Dynamic Predictive Coding: A New Model of Hierarchical Sequence Learning and Prediction in the Cortex

Linxing Preston Jiang[1, 2] and Rajesh P. N. Rao*[1, 2]

[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Center for Neurotechnology, University of Washington
{prestonj,rao}@cs.washington.edu

Abstract

We introduce dynamic predictive coding, a new hierarchical model of spatiotemporal prediction and sequence learning in the cortex. The model assumes that higher cortical levels modulate the temporal dynamics of lower levels, correcting their predictions of dynamics using precision-weighted prediction errors. We tested this model using a two-level neural network, where the top-down modulation is implemented as a low-dimensional mixture of possible temporal dynamics. When trained on natural videos, the first-level neurons developed space-time receptive fields similar to those found in simple cells of the primary visual cortex. The second-level responses spanned longer timescales and showed more stability than those at the first level, mimicking temporal response hierarchies in the cortex. After adapting to a repeated visual sequence, the model displayed full recall of the sequence given only the beginning of the sequence, similar to sequence recall in the visual cortex. Our results suggest that sequence learning and temporal prediction in the cortex can be interpreted as dynamic predictive coding based on a hierarchical generative model of input sequences.

Being able to predict future stimuli and event outcomes is a critical aspect of cognitive function. A fleeing animal needs to make predictions about the future trajectories of a predator based on past and current observations, and plan a sequence of actions for escaping the predator. These abilities are predicated on two requirements: (a) the brain must infer the dynamics of sensory stimuli to make spatiotemporal predictions with an internal model of the world, and (b) the brain's temporal representations must span different timescales to support predictions over both short and long horizons.

Many experimental studies have provided evidence for such computations. Predictive representations of upcoming stimuli have been found in various open and closed-loop paradigms where animals developed experience-dependent visual and auditory expectations [1–6]. Other empirical evidence suggests that cortical representations exhibit a hierarchy of timescales and an increase in stability from lower-order to higher-order areas across both sensory and cognitive regions [7–10]. We asked the question: could such phenomena be explained by the cortex learning a spatiotemporal generative model based on a temporal hierarchy of representations?

Previous models for spatiotemporal predictions were restricted to signal processing in the retina and LGN [11, 12]. Other models such as sparse coding [13] and independent component analysis [14] extract independent space-time filters from image sequences but do not predict future frames or learn temporal hierarchies. Singer et al. trained a hierarchical model by making higher-level neurons predict lower-level activities over a fixed time window [15, 16]. The model required a sequence of images to be provided as a single input and the hierarchy of timescales was hard-coded rather than learned.

Here, we present a new predictive coding model, dynamic predictive coding (DPC) with precision weighting, that addresses the limitations of previous models. Predictive coding provides a unifying framework for understanding perception and prediction in terms of learning hierarchical generative models of the environment [17–21]. In our new formulation, higher-level model neurons predict the *transition dynamics* of lower-level neural activities using a *hypernetwork*, a neural network that predicts the weights of another neural network [22]. Specifically,

*Corresponding Author

in DPC, the transition dynamics of the lower level are defined as a set of mixture weights for a group of learned transition matrices. These weights, which can be regarded as modulating lower level synapses, are predicted by the higher level. Additionally, each top-down prediction of weights is associated with a corresponding "confidence" or *precision*, defined as inverse variance. Precision scales the amplitude of prediction errors and controls the degree of influence exerted by the internal model during perception.

We tested the DPC model using a two-level neural network trained on natural and artificial image sequences, with the objective of minimizing precision-weighted prediction errors. After training, the first-level neurons developed space-time receptive fields similar to those found in simple cells in the primary visual cortex (V1) [23]. Neural responses in the second level learned to capture input dynamics on a longer timescale and exhibited greater stability compared to responses in the first level. Furthermore, after repeated sequence exposure to a fixed input sequence, the model exhibited cue-triggered activity recall, an effect previously reported in rodents [1], human V1 [24–26] and monkey V4 [27]. This effect emerged naturally in our DPC network as a result of prediction error minimization and an increase in precision as a result of learning. Our results suggest that the temporal response hierarchy and activity recall seen in the visual cortex may be a natural consequence of dynamic predictive coding of image sequences based on a hierarchical spatiotemporal generative model.

Results

Dynamic hierarchical predictive coding

The DPC model assumes that spatiotemporal inputs are generated by a hierarchical generative model (Figure 1a). We describe here a two-level hierarchical model, though the model can be readily extended to more levels. The lower-level of the model uses a spatial component which generates the input at each time step t via a set of spatial filters $\mathbf{U}$ and a sparse latent state vector $\mathbf{r}_t$ [28]: $\mathbf{I}_t = \mathbf{U}\mathbf{r}_t + \mathbf{n}$ where $\mathbf{n}$ is zero mean Gaussian white noise. The temporal dynamics of the state $\mathbf{r}_t$ is modeled using K learnable transition matrices $\left\{\mathbf{V}^{(k)}\right\}_{k=1}^{K}$ which can be linearly combined using a set of "modulation" weights given by a K-dimensional mixture vector $\mathbf{w}$. This vector of weights is generated by the higher-level state vector $\mathbf{r}^h$ (Figure 1(b)) using a function $\mathcal{H}$, implemented as a hypernetwork [22]:

$$\mathbf{w} = \mathcal{H}(\mathbf{r}^h) \tag{1}$$

$$\mathbf{V} = \sum_{k=1}^{K} w_k \mathbf{V}^{(k)}. \tag{2}$$

Here, w_k is the kth component of the vector $\mathbf{w}$. The lower-level state vector at time $t+1$ is generated as $\mathbf{r}_{t+1} = \mathrm{ReLU}(\mathbf{V}\mathbf{r}_t) + \mathbf{m}$ where $\mathbf{m}$ is zero mean Gaussian white noise (Figure 1(c)).

Additionally, the higher level also generates a "precision" signal for the lower level via a latent vector $\mathbf{r}^s$ and a function $\mathcal{G}$ (implemented as a neural network): $\beta = \mathcal{G}(\mathbf{r}^s)$ (Figure 1(b)). This precision signal, which is the inverse variance of the temporal prediction error distribution, scales the amplitude of temporal prediction errors during inference (see Methods). This allows the model to balance the influence of the current temporal prediction versus the current input in estimating the state and latent vectors at all levels. We assume that the input precision α, defined as the inverse variance of the input prediction error distribution (see Methods), is shared across all input frames and learned as a model parameter.

When an input sequence is presented, the model takes a Bayesian filtering approach to perform online inference on the latent vectors [29] by minimizing a loss function that includes prediction errors and penalties from prior distributions over the latents (see Methods). Given the model's estimates $\hat{\mathbf{r}}_t$, $\hat{\mathbf{r}}^s$, and $\hat{\mathbf{r}}^h$ at time t, the estimate $\hat{\mathbf{r}}_{t+1}$ of $\mathbf{r}$ at time $t+1$ is computed by gradient descent to minimize the sum of the precision-weighted spatial prediction error $\alpha\|\mathbf{I}_{t+1} - \mathbf{U}\hat{\mathbf{r}}_t\|_2^2$ and the precision-weighted temporal prediction error $\beta\|\mathbf{r}_{t+1} - \mathrm{ReLU}(\mathbf{V}\hat{\mathbf{r}}_t)\|_2^2$ (Figure 1(d)) plus a sparseness penalty. Similarly, the second level estimates $\hat{\mathbf{r}}^h$ and $\hat{\mathbf{r}}^s$ are updated using the temporal prediction error plus a prior-related penalty. The parameters of the model are learned by minimizing the same prediction errors across all time steps and input sequences, further reducing the errors not accounted for by the latent vector estimation process (see Methods).

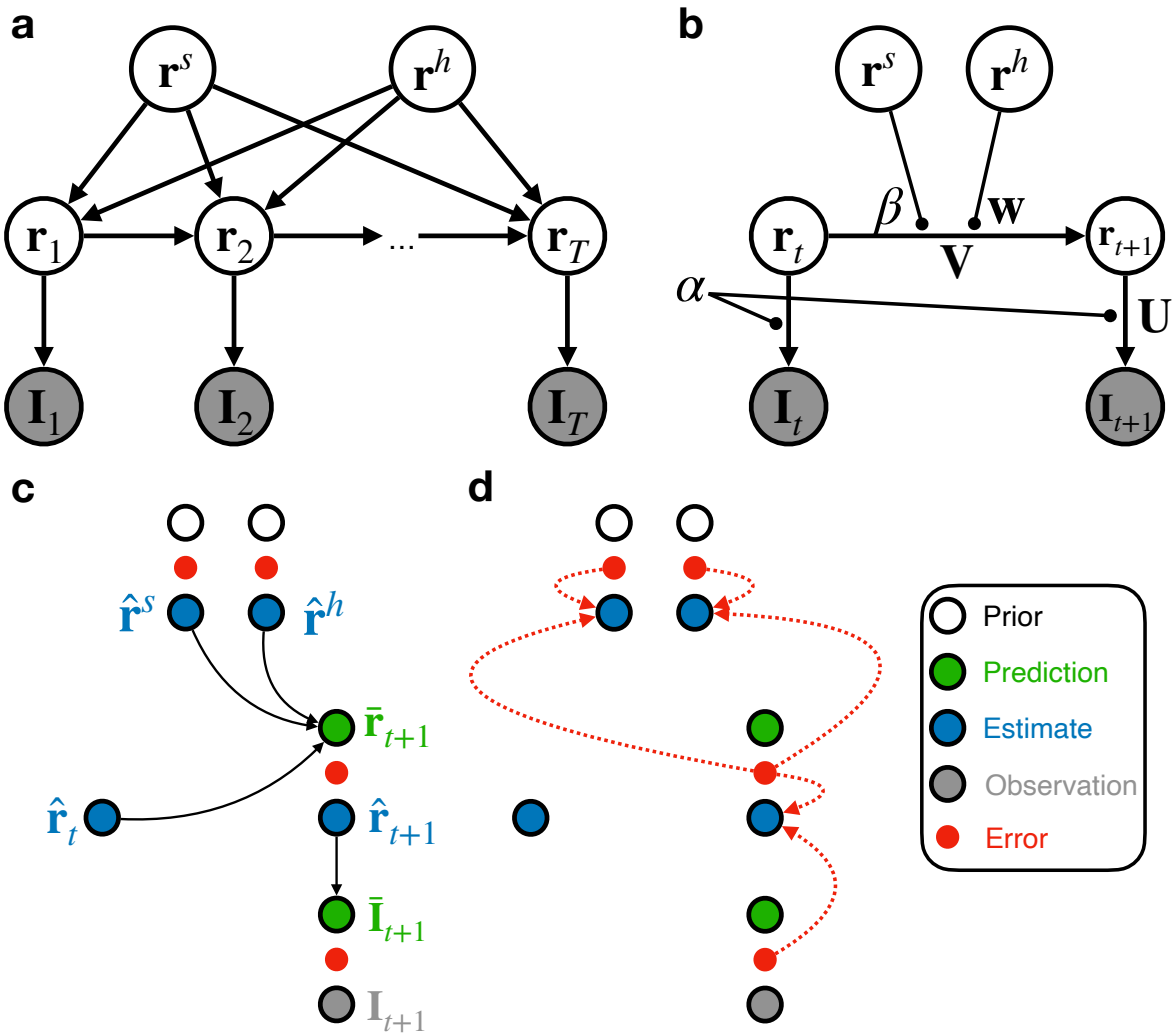

Figure 1: **Dynamic hierarchical predictive coding. (a)** Generative model for dynamic predictive coding. **(b)** Parameterization of the model. **(c)** Schematic depiction of spatiotemporal predictions (green) used for inference in the model. Note that $\bar{\mathbf{r}}_{t+1}$ is shown as a prediction node but it is equivalent to a temporal prior on $\mathbf{r}_{t+1}$. **(d)** Schematic depiction of prediction error computation and its use in correcting the current state estimates at two levels of the hierarchy.

Hierarchical predictive coding of natural videos

We implemented the DPC model described above using a two-level neural network where neural responses represent estimates of the latent vectors and the spatial filters and transition parameters are represented by synaptic weights. Perception corresponds to estimating the latent vectors by updating neural responses (through network dynamics) to minimize prediction errors via gradient descent (see Methods). Updating network parameters to further reduce prediction errors corresponds to learning (slow changes in synaptic weights through synaptic plasticity). We trained the network on thousands of natural image sequences extracted from a video recorded by a person walking on a forest trail (frame size: 10×10 pixels, sequence length: 10 frames (~0.35 seconds)). The frames were spatially and temporally whitened to simulate retinal and lateral geniculate nucleus (LGN) processing [28, 30]. The image sequences reserved for testing did not overlap in space or time with the training sequences (see Methods for details).

