Associative memory via covariance-learning predictive coding networks

Mufeng Tang
MRC Brain Network Dynamics Unit
University of Oxford, UK
mufeng.tang@bndu.ox.ac.uk

Tommaso Salvatori
Department of Computer Science
University of Oxford, UK
tommaso.salvatori@cs.ox.ac.uk

Yuhang Song
Department of Computer Science
University of Oxford, UK
yuhang.song@some.ox.ac.uk

Beren Millidge
MRC Brain Network Dynamics Unit
University of Oxford, UK
beren@millidge.name

Thomas Lukasiewicz
Department of Computer Science
University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Rafal Bogacz*
MRC Brain Network Dynamics Unit
University of Oxford, UK
rafal.bogacz@ndcn.ox.ac.uk

Abstract

Classical models of biological memory assume that associative memory (AM) in the hippocampus is achieved by learning a covariance matrix of simulated neural activities. However, it has been also proposed that AM in the hippocampus could be explained in the predictive coding framework. These two seemingly disparate computational principles pose difficulties for developing a unitary theory of memory storage and recall in the brain. In this work, we address this dichotomy using a family of covariance-learning predictive coding networks (covPCNs). We show that earlier predictive coding networks (PCNs) explicitly learning the covariance matrix perform AM, but their learning rule is non-local and unstable. We propose a novel model that implicitly learns the covariance matrix with Hebbian plasticity and stably converges to the same memory retrieval as the earlier models. We further show that this model can be combined with hierarchical PCNs to model the hippocampo-neocortical interactions. In practice, our models can store a large number of memories of structured images and retrieve them with high fidelity. Our models shed light on how predictive processing can be performed in the recurrent hippocampal network, and also unify two distinct computational principles underlying the modelling of the hippocampus in AM.

*Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, US.

1 Introduction

It is widely believed that the recurrent hippocampal network is crucial for *associative memory* (AM) in the brain [1, 2, 3]. Early computational models of the hippocampus in AM, such as the Hopfield Networks [4] and Correlation Matrix Memories [5], assumed that input patterns are memorized by learning a covariance matrix representing the associations between individual neurons activated by a memory item, which is encoded in the recurrent hippocampal connections. On the other hand, experimental studies have revealed that the hippocampus is capable of predicting ongoing sensory inputs, and has neurons encoding prediction errors [6, 7, 8]. It is thus hypothesized that the computational principle underlying AM is a form of predictive coding (PC), where the hippocampus generates predictions of the neocortical inputs, based on the memories stored in its recurrent connections, and the dynamics of the whole network aim to minimize the prediction errors [9]. Adopting the predictive coding network (PCN) developed by Rao and Ballard [10], a computational model then verified this theoretical hypothesis, showing that a purely hierarchical PCN can perform AM without recurrent connections between the neurons [11].

These two modelling approaches, namely AM via covariance learning and predictive coding, appear to be entirely different and diverging: the covariance-learning models disregard the predictive nature of hippocampal activities, whereas the PC model fails to capture the recurrent structure of the hippocampal network that may encode the useful covariance information. This dichotomy poses difficulties in understanding the hippocampus from a computational perspective. In this work we aim to resolve this dichotomy by unifying these two approaches to modelling AM using a family of covariance-learning PCNs (covPCNs). Specifically, we show that the covariance-learning PCN proposed by Friston [12, 13], which we refer to as the *explicit covPCN*, performs AM. However, we note that the plasticity rule of this model employs non-local information, and is also numerically unstable. To address these issues, we propose in this work a novel recurrent PCN that learns the covariance implicitly, called the *implicit covPCN*. We show that analytically, the memory retrieval of the implicit and the explicit covPCNs are *equivalent* given the same cue, while the implicit model is more stable in practice and only employs local Hebbian plasticity [14]. Furthermore, the implicit model can also be combined with a hierarchical PCN [10, 11] to model the whole hippocampo-neocortical region. Overall, our contribution is twofold: First, we implement PC within a plausible and stable recurrent network, which sheds light on how predictive processing can be performed via the recurrent hippocampal network. Second, we demonstrate that the recurrent PCNs also retains the covariance-learning properties of classical models, thus unifying these classical models with the PC framework. Such a unified description facilitates the theoretical understanding of computations performed by the hippocampus in AM tasks.

