \frac{q(s)p(x)}{p(x,s)}\right] \tag{10}$$

$$= \log p(x) - \mathbb{E}_{s \sim q(s)}\left[\log \frac{p(x,s)}{q(s)}\right] \tag{11}$$

The quantity $\mathbb{E}_{s \sim q(s)}\left[\log \frac{p(x,s)}{q(s)}\right]$ is known as the evidence lower bound (ELBO) because, due to the non-negativity of the KL divergence, it forms a lower bound on the log evidence

$$\log p(x) \geq \mathbb{E}_{s \sim q(s)}\left[\log \frac{p(x,s)}{q(s)}\right] = \mathbb{E}_{s \sim q(s)}[\log p(x|s)] - D_{KL}\Big(q(s)||p(s)\Big) \tag{12}$$

Therefore, minimizing the KL term $D_{KL}\Big(q(s)||p(s|x)\Big)$ amounts to maximizing the ELBO since $\log p(x)$ does not depend on either s or the parameters of the model. When we factor in the assumption in Equation 4, the ELBO reduces to the objective

$$-\mathbb{E}_{s \sim q(s)}\Big[\|x - f(\boldsymbol{W}s)\|_2^2\Big] - D_{KL}\Big(q(s)||p(s)\Big) \tag{13}$$

To see the correspondence between the ELBO in Equation 13 and the predictive coding objective in Equation 7, note that the first term in Equation 13 leads to the minimization of the reconstruction error, while the second term constrains the deviation of the posterior $q(s)$ from the prior $p(s)$. Finally, We can train neural networks to optimize the ELBO in equation 13 using the framework of Variational Autoencoders[2].

Note S2: Survery of related work

Intrinsic motivation Our approach can be regarded as an intrinsically-motivated exploration strategy[6]. In intrinsically-motivated exploration, an agent learns exploratory behavior in the absence of any extrinsic reward signals. Instead of extrinsic reward, exploration is guided by intrinsic value, which in our case is based on the expected uncertainty reduction associated with an action. The uncertainty is measured with respect to the agent's perception model, which is learned in a completely unsupervised manner. Other types of intrinsic signals have been used for autonomous exploration, such as prediction error[7-9], space coverage[10,11], and visitation count[12]. Intrinsic motivation strategies can help build representations that generalize to different tasks in the same environment since there is no dependence on a specific reward function. The closest family of intrinsic motivation approaches to ours are information-theoretic approaches, discussed below.

Information-theoretic exploration in reinforcement learning Information gain has been used to promote autonomous exploration in multiple approaches[13-15]. However, these approaches rely on state-action enumeration to compute information gain, which limits their applicability to settings with discrete state and action spaces. In contrast, our framework is general and can be applied to both discrete and continuous settings. In the discrete setting, our work is most related to Little and Sommer (2013)[16]; we test our model in a maze navigation task similar to the one used there. The main difference between their approach and ours is that we do not assume explicit knowledge about the true generative model of the environment. Instead, the perception component of our architecture learns a generative model through collected experiences in an end-to-end manner. In the continuous setting, our work is most similar to Houthooft et al. (2016)[17] and Mohamed et al. (2015)[18] in deep reinforcement learning (RL). Our approach is different from those two approaches in that it can be applied in model-based settings, since the perception component of our model explicitly learns the transition dynamics of the environment, enabling the generation of *imagined* trajectories that can be used for model-based planning and training. In contrast, those two approaches rely on model-free methods by modifying the reward function to include an information gain component. Finally, some approaches, such as Plan2Explore[19], calculate disagreement between ensembles of predictive models as a surrogate measure of uncertainty that is then used to guide exploration. These ensembles can be trained to predict observations, in which case exploration can be performed without external reward signals, as well as future extrinsic rewards[20].

Active vision and visual attention in machine learning We apply our model to the task of active vision. Here, our work is related to the Recurrent Attention Model (RAM)[21] and the DRAW model[22], but differs from those models in four key aspects. First, the perception and action components of our model are trained in a completely unsupervised, task-independent manner. During the classification task, only one feedforward decision network (separate from the main model) is trained with the classification loss. The learned representations can then be used for arbitrary tasks: to illustrate, we use these representations as input to the decision network to achieve high performance on a downstream image classification task. Second, despite the sequential nature of this task, our model solves it using end-to-end feedforward networks, greatly reducing the amount of computation compared to the recurrent architectures used in RAM[21] and DRAW[22]. Third, in contrast to DRAW[22], our model does not assume access to the full image in the training loss function, which is consistent with the assumption of bandlimited sensing. Finally, our model makes explicit links to ideas in neuroscience that enable the testing of functional hypotheses in a modern machine learning setting.

Our active vision approach is also somewhat related to the framework of Attend, Infer, Repeat (AIR)[23,24]. However, there are some key differences in the underlying modeling assumptions. For example, the latent variables inferred by AIR for a given image (or scene) are assumed to correspond to the attributes of entire objects that decompose the scene, whereas our model makes no restrictions on what the lower-level latent variables represent. Further, for biological plausibility, our approach imposes the restriction that only patches of the image corresponding to the model's bandlimited sensor are used during inference and training. Since AIR is mainly concerned with decomposing scenes into constituent objects, it makes no such restriction and so the underlying approach involves a different inference and training process.

Embodied AI for exploration and learning Embodied AI methods focus on learning perception models in agents that interact with their environment. This is in contrast, for example, to traditional computer vision approaches, where vision is treated as a passive process. Much work has been done in the area of embodied exploration for robotics to find policies (e.g., for movement and sensing) that help agents learn good perceptual representations of their environments[25-28]. For example, Self-supervised Embodied Active Learning (SEAL)[26]

focuses on improving object detection and instance segmentation through actively collecting data that provide the agent with the most knowledge during training, utilizing a curiosity-based objective. Other methods, such as Active Neural SLAM[25], focus on leveraging a learned SLAM module to guide exploration and navigation in unknown environments.