Figure 2(a) illustrates the inference process on an example natural image sequence by the network. The first row displays the ground truth input $\mathbf{I}_t$ for 10 time steps: each frame was shown sequentially to the model. The next row shows the model's predictions $\mathbf{U}\bar{\mathbf{r}}_t$ for each time step t, where $\bar{\mathbf{r}}_t$ was predicted by the previous state estimate $\hat{\mathbf{r}}_{t-1}$. The prediction errors $\mathbf{I}_t - \mathbf{U}\bar{\mathbf{r}}_t$ are shown in the third row. The prediction errors were the largest in the first two steps as the model inferred the spatial features and the transition dynamics from the initial inputs.

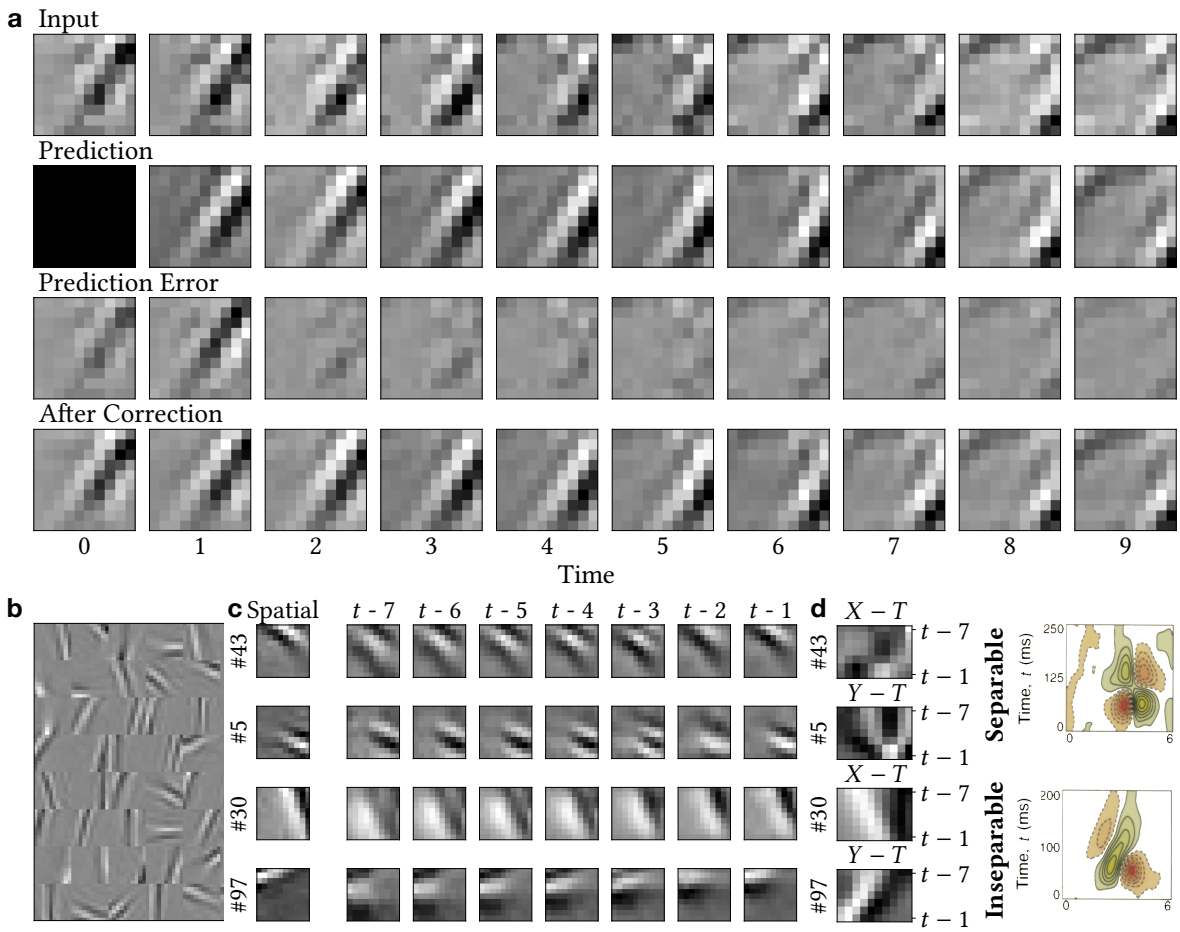

Figure 2: **Predictive coding of natural videos and learned space-time receptive fields**. **(a)** Inference on an example image sequence. Top to bottom: Input sequence ($\mathbf{I}_t$); model's prediction of the current input ($\mathbf{U}\bar{\mathbf{r}}_t$) from the previous step (the first step prediction being zero); prediction error ($\mathbf{I}_t - \mathbf{U}\bar{\mathbf{r}}_t$) visualized as subtracting the predicted input from the ground truth; model's final estimate of the current input after prediction error minimization ($\mathbf{U}\hat{\mathbf{r}}_t$). **(b)** 40 example spatial receptive fields (RFs) learned from natural videos. Each square tile is a column of $\mathbf{U}$ reshaped to a 10×10 image. **(c)** Space-Time RFs (STRFs) of four example first-level neurons. First column: the spatial RFs of the example neurons. Next 7 columns: the STRFs of the example neurons revealed by reverse correlation mapping. **(d)** Left panel: space-time plots of the example neurons in (c). Right panel: space-time plots of the RFs of two simple cells in the primary visual cortex of a cat (adapted from [23]).

The subsequent predictions were more accurate, resulting in minimized prediction errors. Finally, the last row shows the corrected estimates $\mathbf{U}\hat{\mathbf{r}}_t$ after $\bar{\mathbf{r}}_t$ has been updated to $\hat{\mathbf{r}}_t$ through prediction error minimization.

We examined the learned spatial receptive fields (RFs) of the model neurons at the first level and compared them with the spatial RFs of simple cells in primary visual cortex (V1). A subset of the spatial filters (columns of $\mathbf{U}$) learned by the model from our natural videos dataset are shown in Figure 2(b). These filters resemble oriented Gabor-like edge or bar detectors, similar to the localized, orientation-selective spatial RFs found in V1 simple cells [28, 31].

To measure the spatiotemporal receptive fields of the first-level neurons, we ran a reverse correlation experiment [32, 33] with a continuous natural video clip (≈ 47 minutes) extracted from the same forest trail natural video used for training. This video was not shown to the model during either training or testing (see Methods). Figure 2(c) shows the spatiotemporal receptive fields for four example first-level model neurons, computed by weighting input frames from the seven previous time steps $\mathbf{I}_{t-7}, \mathbf{I}_{t-6}, \ldots, \mathbf{I}_{t-1}$ by the response $\hat{\mathbf{r}}_t$ they caused at the current time step t (see Methods). The resulting average spatiotemporal receptive fields are shown as seven-image sequences labeled $t-7, t-6, \ldots, t-1$ (lasting ≈ 250 milliseconds in total). The first column labeled "Spatial RF" shows the spatial RFs of the example neurons.

To compute the space-time receptive fields (STRFs), we took the spatiotemporal $X - Y - T$ receptive field cubes and collapsed either the X or Y dimension, depending on which axis had time-invariant responses. Figure 2(d) left panel shows the $X/Y - T$ receptive fields of these example neurons. For comparison, Figure 2(d) right panel shows the STRFs of simple cells in the primary visual cortex (V1) of a cat (adapted from DeAngelis et al. [23]). DeAngelis et al. categorized the receptive fields of simple cells in V1 to be space-time separable (Figure 2(d) top row) and inseparable (Figure 2(d) bottom row). Space-time separable receptive fields maintain the spatial form of bright/dark-excitatory regions over time but switch their polarization: the space-time receptive field can thus be obtained by multiplying separate spatial and temporal receptive fields. Space-time inseparable receptive fields on the other hand exhibit bright/dark-excitatory regions that shift gradually over time, showing an orientation in the space-time domain. Neurons with space-time inseparable receptive fields are direction-selective, responding to motion in only one direction. As seen in Figure 2(d) left pane, the neurons in the first level of our network learned similar V1-like separable and inseparable STRFs, based on the principle of spatiotemporal prediction error minimization.

Temporal hierarchy through prediction of dynamics

Next, we show that our two-level network learned temporal representations with hierarchical timescales. In our formulation of the model, the second-level states predict the *dynamics* of the first-level state transitions. This means the second-level network neurons would have stable activation for input sequences with consistent dynamics. When a change occurs in input dynamics, the second-level responses switch to a different activation profile to minimize prediction errors. We hypothesize that the different timescales of neural responses observed in the cortex [7–10] could be an emergent property of the cortex learning a similar hierarchical generative model.

We tested our hypothesis in our trained network with natural videos. Figure 3(a) shows the neural activities responding to the natural video sequence example shown in Figure 2(a). As seen in the figure, the first-level responses changed rapidly as the stimulus moved downwards (left panel). The second-level responses (middle panel) and the predicted transition dynamics (right panel) were more stable after the initial adaptation to the motion. Since the stimulus continued to follow roughly the same dynamics (downward motion) after the first two steps, the transition matrix predicted by the second-level neurons continued to be accurate for the rest of the steps, leading to small prediction errors and few changes in the responses. Note that we did not enforce a longer time constant or smoothness constraint for $\mathbf{r}^h$ during inference - the longer timescale and more stable responses are entirely a result of the higher-level neurons learning to predict the lower-level dynamics under the proposed generative model.

We computed the autocorrelation of the first- and second-level responses to unseen natural videos and fitted an exponential decay function (see Methods) to quantify the response timescales. As Figure 3(b) shows, the autocorrelation of the second-level responses $\mathbf{r}^h$ is higher than the first-level response $\mathbf{r}$ and has a slower decay rate (exponential time constant for $\mathbf{r}^h$: 9.49 time steps; for $\mathbf{r}$: 3.67 steps). To factor out the effect of the natural video statistics, we computed the same autocorrelation using Gaussian white noise sequences. We found that the hierarchy of timescales still exists ($\mathbf{r}^h$: 0.79 time steps; $\mathbf{r}$: 0.18 steps). Figure 3(e) shows the autocorrelation of neural responses from the medial-temporal (MT) area in the visual cortex and lateral prefrontal cortex (LPFC) of monkeys for time periods preceding a motion task [7]. These neural responses show a difference in timescales qualitatively similar to the different response timescales in our hierarchical network.

We examined our network's ability to learn hierarchical temporal representations in more detail by training it on the Moving MNIST dataset [34]. Each image sequence contains ten 16×16 pixel frames. Each frame contains a single digit and the digit identity does not change within the sequence. The digit's motion is limited to up, down, left or right directions with a fixed speed. Figure 3(e) illustrates the trained network's inference process on an example image sequence. Similar to the responses to the natural video sequence, the first-level responses displayed fast changes while the second-level responses spanned a longer timescale and showed greater stability (Figure 3(f)). Note that at time $t = 5$, the input dynamics changed as the digit "bounced" against the upper boundary and started to move downwards (Figure 3(e) red dashed box). The second-level neurons predicted an upward motion at $t = 5$, resulting in a large prediction error as the input digit had already begun the downward motion (Figure 3(e) third row). The prediction error caused notable changes in the second-level responses $\mathbf{r}^h$ (Figure 3(c) red dashed boxes). For the rest of the steps, $\mathbf{r}^h$ remained stable and generated accurate predictions of the dynamics (which were also stable).

