Neural networks learn an environment's geometry in latent space by performing predictive coding on visual scenes

James Gornet
Computation and Neural Systems
California Institute of Technology
jgornet@caltech.edu

Matthew Thomson
Biology and Biological Engineering
California Institute of Technology
mthomson@caltech.edu

Abstract

Humans navigate complex environments using only visual cues and self-motion. Mapping an environment is an essential task for navigation within a physical space; neuroscientists and cognitive scientists also postulate that mapping algorithms underlie cognition by mapping concepts, memories, and other nonspatial variables. Despite the broad importance of mapping algorithms in neuroscience, it is not clear how neural networks can build spatial maps exclusively from sensor observations without access to the environment's coordinates through reinforcement learning or supervised learning. Path integration, for example, implicitly needs the environment's coordinates to predict how past velocities translate into the current position. Here we show that predicting sensory observations—called predictive coding—extends path integration from implicitly requiring the environment's coordinates. Specifically, a neural network constructs an environmental map in its latent space by predicting visual input. As the network traverses complex environments in Minecraft, spatial proximity between object positions affects distances in the network's latent space. The relationship depends on the uniqueness of the environment's visual scene as measured by the mutual information between the images and spatial position. Predictive coding extends to any sequential dataset. Observations from paths traversing a manifold can generate such sequential data. We anticipate neural networks that perform predictive coding identify the underlying manifold without requiring the manifold's coordinates.

1 Introduction

After traversing a few paths, humans can learn to map complex environments using only visual scenes and self-motion. Experimental evidence in neuroscience also indicates that humans use mapping to build relationships between concepts given only a few examples. Despite the significance of mapping algorithms in neuroscience, it is not clear how neural networks can build spatial maps exclusively from sensor observations without access to the environment's coordinates. For example, path integration, also called dead-reckoning, estimates the current position x_t given the past velocity v_t. Path integration implicitly uses the environment's coordinates in how the velocity influences the transition function $\mathbb{P}[x_t|v_t]$. Recent approaches such as Banino *et al.* (2018) use coordinates as labels in supervised training. Moreover, studies such as Gupta *et al.* (2019) use the coordinates to determine whether an agent reaches a goal.

We show that predicting sensory observations—called predictive coding—is sufficient to map an environment. Predictive coding extends path integration from requiring the environment's coordinates. Specifically, a neural network constructs an environmental map in its latent space by traversing an environment and predicting visual scenes. The agent traverses various Minecraft environments with

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

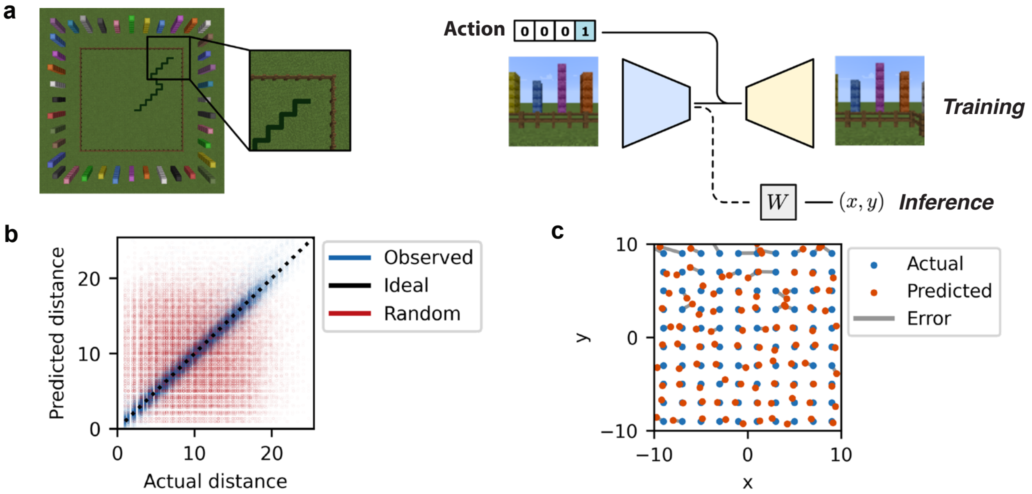

Figure 1: **A predictive coding neural network's latent space constructs an environmental map.** **a,** an agent performs predictive coding by predicting visual scenes while traversing an environment. A predictive coding neural networks encodes the current visual observation and action into a latent space, which the decoder uses to predict the next visual observation. **b,** distances in the neural network's latent space depend on the spatial proximity between object positions. The blue line (🟦) plots distances from the neural network's latent space to the the corresponding physical coordinates. The black line (⬛) plots an ideal decoder from the latent space to physical coordinates. The green line (🟥) plots a random decoder. **c,** the neural network's latent space accurately decodes the physical coordinates. The orange dot (•) plots the predicted coordinates from the network's latent space from a linear model. The blue dot (•) plots the physical coordinate position.

translational motion and receives only visual data. As the agent traverses the environment, distances in the network's latent space depend on the spatial proximity of object positions. Moreover, we demonstrate that the network's latent space predicts spatial position: a linear model can predict the environment's coordinates from the predictive coding network's latent space. The prediction's accuracy depends on the uniqueness of the environment's visual scene as measured by the mutual information between images and spatial position.