2 Models

Explicit covPCNs The distinguishing feature of PCNs is the error neurons encoding the mismatch between internally generated predictions and external inputs [10]. Friston [12, 13] extended the original PCNs by introducing recurrent connections encoding the covariance matrix to weight the error neurons. The model assumes that the input patterns $\{\mathbf{x}(i)\}_{i=1}^N$ are d-dimensional samples from a Gaussian distribution with true mean $\boldsymbol{\mu}_{\text{true}}$ and covariance matrix Σ_{true}, where i indicates the ith sample within the dataset, and the bold font denotes vectors. The plasticity of the model, parameterized by mean $\boldsymbol{\mu}$ and covariance Σ, aims to maximize the log likelihood $\mathcal{F}_{\text{ex}}$ ("ex" for explicit) of observed neural activities $\mathbf{x}(i)$ given this Gaussian distribution via gradient ascent:

$$\Delta\boldsymbol{\mu} \propto \partial\mathcal{F}_{\text{ex}}/\partial\boldsymbol{\mu} = \sum\nolimits_i \boldsymbol{\varepsilon}(i), \quad \Delta\Sigma \propto \partial\mathcal{F}_{\text{ex}}/\partial\Sigma = -N\Sigma^{-1} + \sum\nolimits_i \boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}(i)^T \tag{1}$$

where $\boldsymbol{\varepsilon}(i) = \Sigma^{-1}(\mathbf{x}(i) - \boldsymbol{\mu})$ is the (weighted) error. The above learning rules have a property that both parameters will converge to the maximum likelihood estimate (MLE) of $\boldsymbol{\mu}_{\text{true}}$ and Σ_{true} based on the training data points, i.e. $\boldsymbol{\mu} \to \frac{1}{N}\sum_{i=1}^N \mathbf{x}(i)$ and $\Sigma \to \frac{1}{N}\sum_{i=1}^N (\mathbf{x}(i) - \boldsymbol{\mu})(\mathbf{x}(i) - \boldsymbol{\mu})^T$, by setting $\Delta\boldsymbol{\mu} = \Delta\Sigma = 0$. We denote these MLEs of mean and covariance by $\bar{\mathbf{x}}$ and S respectively. Therefore the parameter Σ will *explicitly* encode the sample covariance of the data S when the learning converges, thus the name explicit covPCNs.

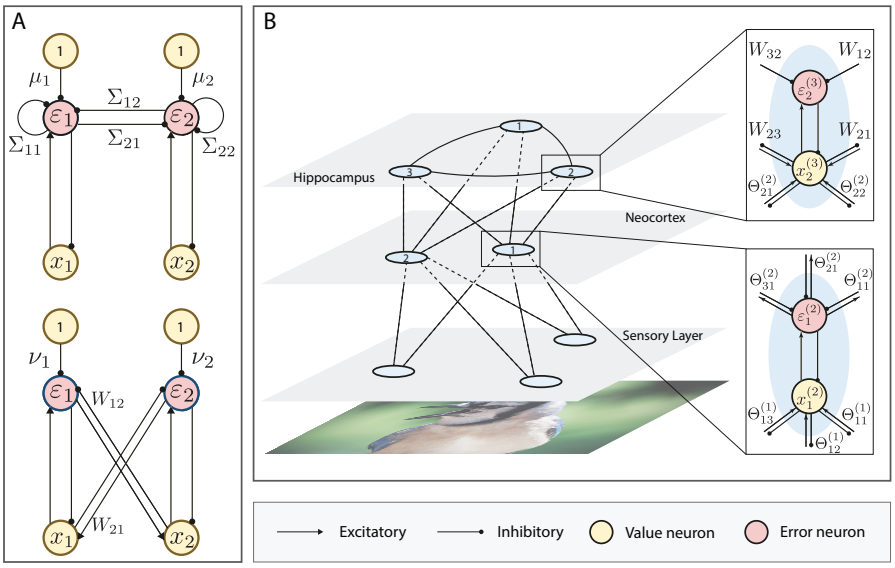

Figure 1: A: The explicit (top) and implicit (bottom) covPCNs with 2-dimensional stimuli. B: The hybrid PCN where the top hippocampal layer is an implicit covPCN, and the neocortical and sensory layers follow the hierarchical PCN for AM in [11]. Expanded boxes present the detailed computations. For simplicity we assume one neocortical layer and the same number of neurons in all layers. Unlabeled connections have strengths 1. Subscripts denote vector indices/neuron numbers, and superscripts with brackets denote layer numbers.