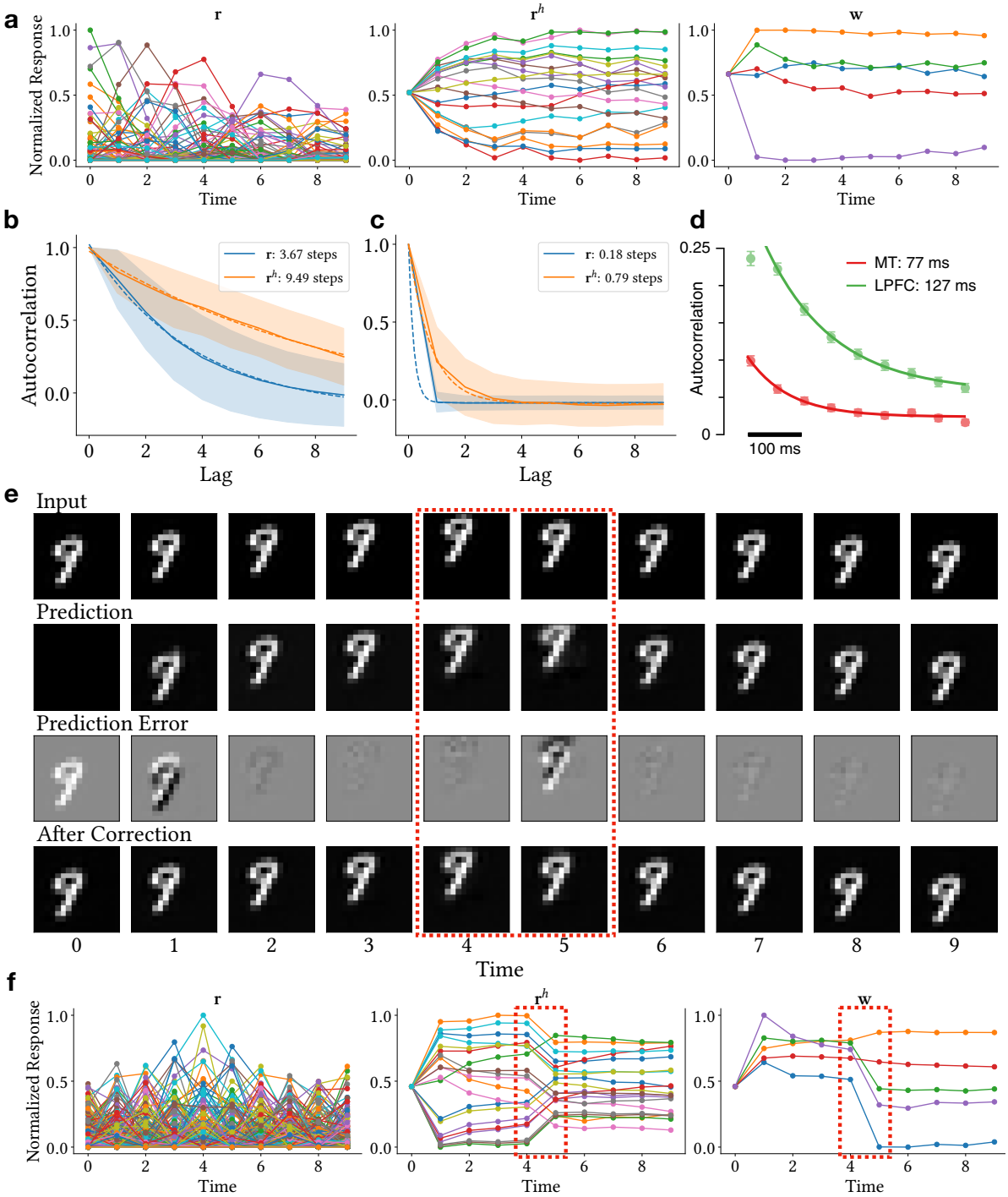

Figure 3: **Hierarchical temporal representation with different timescales. (a)** The trained DPC network's response to the natural image sequence in Figure 2(a). Each plotted line represents the responses of a model neuron over 10 time steps. Left: first-level responses; middle: second-level responses; right: predicted transition dynamics (each line being a mixture weight for a basis transition matrix). **(b)** Autocorrelation of the first- and second-level responses in the trained network for natural videos. Shaded area denotes ±1 standard deviation. Dotted lines show fitted exponential decay functions. **(c)** Same as (b) but for white noise stimuli. **(d)** Autocorrelation of the neural responses recorded from MT and LPFC of monkeys. Adapted from Murray et al. [7] **(e)** Inference for an example Moving MNIST sequence in a trained network. The red dashed box marks the time steps when the dynamics of the input changed. **(f)** The network's responses to the input Moving MNIST sequence in (e). Note the change in the second-level responses after the input dynamics changed (red dashed boxes); this gradient-based change helps to minimize prediction errors.

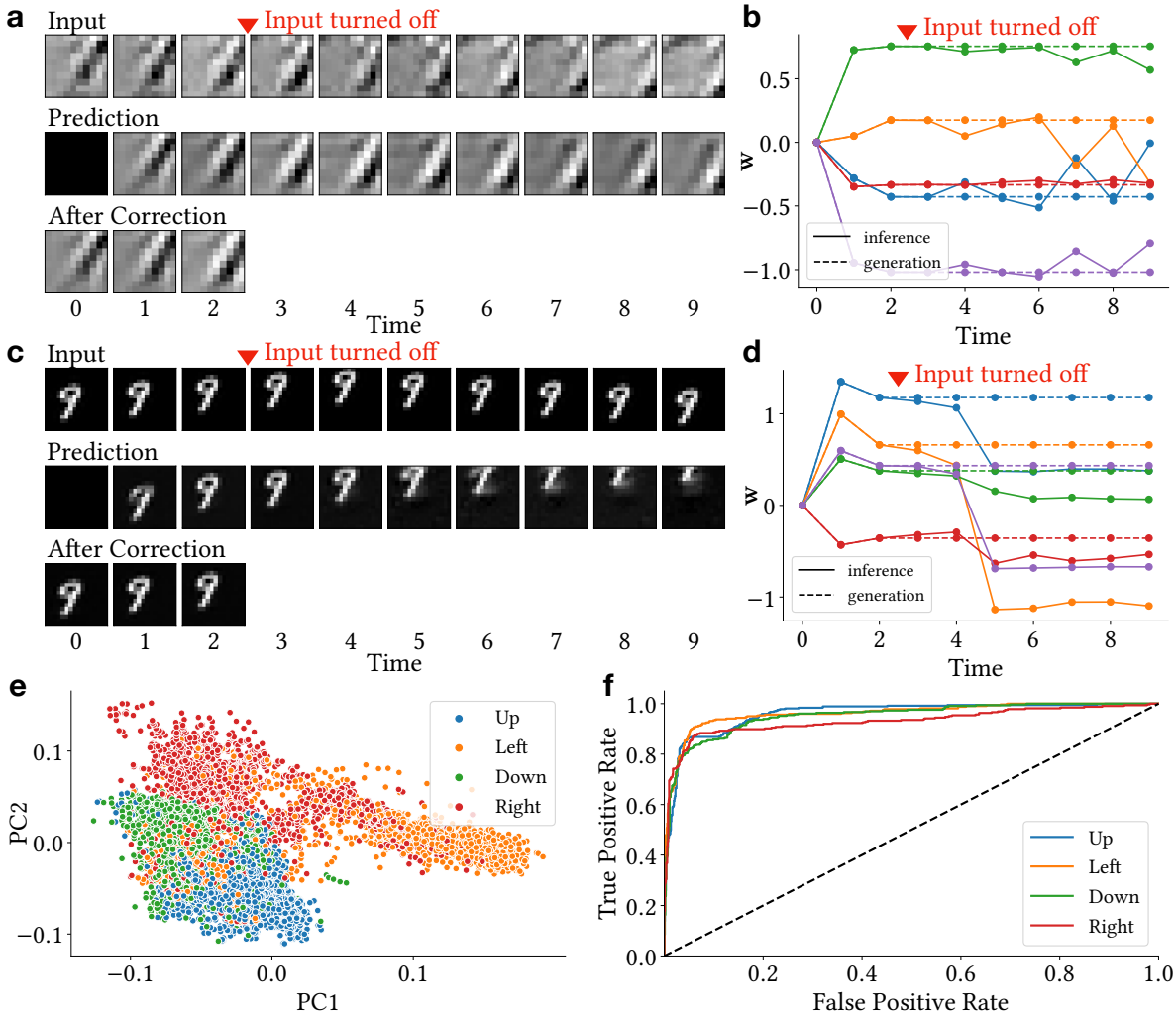

Figure 4: **Generative capabilities of the second-level model neurons. (a)** Second-level responses support longer timescale predictions of natural image sequences. The network predicted the rest of the sequence ($t > 3$) after the input was turned off (red triangle). **(b)** Comparison of the transition dynamics estimated from the full sequence (solid lines) with those inferred from the first three steps (dashed lines). **(c-d)** Same as (a) and (b) but for a Moving MNIST sequence. **(e)** Second-level responses to the Moving MNIST sequences visualized in the 2D space of the first two principal components, colored according to motion direction. **(f)** Receiver operating characteristic (ROC) curve for a logistic regression classifier of motion direction using the second-level responses as the features.

We also investigated the generative capabilities of the trained network by testing how well it can predict sequences without any inputs, after inferring dynamics from the first few frames. Figure 4(a) shows an example: no input frames were given after the initial three steps (marked by the red triangle), after which the network predicted the rest of the seven steps using the inferred transition dynamics $\mathbf{w}$. As Figure 4(b) shows, this inferred $\mathbf{w}$ closely matches the $\mathbf{w}$ for the full sequence, thereby supporting sequence prediction without inputs. In the example Moving MNIST sequence shown in the Figure 4(c), the network predicted a continued upward motion for the digit, given that no prediction error was provided at the bouncing step to correct the network's estimates.

Lastly, we confirmed that input dynamics are indeed encoded in the second-level responses and not just in the first-level response dynamics. We performed principal component analysis (PCA) on the second-level responses $\mathbf{r}^h$ for the Moving MNIST sequences in the test set. Figure 4(e) visualizes these responses in the space of their first two principal components (PCs). The responses clearly formed clusters by input motion direction. Logistic regression on $\mathbf{r}^h$ yielded 81.1% 4-class classification accuracy on digit motion direction in the test set (up, down, left, right). Figure 4(f) displays the receiver operating characteristic (ROC) curve for the logistic regression classifier.

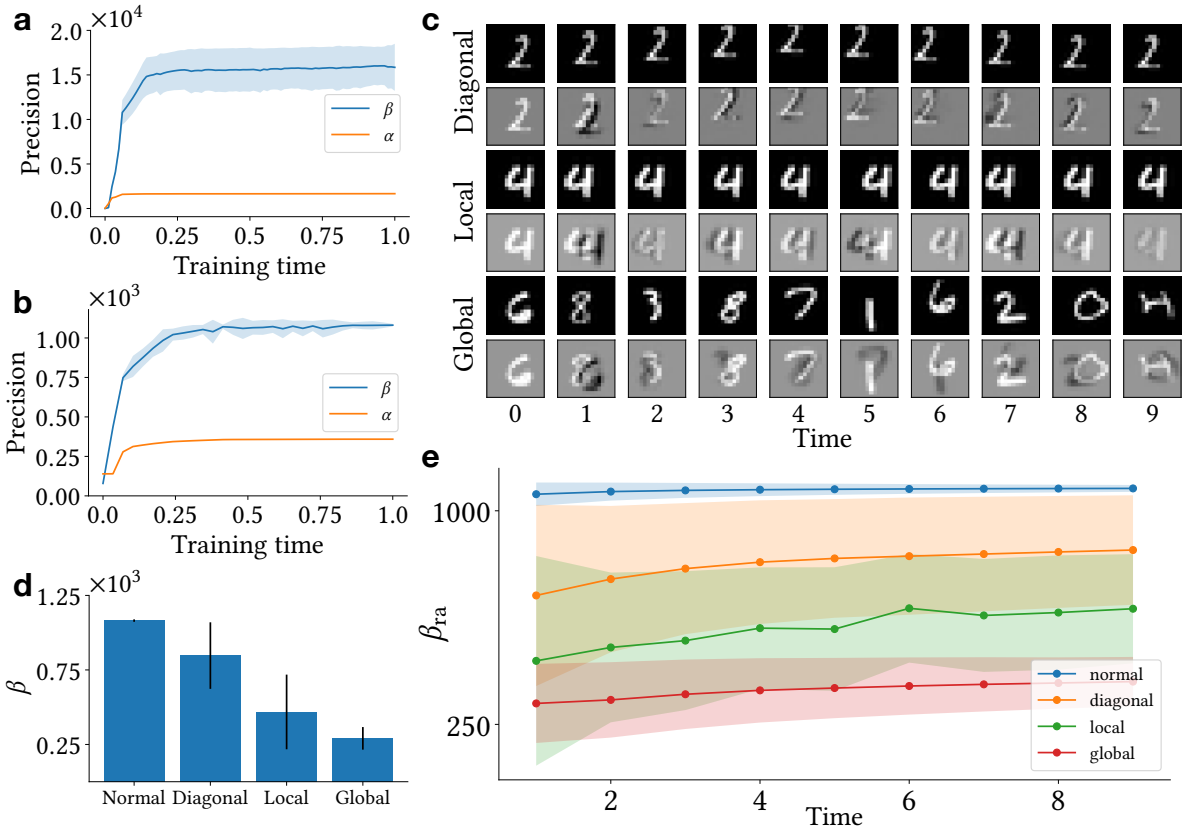

Figure 5: **Predictability of input dynamics affects precision. (a)** α (spatial precision) and β (temporal precision) for the natural video test set as training progresses. The blue line plot denotes average β while the shaded area denotes ± 1 standard deviation. Note that α is a single real value parameter of the model. **(b)** Same as (a) but for the Moving MNIST test set. **(c)** Example pairs of the input sequence under different dynamic modifications, and their corresponding prediction errors. **(d)** Comparison of β across different dynamic modifications on the Moving MNIST test set. **(e)** Running averages of β over time (denoted as β_{ra}) from different dynamics modifications. Shared areas denote ± 1 standard deviation.