Active inference and the free energy principle In general, the theoretical formulation of our approach is most similar to the active inference formulation in neuroscience[29,30]. In this formulation, an agent aims to minimize the Expected Free Energy (EFE) objective, which consists of an epistemic value component that minimizes the agent's surprise and an extrinsic value component that maximizes external rewards from the environment. In this sense, our action selection mechanism can be viewed as a special case of the EFE when no extrinsic reward is provided and when action selection is performed in a greedy fashion. The main difference between our approach and standard implementations of active inference is our amortization of action selection with end-to-end optimization, which avoids expensive trajectory search approaches[31] and extends the applicability of our approach to continuous domains[32].

Predictive coding in machine learning There is a large body of work adapting the theory of predictive coding to machine learning problems, ranging from computer vision[33–35], gradient-based optimization[36,37], lifelong learning[38], and temporal learning[39]. However, these models apply the theory in the context of passive perception. Although some recent work combines predictive coding models with action[40], they do not focus on autonomous exploration. Rao and colleagues have recently introduced the framework of active predictive coding[41–44] but their generative models focus on generating transition and policy functions using hypernetworks, and their policy training approach utilizes a supervised reinforcement learning algorithm without particular focus on autonomous exploration.
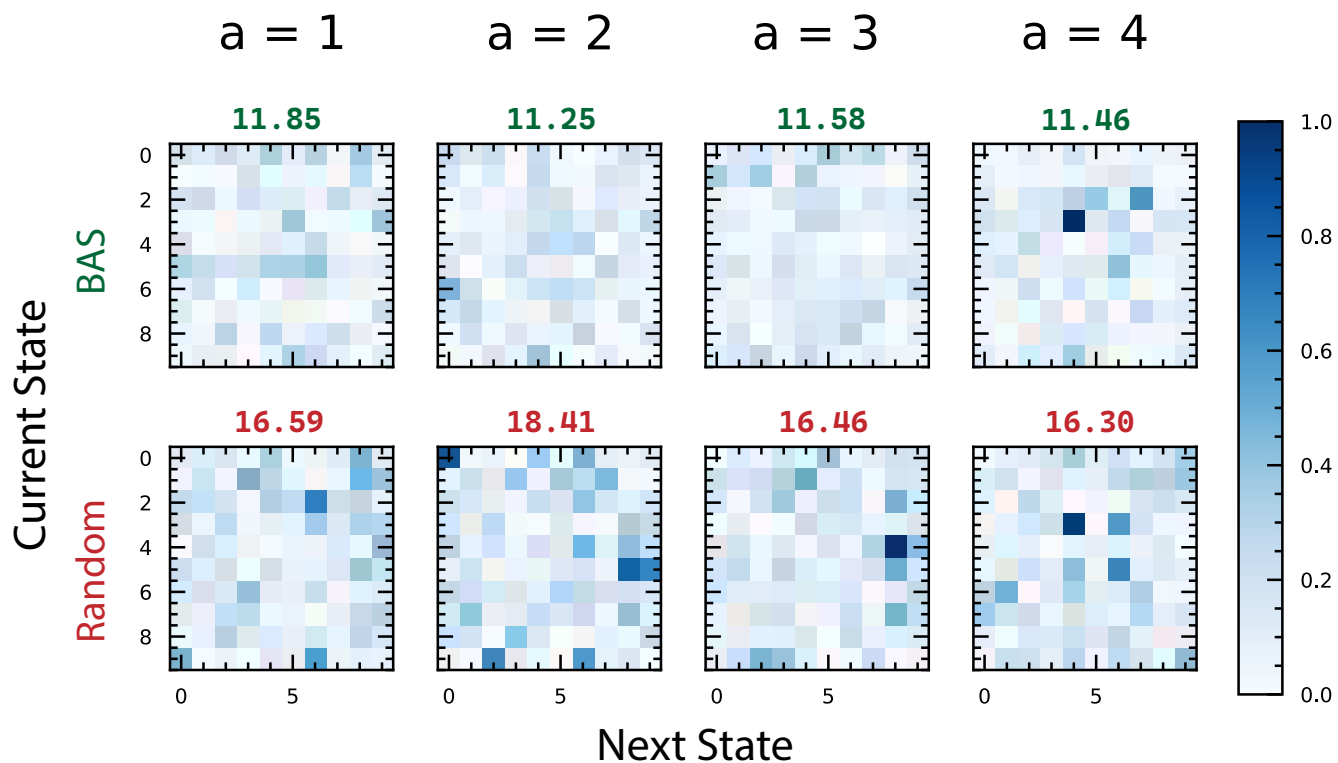
Supplemental Figures

Figure S2: Differences between learned and true transition distributions for our active explorer (BAS, top row) and a random explorer (bottom row) in the Dense Worlds environment. Each image represents the normalized absolute differences between the learned and true transition distributions for a given action (a = 1, 2, 3, 4). Numbers above each image represent the sum of all values in the corresponding matrix. In each case, agents were allowed to explore the environment for 2000 steps.
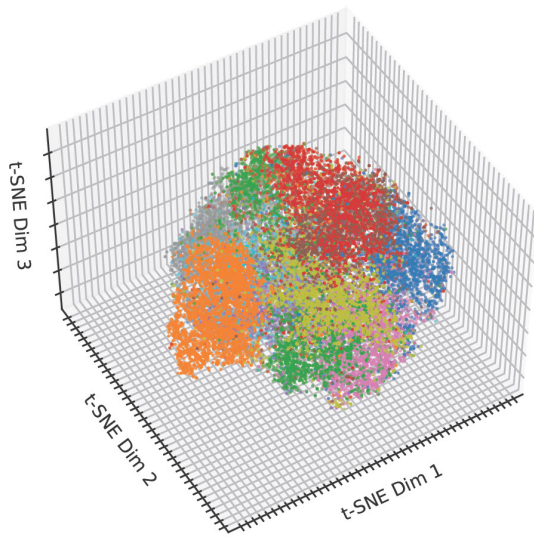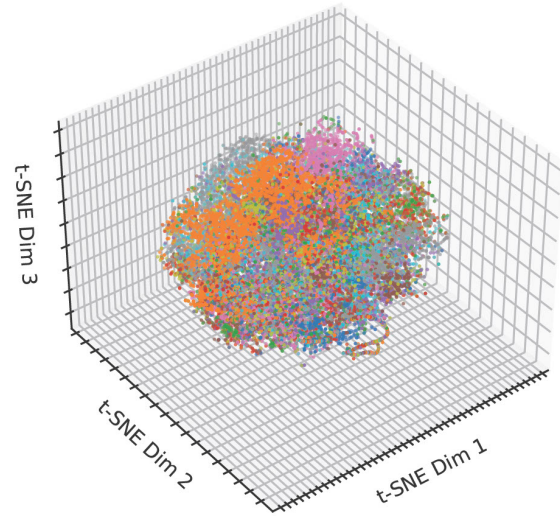
(a)   **(b)**