2 Related work

The hippocampal system not only develops a map of a spatial environment but also organizes sequential experiences (Buzsáki & Tingley, 2018). Experimental evidence suggests that the hippocampal maps sequences and abstract concepts by their relationships on an abstract map. Several studies establish that the hippocampal system also maps nonspatial variables—time (MacDonald *et al.*, 2011; Pastalkova *et al.*, 2008) and auditory frequency (Aronov *et al.*, 2017). Nieh *et al.* (2021) find that the hippocampal system integrates sensory input to determine the environment's state—not just position. Constantinescu *et al.* (2016) extends the principle that the hippocampus is a mapping unit, indicating humans organize abstract concepts using a gridlike neural representation.

Neurological and behavioral evidence indicate that path integration develops a map of an environment. However, it is not clear conceptually how humans map an environment from visual cues and self-motion—without knowledge of their own kinematics or the environment's coordinates. Banino *et al.* (2018), Cueva & Wei (2018), and Zhang *et al.* (2021) demonstrate that artificial neural networks can perform path integration and use a map to navigate. However, these studies require supervised training on the environment's coordinates. Similarly, Duan *et al.* (2016), Gupta *et al.* (2019), Mirowski *et al.* (2018), and Singh Chaplot *et al.* (2020) use visual input for mapping but encode the environment's coordinates either through a goal reward or through supervised training on coordinates. Previous studies implicitly introduce the environment's coordinates into the network by goal rewards or supervised training. It is unclear whether networks can map an environment exclusively from sensory data.

3 Predictive coding

Predictive inference involves framing the problem of learning from a sequence of actions and observations. The learner and decision maker are called the *agent*. With everything the agent interacts is called the *environment*. The *state space* S and the *action space* A are both topologies. The *observation space* O are random variables O dependent only on the current state s in the state space S. For a sequence $(s_1, \dots, s_T)$ of states, we assume the observations are *continuous in distribution*

$$\lim_{t' \to t} \mathbb{P}[O_{t'} | s_{t'}] = \mathbb{P}[O_t | s_t].$$

The learning problem is the following. An agent constructs a sequence of actions $\left(a_1, \dots, a_T\right)$, and observes samples $\left(O(s_1), \dots, O(s_T)\right)$. The goal is for the person to learn the stochastic process

$$\mathbb{P}[O(s_t) | O(s_{t-1}), \dots, O(s_1), a_{t-1}, \dots, a_1].$$

for every state s in S and action a in A.

Consider the example of a person walking through an open field (Figure 1.**a**). The person has a position (x, y). The person can walk around facing north. The state space is the Euclidean space $\mathbb{R}^2$. The action space is the person's velocity (v^x, v^y). Every action (v^x_{t-1}, v^y_{t-1}) deterministically maps a past state (x_{t-1}, y_{t-1}) to a current position (x_t, y_t) with the equations

$$x_t = x_{t-1} + v^x_{t-1} t, \quad y_t = y_{t-1} + v^y_{t-1} t$$

At each position in $\mathbb{R}^2$ there exists the random variable observations $O(x, y)$ the person can see given their state (x, y). A path, as a sequence of actions, generates a sequence of observations $(O_1, \dots, O_T) \sim (\mathbb{P}[O(x_1, y_1)], \dots, \mathbb{P}[O(x_T, y_T)])$, which satisfies a stochastic process. The goal is for the person to learn the stochastic process

$$\mathbb{P}[O(x_t, y_t) | O(x_{t-1}, y_{t-1}), \dots, O(x_1, y_1), (v^x_{t-1}, v^y_{t-1}), \dots, (v^x_1, v^y_1)]$$

for every position (x, y) and action (v^x, v^y).

4 A predictive coding neural network's latent space constructs an environmental map

We study how a neural network—without the environment's coordinates—maps an environment by performing predictive coding using the Malmo environment in Minecraft (Johnson *et al.,* 2016). We construct an open field (Figure 1.**a**) enclosed by a fence of 20 × 20 blocks. Columns of varying heights and colors provide the agent with a unique visual scene at each position. The agent can traverse the environment by moving forwards, backwards, left, and right by one block. We use an encoder-decoder convolutional neural network[†] with an embedding from the action to the latent space (Figure 1.**a**). To solve the predictive coding problem we assume the predictive coding distribution is a multivariate Gaussian. We estimate the density p by solving the maximum likelihood estimator

$$\arg\max_{\Theta} \log p\left(O_t | O_{t-1}, (v^x_{t-1}, v^y_{t-1}), \Theta\right) = \arg\max_{\Theta} \frac{1}{2}\left(O_t - \hat{O}_{\Theta}(O_{t-1}, v^x_{t-1}, v^y_{t-1})\right)^2$$

The neural network approximates predictive coding by minimizing the mean-squared error between the current visual scene and its prediction given the previous image and action.

We first show the latent space's capability to estimate the environment's distances within one block, demonstrating the latent space contains significant information about the physical space. To determine the latent axes that capture the environment's geometry, we fit a linear model that estimates the environment's coordinates from the latent space. We calculate the pairwise distance between the predicted locations as well as the actual locations (Figure 1.**c**). For the 20 × 20 block lattice, the mean error between predicted distances and actual distances is 0.743 ± 0.019 blocks (95% confidence interval).

Furthermore, the latent space captures a linear representation of physical coordinates. Figure 1.**c** shows the linear model's coordinate estimates (●) from the latent space. The grey lines show the error from the physical coordinates (●) with prediction error of 0.779 ± 0.186 blocks. The prediction error increases as the agent travels to the front, indicating the latent space contains less positional information toward the front.

