Predictive learning enables compositional representations

Gauthier Boeshertz[1] and Claudia Clopath[1]

[1]Department of Bioengineering, Imperial College London

Abstract

The brain builds predictive models to plan future actions. These models generalize remarkably well to new environments, but it is unclear how neural circuits acquire this flexibility. Compositional representations, which have been observed in the brain, could explain this adaptability. They enable a process called compositional generalization, where independent modules performing different computations can be selected to perform novel composite tasks. In this work, we show that compositional representations emerge in recurrent neural networks (RNNs) trained solely to predict future sensory inputs. We trained an RNN to predict frames in a visual environment defined by independent latent factors and their corresponding dynamics. We found that the network learned to solve this task by developing a compositional model. Specifically, it had disentangled representations of the latent factors, and formed distinct, modular clusters, each implementing a single dynamic. The network autonomously selected which cluster to use according to the sensory inputs, without task labels, using competitive dynamics between clusters. This modular and disentangled architecture enabled the network to perform compositional generalization, accurately predicting outcomes in novel contexts composed of unseen combinations of dynamics. Our findings explain how an unsupervised mechanism can learn the modular causal structure of an environment in a compositional code.

Significance

The brain can function in environments it has never been in before, an ability called generalization. In our study, we show that when a neural network model is trained to predict future observations, it will develop a modular structure that enables the brain to do one type of generalization, called compositional generalization, where known dynamics can be recomposed in new ways. Indeed, the network develops clusters that each perform a computation representing a dynamic. Importantly, it can recombine these clusters according to the input it is receiving, with no need for other clues. We also show that the network has

distinct clusters of neurons coding for each latent of the environment, consistent with what has been observed experimentally.

1 Introduction

To anticipate the consequences of actions or events and plan accordingly, the brain is widely believed to employ world models [10, 34, 40]. This principle is fundamental to motor control, where the cerebellum is hypothesized to predict sensory outcomes to refine movements [14, 31], and to sensory processing, where predictive coding theories posit that higher-level cortical areas send predictions to lower-level areas to be compared against incoming sensory data [2, 8, 25, 37]. These models generalize remarkably well to new environments, but it is not clear how. Here, we show that predictive learning naturally leads to a type of generalization called compositional generalization [16, 40].

Compositional generalization is the ability to solve new tasks by selectively using and combining learned representations encoding elementary parts of states and rules of an environment. It is thought that the brain operates on this principle [12, 27, 30, 32, 33]. For example, prefrontal networks integrate task rules with sensory inputs [27], motor systems stitch primitives into sequences [33], and hippocampo-cortical loops recombine relational structure to support inference and generalization [30]. To explain how compositionality could arise in the prefrontal cortex, recent work has demonstrated that RNNs trained with multi-task learning can develop compositional solutions, spontaneously organizing into modular subnetworks that handle distinct, elementary computations [5, 36, 41]. This modular architecture facilitates efficient generalization, as novel tasks can be addressed by reassembling these learned components. While this paradigm is a powerful model for goal-directed learning, it is less clear how it applies to processing in sensory systems, which don't have access to many tasks and labels.

Recent work has shown that an RNN trained with a predictive objective in an environment governed by a single latent variable learns to represent that latent in its hidden state activity [26]. However, the type of representations that would emerge in a rich environment, governed by multiple factors and dynamics, is not clear. We hypothesized that for an RNN to successfully predict future observations in such a composite world, it must implicitly learn the underlying causal structure. Therefore, training with a purely predictive objective should drive the emergence of a modular and compositional internal representation, where distinct latent factors and their dynamics are segregated in the network's state space.

To test this hypothesis, we adapted a task used in [33] for our purposes of sequential predictive learning. [33] trained monkeys to draw a collection of shapes with different scales on a screen. We implemented this task with a dataset in which coloured shapes appear at multiple screen positions and at any temporal position in a sequence. We can probe for the invariance of neurons coding for the shape to the other latents, a property also known as disentanglement. To assess compositional generalization, we left some compositions of dynamics that

change the latents out of the training data, and tested the trained network on them.

We trained an RNN with predictive learning in this environment. We found that the network learned to disentangle the latents, and that a modular structure comparable to that achieved through multi-task learning, whose organization did not require explicit task or context labels. Both of these properties enabled the network to generalize compositionally.

2 Results

To investigate how predictive learning enables (Figure **??** A) compositional representations, we constructed a synthetic dataset of videos (Figure **??** B), in which each frame is uniquely determined by three discrete independent latent variables, which encode the object's shape (z_{shape}), colour (z_{colour}), and position ($z_{position}$) as their index in their respective list of possible values. We generated sequences from these images by applying dynamics that independently change one latent. Specifically, each sequence is associated with a context, defined as a tuple of offsets $(\Delta z_{\text{shape}}, \Delta z_{\text{colour}}, \Delta z_{\text{position}})$ which specifies how the corresponding latents evolve across time steps. Their composition forms the context of the sequence. For example, the context $(3, 1, 2)$ increments the shape index by 3, the colour by 1, and the position by 2 (Figure 1 C). We used 6 dynamics for the shapes, and 3 for both colour and position. This resulted in 54 context, which we split into 43 training contexts and 11 generalization contexts. The generalization split has compositions not seen during training, but their individual components were. If the network performs as well on the generalization as it does on the training set, we say it generalizes compositionally.

The whole network is made of a convolutional encoder that processes each image individually; the encodings are then sequentially given to the RNN, which will predict the dynamics. Its activity is fed to a deconvolutional decoder, which reconstructs the image. Importantly, it does not receive any information on the current context, but the network can infer it by integrating two consecutive inputs. The network is trained to output images directly, we do not impose any constraints beyond regularization on weights and activations. This means that the network is free to represent the inputs in any manner. However, certain representations could make it easier for the network to have compositional representations. In particular, having disentangled representations with respect to the latents and having clustered networks that perform the dynamics may help the network capture the underlying structure of the data. In the following, we will show how the predictive model exhibits both properties. We will compare it to an autoencoder, a network with the same structure but trained to output the last observation it received.