Figure S3: Effect of action selection strategy on perceptual latent representations visualized through t-SNE.
t-SNE projections [45] of the learned latent representations are shown for two cases: (a) the BAS-trained perception model,
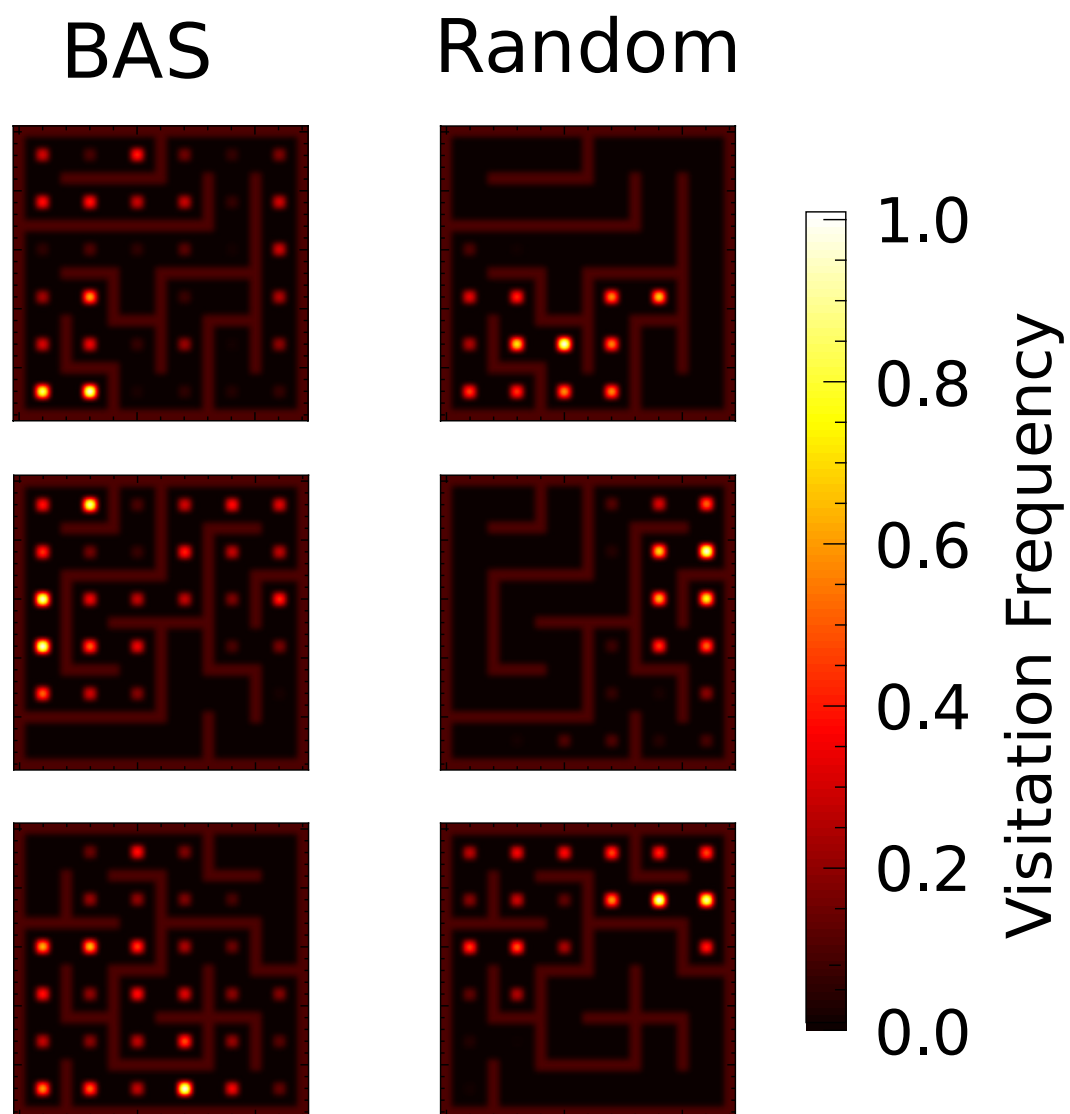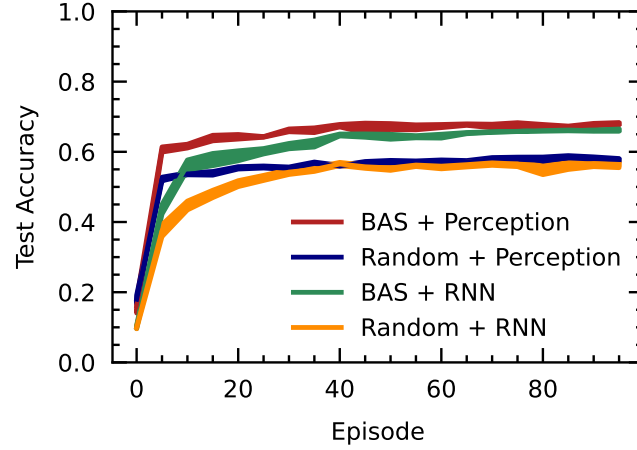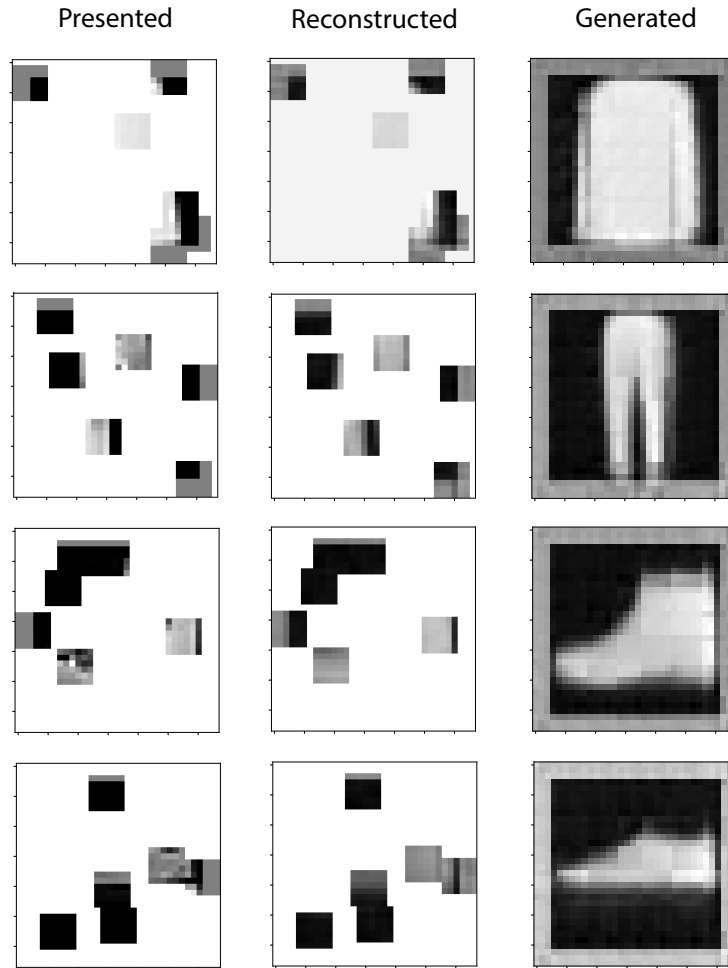and (b) the randomly-trained perception model.