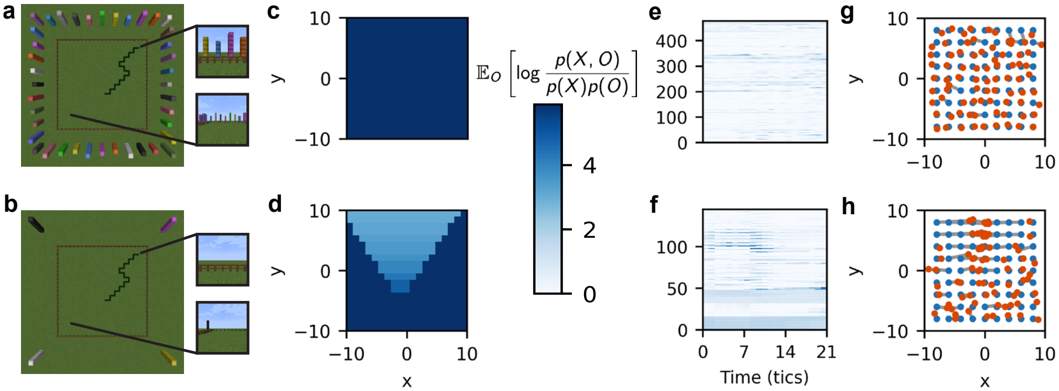

Figure 2: **Predictability of spatial location from the latent space depends on the uniqueness of the environment's visual scene.** **a,** Environment A. **b,** Environment B. **c-d,** the expected log probability difference between the joint probability $p(x \in X, O)$ and marginal probabilities $p(x \in X)$ and $p(O)$ for a position $x \in X$ and visual scenes O. Lower log difference indicates a higher uncertainty of the position from the visual scene. **e-f**, the non-zero latent space units. The more zero latent space units ($145 : 477$) in Environment B corresponds with the decreased mutual information ($1998.03 : 3067.0$) and indicates the latent space stores less spatial information. **g-h,** predicted positions given by a linear model on the neural network's latent space in Environments A and B, respectively. In Environment B, the latent space reconciles unidentifiable areas by minimizing prediction errors of physical coordinates.

5 Predictability of spatial location from the latent space depends on the uniqueness of the environment's visual scene

In section 4, we demonstrate a neural network can map the environment in Figure 1.**a** by learning predictive coding. In this environment (Environment A), positions are relatively identifiable from visual landmarks. Many environments contain areas with no identifiable landmarks. Introducing an environment (Environment B) with sparse landmarks (Figure 2.**b**), we show the neural network reconciles these areas in latent space by minimizing prediction errors of the physical coordinates. In this environment, many locations are not identifiable from landmarks. Between Environments A and B, we calculate the expected log difference $\mathbb{E}_{O\sim\mathbb{P}[O]}\left[\log p(x \in X, O) - \log p(x \in X)p(O)\right]$ between the joint probability $p(X, O)$ and marginal probabilities $p(X)$ and $p(O)$ between the location $x \in X$ and the visual scene O—based on the presence of visual landmarks (Figure 2.**c-d**). The mutual information $I(X; O)$ is the expectation of this parameter, i.e., $\mathbb{E}[\log p(X, O) - \log p(X)p(O)]$. Intuitively, areas in Environment B with a low log difference have locations that are less predictable from visual scenes—with a value of zero if the visual scene and positions are independent.

We visualize the latent space units as the agent traverses the environment (Figure 2.**e-f**). The neural network in Environment B has notably more zero latent units as compared to the neural network in Environment A. Similar to section 4, we use a linear model to estimate the predictability of the environment's positions from the network's latent space (Figure 2.**g-h**). The prediction error depends on the log probability difference $\mathbb{E}_{O\sim\mathbb{P}[O]}\left[\log p(X, O) - \log p(X)p(O)\right]$ between the visual scene and the agent's position; the network reconciles positions without landmarks by guessing in latent space.

6 Discussion

Humans can navigate complex environments using only visual landmarks and self-motion; however, current algorithms require the environment's coordinates or knowledge of their own kinematics. We demonstrated that predictive coding is sufficient to map an environment. By predicting visual scenes, predictive coding generalizes the prediction task of path integration to arbitrary visual scenes. In this paper, we treat predictive coding with the state space $\mathcal{S}$ as a Euclidean space. The state space $\mathcal{S}$ can be any topology, most notably a differentiable manifold. Observations $(O(\gamma(t)))_{t\in T}$ from a path $\gamma(t)$ along the manifold generates a stochastic process. We anticipate a majority of sequential learning problems can be formulated as predictive coding on manifolds. By learning predictive coding, neural networks may identify the underlying manifold without requiring the manifold's coordinates.