After learning, we fix the parameters $\boldsymbol{\mu}$ and Σ, and provide a single cue $\tilde{\mathbf{x}}$ to the network to initialize the memory retrieval, i.e. we initialize the neural activity to $\mathbf{x} = \tilde{\mathbf{x}}$, and the explicit covPCN performs inference by updating $\mathbf{x}$ according to the derivative of the log likelihood:

$$\Delta \mathbf{x} \propto \partial \mathcal{F}_{\text{ex}} / \partial \mathbf{x} = -\boldsymbol{\varepsilon} \tag{2}$$

If $\tilde{\mathbf{x}}$ is a corrupted training data point, the equation above will drive it towards its corresponding training data point to achieve memory recall, as the original data point defines higher log likelihood $\mathcal{F}_{\text{ex}}$. For details of derivations of Eq 1 and 2 see Appendix. The above neural dynamics can be implemented within the network shown in Fig 1A, when $d = 2$. In this network, $\boldsymbol{\varepsilon}$ and $\mathbf{x}$ are encoded in activities of error and value neurons respectively, and parameters $\boldsymbol{\mu}$ and Σ in synaptic weights. The value neurons $\mathbf{x}$ receive the sensory inputs and also receive inhibition from $\boldsymbol{\varepsilon}$ (Eq 2), while the error neurons $\boldsymbol{\varepsilon} = \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})$ receive excitatory inputs from the value neurons, inhibitory inputs encoding the prior expectation $\boldsymbol{\mu}$, and lateral inhibitory inputs encoding the weight Σ. The topmost value neurons have activity 1 so that $\boldsymbol{\mu}$ can be encoded into synaptic strengths, following the implementation in [15]. Notice that if we denote Σ_{ab} as the synapse connecting the a-th and b-th neuron in this explicit model, to update this single synapse ($\Delta\Sigma_{ab}$), non-local synaptic strengths are needed to compute the inverse $(\Sigma^{-1})_{ab}$ (Eq 1), making it biologically implausible. This is because each entry in the matrix inverse depends on all entries in the original matrix, so each synapse in the neural implementation has to have the global knowledge of all other synaptic strengths in the network to update itself. As we will show later, this inverse term also poses significant computational problems in practice.

Implicit covPCNs Notice that another way of encoding recurrent interactions between neurons, while preserving the predictive nature of the network, is to let the neurons predict each other. With this intuition, we parameterize the implicit covPCN with weight matrices W and $\boldsymbol{\nu}$, with W_{ab} representing the synapse connecting the a-th and b-th neurons in the network, and $\boldsymbol{\nu}$ the bias vector. W is zero-diagonal, implying no self-connection/self-prediction in this model. The implicit model tries to maximize the following objective function given the dataset $\{\mathbf{x}(i)\}_{i=1}^{N}$ ("im" for implicit):

$$\mathcal{F}_{\text{im}} = -\sum\nolimits_i \|\mathbf{x}(i) - W\mathbf{x}(i) - \boldsymbol{\nu}\|_2^2/2 \tag{3}$$

In the implicit model, we define the prediction errors as $\boldsymbol{\varepsilon} = \mathbf{x} - W\mathbf{x} - \boldsymbol{\nu}$. Like the explicit model, the implicit covPCN first updates its parameters W and $\boldsymbol{\nu}$ by performing gradient ascent on $\mathcal{F}_{\text{im}}$:

$$\Delta\boldsymbol{\nu} \propto \partial\mathcal{F}_{\text{im}}/\partial\boldsymbol{\nu} = \sum\nolimits_i \boldsymbol{\varepsilon}(i), \quad \Delta W \propto \partial\mathcal{F}_{\text{im}}/\partial W = \left(\sum\nolimits_i \boldsymbol{\varepsilon}(i)\mathbf{x}(i)^T\right)_{diag=0} \tag{4}$$

where the $()_{diag=0}$ notation means "enforcing the diagonal elements to be 0" as we want to keep the 0 diagonal elements of W unchanged. Notice that by setting $\Delta\boldsymbol{\nu}$ and ΔW to 0 i.e., at the convergence of learning, we obtain $\boldsymbol{\nu} = (I - W)\bar{\mathbf{x}}$ and $[(I - W)S]_{diag=0} = 0$. Recall that $\bar{\mathbf{x}}$ and S denote the MLEs of the mean $\boldsymbol{\mu}_{\text{true}}$ and covariance matrix Σ_{true}, and that the parameters of the explicit covPCN, $\boldsymbol{\mu}$ and Σ, converge to $\bar{\mathbf{x}}$ and S. Therefore, the implicit covPCN also learns the mean and the covariance matrix, without encoding them explicitly into its connections.