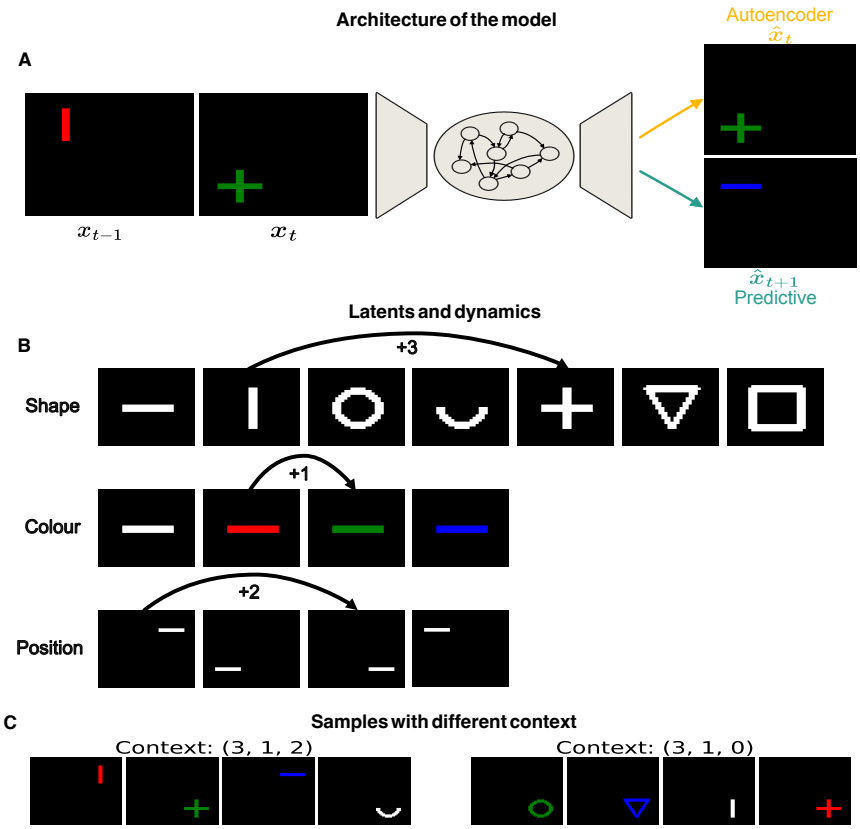

Figure 1: **Training a network to predict the next observation. A** The predictive model in green outputs what will be the next observation, whereas the autoencoder outputs the last image it saw. **B** We defined a number of shapes, colours, and positions. On top, we defined dynamics, shown by the arrows, that each increment by some value the index its associated latent in its list of possible values. This list are circular so the addition goes around the end to the beginning. The dynamics are fixed per sequence. The composition of these dynamics form a context, we show the effect of context (3,1,2). **C** Example of sequences with different contexts.

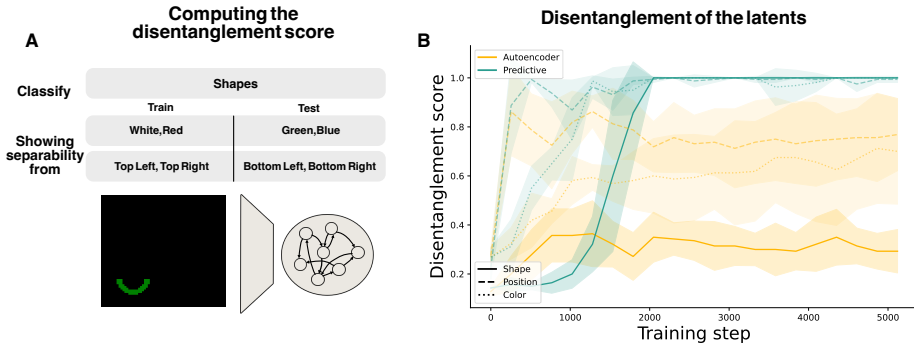

Figure 2: **The predictive model perfectly disentangles the three latents.** **A** Computing the disentanglement of the shape latent. We split the support of the two other latent in half, then encode the images of both splits. We then train a classifier and the test performance is the disentanglement score. **B** Disentanglement performance of the predictive model and autoencoder. The predictive model disentangles the three latents, while the representations of the autoencoder are entangled.

2.1 The predictive model disentangles the latents

Encoding information about the different latents in separate neurons can be beneficial for having compositional dynamics, as it allows for simpler dynamics. When each latent is represented independently, the network can modify one of them without perturbing others, allowing learning to proceed more efficiently. In contrast, if multiple latents are entangled within the same neurons, a change to one latent will affect the representation of others, requiring entangled dynamics to adjust these side effects. This would limit the ability to generalize to unseen combinations of latents.

To measure the level of separability at the single image level, we modified a metric introduced by [1, 13]. We split the support of every other latent in half to create disjoint training and testing data. We then fed the frames from both splits to the model and fitted a linear classifier on the activity of the RNN. We used its accuracy on the test set as the measure of disentanglement. A very entangled model ties the representations of different latents; this means that the values of these latents not included in the training set will affect the representations of the latent we are interested in, and may negatively impact the accuracy of the classifier (Figure 2 A).

We computed this metric for the three latents for both the autoencoder and the predictive model in (Figure 2 B). The predictive model disentangles perfectly the three latents, whereas the autoencoder is slightly above chance for the shape and is not perfect for the two other latents. This is a strong feature for predictive learning. We ran an extensive hyperparameter grid search on the size of the RNN, and the regularization on both the weights of the whole network

and activity following [38], and never found a model that could disentangle the shape.