Figure S4: More examples of visitation frequency heatmaps for our BAS explorer model versus a random exploration strategy in 6×6 mazes.

(a)

Presented   Reconstructed   Generated

(b)

Figure S5: Results on Fashion MNIST. (a) Classification performance on the fashion MNIST dataset during the active vision task. (b) Examples demonstrating the generative ability of the perception model. The original patches presented are shown on the left and their reconstructions are shown in the middle. Right shows images generated by first generating small patches at various locations (not seen during presentation) and combining them to form the final image. These results show the perception model is able to capture the spatial relationships associated with elements in the dataset.
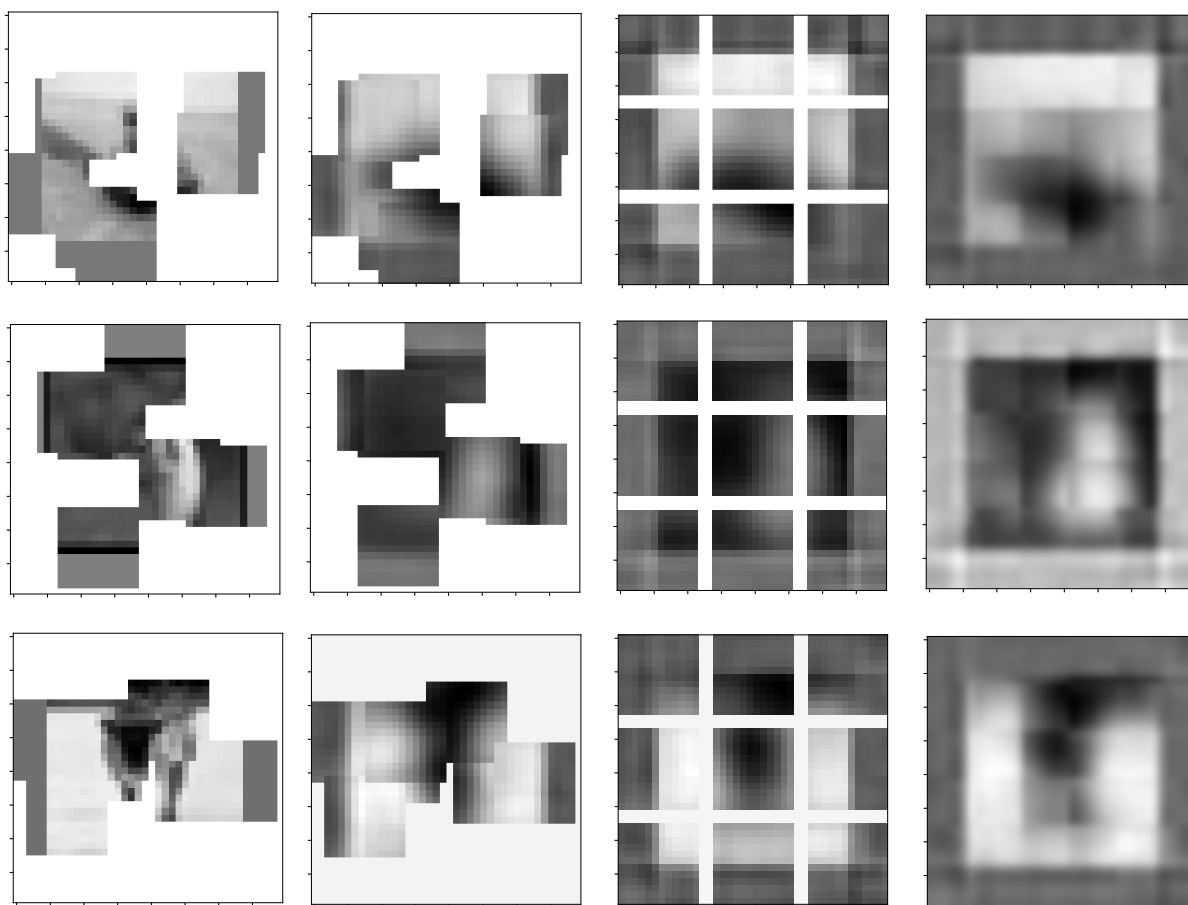
Figure S6: Results on grayscale CIFAR-10. Original presented patches are shown on the left and their reconstructions are shown on the second column. Third and fourth column show images generated by the perception model combining smaller patches at different locations. The last column has more patches at continguous locations followed by bicubic smoothing for illustration. These results show that, despite the simplicity of our model's architecture, it is still able to capture the overall structure and statistics in natural images.