In summary, these results show that a two-level network implementing the proposed generative model can successfully learn hierarchical temporal representations, with the second-level response vector representing entire sequences at the first level (by generating the transition dynamics weights), thereby showing greater stability than the first-level. Our results suggest that the increasing response timescales observed along the cortical hierarchy could be a result of the cortex learning a hierarchical spatiotemporal generative model of natural image sequences.

Precision reflects predictability of input dynamics by the model

The previous section has shown that the second-level latent vector $\mathbf{r}^h$ can learn to generate transition dynamics of the first-level states. We now turn to investigate the other second-level latent vector $\mathbf{r}^s$ and the top-down precision β it generates. Precision serves as a multiplicative gain on prediction errors and is defined as the inverse variance of the first-level temporal prediction errors. During inference, $\mathbf{r}^s$ is updated by minimizing the sum of precision-weighted temporal prediction errors $\beta\|\mathbf{r}_t - \hat{\mathbf{r}}_t\|_2^2$ and a regularization term (see Methods).

By definition, a higher value for top-down precision β indicates better prediction of lower level sequences as the difference between the predicted lower-level state and the actual state is reduced. Figure 5 shows the evolution of the average β predicted by the network when training on natural sequences (a) and Moving MNIST sequences (b). For comparison, the input precision α, which is a model parameter and learned across all input sequences, is also shown. As training proceeded, both precision values increased as the network generated more accurate predictions. The standard deviation of β for the Moving MNIST test set was much smaller than that for the natural video test set, reflecting the less noisy nature of the Moving MNIST dynamics.

What happens when the network encounters input dynamics that it has not been trained on? Intuitively, the temporal prediction errors will be large as the network is not adapted to these dynamics, which will then lead to reduced temporal precision β, encouraging the network to rely on the spatial prediction errors for inference. To test this hypothesis, we exposed the network trained on Moving MNIST sequences to three novel kinds of digit motion not in the training data: diagonal motion (motion in four diagonal directions), local shuffling (the Moving MNIST test dataset sequences were temporally shuffled), and global shuffling (both the digit identities and the temporal order were shuffled). Figure 5(c) shows three such example input sequences and the corresponding prediction errors for these sequences. As Figure 5(d) shows, the average β decreased monotonically going from the diagonal motion to local shuffling and global shuffling conditions. Figure 5(e) shows the running averages of β across time (denoted as β_{ra}) across sequences. These results show that the precision component can reflect the uncertainty of the network in the input dynamics and decrease the influence of the network's temporal prediction during inference when the input dynamics are unfamiliar.

Our results thus far demonstrate how a dynamic predictive coding network can (1) hierarchically predict the input dynamics and (2) estimate the precision of that prediction. We next show how these two components of the model provide a principled explanation for sequence learning and cue-triggered activity recall in V1 [1, 24].

High precision supports cue-triggered recall

A number of experiments in rodents have shown that primary visual cortex (V1) encodes predictive representations of upcoming stimuli [1–4, 35]. In a notable study, Xu et al. demonstrated that after exposing rats repeatedly to a moving dot visual sequence, only displaying the starting dot stimulus triggered sequential firing in V1 in the same order as when displaying the complete sequence [1]. Similar effects have been reported in monkey [27] and human [24–26] visual cortical areas as well. This phenomenon of cue-triggered activity recall poses an intriguing challenge for predictive coding models: why does V1 continue to recall the visual sequence when the observations are omitted, eliciting large input prediction errors? Why don't these input prediction errors correct the representations in V1 to predict the empty frames?

We hypothesize that this effect occurs in a dynamic predictive coding network due to prediction of dynamics with high temporal precision: the specific dynamics of a reliable and often repeated sequence is learned well by the network and linked to a particular higher-level state $\mathbf{r}^h$. Because of minimized temporal prediction errors after learning the sequence, the network also learns to predict (via $\mathbf{r}^s$) a high precision value β for the sequence. When the starting cue is displayed, the higher level converges quickly to the $\mathbf{r}^h$ and $\mathbf{r}^s$ for the trained sequence. These higher level representations generate the full dynamics of the sequence and the associated high precision value respectively, suppressing the input prediction errors due to the blank stimuli and resulting in full recall of the sequence.

To test this hypothesis, we simulated the Xu et al. experiment using a network trained on natural videos and compared the network's responses to experimental data. In the experimented setup by Xu et al., a dot motion sequence from a START location to an END location was repeatedly shown to rats during conditioning (Figure 6(a)); this evoked sequential firing of V1 neurons whose RFs were centered along the motion trajectory (Figure 6(b), RF locations shown in Figure 6(a)). After several minutes of conditioning, this sequential firing was re-activated when the dot at the starting location was briefly displayed [1].

We simulated the conditioning stimuli using a moving bar sequence shown in Figure 5(c), with three initial empty frames to simulate the resting period before each trial. During conditioning, the DPC network trained on natural videos evoked a sequential firing order in the first-level neurons (Figure 6(d) upper left panel, normalized activation of eight most active neurons are shown). Conditioning involved adapting the temporal components of the network through the same prediction error minimization procedure that was used for training the network on natural videos. The spatial components of the network were kept fixed (see Methods for details).

After conditioning (75 repetitions of the same moving bar sequence shown in Figure 6(c)), the network was tested with the starting frame only (red box), the middle frame only (blue box), and the end frame only (green box) followed by seven empty frames. We found that the first-level model neurons had developed cue-triggered activity recall given only the start frame of the sequence (Figure 6(d) upper right panel). Cueing the network with the middle or end frame of the sequence did not trigger the recall, consistent with findings by Xu et al. [1]. To understand the genesis of this activity recall behavior, we examined model neuron responses at the second-

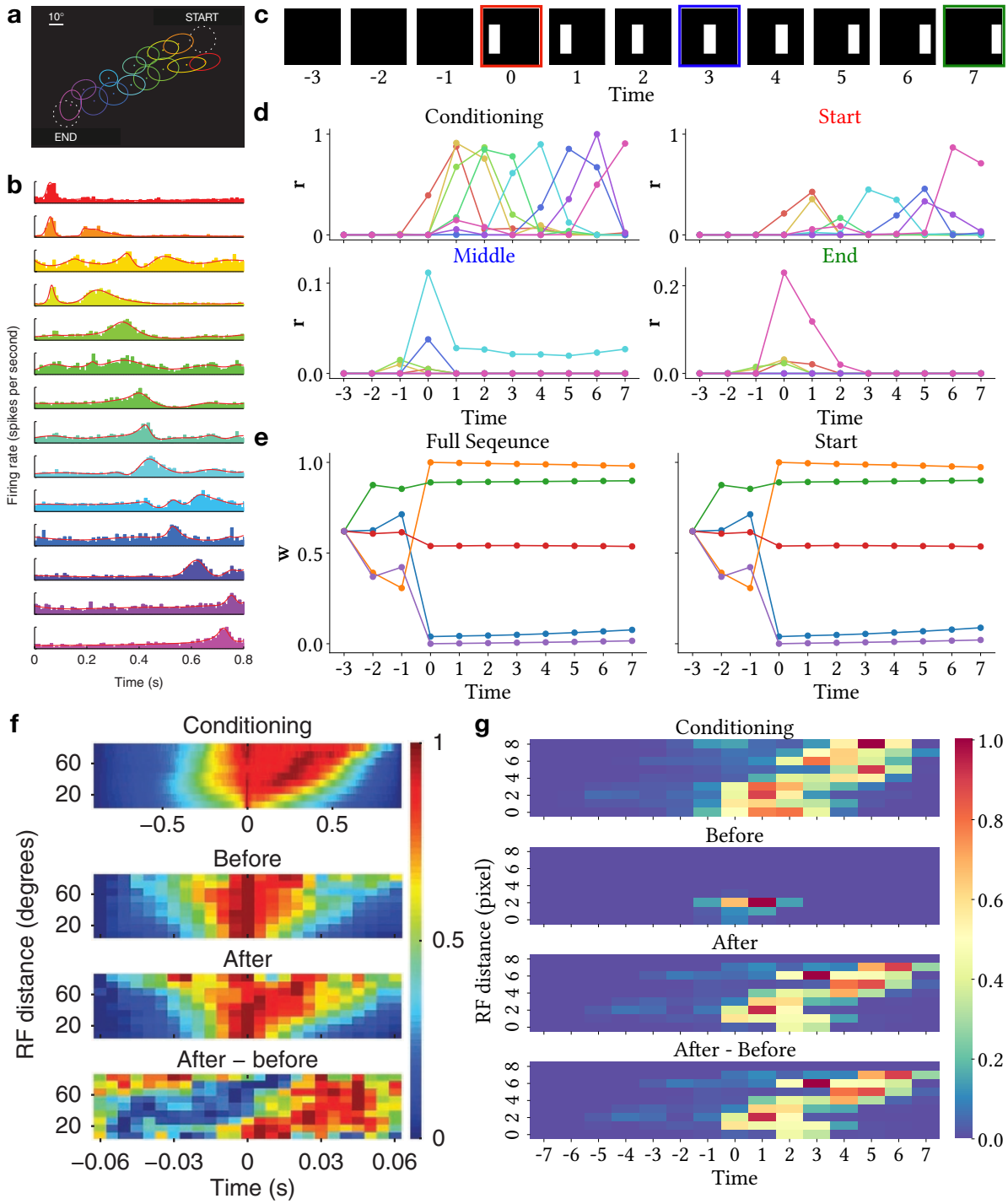

Figure 6: **Cue-triggered activity recall in the first-level network neurons. (a)** The experimental setup by Xu et al. (Adapted from [1]). A bright dot stimulus moved from START to END repeatedly during conditioning. Neurons whose receptive fields were along the dot trajectory (colored ellipses) were recorded. **(b)** Peristimulus time histograms (PSTHs) of the neurons shown in a). A sequential firing order was evoked by the moving dot (adapted from [1]). **(c)** The image sequence used to simulate conditioning and testing for our network. The first three empty frames simulate the resting period before the actual image sequence is presented. **(d)** Responses of the first-level neurons of the network. Colored lines represent the eight most active neurons of the first level. Upper left panel: neural responses during conditioning. Rest panels: testing the network with single frames (start: red box; middle: blue box; end: green box in (c)). **(e)** Transition dynamics predicted by the network after conditioning, when the full sequence (left panel) or just the starting frame (right panel) was shown. **(f, g)** Normalized pairwise cross correlation of V1 neurons **(f)** (Adapted from [1]) and the first-level network neurons **(g)**. Top: during conditioning; middle two: testing with the starting stimulus, before and after conditioning; bottom: the differences of cross correlations, after minus before conditioning.