2.2 The predictive model has clustered dynamics representations

Having distinct populations that each compute one dynamic could help the network be compositional, as it enables it to choose which population to use according to the information it is integrating. In contrast, there are two other ways to solve our task that could work but would not lead to compositional representations. The first is having a cluster for each context encountered during training, the second is having a single, large population in which all dynamics are intermixed. In the first case, the network effectively memorizes context-specific solutions rather than learning reusable components, which limits generalization to unseen contexts. In the second, the lack of specialization forces all computations to be shared across latents and contexts, making it difficult to isolate and recombine specific dynamics. While both approaches could achieve low training error, neither would produce the compositional structure that would generalize to unseen contexts.

To test the compositional properties of the network and assess which of the solutions explained above the network adopted, we measured the degree of clustering of the representations using a method from [41]. Specifically, we first computed the normalized variance of each neuron across contexts, which serves as a measure of context selectivity. These selectivity profiles were then partitioned into clusters, with the number of clusters chosen to best explain the data. We computed the number of clusters for the predictive model (Figure 3 A) and for the autoencoder (Figure 3 C) and sorted them according to the latent they affected. We found that the predictive model had clear clusters that each perform one dynamic. The autoencoder did not have clusters because it did not need to compute any dynamics. To understand the causal role of the clusters, we computed (Figure 3 B) the difference in loss when lesioning one cluster of neurons from the original loss. The differences are similar to the selectivity measure, confirming that a cluster is only active when its selected context is active. Furthermore, removing a cluster from the network impacts only the aspect of the object related to the dynamic computed by the cluster (Figure 3 E). For example, cluster 4 changes the shape of the object; lesioning it affects only the shape of the object in the output. Indeed, the position and colour are both correct, but the shape looks like the mean of all shapes.

2.3 Mechanism of cluster selection

Having established that the predictive model develops clustered dynamics, we investigated the neural mechanisms that enable these properties. First, we found that the network employs a selective activation mechanism. When processing an input sequence, only the clusters selective to the sequence's context show significant activity, while the others show very little activity, if any (Figure 4