Supplemental Experimental Procedures

A Training of active exploration model in controllable Markov chains

Algorithm 1 describes the Bayesian Action Selection (BAS) algorithm in controllable Markov chains (CMCs) with discrete action and state spaces. The training algorithm for our full active sensor model (including perception) in CMCs is outlined in Algorithm 2. After the model is trained for T time steps in the environment, the learned transition distribution $\hat{p}(:|s,a)$ for a given state s and action a is found as the mean of the Dirichlet posterior output by the perception network at the end of training. That is,

$$q(z_{s,a}) = \text{Dir}\Big(\phi_T(s,a,\boldsymbol{h}_{s,a})\Big) \tag{14}$$

$$\hat{p}(:|s,a) = \mathbb{E}_q[z_{s,a}] \tag{15}$$

The ELBO objective in Equation 5 of the main text can be derived as follows.

$$\log p(\mathcal{H}) = \log \int_z q(z|\mathcal{H}) \frac{p(z,\mathcal{H})}{q(z|\mathcal{H})}) \, dz \tag{16}$$

$$= \log \mathbb{E}_q \left[\frac{p(z,\mathcal{H})}{q(z|\mathcal{H})} \right] \tag{17}$$

$$\geq \mathbb{E}_q \left[\log \frac{p(\mathcal{H}|z)p(z)}{q(z|\mathcal{H})} \right] \tag{18}$$

$$= \mathbb{E}_q \left[\log p(\mathcal{H}|z) \right] - D_{KL} \left(q(z|\mathcal{H}) || p(z) \right) = \mathcal{L}_{ELBO} \tag{19}$$

where Equation 18 follows from Jensen's inequality. In practice, we can multiply the KL term in 19 by a scalar β to control the balance between the regularity of the learned distributions and how well they explain past experiences[46].

Algorithm 1 Bayesian Action Selection in Controllable Markov Chains

---

Input: current state s_t, history vectors $\boldsymbol{h}_{s,a}$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$, perception network ϕ

Initialize $bestAction = a_0$

Initialize $bestValue = -\infty$

for each action a_i in $\mathcal{A}$ **do**

  $\boldsymbol{\alpha} = \phi(s_t, a_i, \boldsymbol{h}_{s_t,a_i})$

  $q(z_{s_t,a_i}) = \text{Dir}(\boldsymbol{\alpha})$

  Current Entropy $= H\Big(q(z_{s_t,a_i})\Big)$

  Draw reparameterized sample $\tilde{z}_{s_t,a_i}$ from $q(z_{s_t,a_i}|\mathcal{H})$

  Expected Entropy $= 0$

  Expected Future Uncertainty $= 0$

  for each state s_j in $\mathcal{S}$ **do**

    $\boldsymbol{h}^+_{s_t,a_i} = \boldsymbol{h}_{s_t,a_i} + s_j$

    $q^+(z_{s_t,a_i}) = \text{Dir}(\phi(s_t,a_i,\boldsymbol{h}^+_{s_t,a_i}))$

    Expected Entropy $=$ Expected Entropy $+ \tilde{z}_{s_t,a_i}(j) \times H\Big(q^+(z_{s_t,a_i})\Big)$

    Future Uncertainty $= 0$

    for each action $a_k \in \mathcal{A}$ **do**

      $q(z_{s_j,a_k}) = \text{Dir}(\phi(s_j,a_k,\boldsymbol{h}_{s_j,a_k}))$

      Future Uncertainty $=$ Future Uncertainty $+ H\Big(q(z_{s_j,a_k})\Big)$

    end for

    Expected Future Uncertainty $=$ Expected Future Uncertainty $+ \tilde{z}_{s_t,a_i}(j) \times$ Future Uncertainty

  end for

  $value = ($Current Entropy $-$ Expected Entropy$) +$ Expected Future Uncertainty

  if $value > bestValue$ **then**

    $bestAction = a_i$

  end if

end for

Algorithm 2 Active Sensing in Controllable Markov Chains

Initialize the perception network parameters ϕ_0
Initialize the history vectors $\boldsymbol{h}_{s,a} = \boldsymbol{0}$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$
Get initial state s_0 from the environment
for each time step $t \geq 0$ **do**
    Select action a_t with BAS using Algorithm 1
    Execute action a_t and receive the updated state s_{t+1} from the environment
    $\boldsymbol{h}_{s_t,a_t} = \boldsymbol{h}_{s_t,a_t} + s_{t+1}$
    $q(z_{s_t,a_t}) = \mathrm{Dir}\Big(\phi_t(s_t, a_t, \boldsymbol{h}_{s_t,a_t})\Big)$
    Draw reparameterized sample $\tilde{z}_{s_t,a_t}$ from $q(z_{s_t,a_t})$
    $\mathcal{L}_{ELBO} = -\sum_{i=1}^{|\mathcal{S}|} \boldsymbol{h}_{s_t,a_t}(i) \log \tilde{z}_{s_t,a_t}(i) + D_{KL}\Big(q(z_{s_t,a_t})||\mathrm{Dir}(\boldsymbol{1})\Big)$
    Update perception network parameters with gradient descent: $\phi_{t+1} = \phi_t + \mu \nabla_{\phi_t} \mathcal{L}_{ELBO}$
end for