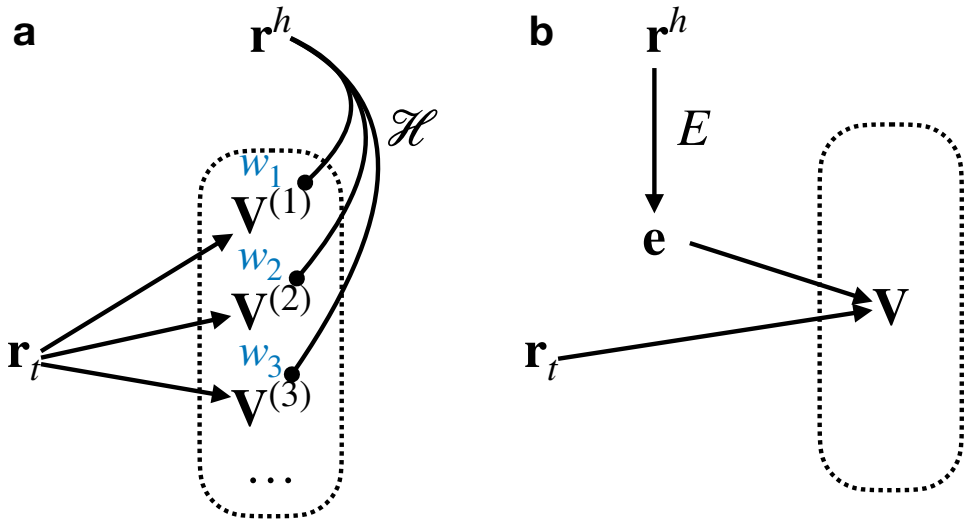

Figure 7: **Two different neural implementations of top-down modulation of recurrent activities. (a)** Hypernetwork-based approach: higher-level response vector $\mathbf{r}^h$ generates, via a hypernetwork $\mathcal{H}$, the top-down inputs $w_1, w_2, w_3, \ldots$, which modulate the outputs of lower-level recurrent networks with weights $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}, \ldots$ respectively, before being summed to produce an output (see Equations 8 and 9). **(b)** Alternate embedding-based approach: $\mathbf{r}^h$ generates a top-down embedding vector $\mathbf{e}$ through a network E, which is then given to a single lower-level recurrent network as an additional input.

level level. We found that the network had learned to recognize the starting frame of the conditioning sequence and rapidly update its second-level representation to predict the dynamics of the complete sequence (Figure 6(e) left panel). The predicted dynamics were not interrupted in the following steps when the subsequent stimuli in the sequence were omitted (due to the high precision learned for this sequence); these dynamics (captured by $\mathbf{w}$) were nearly identical to those for the complete sequence (Figure 6(e) left panel).

Lastly, we plotted the pairwise cross-correlation of the first-level model neurons as a function of their spatial RF distances when tested with the starting frame of sequence, following the analysis done by Xu et al. [1] (see Methods). As Figure 6(g) shows, the peaks of the correlation showed a clear rightward slant after conditioning, consistent with the experimental results (Figure 6(f)). This indicates a strong sequential firing order in the first-level model neurons elicited by the starting cue, where neurons farther apart have longer lags in response cross-correlations, a phenomenon that was nonexistent before conditioning (Figure 6(g)). In line with the experimental results [1], the cue-triggered activity recall also emerged when the direction of the conditioning stimuli was reversed, ruling out the effect of the stimulus dynamics on the recall (Supplementary Figure S1). To confirm the role of the temporal precision β in mediating activity recall in the first-level model neurons, we repeated the conditioning experiment but did not adapt the precision network $\mathcal{G}$ (other temporal components of the model were updated as in the previous experiment). The activity recall phenomenon disappears in this model (Supplementary Figure S2). These results demonstrate that cue-triggered activity recall can be explained by our DPC network as spatiotemporal prediction with high precision.

Discussion

Our results suggest that a dynamic predictive coding (DPC) network can learn hierarchical temporal representations from natural visual sequences. Specifically, we showed that through minimizing prediction errors on image sequences, a two-layer neural network develops V1-like separable and inseparable space-time receptive fields [23] in the first level, and representations with a longer timescale and greater stability in the second level. Such a model also offers a normative explanation for sequence learning and cue-triggered activity recall in terms of learning to predict the dynamics of a reliable and repeated sequence with high precision.

The key to the DPC model's ability to capture lower-level dynamics with relatively stable higher-level response vectors $\mathbf{r}^h$ is the use of a hypernetwork [22], which generates the weights $\mathbf{w}$ characterizing the transition dy-

namics of an entire lower level state sequence. There has been increasing interest in neuroscience in the role of modulatory inputs (*e.g.* top-down contextual information) in shaping the dynamics of recurrent neural networks in the brain [36–38]. The DPC model ascribes an important role to these modulatory inputs in enabling cortical circuits to learn temporal hierarchies. While the neural implementation in this paper used the weights $\mathbf{w}$ to modulate a set of recurrent network weights $\mathbf{V}^{(k)}$, there exists an alternate implementation that can be shown to be equivalent (but perhaps not as efficient) [39]: feeding $\mathbf{r}^h$ to an embedding network whose output is then fed as top-down input to a single lower-level recurrent network (Figure 7). This alternative strategy could be biologically implemented via top-down cortical feedback connections that target the apical dendrites of cortical pyramidal neurons, thereby modulating their recurrent activities (as suggested by, for example, [40]).

Many studies have shown an increase in temporal representation stability and response timescales as we go from lower-order to higher-order areas in the visual and other parts of the cortex [7–10, 41, 42]. We postulate that this is an emergent property of the cortex implementing a hierarchical generative model of temporal dynamics in its inputs. In the DPC model, higher-level model neurons encode the probability distribution of lower-level transition dynamics using $\mathbf{r}^h$ and $\mathbf{r}^s$ (parameterizing mean and (inverse) variance). Our simulation results rely on two implications of this generative model: (1) for lower-level sequences with roughly consistent dynamics within a window of time, the higher-level state remains stable to predict this lower-level dynamics; (2) perception corresponds to inferring a particular $\mathbf{r}^h$ to capture the input dynamics, and a particular $\mathbf{r}^s$ to predict the precision of the dynamics, both processes relying on minimizing the prediction errors.

There are several directions worthy of further study. For instance, in sequence learning, our model predicts that responses in higher cortical areas capture the transition dynamics of complete lower-level sequences and modulate lower-level responses on a longer timescale. It remains to be shown that the transition dynamics of the higher level temporal representation (which changes more slowly than lower levels) can be captured by a yet higher level representation. This formulation of temporal hierarchy allows the timescales of the representations to reflect the transition dynamics of the the environment rather than being a fixed, static quantity [15, 16, 43]. Furthermore, transitions of representations at each level could potentially implement a hierarchical segmentation of continuous events based on the the temporal scale of the representations at that level [44–46].

Another aspect to be considered is how the speed of motion in visual sequences affects learning and cue-triggered recall. Xu et al. showed that moving dot stimuli with low speed did not cause an increase in sequential firing and recall like other faster visual sequences [1]. In a related study, Gavornik and Bear [3] showed that the V1 showed stronger responses when the timing of each frame onset matches that of the training sequence, and the responses were reduced when the timing was random. An interesting future direction is to investigate how the speed of visual sequences used for training the model determines its ability to learn conditioning stimuli with different speeds.

Our results provide a demonstration of how precision signals could support activity recall in the visual cortex. Similar mechanisms have been postulated to support hippocampal replay of episodic memories [47, 48]. Such top-down precision-based modulation could be implemented by modulating the superficial pyramidal cells signaling weighted prediction errors $\beta(\mathbf{r}_t - \hat{\mathbf{r}}_t)$ where β acts as a disinhibiting multiplicative gain, generated by descending circuits from higher level (or hippocampus). Connecting theories of episodic memory formation and recall with the DPC model could be a fruitful direction to explore to understand hippocampal-neocortical interactions from a probabilistic and normative perspective.

Extending the DPC model to action-conditioned prediction and hierarchical planning is another interesting direction for future research. There is a growing body of evidence that neural activity in sensory cortex is predictive of the sensory consequences of an animal's own actions [2, 4, 35, 49, 50]. The DPC model can be augmented to incorporate actions, e.g., by augmenting the higher-level state with a higher-level action input to predict the dynamics of lower-level sensory consequences of the action. Such a model also opens the door to using probabilistic inference for hierarchical planning [18, 51–53], where actions are selected to minimize the sensory prediction errors with respect to preferred goal states (see also [54]). Such a model could also potentially be used to understand the neural basis of navigation and planning [10, 55–57] as an emergent property of prediction error minimization.

Methods

Hierarchical generative model

We assume that the observation $\mathbf{I}_t \in \mathbb{R}^M$ at time t is generated by a first-level latent variable $\mathbf{r}_t \in \mathbb{R}^N$. The latent variable $\mathbf{r}_t$ is generated by the previous step latent variable $\mathbf{r}_{t-1}$ and two second-level latent variables, $\mathbf{r}^h \in \mathbb{R}^{N_h}$ and $\mathbf{r}^s \in \mathbb{R}^{N_s}$. Together, the generative model factorizes as follows:

$$p(\mathbf{I}_{1:T}, \mathbf{r}_{1:T}, \mathbf{r}^h, \mathbf{r}^s) = p(\mathbf{r}^h)p(\mathbf{r}^s)p(\mathbf{r}_1)\prod_{t=1}^{T} p(\mathbf{I}_t|\mathbf{r}_t)\prod_{t=2}^{T} p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{r}^h, \mathbf{r}^s). \tag{3}$$

Each component of the factorization is parameterized as follows:

$$\mathbf{r}^h \sim \mathcal{N}(0, 1) \tag{4}$$

$$\mathbf{r}^s \sim \mathcal{N}(0, 1) \tag{5}$$

$$\mathbf{r}_t|(\mathbf{r}_{t-1}, \mathbf{r}^h, \mathbf{r}^s) \sim \mathcal{N}(f(\mathbf{r}^h, \mathbf{r}_{t-1}), (\mathcal{G}_\phi(\mathbf{r}^s))^{-1}I) \tag{6}$$

$$\mathbf{I}_t|\mathbf{r}_t \sim \mathcal{N}(\mathbf{U}\mathbf{r}_t, \alpha^{-1}I), \tag{7}$$

where $\mathcal{N}$ denotes the normal distribution, $\mathcal{G}_\phi$ is a function parameterized by ϕ, and I is the identity matrix. The mean $\mathbf{r}_t^\mu = f(\mathbf{r}^h, \mathbf{r}_{t-1})$ is given by:

$$\mathbf{r}_t^\mu = \mathrm{ReLU}(\mathbf{V}\mathbf{r}_{t-1}) \tag{8}$$

$$\mathbf{V} = \sum_{k=1}^{K} w_k \mathbf{V}^{(k)} \tag{9}$$

$$\mathbf{w} = \mathcal{H}_\theta(\mathbf{r}^h). \tag{10}$$

Here, $\mathcal{H}_\theta$ is a function parameterized by θ. Both $\mathcal{G}_\phi$ and $\mathcal{H}_\theta$ are implemented using neural networks in our model.

To sum up, the trainable parameters of the model include spatial filters $\mathbf{U}$, input precision α, K transition matrices $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(K)}$, and the two sets of neural network parameters θ and ϕ. The latent variables are $\mathbf{r}_{1:T}$, $\mathbf{r}^h$, and $\mathbf{r}^s$. See Supplementary Materials for a more detailed description of the model architecture.

Prediction error minimization

Here, we derive the loss function used for inference and learning under the assumed generative model. We focus on finding the *maximum a posteriori* (MAP) estimates of the latent variables using a Bayesian filtering approach. At time t, the posterior of $\mathbf{r}_t$ conditioned on the input observations up to time t, $\mathbf{I}_{1:t}$, and the higher level variables $\mathbf{r}^h, \mathbf{r}^s$ can be written as follows using Bayes' theorem:

$$p(\mathbf{r}_t|\mathbf{I}_{1:t}, \mathbf{r}^h, \mathbf{r}^s) \propto p(\mathbf{I}_t|\mathbf{r}_t)p(\mathbf{r}_t|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s), \tag{11}$$

where the first term on the right hand side is the likelihood function defined by Equation 7. The second term is the posterior of $\mathbf{r}_t$ predicted by the optimal estimate from the previous step:

$$p(\mathbf{r}_t|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s) = \int p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{r}^h, \mathbf{r}^s)p(\mathbf{r}_{t-1}|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s)\mathrm{d}\mathbf{r}_{t-1}, \tag{12}$$

where the first term inside the integral is the first level transition dynamics defined by Equation 6. The parameterization of the transition distribution is generated by the second level latent variables.