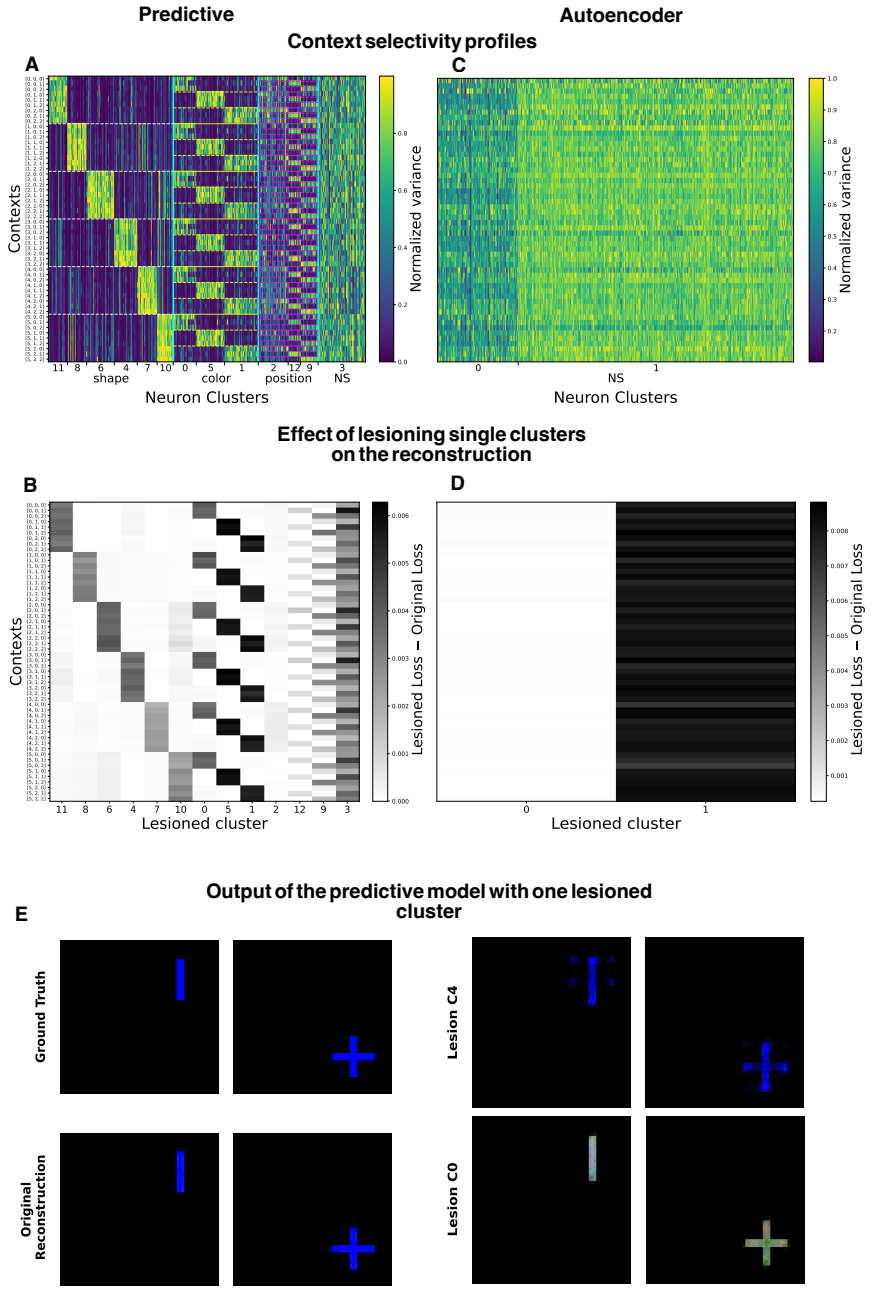

Figure 3: **Clustered dynamics in the predictive model. A** Sorted context selectivity of the predictive model. Each dynamic has its own cluster that represents it. We marked each group of clusters by the latent it affects, and NS for non-selective to a cluster that was not selective to any cluster. **B** Effect of lesioning a cluster on the output of the predictive model, measured by the difference in loss between the lesioned model and the original. The effect of lesioning a cluster is mostly constrained to the context it is selective to. **C** Same as **A** but for the autoencoder. There are only 2 active clusters. **D** Same as **B** for the autoencoder, where lesioning cluster 1 affects every context. **E** Outputs of the predictive model when lesioning the relevant cluster to the context, here context $[3, 0, 2]$. The lesion in cluster 5, which computes the dynamics of the shape, results in a degenerate shape that resembles the mean of all shapes. For cluster 7, which computes the colour of the object, the effect is also isolated to the colour.

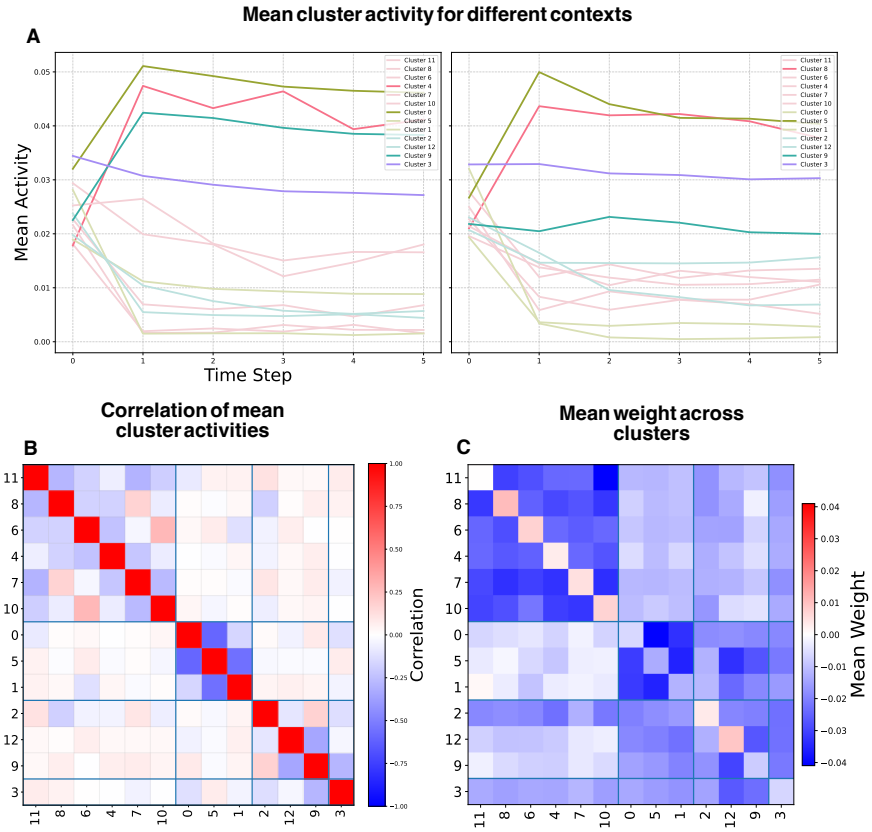

Figure 4: **Analysis of the activity and weights of the predictive model.** **A** Example of the mean activity for inputs of a context. The most active clusters for each dynamics are the ones selective to the context. **B** Correlation of the activity of each cluster for every input. Clusters performing dynamics for one latent are anti-correlated, whereas they have almost no correlation across latents. **C** Mean weight across clusters and intra-cluster. The weights between clusters coding for dynamics of one latent are more negative than those of clusters across latents.

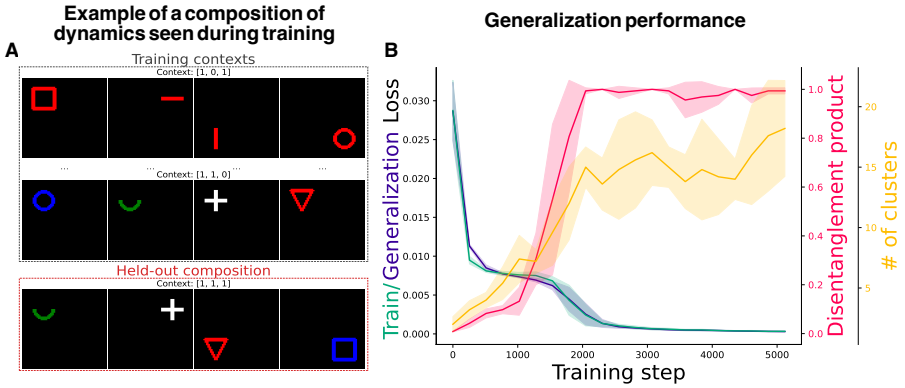

Figure 5: **The predictive model generalizes compositionally. A** We held out some of the composition of the dynamics during training time to assess compositional generalization. **B** The model accurately reconstructs sequences with contexts not seen during training, at a time that coincides with the disentanglement and the number of clusters reaching their limit.

A). The network organizes itself according to the context at hand, and each cluster inhibits the other clusters relevant to the same latent. This selection is implemented through a competitive process between clusters associated with the same latent variable. We observed that the activity of these intra-latent clusters is strongly anti-correlated (Figure 4 B), suggesting they mutually inhibit one another. The RNN's recurrent weight matrix confirms this, weights between competing intra-latent clusters are strongly negative, while recurrent weights within each cluster are positive (Figure 4 C). This connectivity pattern supports a winner-take-all dynamic, where the contextually appropriate cluster becomes active, suppresses its direct competitors, and sustains its own activity. Finally, we found that this modularity extends to how the selected dynamic is applied. Each cluster projects preferentially to the specific disentangled neurons that encode the latent variable it modifies (Figure 4 C). For example, clusters that implement shape dynamics are most strongly connected to the neurons representing shape. This targeted connectivity ensures that an activated dynamic only affects its corresponding feature, thereby preserving the independence of the other latents.