B Derivation of the ELBO for the perception model

The generative graphical model for active vision (Figure 1C) admits the following factorization of the joint likelihood

$$p(x_{1:T}, z_{1:T}, s|l_{0:T-1}) = p(x_{1:T}|z_{1:T})p(z_{1:T}|l_{0:T-1}, s)p(s) \tag{20}$$

$$= p(s) \prod_{t=1}^{T} p(x_t|z_t) \prod_{t=1}^{T} p(z_t|l_{t-1}, s) \tag{21}$$

Similarly, the joint posterior factorizes as follows

$$q(z_{1:T}, s|x_{1:T}, l_{0:T-1}) = q_1(z_{1:T}|x_{1:T}, l_{0:T-1})q_2(s|z_{1:T}, x_{1:T}, l_{0:T-1}) \tag{22}$$

$$= q_1(z_{1:T}|x_{1:T}, l_{0:T-1})q_2(s|z_{1:T}, l_{0:T-1}) \tag{23}$$

$$= q_2(s|z_{1:T}, l_{0:T-1}) \prod_{t=1}^{T} q_1(z_t|x_t, l_{t-1}), \tag{24}$$

We can, therefore, express the log likelihood and posterior probabilities as

$$\log p(x_{1:T}, z_{1:T}, s|l_{0:T-1}) = \sum_{t=1}^{T} \log p(x_t|z_t) + \sum_{t=1}^{T} \log p(z_t|l_{t-1}, s) + \log p(s) \tag{25}$$

$$\log q(z_{1:T}, s|x_{1:T}, l_{0:T-1}) = \log q_2(s|z_{1:T}, l_{0:T-1}) + \sum_{t=1}^{T} \log q_1(z_t|x_t, l_{t-1}) \tag{26}$$

Using the log joint likelihood in Equation 25 and the log posterior in Equation 26, we can obtain the ELBO on the log marginal likelihood as follows

$$\log p(x_{1:T}|l_{0:T-1}) = \log \mathbb{E}_q \left[\frac{p(x_{1:T}, z_{1:T}, s|l_{0:T-1})}{q(s, z_{1:T}|x_{1:T}, l_{0:T-1})} \right] \tag{27}$$

$$\geq \mathbb{E}_q \left[\log p(x_{1:T}, z_{1:T}, s|l_{0:T-1}) - \log q(s, z_{1:T}|x_{1:T}, l_{0:T-1}) \right] \tag{28}$$

$$= \mathbb{E}_q \left[\sum_{t=1}^{T} \log p(x_t|z_t) + \sum_{t=1}^{T} \log p(z_t|l_{t-1}, s) + \log p(s) \right.$$
$$\left. - \sum_{t=1}^{T} \log q_1(z_t|x_t, l_{t-1}) - \log q_2(s|z_{1:T}, l_{0:T-1}) \right] \tag{29}$$

$$= \sum_{t=1}^{T} \mathbb{E}_q \left[\log p(x_t|z_t) \right] + \sum_{t=1}^{T} \mathbb{E}_q \left[\log \frac{p(z_t|l_{t-1}, s)}{q_1(z_t|x_t, l_{t-1})} \right]$$
$$+ \mathbb{E}_q \left[\log \frac{p(s)}{q_2(s|z_{1:T}, l_{0:T-1})} \right], \tag{30}$$

where Equation 28 follows from Jensen's inequality.

C Experiment settings and hyperparameters for controllable Markov chains

Dense World

In the dense world experiments, we let the models explore an environment with 10 states and 4 actions for a total of 2000 timesteps. The transition distributions in the environment were independently drawn from a Dirichlet distribution with parameter $\boldsymbol{\alpha} = 1$. That is, for a given state s and action a, the transition distribution $p(: |s, a)$ is given by

$$p(: |s, a) \sim \text{Dir}(1) := \frac{\Gamma(N)}{\Gamma(1)^N} \tag{31}$$

$$\Gamma(x) := \int_0^\infty t^{x-1} e^{-t} \, dt \tag{32}$$

For both the BAS and the random agents, the perception network was a two-layer feedforward network with 16 hidden units each followed by a softplus nonlinearity. The learning rate for both agents was fixed at 0.001.

Mazes

In the maze experiments, the agents explored a maze for a total of 3000 time steps. We did experiments with different maze sizes (6×6, 8×8, and 12×12) and observed qualitatively the same results. We used the same network architecture and learning rates for all agents as those used in the Dense World experiments.

D   Experiment settings and hyperparameters for active vision

MNIST classification

We trained our model on the regular MNIST dataset, the translated MNIST dataset (described in the main text), and the fashion MNIST dataset. First, the perception model was pre-trained in a completely unsupervised manner with randomly selected fixation locations. Afterwards, we continued to train the perception model with the unsupervised loss while actions were selected using our BAS strategy. At the same time, a separate decision network was trained to take as input the inferred state s at the end of trial output a class label at the end of the trial. Gradients from classification loss were only used to update the parameters of the decision network. For all experiments, the encoder and decoder networks of both the the lower-level and the higher-level VAEs were feedforward networks with two layers, each with 256 hidden units followed by rectified linear unit (ReLU) activation functions. The action network was a two-layer feedforward network with 64 and 32 hidden units, respectively. When perception states were used for decision making, the decision network was a two-layer feedforward network with 256 hidden units each. When an RNN was used to integrate past observations for decision-making, the hidden size of the RNN decision network was chosen to be the same as the dimensionality of the abstract state s. Table S1 lists the hyperparameters used for each type of experiment. Hyperparameters were adjust ad hoc based on the resulting accuracy obtained on a separate validation set. In these experiments, we also use a regularization hyperparameter β as a scalar multiplying the KL term in the ELBO objective for the perception model[46].

Table S1: Settings for Centered, Translated, and Fashion MNIST Experiments

Hyper-parameter	Centered MNIST	Translated MNIST	Cluttered MNIST	Fashion MNIST
# pre-training episodes	0	10	20	10
# fixations (n)	3	4	12	5
Patch dim (d)	8	12	12	6
# foveated patches (N_{fov})	1	3	3	1
Foveation scale	—	2	2	—
z dim	32	64	64	64
s dim	64	128	128	128
σ_{action}	0.15	0.15	0.15	0.05
Action network lr	0.001	0.001	0.001	0.001
Perception model lr	0.001	0.001	0.001	0.001
Decision network lr	0.001	0.001	0.001	0.001
β	0.1	0.1	0.1	1.1
Batch size	64	64	128	64

Grayscale CIFAR-10

We tested our perception model on grayscale CIFAR-10 images to see if it can capture the overall structure and statistics of natural images. Table S2 lists the hyperparameters used for these experiments.