Like the explicit model, after learning, the implicit model performs inference after initializing the activity to a cue input $\mathbf{x} = \tilde{\mathbf{x}}$ following the derivative of $\mathcal{F}_{\text{im}}$ with respect to $\mathbf{x}$:

$$\Delta\mathbf{x} \propto \partial\mathcal{F}_{\text{im}}/\partial\mathbf{x} = -\boldsymbol{\varepsilon} + W^T\boldsymbol{\varepsilon} \tag{5}$$

The above dynamics can be implemented in the network model in Fig 1B, by replacing the lateral connections Σ with W that projects the predictions from all other value neurons into each error neuron. Notice that the learning rule for the connection W_{ab} is Hebbian, as it is simply a product $\varepsilon_a x_b$ of the pre- and post-synaptic activities.

Hybrid PCNs We further use a hierarchical PCN [11] to model the cortical pre-processing of raw sensory inputs for the implicit covPCN. According to a recent theory, the recurrent hippocampus functions as a generative model that accumulates prediction errors from sensory and neocortical neurons lower in the hierarchy, and sends descending predictions to the neocortex, to correct the prediction errors in the neocortical neurons [9]. A demonstration of this hybrid PCN is shown in Fig 1C. The layers in this hybrid model are connected by $\Theta_{ab}^{(l)}$ denoting the synapse between neuron b in the lth layer ($x_b^{(l)}$) and neuron a in the $(l+1)$th layer ($x_a^{(l+1)}$). This model captures both the recurrent structure of the hippocampus and the predictive hippocampo-neocortical relationship [9]. For details of derivation and neural implementation of the hierarchical PCNs see [11].

3 Results

The equivalence of the explicit and implicit covPCNs in AM We now present the key result of our work, that both explicit and implicit covPCNs perform AM, and that their retrievals are *equivalent* given the same corrupted cue of the memories. First, we can show that the following theorem holds:

Theorem 1 *After training has converged, given a partial cue of a memory* $\mathbf{x} = \begin{bmatrix}\mathbf{x}_k & \mathbf{x}_m\end{bmatrix}^T$*, where* $\mathbf{x}_k$ *is the* k*-dimensional corrupted part and* $\mathbf{x}_m$ *the* m*-dimensional intact part, the inferential dynamics of* **both** *the explicit and implicit models (Eq 2, 5) on the corrupted part converge to:*

$$\hat{\mathbf{x}}_k = S_{km}S_{mm}^{-1}(\mathbf{x}_m - \bar{\mathbf{x}}_m) + \bar{\mathbf{x}}_k \tag{6}$$

where $\hat{\mathbf{x}}_k$ is the final retrieval of the corrupted $\mathbf{x}_k$, and S_{pq} denotes the p by q submatrix when we express the MLE S of the covariance matrix as a 2×2 block matrix. $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{x}}_m$ denotes the top k and bottom m elements of the MLE $\bar{\mathbf{x}}$ of the mean $\boldsymbol{\mu}$. See Appendix for details of derivation. Empirically, we found that both explicit and implicit models can memorize 5×5 random Gaussian patterns, and their retrievals given a partially corrupted cue are identical (Fig 2A), consistent with the theorem.

Model performances with structured image data We next examine whether the models can perform stable AM with more complex, structured data such as MNIST [16] and CIFAR10 [17]. The results with the explicit, implicit and hybrid models are shown in Fig 2B. For these experiments,