2.4 The predictive model generalizes compositionally

Systematic compositionality is defined as the capacity to assemble a finite set of known elements into any novel combination [4, 7, 16]. In our work, these elements are the dynamics of the latents, and we say that a model generalizes compositionally if it performs as well on compositions of components not seen during learning as it does on the compositions it was trained on. We train the network on 80% of the possible compositions of the dynamics during training

and save the rest for testing this compositional generalization (Figure 5 A). We computed the training and generalization loss (Figure 5 B), and found that the two losses go down in unison, where the generated images match the ground truth. We concluded that the model composes its clusters, using only sensory inputs, to accurately predict the next observations, and that it can do so in contexts not seen during training. We stress that it does not require any task labels, and can select which cluster to use autonomously, as opposed to common multi-task learning procedures, which give information about the current task to the network [5, 28, 41]. We also found that this ability emerges at around the same time during training as when the model perfectly disentangles the latents, as shown by the product of the score of disentangling the three latents, and when the number of clusters increases (Figure 5 B). This supports the fact that the two properties described above help the network generalize compositionally.

3 Discussion

We have demonstrated that an RNN trained with predictive learning in a rich environment generalizes compositionally by disentangling the generative factors and composing the dynamics. The model trained to predict the next observation separates all latents perfectly, each being encoded in distinct groups of neurons, whereas an autoencoder has them entangled. The predictive model also has multiple clusters of neurons performing one dynamic on one latent, which can be composed according to the dynamics the inputs follow.

Our work builds upon previous studies showing that RNNs trained with multi-task learning can develop modular, compositional solutions [5, 28, 41]. However, a key distinction of our model is that this structure emerges autonomously from a single objective, without the need for explicit task labels or context signals. This is particularly relevant for sensory systems, which must learn the structure of the world largely through unsupervised observation, rather than through a curriculum of distinct, labeled tasks [9, 21]. Our results reinforce the idea that predicting future sensory input can lead to informative representations, and could be used in the brain [3, 6, 9, 17, 26, 39]. Notably, predictive learning is also the principal training objective in large language models [24, 35], typically instantiated as next-token prediction, showing the generality of this learning paradigm.

The emergence of disentangled representations of the latent factors is a cornerstone of this process, as it allows for simple dynamics that can easily modify one latent only. Such invariance is one of the necessary properties of one type of compositional representation, called symbolic representations, which have been observed in the monkey brain [33]. In the direction of separability and selectivity, we predict that neural networks trained with predictive learning will develop clusters of very selective neurons that react to distinct factors of the inputs, similar to what is learned with multi-task learning in feedforward neural networks [13]. On the other hand, our simple autoencoder developped mixed neurons. More complex autoencoders, models with probabilistic latent spaces

like variational autoencoders (VAEs), have been shown to learn disentangled representations [11], and have responses similar to those of biological circuits [11]. However, this success often depends on tuning the objective function to explicitly penalize disentanglement in the latent space, thereby encouraging factorization. Other methods building on top of VAEs also use temporal information by defining transitions between images [15, 18, 19, 29], but they have not analyzed simple RNNs with only a predictive objective. Our findings present a complementary perspective where temporal prediction is sufficient to drive the emergence of a disentangled code. The failure of our standard autoencoder to disentangle these factors highlights that merely reconstructing the present is not a strong enough constraint; the pressure to anticipate the future is what drives the network to learn the underlying causal structure of the environment. In addition to informative representations, learning the dynamics also allows planning and simulation of the environment.

Mechanistically, our model selects its clusters through competition between clusters impacting the same latent. These clusters compete via inhibitory connections, creating winner-takes-all dynamics where the active cluster is determined by the ongoing sequence context. This allows the network to flexibly select and apply the correct dynamic transformation while leaving other, irrelevant latent representations untouched. This functional segregation is further supported by our connectivity analysis, which reveals that clusters are preferentially connected to the neurons encoding the specific latent they manipulate. This suggests the formation of distinct processing pathways, characteristic of modular computation in artificial and biological neural networks.

Our work opens several research directions. Our synthetic environment is deterministic and relatively simple. Future studies should investigate whether these principles scale to more complex, naturalistic environments with stochastic dynamics and hierarchical structure. It would also be interesting to study how changing the environment during learning affects the network. For example, one could add new factors, such as a scale to the objects, and test whether the network keeps its representations and adds new ones for the new latents. Also, when changing the space of the latents, would a new color be encoded in the same neurons as the ones encoding the other colors? It would also be valuable to explore how these learned representations could be leveraged by a downstream decision-making network, linking predictive learning directly to goal-directed behavior.

Altogether, our results demonstrate that the simple objective of predicting future states is a powerful engine for building compositional world models. This process autonomously discovers the underlying generative factors of an environment, segregates them into disentangled representations, and learns their dynamics in a modular fashion. These findings provide a mechanistic account of how the brain might develop world models that support flexible generalization.

4 Methods

4.1 Network Architecture

As stated in the results section, the whole network consists of a convolutional encoder, an RNN, and a deconvolutional decoder. The encoder has 3 convolutional layers with a channel size 16 with ReLU activations, followed by a linear layer with a size of 512. It then projects into the RNN, which also has size 512. The decoder has the same structure as the encoder, but mirrored. The RNN update rule is:

$$h_t = ReLU(W_{rec}h_{t-1} + W_{in}x_t + b + \sqrt{\sigma^2}\xi)$$

where h_t is the activity of the RNN at time t, W_{rec} are the recurrent weights, W_{in} the inputs weights, x_t the inputs, b the biases, σ^2 is a variance scaling applied to the independent gaussian noiseξ with mean zero and unit variance.