Putting Equation 11 and 12 together, we get

$$p(\mathbf{r}_t|\mathbf{I}_{1:t}, \mathbf{r}^h, \mathbf{r}^s) \propto p(\mathbf{I}_t|\mathbf{r}_t)\int p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{r}^h, \mathbf{r}^s)p(\mathbf{r}_{t-1}|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s)\mathrm{d}\mathbf{r}_{t-1}, \tag{13}$$

which defines a recursive way to infer the posterior of $\mathbf{r}_t$ at time t. In this model, we only maintain a single point (MAP) estimate of the posterior at each time step, so we simplify the posterior distribution $p(\mathbf{r}_{t-1}|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s)$ as a Dirac delta function:

$$p(\mathbf{r}_{t-1}|\mathbf{I}_{1:t-1}, \mathbf{r}^h, \mathbf{r}^s) \approx \delta(\mathbf{r}_{t-1} - \hat{\mathbf{r}}_{t-1}), \tag{14}$$

where $\hat{\mathbf{r}}_{t-1}$ is the MAP estimate from the previous step. Now we can further simplify Equation 13 as

$$p(\mathbf{r}_t|\mathbf{I}_{1:t}, \mathbf{r}^h, \mathbf{r}^s) \propto p(\mathbf{I}_t|\mathbf{r}_t)p(\mathbf{r}_t|\hat{\mathbf{r}}_{t-1}, \mathbf{r}^h, \mathbf{r}^s). \tag{15}$$

This gives the posterior of all the latent variables at time t as

$$p(\mathbf{r}_t, \mathbf{r}^h, \mathbf{r}^s|\mathbf{I}_{1:t}) \propto p(\mathbf{I}_t|\mathbf{r}_t)p(\mathbf{r}_t|\hat{\mathbf{r}}_{t-1}, \mathbf{r}^h, \mathbf{r}^s)p(\mathbf{r}^h)p(\mathbf{r}^s). \tag{16}$$

We can find the MAP estimates of the latent variables by minimizing the negative log of Equation 16. Substituting the generative assumptions (Equation 4-7), we get:

$$\beta = \mathcal{G}_\phi(\mathbf{r}^s) \tag{17}$$

$$\bar{\mathbf{r}}_t = f(\mathbf{r}^h, \hat{\mathbf{r}}_{t-1}) \tag{18}$$

$$\mathcal{L}_t = \alpha\|\mathbf{I}_t - \mathbf{U}\mathbf{r}_t\|_2^2 + \beta\|\mathbf{r}_t - \bar{\mathbf{r}}_t\|_2^2 + M\log\alpha^{-1} + N\log\beta^{-1} + \lambda\|\mathbf{r}_t\|_1 + \lambda_h\|\mathbf{r}^h\|_2^2 + \lambda_s\|\mathbf{r}^s\|_2^2, \tag{19}$$

where λ_h and λ_s are the Gaussian prior penalties for $\mathbf{r}_h$ and $\mathbf{r}_s$ respectively, and λ is an additional sparsity penalty for $\mathbf{r}_t$. Using Equation 19, we compute the MAP estimate of $\mathbf{r}_t$ at time t as

$$\hat{\mathbf{r}}_t = \arg\min_{\mathbf{r}_t} \mathcal{L}_t. \tag{20}$$

At each time step, we update the current estimate of $\mathbf{r}^h$ and $\mathbf{r}^s$ to minimize $\mathcal{L}_t$ as well:

$$\mathbf{r}^h = \arg\min_{\mathbf{r}^h} \mathcal{L}_t \tag{21}$$

$$\mathbf{r}^s = \arg\min_{\mathbf{r}^s} \mathcal{L}_t. \tag{22}$$

To begin the recursive estimation (without the temporal prediction from the previous step), we compute the MAP estimate of the first step latent variable $\mathbf{r}_1$ using the following reduced loss

$$\mathcal{L}_1 = \alpha\|\mathbf{I}_1 - \mathbf{U}\mathbf{r}_1\|_2^2 + M\log\alpha^{-1} + \lambda\|\mathbf{r}_1\|_1 \tag{23}$$

$$\hat{\mathbf{r}}_1 = \arg\min_{\mathbf{r}_1} \mathcal{L}_1. \tag{24}$$

The parameters of the model can be optimized by minimizing the same prediction errors summed across time and averaged across different sequences, using the MAP estimates of the latent variables. See Supplementary Materials for more detailed pseudocode describing the inference and learning procedure.

Data and preprocessing

For the natural video dataset, we extracted 14000 image sequences from a YouTube video (link here) recorded by a person walking on a forest trail (image size: 10×10 pixels, sequence length: 10 frames (≈ 0.35 seconds, uniformly sampled in time). The image sequences do not overlap with each other spatially or temporally. Each sequence was spatially and temporally whitened to simulate retinal and LGN processing following the methods in Olshsausen & Field [28] and Dong & Atick [30]. 10,500 sequences were used to train the model and the remaining 3,500 were reserved for testing.

For the Moving MNIST dataset [34], we used 10000 image sequences (image size: 16×16 pixels, sequence length: 10 frames), each sequence containing a fixed digit moving in a particular direction. The motion of the digits was restricted to upward, downward, leftward, or rightward direction. When a digit hit the boundary, its motion direction was inverted (leftward to rightward, upward to downward, and vice versa). No whitening procedures were performed on the MNIST sequences. 9,000 sequences were used to train the model and the remaining 1,000 were reserved for testing.

Reverse correlation for computing space-time receptive fields

The reverse correlation stimuli for deriving the space-time receptive fields (Figure 2) were extracted from the same natural video without any spatial and temporal overlapping with the training and test set. We used 50 continuous image sequences with 80,000 steps ($\approx$ 47 minutes, spanned the same time range but no spatial overlaps) and computed the space-time receptive fields (STRFs) as the firing-rate-weighted average of input frame sequences of length seven ($\approx$ 250 ms), across time and sequences.

Formally, let $\left\{\mathbf{I}_{1:T}^{(j)}\right\}_{j=1}^{J}$ be the J stimulus sequences of length T and let τ be the length of the STRFs (here, $J = 50, T = 80,000, \tau = 7$). For each neuron i, its space-time receptive field STRF_i has dimensions $M \times \tau$, where M is the dimensionality of a single image frame vector (here, $M = 100$ after vectorizing the 10×10 image frame). We compute the STRF of neuron i as follows

$$\text{STRF}_i = \frac{1}{J(T-\tau)} \sum_{j=1}^{J} \sum_{t=\tau+1}^{T} \hat{r}_{t,i}^{(j)} \mathbf{I}_{t-\tau:t-1}^{(j)}, \tag{25}$$

where $\hat{r}_{t,i}^{(j)}$ is the predicted firing rate of neuron i at time t in sequence j and $\mathbf{I}_{t-\tau:t-1}^{(j)}$ is the image sequence from time $t-\tau$ to $t-1$ in sequence j. This procedure is analogous to the calculation of the spike-triggered average widely used in neurophysiology [32]; in this case, we computed the average of input sequences weighted by the activity $\hat{r}$ caused by the sequence.

Autocorrelation for quantifying timescales

Since our model responds deterministically to the same sequence (MAP estimation), we cannot follow the exact approach of Murray et al. [7] that relies on across-trial variability to the same stimulus. We computed the autocorrelation of single neuron responses to natural videos and Gaussian white noise sequences. We averaged the single-neuron autocorrelations across the first or second level, and trials. Formally, let $r_{t,i,j}$ is the response of neuron i at time t during trial j. We computed the autocorrelation with lag k as follows:

$$\mu_{i,j} = \frac{1}{T} \sum_{t=1}^{T} r_{t,i,j} \tag{26}$$

$$\sigma_{i,j} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (r_{t,i,j} - \mu_{i,j})^2} \tag{27}$$

$$\rho_{i,j}(k) = \frac{\sum_{t=1}^{T-k} (r_{t,i.j} - \mu_{i,j})(r_{t+k,i,j} - \mu_{i,j})}{(T-k)\sigma_{i,j}\sigma_{i,j}} \quad \forall i = 1 \ldots N. \tag{28}$$

To compute the autocorrelation for an entire population at lag k, we took the average of the autocorrelations across all N neurons and J trials:

$$\rho(k) = \frac{1}{NJ} \sum_{i=1}^{N} \sum_{j=1}^{J} \rho_{i,j}(k). \tag{29}$$

For the results shown in Figure 3(b) and (c), we choose $J = 500$, $T = 50$ and computed the autocorrelation with lag $k = 0 \ldots 9$ for the first and second level neurons. White noise pixels were i.i.d. samples from $\mathcal{N}(0, 0.0075)$. We also computed the autocorrelation for both levels with natural videos selected from the same stimuli used for the reverse correlation analysis (note though with natural videos the stationary mean and variance assumption is less valid). To quantify the timescale of the autocorrelation decay, we fitted an exponential decay function $\rho_{\text{fit}}(k) = a \exp(-k/\tau) + b$ to the autocorrelation data on each level (through Scipy `optimize.curve_fit` function [58]), where a, b, and τ are fitted parameters of the function and τ represents the response timescale following the definition by Murray et al. [7].

Sequence learning and recall simulation

To simulate the sequence learning experiment of Xu et al. [1], we created an eight-step image sequence with a white bar stimulus on a black background (Figure 6(c). The image size was 10×10 pixels and the bar size was 5×2 pixels. To be consistent with the natural video data the model was trained on, we applied the same spatiotemporal whitening procedure to the moving bar sequence. Additionally, three empty frames (with uniform pixel value 0.01) were added at the beginning of the sequence to simulate the rest period ahead of conditioning.

We exposed the model trained on natural videos to the moving bar sequence repeatedly and adapted the weights of the model using the same prediction error minimization procedure used for training (Equation 19). We only adapted the temporal components of the model (transition matrices $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(K)}$, top-down connections $\mathcal{H}_\theta$ and $\mathcal{G}_\phi$) and kept the spatial components (spatial filters $\mathbf{U}$, spatial precision α) fixed. For the model without precision adaptation, the precision network $\mathcal{G}_\phi$ was kept fixed as well. The model was adapted for 200 exposures to the bar sequence (200 gradient updates) but because the model developed the recall patterns after 70 steps, we used the model after 75 steps for the visual recall simulations.

We simulated the visual recall condition as follows. The initial three empty frames remained ($t = -3, -2, -1$) as the rest period, followed by an one-step display of a visual cue: the starting frame ($t = 0$, red), the middle frame ($t = 3$, blue), or the end frame ($t = 7$, green) (see Figure 6(c). The visual cue was followed by seven steps of empty frames (same as the ones in the initial rest period). The cross correlation plot was computed following the same procedure as the one described in Xu et al. [1].

Data Availability

Code for reproducing all simulations in the paper will be published on GitHub upon acceptance.

ACKNOWLEDGMENTS

We thank Dimitrios Gklezakos for his help with hypernetworks in an earlier formulation of the model, and Ares Fisher for his feedback on neuroscience-related aspects. LPJ also thanks Jieyu Zhang, Sirui Lu, and Weijia Shi for various inspiring discussions. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0021, a Weill Neurohub Investigator award, a grant from the Templeton World Charity Foundation, and a Cherng Jia & Elizabeth Yun Hwang Professorship to RPNR. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the funders.

References

1. Xu, S., Jiang, W., Poo, M.-m. & Dan, Y. Activity recall in a visual cortical ensemble. *Nature Neuroscience* **15,** 449–455 (Mar. 2012).
2. Keller, G. B., Bonhoeffer, T. & Hübener, M. Sensorimotor Mismatch Signals in Primary Visual Cortex of the Behaving Mouse. *Neuron* **74,** 809–815 (June 2012).
3. Gavornik, J. P. & Bear, M. F. Learned spatiotemporal sequence recognition and prediction in primary visual cortex. *Nature Neuroscience* **17,** 732–737 (May 2014).
4. Fiser, A. *et al.* Experience-dependent spatial expectations in mouse visual cortex. *Nature Neuroscience* **19,** 1658–1664 (Dec. 2016).
5. Schneider, D. M., Sundararajan, J. & Mooney, R. A cortical filter that learns to suppress the acoustic consequences of movement. *Nature* **561,** 391–395 (Sept. 2018).
6. Gillon, C. J. *et al.* Learning from unexpected events in the neocortical microcircuit. *bioRxiv,* 2021.01.15.426915 (Jan. 2021).

7. Murray, J. D. *et al.* A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience* **17,** 1661–1663 (Dec. 2014).

8. Runyan, C. A., Piasini, E., Panzeri, S. & Harvey, C. D. Distinct timescales of population coding across cortex. *Nature* **548,** 92–96 (Aug. 2017).