References

1. Aronov, D., Nevers, R. & Tank, D. W. Mapping of a Non-Spatial Dimension by the Hippocampal–Entorhinal Circuit. *Nature* **543.** 719–722 (Mar. 2017).
2. Banino, A. *et al.* Vector-Based Navigation Using Grid-like Representations in Artificial Agents. en. *Nature* **557.** 429–433 (May 2018).
3. Buzsáki, G. & Tingley, D. Space and Time: The Hippocampus as a Sequence Generator. *Trends in Cognitive Sciences* **22.** 853–869 (Oct. 2018).
4. Constantinescu, A. O., O'Reilly, J. X. & Behrens, T. E. J. Organizing Conceptual Knowledge in Humans with a Gridlike Code. *Science* **352.** 1464–1468 (June 2016).
5. Cueva, C. J. & Wei, X.-X. Emergence of Grid-like Representations by Training Recurrent Neural Networks to Perform Spatial Localization. *arXiv:1803.07770 [cs, q-bio, stat].* arXiv: `1803.07770 [cs, q-bio, stat]` (Mar. 2018).
6. Duan, Y. *et al.* RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning (Nov. 2016).
7. Gupta, S. *et al. Cognitive Mapping and Planning for Visual Navigation* Feb. 2019. arXiv: `1702.03920 [cs]`.
8. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* Dec. 2015. arXiv: `1512.03385 [cs]`.
9. Johnson, M., Hofmann, K., Hutton, T. & Bignell, D. *The Malmo Platform for Artificial Intelligence Experimentation* in *Proc. 25th International Joint Conference on Artificial Intelligence* (AAAI Press, 2016), 4246.
10. MacDonald, C. J., Lepage, K. Q., Eden, U. T. & Eichenbaum, H. Hippocampal "Time Cells" Bridge the Gap in Memory for Discontiguous Events. *Neuron* **71.** 737–749 (Aug. 2011).
11. Mirowski, P. *et al. Learning to Navigate in Cities Without a Map* in *Advances in Neural Information Processing Systems* **31** (Curran Associates, Inc., 2018).
12. Nieh, E. H. *et al.* Geometry of Abstract Learned Knowledge in the Hippocampus. *Nature* **595.** 80–84 (July 2021).
13. Pastalkova, E., Itskov, V., Amarasingham, A. & Buzsáki, G. Internally Generated Cell Assembly Sequences in the Rat Hippocampus. *Science* **321.** 1322–1327 (Sept. 2008).
14. Ronneberger, O., Fischer, P. & Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* May 2015. arXiv: `1505.04597 [cs]`.
15. Singh Chaplot, D., Salakhutdinov, R., Gupta, A. & Gupta, S. *Neural Topological SLAM for Visual Navigation* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Seattle, WA, USA, June 2020), 12872–12881.
16. Zhang, T., Rosenberg, M., Perona, P. & Meister, M. Endotaxis: A Universal Algorithm for Mapping, Goal-Learning, and Navigation. *bioRxiv : the preprint server for biology.* eprint: `https://www.biorxiv.org/content/early/2021/09/25/2021.09.24.461751.full.pdf` (2021).

Supplemental Information

6.1 Detailed experimental setup

For all experiments, the Malmo framework in Minecraft (Johnson *et al.,* 2016) simulate the environments. The agent traverses the environment by a directed random walk, which is discretized on the lattice. The actions are represented by a one-hot vector (left, right, forward, backward). Visual observations are represented by 64×64 RGB images.

For training, we use an U-Net architecture (Ronneberger *et al.,* 2015) with a ResNet38 encoder (He *et al.,* 2015) encoder and the corresponding ResNet38 with transposed convolutions for the decoder. A linear transform maps the action one-hot vector $\{0, 1\}^4$ to 512 channels of $\mathbb{R}^{2\times2}$ units. The $\mathbb{R}^{2\times2\times512}$ output units are concatenated to the output of the ResNet38 encoder to form a $\mathbb{R}^{2\times2\times1024}$ array. The $\mathbb{R}^{2\times2\times1024}$ is input into the corresponding ResNet38 decoder.

For Environment A and Environment B, the neural network trained on $20,000$ batches with a batch size of 128. The neural network minimizes the mean-squared error between the predicted and actual image using RMSprop with a learning rate of 10^{-3} with a decay of 0.1 every $5,000$ batches and a ℓ_2 weight decay of 5×10^{-6}.

6.2 Extended figures

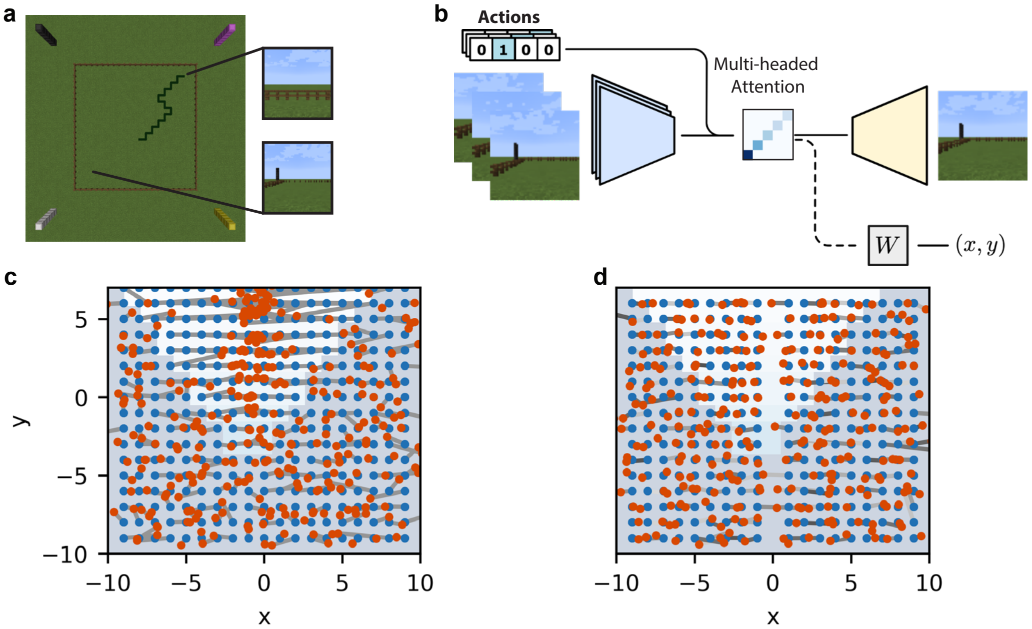

Figure 3: **Predictive coding with multiple past observations eliminates uncertainty in spatial location. a,** environment with locations unidentifiable from the visual scene (Environment B). **b,** a predictive coding neural network incorporates multi-head attention to encode multiple past observations and actions. **c,** a predictive coding neural network without attention cannot recover areas unidentifiable from the visual scene. The orange dot (●) plots the predicted coordinates from the network's latent space from a linear model. The blue dot (●) plots the physical coordinate position. **d,** a predictive coding neural network that incorporates multiple past observations and actions can recover visually unidentifiable areas. Predicted coordinates overlaps closely with the physical coordinate positions.