Table S2: Settings for Grayscale CIFAR-10 Experiments

Hyper-parameter	Setting
# fixations (n)	6
Patch dim (d)	12
# foveated patches (N_{fov})	1
z dim	32
s dim	64
Perception model lr	0.001
β	0.01
Batch size	64

Learning speed and data efficiency comparisons

Learning speed and data efficiency tests were performed on the translated MNIST dataset. Our approach (BAS + FF), described in the main text, was compared to four baselines: the Recurrent Attention Model (RAM)[21], VIME[17], Plan2Explore[19], and a feedforward (FF) neural network receiving full images as input (Full Images + FF). In all these cases, the decision network (the network that outputs class labels) consisted of two hidden layers each with 128 hidden units followed by ReLU activation functions.

BAS + FF, VIME, BYOL-Explore and Plan2Explore were all pre-trained unsupervised for 10 epochs with a random action selection strategy. For BAS + FF, the architectures of the perception model and the action network were the same as those described in Section D. For RAM, the hidden size of the RNN was chosen to match the dimensionality of the abstract representation s in our perception model, which was 128. In all models (except Full Images + FF), the location was drawn from a two-component Gaussian (with a pre-determined fixed variance) parameterized by the output of the location network. Hyperparameters for all models were adjusted ad hoc to optimize performance on a validation set (separate from the MNIST test set).

All hyperparameters are listed in Tables S3, S4, S5, S6, and S8. In the Full Images + FF case, the only hyperparameters adjusted were the batch size and the learning rate which were fixed at 64 and 0.001, respectively.

Table S3: Global Hyperparameters

Hyper-parameter	Value
# fixations (n)	3
Patch dim (d)	12
# foveated patches (N_{fov})	3
Foveation scale	2
Batch size	64

Table S4: Hyperparameters for BAS + FF

Hyper-parameter	Value
z dim	64
s dim	128
σ_{action}	0.15
Action network hidden layers	[64]
Action network lr	0.001
Perception model lr	0.001
Decision network lr	0.001
β	0.1

Table S5: Hyperparameters for RAM

Hyper-parameter	Value
h_g	64
h_l	64
RNN hidden size	128
σ_{action}	0.05
Action network hidden layers	[64]
Action network lr	0.001
Decision network lr	0.001
Core and glimpse networks lr	0.001

Table S6: Hyperparameters for VIME

Hyper-parameter	Value
Action network hidden layers	[256]
σ_{action}	0.05
BNN hidden size	256
BNN prior std	0.5
BNN likelihood std	5.0
BNN lr	0.0001
Action network lr	0.001
Decision network lr	0.001

Table S7: Hyperparameters for BYOL-Explore

Hyper-parameter	Value
Latent dim	256
Encoder hidden layers	[256, 256]
Action network hidden layers	[64, 32]
Critic network hidden layers	[256, 256]
World model hidden layers	[256, 256]
σ_{action}	0.05
lr	0.001

Table S8: Hyperparameters for Plan2Explore

Hyper-parameter	Value
Number of ensemble models	10
One step model layers	[256, 256]
Horizon length	15
Encoder layers	[256, 256]
Decoder layers	[256, 256]
Action network hidden layers	[256, 256]
Critic network hidden layers	[256, 256]
Reward network layers	[32, 32]
σ_{action}	0.01
Action network lr	0.001
Critic network lr	0.001
One step model lr	0.001
Decision network lr	0.001

References

1. Rao, R. P. N., and Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience *2*, 79–87.

2. Kingma, D. P., and Welling, M. Auto-encoding variational bayes (2013). doi:10.48550/ARXIV.1312.6114.

3. Jiang, L. P., and Rao, R. P. N. Predictive coding theories of cortical function (2021). URL: https://arxiv.org/abs/2112.10048. doi:10.48550/ARXIV.2112.10048.

4. Marino, J. (2022). Predictive Coding, Variational Autoencoders, and Biological Connections. Neural Computation *34*, 1–44. doi:10.1162/neco_a_01458.

5. Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An Introduction to Variational Methods for Graphical Models (105–161). Cambridge, MA, USA: MIT Press. ISBN 0262600323 (1999):(105–161).

6. Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. Intrinsically motivated learning in natural and artificial systems (17–47).

7. Schmidhuber, J. A possibility for implementing curiosity and boredom in model-building neural controllers. In: *Proc. of the international conference on simulation of adaptive behavior: From animals to animats* (1991):(222–227).

8. Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. CoRR *abs/1705.05363*. URL: http://arxiv.org/abs/1705.05363. arXiv:1705.05363.

9. Guo, Z., Thakoor, S., Pîslar, M., Avila Pires, B., Altché, F., Tallec, C., Saade, A., Calandriello, D., Grill, J.-B., Tang, Y. et al. (2022). Byol-explore: Exploration by bootstrapped prediction. Advances in neural information processing systems *35*, 31855–31870.

10. Hazan, E., Kakade, S. M., Singh, K., and Soest, A. V. (2018). Provably efficient maximum entropy exploration. CoRR *abs/1812.02690*. URL: http://arxiv.org/abs/1812.02690. arXiv:1812.02690.

11. Amin, S., Gomrokchi, M., Aboutalebi, H., Satija, H., and Precup, D. (2020). Locally persistent exploration in continuous control tasks with sparse rewards. CoRR *abs/2012.13658*. URL: https://arxiv.org/abs/2012.13658. arXiv:2012.13658.

12. Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. (2020). Fast active learning for pure exploration in reinforcement learning. CoRR *abs/2007.13442*. URL: https://arxiv.org/abs/2007.13442. arXiv:2007.13442.

13. Storck, J., Hochreiter, S., Schmidhuber, J. et al. Reinforcement driven information acquisition in non-deterministic environments. In: *Proceedings of the international conference on artificial neural networks, Paris* vol. 2 (1995):(159–164).

14. Sun, Y., Gomez, F. J., and Schmidhuber, J. (2011). Planning to be surprised: Optimal bayesian exploration in dynamic environments. CoRR *abs/1103.5708*. URL: http://arxiv.org/abs/1103.5708. arXiv:1103.5708.

15. Still, S., and Precup, D. (2012). An information-theoretic approach to curiosity-driven reinforcement learning. Theory in Biosciences *131*, 139–148.

16. Little, D. Y., and Sommer, F. T. (2013). Learning and exploration in action-perception loops. Frontiers in neural circuits *7*, 37.