9. Siegle, J. H. *et al.* Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature* **592,** 86–92 (Apr. 2021).

10. Brunec, I. K. & Momennejad, I. Predictive Representations in Hippocampal and Prefrontal Hierarchies. *Journal of Neuroscience* **42,** 299–312 (Jan. 2022).

11. Srinivasan, M. V., Laughlin, S. B. & Dubs, A. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **216,** 427–459 (1982).

12. Dong, D. W. & Atick, J. J. Statistics of natural time-varying images. *Network: Computation in Neural Systems* **6,** 345–358 (Jan. 1995).

13. Olshausen, B. A. Sparse coding of time-varying natural images. *Journal of Vision* **2,** 130–130 (Nov. 2002).

14. Van Hateren, J. H. & Ruderman, D. L. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences* **265,** 2315–2320 (Dec. 1998).

15. Singer, Y. *et al.* Sensory cortex is optimized for prediction of future input. *eLife* **7** (eds Gallant, J. L. & Kastner, S.) e31557 (June 2018).

16. Singer, Y., Willmore, B. D. B., King, A. J. & Harper, N. S. *Hierarchical temporal prediction captures motion processing from retina to higher visual cortex* preprint (Neuroscience, Mar. 2019).

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

18. Friston, K. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience* **11,** 127–138 (Feb. 2010).

19. Huang, Y. & Rao, R. P. N. Predictive coding. *WIREs Cognitive Science* **2,** 580–593 (2011).

20. Bharioke, A. & Chklovskii, D. B. Automatic Adaptation to Fast Input Changes in a Time-Invariant Neural Circuit. *PLOS Computational Biology* **11,** e1004315 (Aug. 2015).

21. Friston, K. Does predictive coding have a future? *Nature Neuroscience* **21,** 1019–1021 (Aug. 2018).

22. Ha, D., Dai, A. M. & Le, Q. V. *HyperNetworks* in *5th international conference on learning representations, ICLR 2017, toulon, france, april 24-26, 2017, conference track proceedings* (OpenReview.net, 2017).

23. DeAngelis, G. C., Ohzawa, I. & Freeman, R. D. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences* **18,** 451–458 (Oct. 1995).

24. Ekman, M., Kok, P. & de Lange, F. P. Time-compressed preplay of anticipated events in human primary visual cortex. *Nature Communications* **8,** 15276 (May 2017).

25. Bang, J. W., Sasaki, Y., Watanabe, T. & Rahnev, D. Feature-Specific Awake Reactivation in Human V1 after Visual Training. *Journal of Neuroscience* **38,** 9648–9657 (Nov. 2018).

26. Lu, J., Luo, L., Wang, Q., Fang, F. & Chen, N. Cue-triggered activity replay in human early visual cortex. *Science China Life Sciences* **64,** 144–151 (Jan. 2021).

27. Eagleman, S. L. & Dragoi, V. Image sequence reactivation in awake V4 networks. *Proceedings of the National Academy of Sciences* **109,** 19450–19455 (Nov. 2012).

28. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381,** 607–609 (June 1996).

29. Rao, R. P. An optimal estimation approach to visual perception and learning. *Vision Research* **39,** 1963–1989 (June 1999).

30. Dong, D. W. & Atick, J. J. Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in neural systems* **6,** 159–178 (1995).

31. Hubel, D. H. & Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology* **148,** 574–591 (Oct. 1959).

32. Ringach, D. & Shapley, R. Reverse correlation in neurophysiology. *Cognitive Science* **28,** 147–166 (Mar. 2004).

33. Talebi, V. & Baker, C. L. Natural versus Synthetic Stimuli for Estimating Receptive Field Models: A Comparison of Predictive Robustness. *Journal of Neuroscience* **32,** 1560–1576 (Feb. 2012).

34. Srivastava, N., Mansimov, E. & Salakhudinov, R. *Unsupervised Learning of Video Representations using LSTMs* in *Proceedings of the 32nd International Conference on Machine Learning* (PMLR, June 2015), 843–852.

35. Jordan, R. & Keller, G. B. Opposing Influence of Top-down and Bottom-up Input on Excitatory Layer 2/3 Neurons in Mouse Primary Visual Cortex. *Neuron* **108,** 1194–1206.e5 (Dec. 2020).

36. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503,** 78–84 (Nov. 2013).

37. Stroud, J. P., Porter, M. A., Hennequin, G. & Vogels, T. P. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience* **21,** 1774–1783 (Dec. 2018).

38. Masse, N. Y., Grant, G. D. & Freedman, D. J. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences* **115,** E10467–E10475 (Oct. 2018).

39. Galanti, T. & Wolf, L. *On the Modularity of Hypernetworks* in *Advances in Neural Information Processing Systems* **33** (Curran Associates, Inc., 2020), 10409–10419.

40. Crick, F. & Koch, C. Constraints on cortical and thalamic projections: the no-strong-loops hypothesis. *Nature* **391,** 245–250 (Jan. 1998).

41. Piasini, E. *et al.* Temporal stability of stimulus representation increases along rodent visual cortical hierarchies. *Nature Communications* **12,** 4448 (July 2021).

42. Henin, S. *et al.* Learning hierarchical sequence representations across human cortex and hippocampus. *Science Advances* **7** (Feb. 2021).

43. Heeger, D. J. Theory of cortical function. *Proceedings of the National Academy of Sciences* **114,** 1773–1782 (Feb. 2017).

44. Zacks, J. M. *et al.* Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience* **4,** 651–655 (June 2001).

45. The Hippocampal Film Editor: Sensitivity and Specificity to Event Boundaries in Continuous Experience. *Journal of Neuroscience* **38** (eds Ben-Yakov, A. & Henson, R. N.) 10057–10068 (Nov. 2018).

46. Teşileanu, T., Golkar, S., Nasiri, S., Sengupta, A. M. & Chklovskii, D. B. Neural Circuits for Dynamics-Based Segmentation of Time Series. *Neural Computation* **34,** 891–938 (Mar. 2022).

47. Foster, D. J. Replay Comes of Age. *Annual Review of Neuroscience* **40,** 581–602 (2017).

48. Barron, H. C., Auksztulewicz, R. & Friston, K. Prediction and memory: A predictive coding account. *Progress in Neurobiology* **192,** 101821 (Sept. 2020).

49. Keller, G. B. & Mrsic-Flogel, T. D. Predictive Processing: A Canonical Cortical Computation. *Neuron* **100,** 424–435 (Oct. 2018).

50. Zmarz, P. & Keller, G. B. Mismatch Receptive Fields in Mouse Visual Cortex. *Neuron* **92,** 766–772 (Nov. 2016).

51. Attias, H. *Planning by Probabilistic Inference* in *International Workshop on Artificial Intelligence and Statistics* (PMLR, Jan. 2003), 9–16.

52. Verma, D. & Rao, R. P. Goal-Based Imitation as Probabilistic Inference over Graphical Models. *Advances in Neural Information Processing Systems* **18** (2005).

53. Botvinick, M. & Toussaint, M. Planning as inference. *Trends in Cognitive Sciences* **16,** 485–488 (Oct. 2012).

54. Gklezakos, D. C. & Rao, R. P. N. *Active Predictive Coding Networks: A Neural Solution to the Problem of Learning Reference Frames and Part-Whole Hierarchies* tech. rep. arXiv:2201.08813 (arXiv, Jan. 2022).

55. Momennejad, I. Learning Structures: Predictive Representations, Replay, and Generalization. *Current Opinion in Behavioral Sciences. Understanding memory: Which level of analysis?* **32,** 155–166 (Apr. 2020).

56. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nature Neuroscience* **20,** 1643–1653 (Nov. 2017).

57. Papadimitriou, C. H., Vempala, S. S., Mitropolsky, D., Collins, M. & Maass, W. Brain computation by assemblies of neurons. *Proceedings of the National Academy of Sciences* **117,** 14464–14472 (June 2020).

58. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17,** 261–272 (Mar. 2020).

59. Ba, J. L., Kiros, J. R. & Hinton, G. E. *Layer Normalization* tech. rep. arXiv:1607.06450 (arXiv, July 2016).

60. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. *Fast and accurate deep network learning by exponential linear units (ELUs)* in *4th international conference on learning representations, ICLR 2016, San Juan, Puerto Rico, may 2-4, 2016, conference track proceedings* (eds Bengio, Y. & LeCun, Y.) (2016).

Supplementary Information

Cue-triggered activity recall with reversed direction image sequence

We performed the cue-triggered activity recall experiment in Figure 6 in the Main Text, but with the direction of the moving bar sequence reversed. After the same conditioning procedure as used for the original moving bar sequence (105 repetitions of the sequence shown in Figure S1(a)), the network was tested with the starting frame only (red box), the middle frame only (blue box), and the end frame only (green box) followed by seven empty frames. We found that the first-level model neurons had developed cue-triggered activity recall given only the start frame of the sequence (Figure S1(b) upper right panel) and not with the middle and end frames (Figure S1(b) lower panels). The second-level representation also converged to predict the dynamics of the complete sequence upon presentation of the start frame (Figure S1(c)). Lastly, the peaks of the pairwise cross-correlation showed a clear rightward slant after conditioning, indicating a strong sequential firing order in the first-level model neurons elicited by the starting cue [1]. Together with the results in the Main Text (Figure 6), these results show that the dynamic predictive coding (DPC) model exhibits cue-triggered activity recall regardless of the specificity of the input dynamics, similar to cue-triggered activity recall observed in the visual cortex [1, 24–27].

Temporal precision supports activity recall in first-level model neurons

To investigate the role of the temporal precision β in mediating activity recall in the first-level model neurons, we repeated the conditioning experiment described in the Main Text but did not adapt the precision network $\mathcal{G}$ (other temporal components of the model were updated as in the previous experiment). We refer to this model as the "β-unconditioned model". Figure S2(a) compares the temporal precision during testing predicted by the β-unconditioned network versus the conditioned network. The former's predicted precision is roughly an order of magnitude lower than that predicted by the conditioned network. For the β-unconditioned network, a decrease in precision occurred at $t = 1$ when the second frame of the training sequence was predicted but the actual input was blank, inducing a large prediction error. For the β-unconditioned network, first-level activity recall did not occur when the starting stimulus was shown (Figure S2(b) top panel) and the predicted dynamics $\mathbf{w}$ was different (Figure S2(b) bottom panel) compared to the conditioned $\mathbf{w}$ (see Figure 6(e) right panel in Main Text). The pairwise cross-correlation also did not show a rightward slant (Figure S2(c)) as seen in the conditioned network and rat V1 [1]. These results demonstrate the role of top-down precision in supporting sequence recall when input stimuli are absent – since conditioning involved a highly reliable sequence of stimuli, the network generates a high precision signal given the starting frame and the learned internal model's prediction dominates state estimation and suppresses the input prediction errors caused by missing observations.

Summary of the DPC generative model

Table S1 summarizes the parameters and values used in the DPC generative model. The top-down dynamics prediction network $\mathcal{H}$ (a hypernetwork) and the precision network $\mathcal{G}$ are multi-layer perceptrons (MLPs) and have the same architecture (except the output dimensions) as follows. The first layer of the MLP contains 10 hidden neurons, followed by a LayerNorm layer [59] and an ELU activation function [60]. The last two layers of the MLP are comprised of (1) a linear layer with 10 hidden neurons, and (2) an output layer that maps this hidden activity to $\mathbf{w} \in \mathbb{R}^K$ ($\mathcal{H}$ network) or $\beta \in \mathbb{R}$ ($\mathcal{G}$ network).

Inference and learning

Table S2 summarizes the optimizers and learning rates used for inference and learning. Algorithm 1 describes the inference (finding the MAP estimates of the latent variables) and learning (parameter estimation) process.