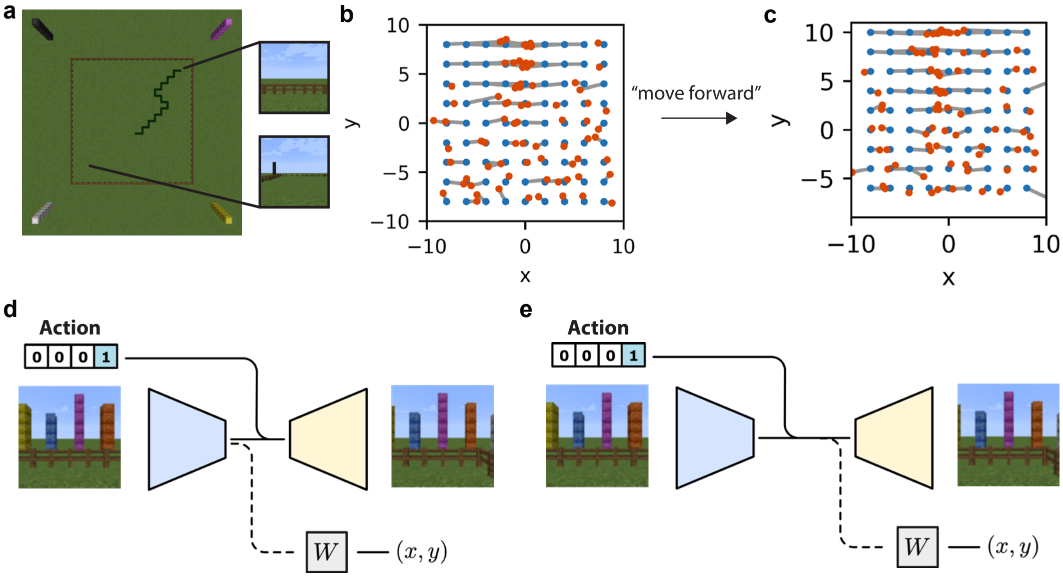

Figure 4: **Actions correspond to coordinate transformations in latent space. a,** environment with locations unidentifiable from the visual scene (Environment B). **b,** Predicted coordinates from the network's latent space before incorporating the current action. **c,** Predicted coordinates from the network's latent space after incorporating the current action. The predicted coordinates demonstrate a translation forward. The neural network—without knowledge of its kinematics—is able learn its motion in the environment by only predicting visual observations.

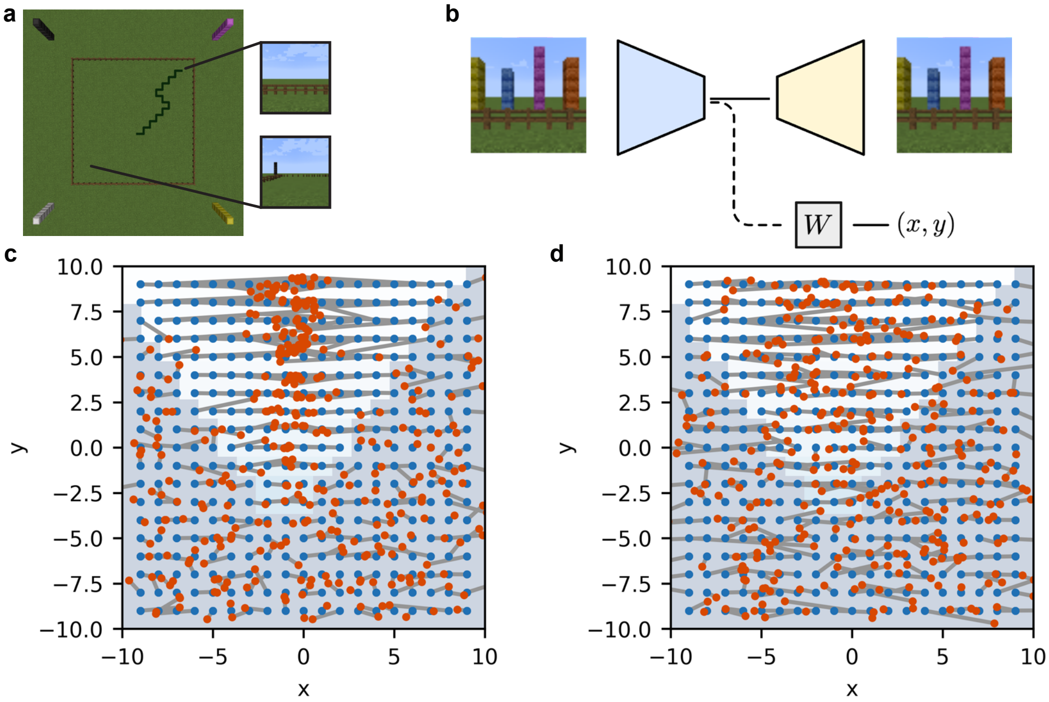

Figure 5: **Predictive coding—compared to auto-encoding—builds a more robust environmental map. a,** environment with locations unidentifiable from the visual scene (Environment B). **b,** Rather than predicting the current observation, auto-encoding predicts the same image by encoding it into a low dimensional space then decoding it to its original image. **c,** the predictive coding neural network's predicted coordinates. **d,** the auto-encoding neural network's predicted coordinates. The auto-encoder recovers the physical coordinates with a much larger error compared to the predictive coding neural network. In addition, the predictive coding neural network reconciles unidentifiable areas by minimizing prediction errors of physical coordinates, which is absent in the auto-encoder.