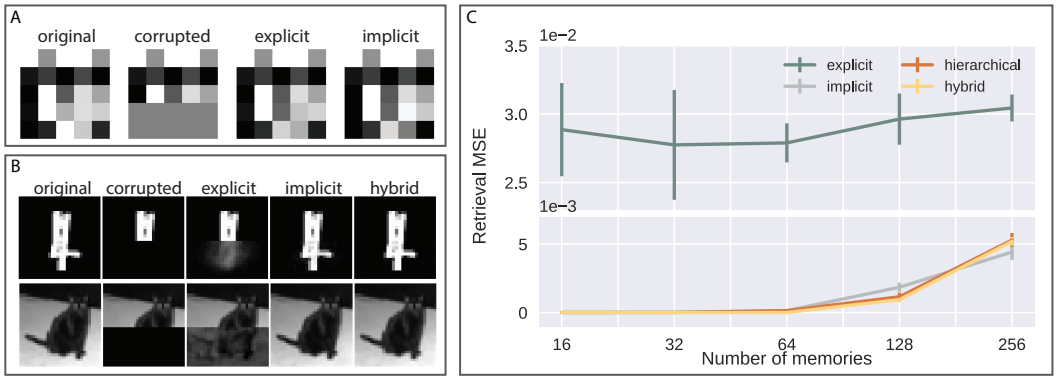

Figure 2: A: Examples of the retrieval of explicit and implicit covPCNs in AM with random 5×5 patterns sampled from a Gaussian distribution. B: Examples of the model (explicit, implicit and hybrid) performances with MNIST [16] and grayscale CIFAR10 [17]. C: Mean squared errors (MSE) of retrieval when the models are trained with different numbers of memories N.

we trained the networks to memorize 64 MNIST and grayscale CIFAR10 images, and provide half-covered cues to the models to initialize the retrieval. Particularly, since the single-layer models are recurrent, the number of value-error neuron pairs in these networks is the same as the number of image pixels d and thus the number of parameters in these recurrent models is d^2. To ensure that the hybrid model have approximately the same number of parameters, we construct the hybrid model to be a 3-layer network with d sensory neurons, $d/2$ hidden neurons and $d/2$ neurons in the topmost implicit layer. As can be seen, the implicit and hybrid model performed well with these datasets, whereas the retrievals of the explicit model are blurry.

Fig 2C shows a systematic investigation into the mean squared error (MSE) between the original and retrieved images, across different number of memories of grayscale CIFAR10. To compare with prior arts, we also trained a purely hierarchical PCN [11] with d sensory neurons and 3 hidden layers, all with $d/2$ neurons, such that the number of parameters is the same as the networks mentioned above. The implicit, hybrid and hierarchical models performed identically well, whereas the explicit model have much higher retrieval MSE and unstable performance across different samples. This is due to the necessity of computing the inverse Σ^{-1} during learning (Eq 1), which may introduce numerical instability especially when the training data is high-dimensional, and has ill-conditioned covariance matrices.

4 Discussion

Classical covariance-learning models of biological memory failed to capture the predictive nature of hippocampal activities, whereas recent PC models for memory overlooked the covariance-encoding recurrent structure of the hippocampus. In this work we have shown that 1) the predictive activities of the hippocampal neurons can be implemented within a recurrent network and 2) the dichotomy between covariance-learning and PC, two disparate computational principles, can be resolved by implicitly encoding the covariance into recurrent PCN synapses. Such a unified description greatly facilitates the theoretical understanding of computations performed by the hippocampus in AM tasks. Looking ahead, such a PC-based model is potentially scalable to modern machine learning models due to the close relationship between PC and deep learning, which has been revealed by recent works [18, 19, 20, 21]. Its computational capability may thus facilitate the modelling of other hippocampal functions, such as representation learning, navigation and temporal predictions, guiding future computational and experimental research into these more complex tasks performed by the hippocampus and thus inspiring more powerful artificial memory systems.

References

[1] William Beecher Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11, 1957.

[2] Larry R Squire and Stuart Zola-Morgan. The medial temporal lobe memory system. *Science*, 253(5026):1380–1386, 1991.

[3] Larry R Squire. Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *Psychological review*, 99(2):195, 1992.

[4] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[5] Teuvo Kohonen. Correlation matrix memories. *IEEE transactions on computers*, 100(4):353–359, 1972.

[6] John Lisman and A David Redish. Prediction, sequences and the hippocampus. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1193–1201, 2009.

[7] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

[8] Sylvia Wirth, Emin Avsar, Cindy C Chiu, Varun Sharma, Anne C Smith, Emery Brown, and Wendy A Suzuki. Trial outcome and associative learning signals in the monkey hippocampus. *Neuron*, 61(6):930–940, 2009.

[9] Helen C Barron, Ryszard Auksztulewicz, and Karl Friston. Prediction and memory: A predictive coding account. *Progress in neurobiology*, 192:101821, 2020.

[10] 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.