17. Houthooft, R., Chen, X., Duan, Y., Schulman, J., Turck, F. D., and Abbeel, P. (2016). Curiosity-driven exploration in deep reinforcement learning via bayesian neural networks. CoRR *abs/1605.09674*. URL: http://arxiv.org/abs/1605.09674. arXiv:1605.09674.

18. Mohamed, S., and Rezende, D. J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. Advances in neural information processing systems *28*.

19. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In: *ICML* (2020):.

20. Seyde, T., Schwarting, W., Karaman, S., and Rus, D. (2020). Learning to plan optimistically: Uncertainty-guided deep exploration via latent model ensembles. arXiv preprint arXiv:2010.14641.

21. Mnih, V., Heess, N., Graves, A., and kavukcuoglu, k. Recurrent models of visual attention. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., eds. *Advances in Neural Information Processing Systems* vol. 27 (2014):.

22. Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. Draw: A recurrent neural network for image generation. In: Bach, F., and Blei, D., eds. *Proceedings of the 32nd International Conference on Machine Learning* vol. 37 of *Proceedings of Machine Learning Research*. Lille, France: PMLR (2015):(1462–1471). URL: https://proceedings.mlr.press/v37/gregor15.html.

23. Eslami, S., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G. E. et al. (2016). Attend, infer, repeat: Fast scene understanding with generative models. Advances in neural information processing systems *29*.

24. Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. (2018). Sequential attend, infer, repeat: Generative modelling of moving objects. Advances in Neural Information Processing Systems *31*.

25. Chaplot, D. S., Gandhi, D., Gupta, S., Gupta, A., and Salakhutdinov, R. (2020). Learning to explore using active neural slam. arXiv preprint arXiv:2004.05155.

26. Chaplot, D. S., Dalal, M., Gupta, S., Malik, J., and Salakhutdinov, R. R. (2021). Seal: Self-supervised embodied active learning using exploration and 3d consistency. Advances in neural information processing systems *34*, 13086–13098.

27. Scarpellini, G., Rosa, S., Morerio, P., Natale, L., and Del Bue, A. (2023). Look around and learn: self-improving object detection by exploration. arXiv preprint arXiv:2302.03566.

28. Jing, Y., and Kong, T. Learning to explore informative trajectories and samples for embodied perception. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023):(6050–6056). doi:10.1109/ICRA48891.2023.10160951.

29. Friston, K. (2010). The free-energy principle: a unified brain theory? Nature reviews neuroscience *11*, 127–138.

30. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: a process theory. Neural computation *29*, 1–49.

31. Kaplan, R., and Friston, K. J. (2018). Planning and navigation as active inference. Biological cybernetics *112*, 323–343.

32. Da Costa, L., Parr, T., Sajid, N., Veselic, S., Neacsu, V., and Friston, K. (2020). Active inference on discrete state-spaces: A synthesis. Journal of Mathematical Psychology *99*, 102447.

33. Lotter, W., Kreiman, G., and Cox, D. D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. CoRR *abs/1605.08104*. URL: http://arxiv.org/abs/1605.08104. arXiv:1605.08104.

34. Han, K., Wen, H., Zhang, Y., Fu, D., Culurciello, E., and Liu, Z. (2018). Deep predictive coding network with local recurrent processing for object recognition. CoRR *abs/1805.07526*. URL: http://arxiv.org/abs/1805.07526. arXiv:1805.07526.

35. Ororbia, A., and Mali, A. Convolutional neural generative coding: Scaling predictive coding to natural images (2023). arXiv:2211.12047.

36. Salvatori, T., Song, Y., Lukasiewicz, T., Bogacz, R., and Xu, Z. (2021). Predictive coding can do exact backpropagation on convolutional and recurrent neural networks. CoRR *abs/2103.03725*. URL: https://arxiv.org/abs/2103.03725. arXiv:2103.03725.

37. Millidge, B., Song, Y., Salvatori, T., Lukasiewicz, T., and Bogacz, R. Backpropagation at the infinitesimal inference limit of energy-based models: Unifying predictive coding, equilibrium propagation, and contrastive hebbian learning (2022). arXiv:2206.02629.

38. Ororbia, A., Mali, A. A., Kifer, D., and Giles, C. L. (2019). Lifelong neural predictive coding: Sparsity yields less forgetting when learning cumulatively. CoRR *abs/1905.10696*. URL: http://arxiv.org/abs/1905.10696. arXiv:1905.10696.

39. Ororbia, A., Mali, A., Giles, C. L., and Kifer, D. (2020). Continual learning of recurrent neural networks by locally aligning distributed representations. IEEE Transactions on Neural Networks and Learning Systems *31*, 4267–4278. doi:`10.1109/TNNLS.2019.2953622`.

40. Ororbia, A., and Mali, A. A. (2021). Backprop-free reinforcement learning with active neural generative coding. CoRR *abs/2107.07046*. URL: `https://arxiv.org/abs/2107.07046`. `arXiv:2107.07046`.

41. Gklezakos, D. C., and Rao, R. P. N. (2022). Active predictive coding networks: A neural solution to the problem of learning reference frames and part-whole hierarchies. CoRR *abs/2201.08813*. URL: `https://arxiv.org/abs/2201.08813`. `arXiv:2201.08813`.

42. Rao, R. P. N., Gklezakos, D. C., and Sathish, V. Active predictive coding: A unified neural framework for learning hierarchical world models for perception and planning (2022). `arXiv:2210.13461`.

43. Rao, R. P. N. (2022). A sensory-motor theory of the neocortex based on active predictive coding. bioRxiv. URL: `https://www.biorxiv.org/content/early/2022/12/31/2022.12.30.522267`. doi:`10.1101/2022.12.30.522267`. `arXiv:https://www.biorxiv.org/content/early/2022/12/31/2022.12.30.522267.full.pdf`.

44. Fisher, A., and Rao, R. P. N. Recursive neural programs: Variational learning of image grammars and part-whole hierarchies (2022). `arXiv:2206.08462`.

45. Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. Journal of machine learning research *9*.

46. Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In: *International Conference on Learning Representations* (2016):.