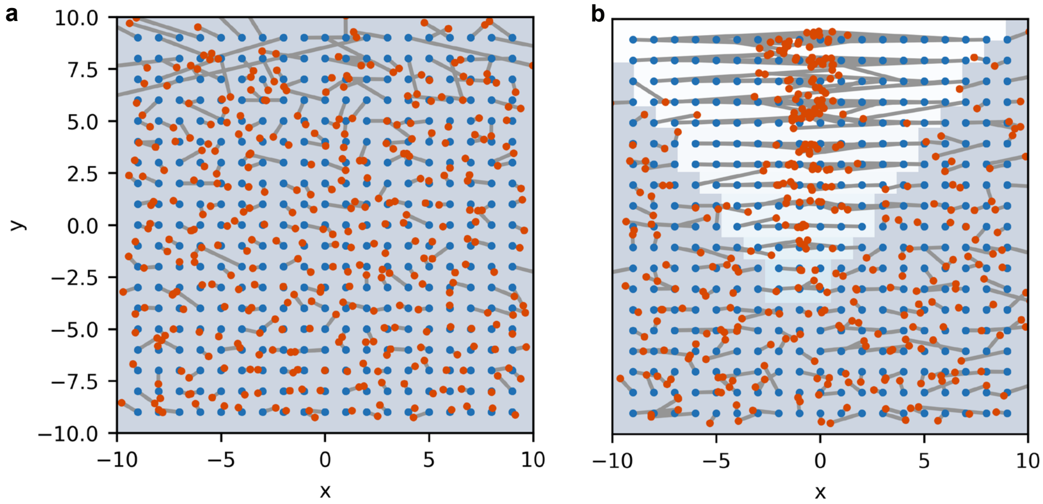

Figure 6: **Errors in predicted positions correspond to the log probability difference between visual scenes and positions.** A linear model fits the neural network's latent space to predict the coordinates in Environment A (**a**) and Environment B (**b**). The log probability difference between the joint probability $p(X, O)$ and the marginal probabilities $p(X)$ and $p(O)$ for positions X and visual scenes O.

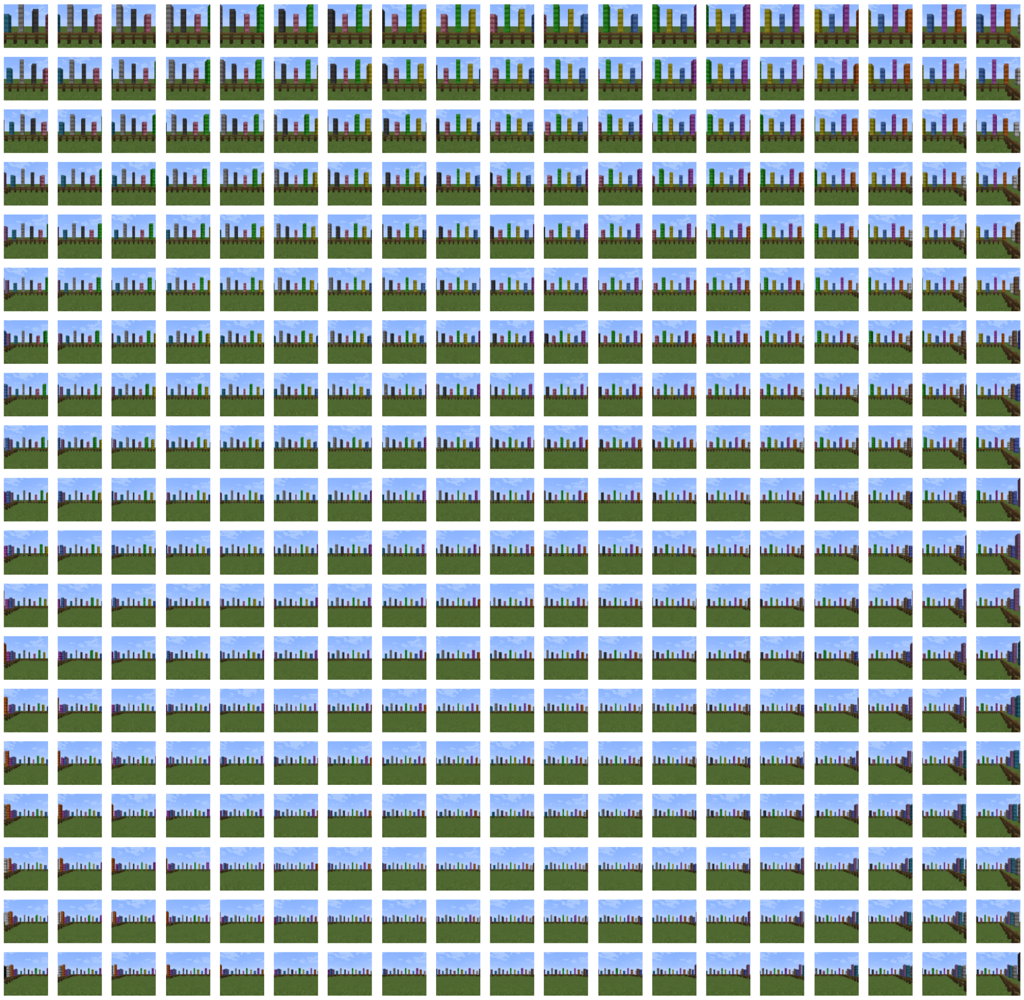

Figure 7: **Visual scenes in Environment A.** Each image corresponds the agent’s visual observation from its position in Environment A.