4.2 Training the model

We trained the predictive model to minimize the squared error of the decoder output, $\hat{o}_t$, and the observation of the next-time step o_{t+1}, $\frac{1}{(T-1)P}\sum_{t=2}^{T}\|\mathbf{o_{t+1}} - \mathbf{\hat{o}_t}\|_2^2$, where T is the length of the sequence, and P is the number of pixels in one image. The evaluation starts from the second frame (t=2) because the model requires at least two frames to infer the sequence's dynamics. We add L2 regularization on the weights and activity of the RNN, with weights $\lambda_{weights}$ and λ_{act}. We didn't find stereotypical effect of increasing or lowering these parameters, similar to [41]. We trained the whole network for 5000 optimization steps, using the AdamW [20] optimizer with a learning rate of 0.001, and momentum parameters $\beta_1 = 0.9, beta_2 = 0.999$, a batch size of 128, using the Pytorch library [22].

4.3 Hyperparameter search

We ran a grid search on the two regularization terms, both with values [0, $0.001, 0.01, 0.1, 1, 10$], and size of the RNN with values $[128, 512]$ for the predictive model and $[8, 32, 128, 512]$ for the autoencoder. We used $\lambda_{weights} = 0.001, \lambda_{act} = 0.01$ and a size of 512 for the RNN for the predictive model because they gave the cleanest clusters. We also show the results with the lowest validation loss in the supplementary S3, with parameters $\lambda_{weights} = 0.1, \lambda_{act} = 0$. For the autoencoder, we found that $\lambda_{weights} = 1, \lambda_{act} = 0$, and size 512 achieved both the lowest generalization loss and the best disentanglement product. We ran the search with 2 seeds for each combination of hyperparameters, and then ran the selected hyperparameters with 4 seeds to make Figure 2 and Figure 5. The plots for Figure 3 and Figure 4 where made with the seed 0 of these models.

4.4 Dataset

The dataset is made of sequences of frames of size 64x64. Each is fully determined by three discrete independent latent variables, which encode an object's

shape (z_{shape}), position ($z_{position}$), and colour (z_{colour}), as their index in their respective list of possible values. There are seven possible shapes: a wide rectangle, a tall rectangle, a circle, a semi-circle in a U-shape, a cross, a triangle, and a square. There are four possible colours: white, red, green, and blue. There are 4 positions, the center of the four quadrants of the frame. To create a video, we sample one value for each latent for the initial conditions and then apply independent dynamics to them, which we call the context of the sequence. They are defined by a tuple $(\Delta z_{\text{shape}}, \Delta z_{\text{position}}, \Delta z_{\text{colour}})$ For example, the context $(1, 0, 2)$ increments the shape index by 1, leaves the position unchanged, and increments the colour by 2. We used 6 dynamics for the shapes, and 3 for both colour and position. This resulted in 54 dynamics, which we split into 43 training dynamics and 11 testing dynamics.

4.5 Disentanglement

To measure disentanglement, we modified a metric introduced in [1, 13]. We constructed disjoint training and testing splits by dividing the support of all other latents in half. We then presented the frames from both splits to the model and extracted the activity of the RNN. After fitting a linear max margin classifier to classify the latent z from the RNN activity, we used the accuracy of the classifier on the test set as the disentanglement score. We used the LinearSVC model, from the scikit-learn library [23], with an L1 penalty of weight equal to 10.

4.6 Cluster analysis

We first calculated a context selectivity score for each RNN neuron, following the task variance analysis method from [41]. We computed the variance of each neuron across all contexts and discarded any with a summed variance below 0.001. The activity of the remaining neurons was then normalized by their individual maximum variance across the contexts.

Using these selectivity scores, we clustered the neurons with the K-Means algorithm, using the scikit-learn library [23]. To determine the optimal number of clusters, we performed a sweep from 2 to 25 clusters, selecting the configuration that minimized intra-cluster distance while maximizing inter-cluster distance. We then sorted the clusters based on the latent dynamic that elicited the strongest response.

Acknowledgements

This work was supported by Wellcome Trust 200790/Z/16/Z, Simons Foundation 564408 and EPSRC EP/R035806/1.

References

[1] Silvia Bernardi, Marcus K Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967, 2020.

[2] Charlotte Caucheteux, Alexandre Gramfort, and Jean-Rémi King. Evidence of a predictive coding hierarchy in the human brain listening to speech. *Nature human behaviour*, 7(3):430–441, 2023.

[3] Yusi Chen, Huanqiu Zhang, Mia Cameron, and Terrence Sejnowski. Predictive sequence learning in the hippocampal formation. *Neuron*, 112(15):2645–2658, 2024.

[4] Noam Chomsky. *Aspects of the Theory of Syntax*. Number 11. MIT press, 2014.

[5] Laura N Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, 2024.

[6] Ching Fang and Kimberly L Stachenfeld. Predictive auxiliary objectives in deep rl mimic learning in the brain. *arXiv preprint arXiv:2310.06089*, 2023.

[7] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

[8] Shohei Furutachi, Alexis D Franklin, Andreea M Aldea, Thomas D Mrsic-Flogel, and Sonja B Hofer. Cooperative thalamocortical circuit mechanism for sensory prediction errors. *Nature*, 633(8029):398–406, 2024.

[9] Manu Srinath Halvagal and Friedemann Zenke. The combination of hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature neuroscience*, 26(11):1906–1915, 2023.

[10] Demis Hassabis and Eleanor A Maguire. The construction system of the brain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1263–1271, 2009.