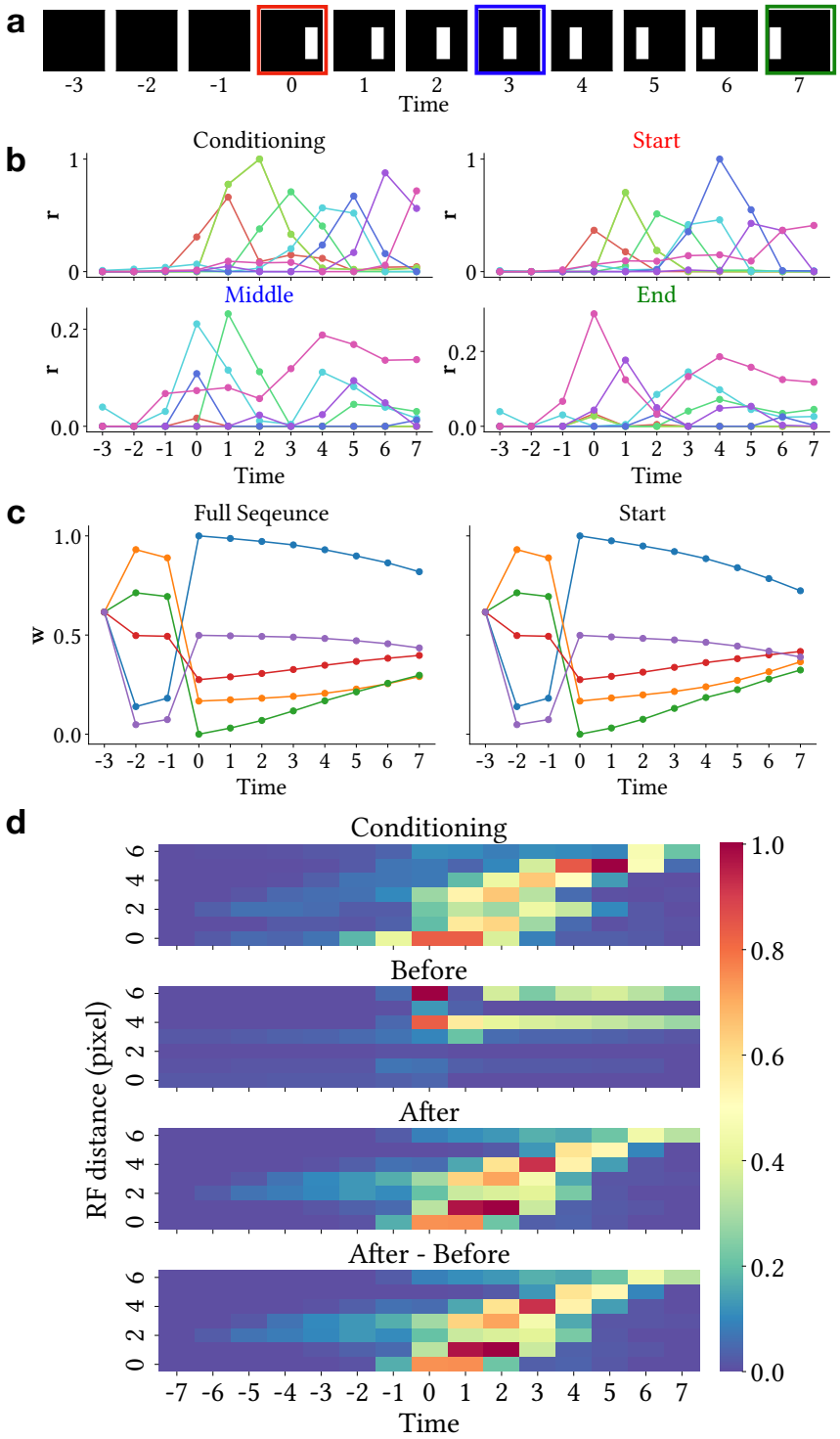

Figure S1: **Cue-triggered activity recall with reversed direction image sequence. (a)** The reversed image sequence (compared to the sequence in the Main Text) used for conditioning and testing the DPC network. The first three empty frames simulate the resting period before the actual image sequence is presented. **(b)** Responses of the first-level neurons of the network. Colored lines represent the eight most active neurons of the first level. Upper left panel: neural responses during conditioning. Rest panels: testing the network with single frames (start: red box; middle: blue box; end: green box in (a)).**(c)** Transition dynamics predicted by the network after conditioning, when the full sequence (left panel) or just the starting frame (right panel) was shown. **(d)** Normalized pairwise cross correlation between first-level network neurons. Top panel: during conditioning; middle two: testing with the starting stimulus, before and after conditioning; bottom: difference in cross correlations (after conditioning minus before conditioning).

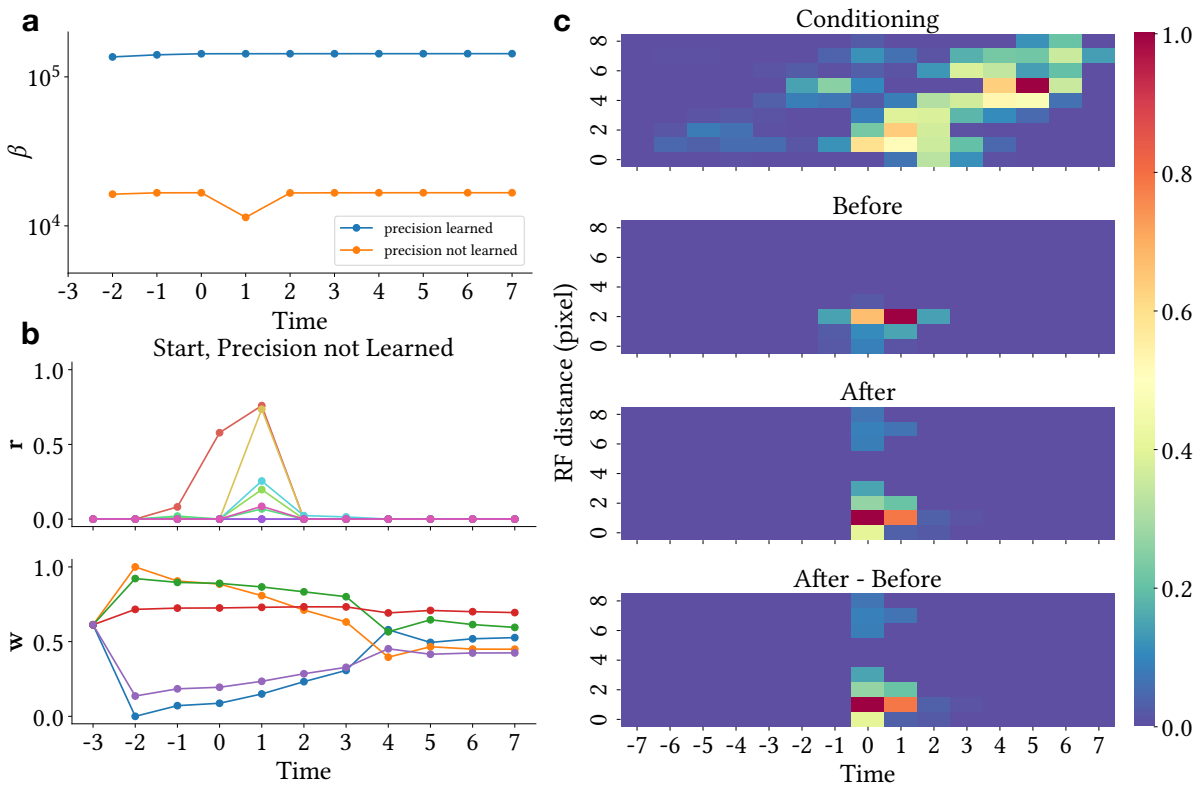

Figure S2: **Activity recall disappears when precision is not learned**. **(a)** Comparison of the predicted precision values for the conditioning sequence during testing when precision is learned versus not learned. **(b)** No sequential firing pattern (after presentation of the start stimulus) is observed in the first-level neurons when precision is not learned (top). The second-level neurons do not predict the transition dynamics for the full sequence when input stimuli after the start stimulus are omitted (bottom). **(c)** No clear rightward-slant in the peaks of the pairwise cross-correlation is observed after conditioning.

Parameter	Description	Value (Natural/MNIST)
$\mathbf{r}_t \in \mathbb{R}^N \quad \forall t = 1 \dots T$	First-level latent variable from time 1 to T	$N = 200/512$
$\mathbf{r}^h \in \mathbb{R}^{N_h}$	Second-level latent variable	$N_h = 20/20$
$\mathbf{r}^s \in \mathbb{R}^{N_s}$	Second-level latent variable	$N_s = 20/20$
$\mathbf{I}_t \in \mathbb{R}^M \quad \forall t = 1 \dots T$	Input observations from time 1 to T	$M = 100/256$
$\mathbf{U} \in \mathbb{R}^{M \times N}$	Learnable spatial filters	See text
$\alpha \in \mathbb{R}$	Spatial precision	Real number/Real number
K	Number of transition matrices	$K = 5/5$
$\mathbf{V}^{(k)} \in \mathbb{R}^{N \times N} \quad \forall k = 1 \dots K$	Learnable transition matrices	See text
$\mathcal{H}$	Top-down dynamics network	See text
$\mathcal{G}$	Top-down precision network	See text
$\mathbf{w} = \mathcal{H}_\theta(\mathbf{r}^h) \in \mathbb{R}^K$	Predicted transition dynamics	See text
$\beta = \mathcal{G}_\phi(\mathbf{r}^s) \in \mathbb{R}$	Predicted transition precision	Real number/Real number
λ	Sparsity penalty for $\mathbf{r}_t$	$5e-4/1e-3$
λ_h	Prior penalty for $\mathbf{r}^h$	$5e-6/5e-6$
λ_s	Prior penalty for $\mathbf{r}^s$	$5e-6/5e-6$

Table S1: DPC generative model parameters and values.

Optimizee	Optimizer (learning rate: Natural/MNIST)
	Inference
$\mathbf{r}_t \quad \forall t = 1 \dots T$	SGD (12/15)
$\mathbf{r}^h$	ADAM (1e − 3/1e − 3)
$\mathbf{r}^s$	ADAM (1e − 3/1e − 3)
	Learning
$\mathbf{U}$	SGD (20/5)
α	ADAM (5e − 2/5e − 2)
$\mathbf{V}^{(k)} \in \mathbb{R}^{N \times N} \quad \forall k = 1 \dots K$	ADAM (1e − 3/1e − 3)
$\mathcal{H}_\theta$	ADAM (1e − 3/1e − 3)
$\mathcal{G}_\phi$	ADAM (1e − 3/1e − 3)

Table S2: Optimizers and learning rates used for inference and learning.

Algorithm 1 Inference & learning process

Input: Image sequences dataset $\mathcal{D}$
Parameters: $\mathbf{U}, \alpha, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta, \phi$

while the parameters have not converged **do**
  Sample minibatch $\{\mathbf{I}_{1:T}^{(m)}\}_{m=1}^{M}$ from dataset $\mathcal{D}$
  Initialize total loss $\mathcal{L} = 0$
  `// Inference`
  for each sequence $\mathbf{I}_{1:T}$ in the minibatch **do**
    Initialize $\mathbf{r}_1, \mathbf{r}^h, \mathbf{r}^s$ as zero vectors
    $\hat{\mathbf{r}}_1 = \arg\min_{\mathbf{r}_1} \mathcal{L}_1(\mathbf{r}_1, \alpha, \mathbf{U}, \mathbf{I}_1)$
    Update total loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_1(\hat{\mathbf{r}}_1, \alpha, \mathbf{U}, \mathbf{I}_1)$
    for $t = 2 \dots T$ **do**
      Initialize $\mathbf{r}_t$ as a zero vector
      $\hat{\mathbf{r}}_t = \arg\min_{\mathbf{r}_t} \mathcal{L}_t(\mathbf{r}_t, \hat{\mathbf{r}}_{t-1}, \hat{\mathbf{r}}_h, \hat{\mathbf{r}}_s, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta, \phi, \mathbf{I}_t)$
      $\hat{\mathbf{r}}^h, \hat{\mathbf{r}}^s = \arg\min_{\mathbf{r}^h, \mathbf{r}^s} \mathcal{L}_t(\hat{\mathbf{r}}_t, \hat{\mathbf{r}}_{t-1}, \mathbf{r}_h, \mathbf{r}_s, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta, \phi, \mathbf{I}_t)$
      Update total loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_t(\hat{\mathbf{r}}_t, \hat{\mathbf{r}}_{t-1}, \hat{\mathbf{r}}_h, \hat{\mathbf{r}}_s, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta, \phi, \mathbf{I}_t)$
    end for
  end for
  `// Learning`
  Update parameters using gradients calculated from $\frac{1}{M}\mathcal{L}$ and their corresponding optimizers from Table S2
end whilereturn $\mathbf{U}, \alpha, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta, \phi$