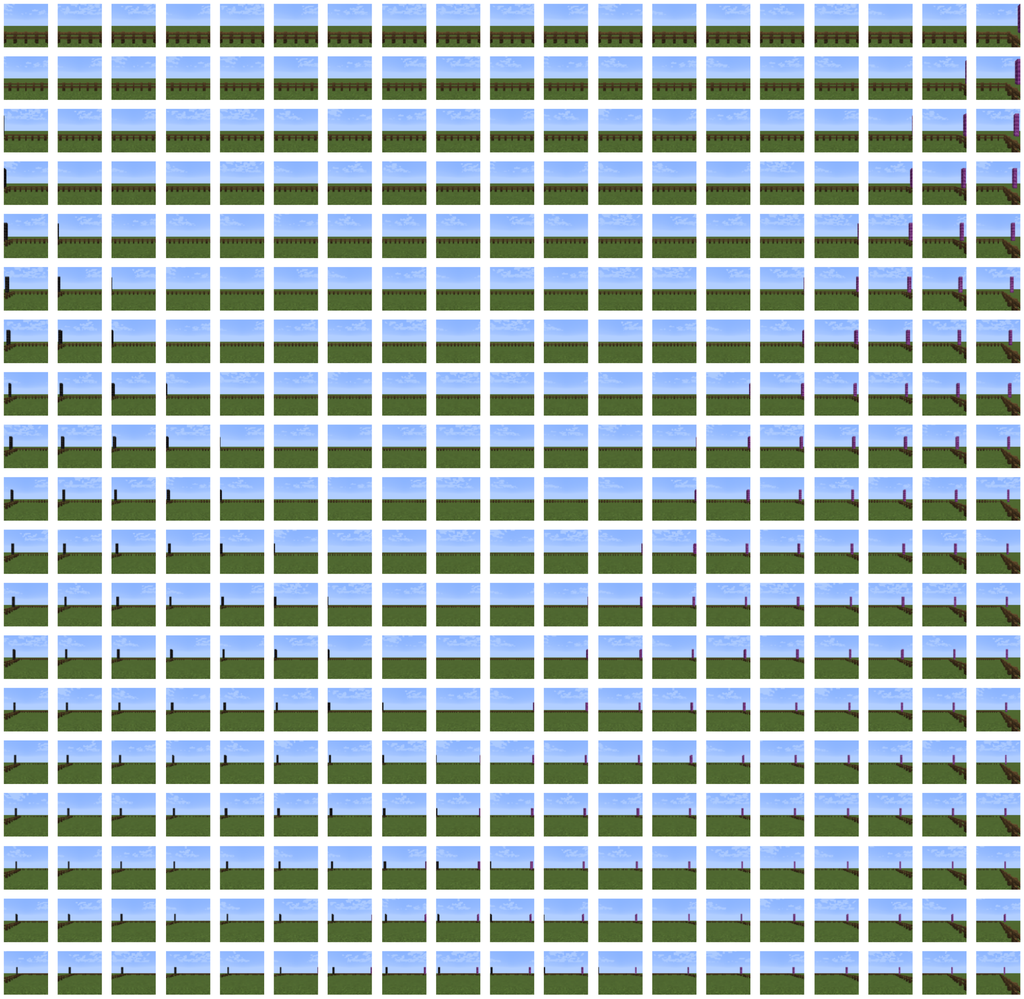

Figure 8: **Visual scenes in Environment B.** Each image corresponds the agent's visual observation from its position in Environment B.