[11] Tommaso Salvatori, Yuhang Song, Yujian Hong, Lei Sha, Simon Frieder, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. Associative memories via predictive coding. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] Karl Friston. Learning and inference in the brain. *Neural Networks*, 16(9):1325–1352, 2003.

[13] Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005.

[14] Donald Hebb. *The Organization of Behavior.* Wiley, New York, 1949.

[15] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.

[16] Yann LeCun, Corinna Cortes, and Christopher J Burges. Mnist handwritten digit database. 2010. *URL http://yann. lecun. com/exdb/mnist*, 7(23):6, 2010.

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

[18] James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5):1229–1262, 2017.

[19] Yuhang Song, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz. Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems*, 33:22566–22579, 2020.

[20] Tommaso Salvatori, Yuhang Song, Zhenghua Xu, Thomas Lukasiewicz, and Rafal Bogacz. Reverse differentiation via predictive coding. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, volume 10177, pages 507–524. AAAI Press, February 2022.

[21] Robert Rosenbaum. On the relationship between predictive coding and backpropagation. *Plos one*, 17(3):e0266102, 2022.

[22] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Appendix

Explicit covPCN dynamics The objective function $\mathcal{F}_{\text{ex}}$ is also the log likelihood of the assumed Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{true}}, \Sigma_{\text{true}})$, given the data points $\{\mathbf{x}(i)\}_{i=1}^{N}$:

$$\mathcal{F}_{\text{ex}} = -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum\nolimits_i (\mathbf{x}(i) - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}(i) - \boldsymbol{\mu}) \tag{7}$$

Eq 1 and 2 naturally follows by taking the derivative of $\mathcal{F}_{\text{ex}}$ with respect to $\boldsymbol{\mu}$, Σ and $\mathbf{x}$.

The equivalent memory retrieval by explicit and implicit covPCNs Here we provide the details of the derivation of Eq 6, showing that both explicit and implicit models obtain the same retrieval at convergence. Recall that the parameters $\boldsymbol{\mu}$ and Σ of the explicit model converges to the MLEs $\bar{\mathbf{x}}$ and S at the convergence of learning, and we also assume that the inferential dynamics Eq 2 converges. That is:

$$\boldsymbol{\varepsilon} = S^{-1}(\mathbf{x} - \bar{\mathbf{x}}) = 0 \tag{8}$$

This equation can be written into its 2×2 block matrix form:

$$\begin{bmatrix} S_{kk} & S_{km} \\ S_{mk} & S_{mm} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_k - \bar{\mathbf{x}}_k \\ \mathbf{x}_m - \bar{\mathbf{x}}_m \end{bmatrix} = 0 \tag{9}$$

The inverse of S in terms of its submatrices is:

$$\begin{bmatrix} S_{kk} & S_{km} \\ S_{mk} & S_{mm} \end{bmatrix}^{-1} = \begin{bmatrix} Q^{-1} & -Q^{-1}S_{km}S_{mm}^{-1} \\ -S_{mm}^{-1}S_{km}^T Q^{-1} & S_{mm}^{-1} + S_{mm}^{-1}S_{km}^T Q^{-1} S_{km} S_{mm}^{-1} \end{bmatrix} \tag{10}$$

where Q is called the Schur complement of S_{mm} in S [22] and $Q = S_{kk} - S_{km}S_{mm}^{-1}S_{km}^T$. Since only $\mathbf{x}_k$ is relaxed during retrieval, we get:

$$Q^{-1}(\mathbf{x}_k - \bar{\mathbf{x}}_k) - Q^{-1}S_{km}S_{mm}^{-1}(\mathbf{x}_m - \bar{\mathbf{x}}_m) = 0 \tag{11}$$

which immediately gives us the retrieval dynamics in Eq 6 by multiplying Q on both sides of the equation and rearrange.

Now we show that the retrieval of the implicit model also follows Eq 6 at the convergence of inference. Setting Eq 5 to 0 gives us $\boldsymbol{\varepsilon} = \mathbf{x} - W\mathbf{x} - \boldsymbol{\nu} = 0$, since W is a zero-diagonal matrix and thus cannot be equal to I. Splitting it into blocks corresponding to $\mathbf{x}_k$ and $\mathbf{x}_m$ we have:

$$\begin{bmatrix} W_{kk} & W_{km} \\ W_{mk} & W_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_m \end{bmatrix} + \begin{bmatrix} \boldsymbol{\nu}_k \\ \boldsymbol{\nu}_m \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_m \end{bmatrix} \tag{12}$$