[11] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

[12] Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks. *Advances in neural information processing systems*, 35:32225–32239, 2022.

[13] W Jeffrey Johnston and Stefano Fusi. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nature Communications*, 14(1):1040, 2023.

[14] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

[15] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

[16] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR, 2018.

[17] Daniel Levenstein, Aleksei Efremov, Roy Henha Eyono, Adrien Peyrache, and Blake Richards. Sequential predictive learning is a unifying theory for hippocampal representation and replay. *bioRxiv*, pages 2024–04, 2024.

[18] Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

[19] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International conference on machine learning*, pages 6348–6359. PMLR, 2020.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[21] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[25] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[26] Stefano Recanatesi, Matthew Farrell, Guillaume Lajoie, Sophie Deneve, Mattia Rigotti, and Eric Shea-Brown. Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nature communications*, 12(1):1417, 2021.

[27] Carlo Reverberi, Kai Görgen, and John-Dylan Haynes. Compositionality of rule representations in human prefrontal cortex. *Cerebral cortex*, 22(6):1237–1246, 2012.

[28] Reidar Riveland and Alexandre Pouget. Natural language instructions induce compositional generalization in networks of neurons. *Nature Neuroscience*, 27(5):988–999, 2024.

[29] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[30] Philipp Schwartenbeck, Alon Baram, Yunzhe Liu, Shirley Mark, Timothy Muller, Raymond Dolan, Matthew Botvinick, Zeb Kurth-Nelson, and Timothy Behrens. Generative replay underlies compositional inference in the hippocampal-prefrontal circuit. *Cell*, 186(22):4885–4897, 2023.

[31] Arseny A Sokolov, R Chris Miall, and Richard B Ivry. The cerebellum: adaptive prediction for movement and cognition. *Trends in cognitive sciences*, 21(5):313–332, 2017.

[32] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[33] Lucas Y Tian, Kedar U Garzón, Adam G Rouse, Mark AG Eldridge, Marc H Schieber, Xiao-Jing Wang, Joshua B Tenenbaum, and Winrich A Freiwald. Neural representation of action symbols in primate frontal cortex. *bioRxiv*, 2025.

[34] Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[36] Elia Turner and Omri Barak. The simplicity bias in multi-task rnns: shared attractors, reuse of dynamics, and geometric representation. *Advances in Neural Information Processing Systems*, 36:25495–25507, 2023.

[37] Catherine Wacongne, Etienne Labyt, Virginie Van Wassenhove, Tristan Bekinschtein, Lionel Naccache, and Stanislas Dehaene. Evidence for a hierarchy of predictions and prediction errors in human cortex. *Proceedings of the National Academy of Sciences*, 108(51):20754–20759, 2011.

[38] James CR Whittington, Will Dorrell, Surya Ganguli, and Timothy EJ Behrens. Disentanglement with biological constraints: A theory of functional cell types. *arXiv preprint arXiv:2210.01768*, 2022.

[39] James CR Whittington, William Dorrell, Timothy EJ Behrens, Surya Ganguli, and Mohamady El-Gaby. A tale of two algorithms: Structured slots explain prefrontal sequence memory and are unified with hippocampal cognitive maps. *Neuron*, 113(2):321–333, 2025.

[40] James CR Whittington, David McCaffary, Jacob JW Bakermans, and Timothy EJ Behrens. How to build a cognitive map. *Nature neuroscience*, 25(10):1257–1272, 2022.

[41] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

Supplementary

S1 Training performance

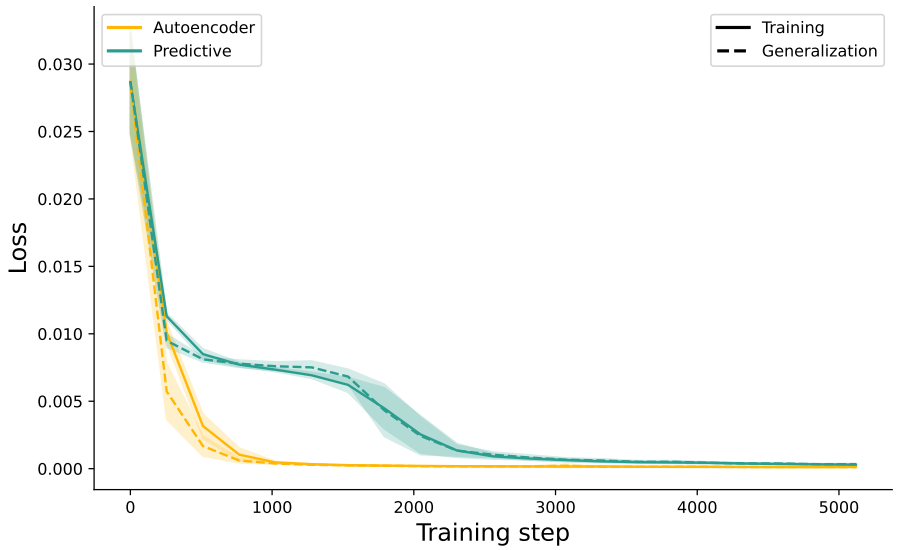

Figure S1: **Comparing the training performance of the predictive model and the autoencoder.**

We computed the training and generalization losses every 256 gradient steps during training (Figure S1). The autoencoder losses converge very quickly, whereas the predictive model is slower. This is due to the fact that predicting the future observation is much harder than reconstructing the last observation seen. However, having the harder task also means that structured representations will emerge. Additionally, as the autoencoder does not take into account dynamics, training and generalization can be seen as the same task.