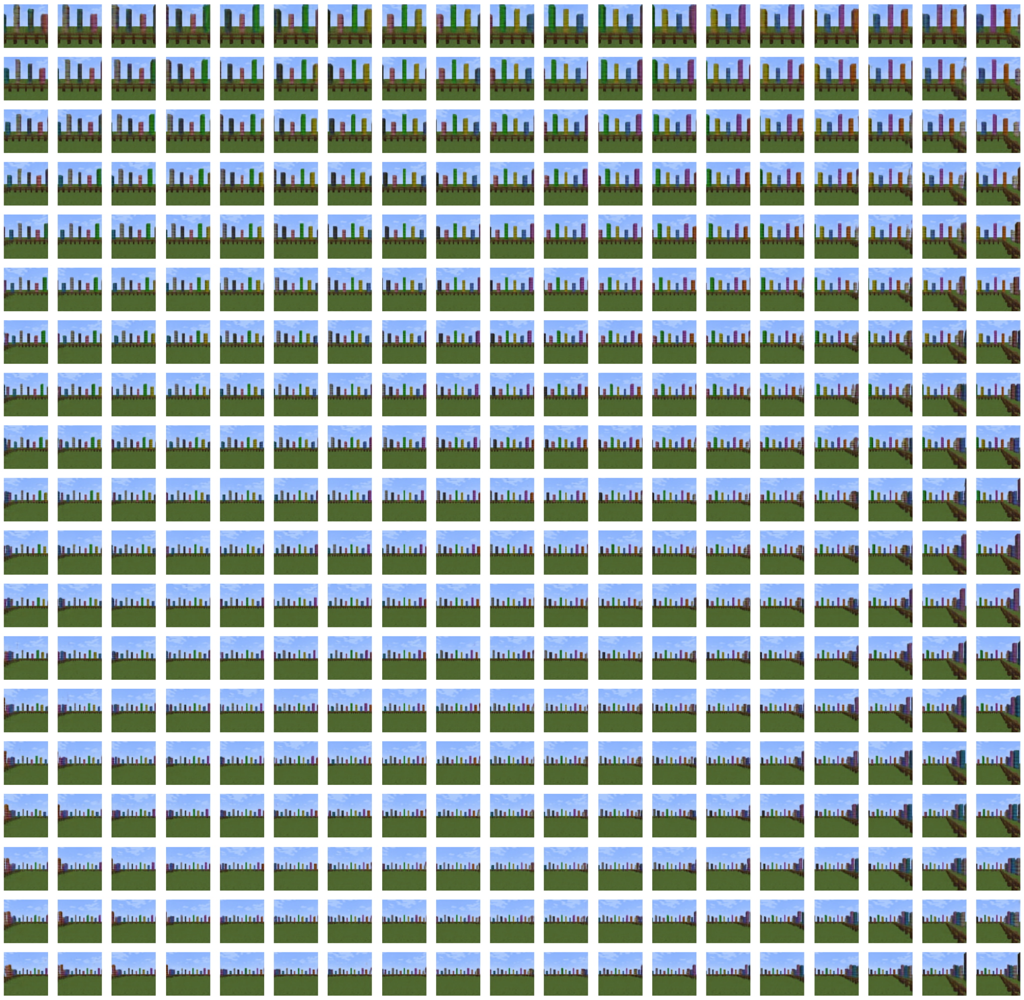

Figure 9: **Predicted visual scenes in Environment A.** Each image corresponds the agent's predicted visual observation by moving right from its position in Environment A.

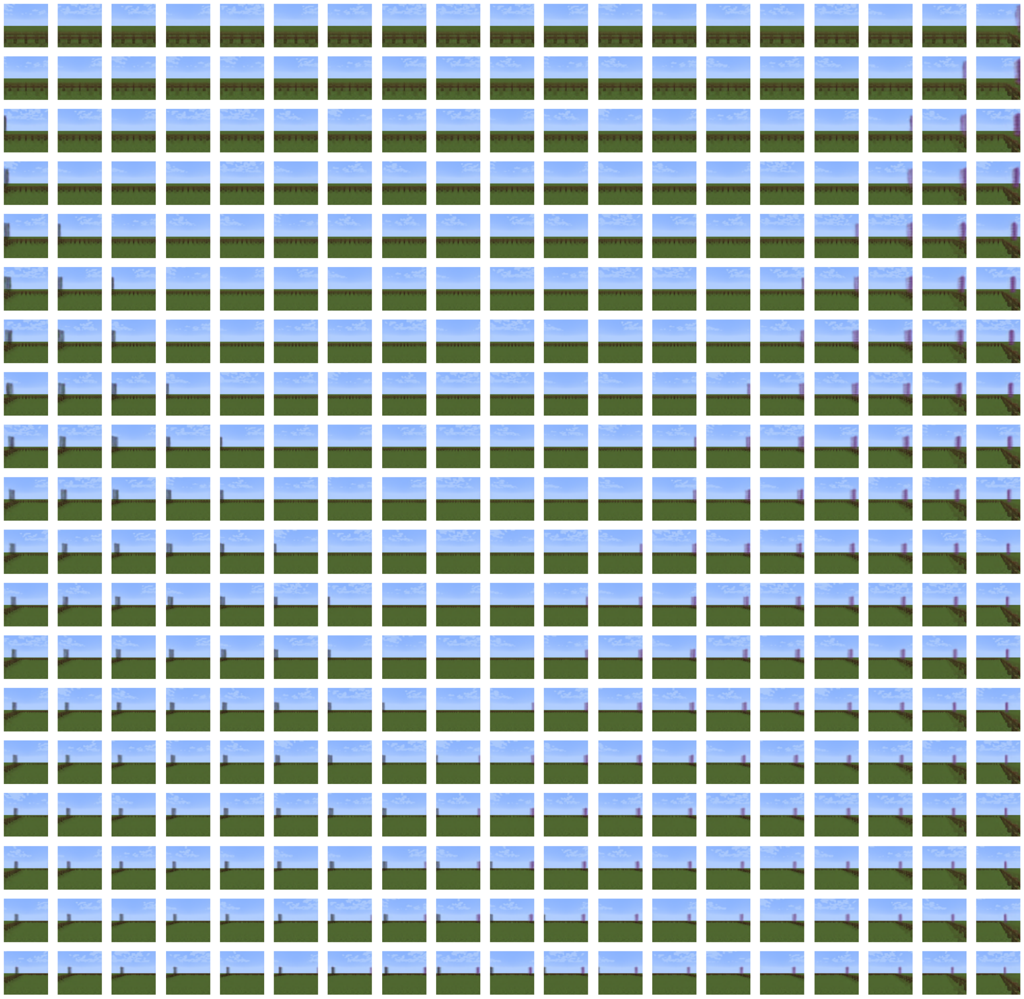

Figure 10: **Predicted visual scenes in Environment B.** Each image corresponds the agent's predicted visual observation by moving right from its position in Environment B.