Since we only perform inference on the top k corrupted entries in $\mathbf{x}$, we have:

$$W_{kk}\mathbf{x}_k + W_{km}\mathbf{x}_m + \boldsymbol{\nu}_k = \mathbf{x}_k \Rightarrow (I_{kk} - W_{kk})\mathbf{x}_k = W_{km}\mathbf{x}_m + \boldsymbol{\nu}_k \tag{13}$$

We now investigate the parameter values in the above equation. Notice that we assumed the convergence of parameter learning at the time of retrieval, which gives us $\boldsymbol{\nu} = (I - W)\bar{\mathbf{x}}$ and $[(I - W)S]_{diag=0} = 0$. By splitting the first equation into blocks we have:

$$\boldsymbol{\nu}_k = (I_{kk} - W_{kk})\bar{\mathbf{x}}_k - W_{km}\bar{\mathbf{x}}_m \tag{14}$$

Substituting the $\boldsymbol{\nu}_k$ in Eq 13 with this relationship we get:

$$W_{km}(\mathbf{x}_m - \bar{\mathbf{x}}_m) = (I_{kk} - W_{kk})(\mathbf{x}_k - \bar{\mathbf{x}}_k) \tag{15}$$

Notice that $[(I - W)S]_{diag=0} = 0$ connects the value of W and S, which can also be written into the block matrix form:

$$\left(\begin{bmatrix} W_{kk} & W_{km} \\ W_{mk} & W_{mm} \end{bmatrix} \begin{bmatrix} S_{kk} & S_{km} \\ S_{mk} & S_{mm} \end{bmatrix} - \begin{bmatrix} S_{kk} & S_{km} \\ S_{mk} & S_{mm} \end{bmatrix} \right)_{diag=0} = 0 \tag{16}$$

which gives us two useful relationships:

$$W_{kk}S_{km} + W_{km}S_{mm} = S_{km} \Rightarrow W_{km} = (I_{kk} - W_{kk})S_{km}S_{mm}^{-1} \tag{17}$$

$$(W_{kk}S_{kk} + W_{km}S_{mk} - S_{kk})_{diag=0} = 0 \tag{18}$$

Substituting the expression of W_{km} in terms of W_{kk} into Eq 15 we have:

$$(I_{kk} - W_{kk})S_{km}S_{mm}^{-1}(\mathbf{x}_m - \bar{\mathbf{x}}_m) = (I_{kk} - W_{kk})(\mathbf{x}_k - \bar{\mathbf{x}}_k) \tag{19}$$

The above expression is already close to Eq 6 which we seek to prove, i.e. we could obtain Eq 6 by cancelling $I_{kk} - W_{kk}$ on both sides of Eq 19, hence we will now show that $I_{kk} - W_{kk}$ is invertible. To do it we first observe that according to Eq 18, $W_{kk}S_{kk} + W_{km}S_{mk} - S_{kk}$ is a diagonal matrix, which we call D. Therefore we have:

$$\begin{aligned} D &= (I_{kk} - W_{kk})S_{kk} - W_{km}S_{mk} \\ &= (I_{kk} - W_{kk})S_{kk} - (I_{kk} - W_{kk})S_{km}S_{mm}^{-1}S_{mk} \\ &= (I_{kk} - W_{kk})(S_{kk} - S_{km}S_{mm}^{-1}S_{mk}) \\ &= (I_{kk} - W_{kk})Q \end{aligned} \tag{20}$$

Since Q is the Schur complement of S_{mm} in S, a sample covariance matrix that we assume to be positive definite, it is also positive definite and thus invertible [22]. Therefore $I_{kk} - W_{kk} = DQ^{-1}$. Notice that since W_{kk} has all its diagonal elements equal to 0, the matrix DQ^{-1} must have 1's on its diagonal, which prevents the diagonal matrix D from having 0's on its diagonal (for $\left(DQ^{-1}\right)_{ii} = D_{ii}\left(Q^{-1}\right)_{ii} = 1$, D_{ii} cannot be 0). Therefore, D is also invertible, which makes $I_{kk} - W_{kk}$ an invertible matrix as well. This enables us to "cancel out" the $I_{kk} - W_{kk}$ on both sides of Eq 19 and establish the equivalence to the retrieval of the explicit covPCN (Eq 6).

Code availability Code to reproduce the experiments is available at:

https://github.com/C16Mftang/covariance-learning-PCNs