S2 Comparing to an autoencoder without an RNN in the middle

Having an RNN in the middle of the encoder and decoder may affect the disentanglement. Therefore, to get another control, we computed the disentanglement score of a feedforward autoencoder with no recurrency (Figure S2). It will not use temporal information, which is fine since it only reconstructs the last observation. We ran hyperparameter sweeps with different sizes of bottlenecks, again following [38], with different levels of L2 regularization on both weight

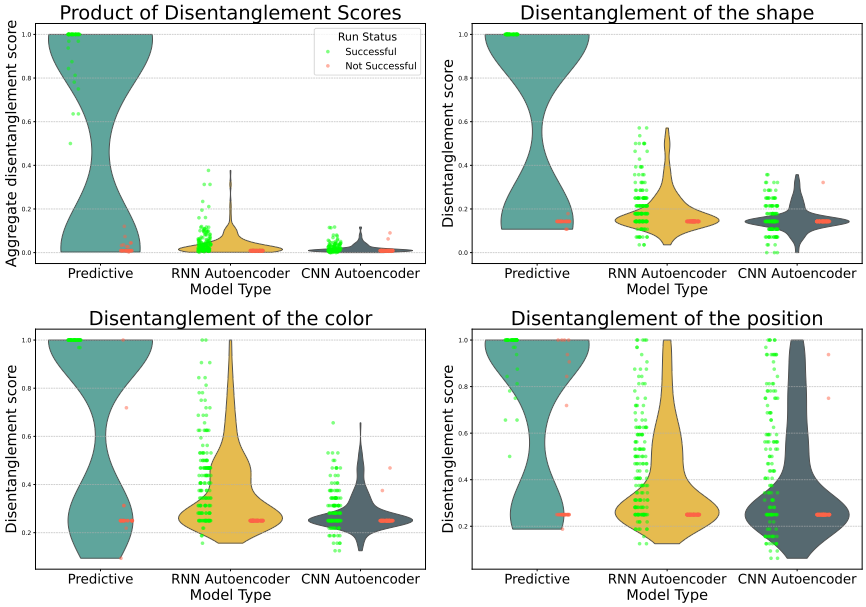

Figure S2: **Comparing the disentanglement of the predictive model, a recurrent autoencoder, and a feedforward autoencoder.** The product of the disentanglement score is the product of the three disentanglement scores shown here. The colour of the dots indicates whether the generalization loss of the model was lower than 0.006, which we defined as a successful run. For the predictive model, a successful run almost always means that the three latents are perfectly disentangled, whereas neither autoencoder models perfectly disentangle all of the latents. There are fewer points for the predictive model because lower sizes of bottlenecks were never successful, so we didn't include them in the hyperparameter search.

and bottleneck activity. The RNN autoencoder disentangles the latents better on average than the CNN autoencoder.

S3 Results for the lowest generalization loss model

We chose the model with the clearest cluster for the main results, and selected the model with the lowest generalization loss for the autoencoder, which coincided with the best disentanglement as well. We also computed the clustering and the summary statistics of the correlation between clusters and mean weights across clusters for the predictive model with the lowest generalization loss Figure S3. We still find a clear cluster structure (Figure S3 A), but lesioning them as a greater effect in contexts they are not selective to (Figure S3 B). We still find evidence for competition across clusters computing dynamics for the same

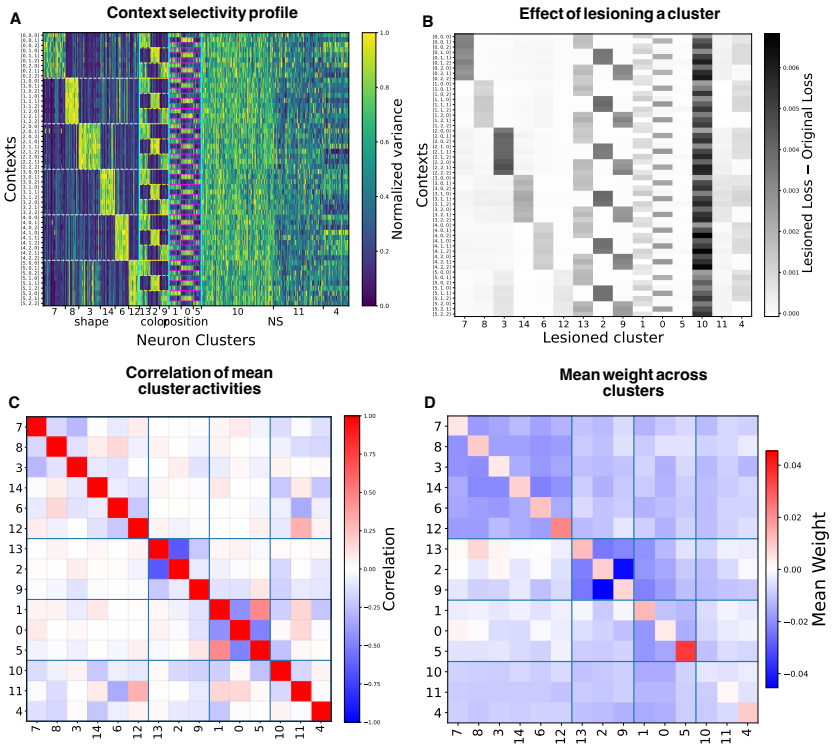

Figure S3: **Results for the model with lowest generalization loss. A** Clustering analysis of the model. The model still has distinct clusters for the dynamics. **B** Effect of lesioning a cluster. Even though the model has distinct cluster for the dynamics, lesioning a cluster also has effects on contexts it is not selective to. For example, lesioning any cluster computing dynamics for the shape worsens the reconstruction for contexts associated with moving the shape by 5, even though they are not selective to it. **C** Correlation of the mean activity of clusters. There still is a negative correlation between cluster computing dynamics for the same latent. **D** Mean weight across clusters. We still find that weights across clusters computing dynamics for the same latents are negative, whereas those intra-cluster are positive.

latents (Figure S3 C,D).