CENTER FOR
Brains
Minds+
Machines

**CBMM Memo No. 137** **July 6, 2022**

# Understanding the Role of Recurrent Connections in Assembly Calculus

**Akshay Rangamani and Yi Xie**
MIT Center for Brains, Minds, and Machines
MIT Computer Science & Artificial Intelligence Laboratory
Massachusetts Institute of Technology

## Abstract

In this note, we explore the role of recurrent connections in Assembly Calculus through a number of experiments conducted on models with and without recurrent connections. We observe that assemblies can be formed even in the absence of recurrent connections, but also find that models with recurrent connections are more robust to noisy inputs. We also investigate the spectral structure of the synaptic weights and find intriguing similarities between models of neural assemblies and associative memories.

**This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.**

# Understanding the Role of Recurrent Connections in Assembly Calculus

**Akshay Rangamani and Yi Xie**

MIT Center for Brains, Minds and Machines
MIT Computer Science & Artificial Intelligence Laboratory
Massachusetts Institute of Technology

{arangam, xieyi}@mit.edu

**Abstract**

In this note, we explore the role of recurrent connections in Assembly Calculus through a number of experiments conducted on models with and without recurrent connections. We observe that assemblies can be formed even in the absence of recurrent connections, but also find that models with recurrent connections are more robust to noisy inputs. We also investigate the spectral structure of the synaptic weights and find intriguing similarities between models of neural assemblies and associative memories.

## 1 Introduction

There have been many models proposed to understand the brain, at different levels of abstraction - ranging from molecular models of neurotransmission, to models of neurons, to whole brain models studied in cognitive science. Recently, *assemblies of neurons* [1] have been proposed as a computational model at a level of abstraction between models of neurons [2, 3] and whole brain models. Neural assemblies were introduced by Hebb [4] nearly seventy years ago and have recently been experimentally established [5, 6, 7, 8] as existing in mammalian brains. While modern deep neural networks are inspired by models of the sensory parts of the brain, especially the visual cortex [9], assemblies of neurons are meant to model intermediate levels of computation beyond sensory information processing. They are believed to have the ability to represent memories and cognitive concepts. It is likely that this level is the one at which higher cognitive functions like language are implemented.

The model of the brain used in Assembly Calculus consists of several brain areas with $n$ excitatory neurons each. The neurons do not have any internal structure and can either be firing or not firing (neuron set to $0$ or $1$). The synaptic connections between neurons in different areas are drawn independently at random with probability $p$. This means they have the structure of a random bipartite graph. Synpatic connections between neurons in the same area follow the structure of an Erdos-Renyi graph $G_{n,p}$. While brains have excitatory as well as inhibitory neurons, this model only has one type of neuron that fires depending upon its total synaptic input. The function of inhibitory neurons is instead implemented by a *cap* operation, which ensures that no more than $k$ neurons out of the $n$ in a specific brain area will fire at a particular timestep. The final key component in this model is the multiplicative Hebbian plasticity, that increases the strength of connection between neurons that fire in consecutive timesteps by a factor of $\beta$.

In this note, we investigate the role of recurrent connections in the model of Assembly Calculus as posited in [1]. We consider two models in the Assembly Calculus - one with recurrent synaptic connections within a brain area and another containing only feedforward synaptic connections. We evaluate their performance on two tasks - the formation of assemblies and their ability to classify stimuli from different distributions. We observe that in both cases, models without recurrent connections are able to perform the tasks as well as those with recurrent connections. We then investigate the robustness of neural assembly models to perturbations in the input and observe that recurrent connections make the models more

robust to perturbations in the input.

Finally we investigate the spectral structure of the synaptic weights of the neural assembly models and observe that one can recover the stimuli and corresponding neural assembly from the right and left singular vectors respectively. This suggests that the model of the brain proposed in the Assembly Calculus is close to an associative memory.

# 2 Creation of Assemblies

## 2.1 Single stimulus

Assembly projection is the primary operation of the Assembly Calculus. It entails that through projection of a stimulus assembly $x$ in area $S$, an assembly $y$ that can be thought of as a "copy" of $x$ forms in a downstream area $A$, and henceforth $y$ fires every time $x$ fires [1]. The idea is that, with repeated firing of $x$, afferent synaptic connectivity from area $S$ to area $A$ excites a sequence of $\{y^{(t)}\}$ of sets of neurons of size $k$ in area $A$. With large enough parameters and high enough plasticity, this process converges exponentially fast, with high probability, to create an assembly $y$ as a result of the projection, as proven by previous literature [1].

More formally, If we consider a brain area $A$ receiving synaptic input from an assembly $x$ in stimulus area $S$, we can write the dynamics of the projection operation as:

$$\begin{aligned} y^{(t+1)} &= \text{cap}_k\left(W_{AA}^{(t)} y^{(t)} + W_{AS}^{(t)} x\right) \\ W_{AS}^{(t+1)} &= W_{AS}^{(t)} + \beta y^{(t+1)} x^\top \odot W_{AS}^{(t)} \\ W_{AA}^{(t+1)} &= W_{AA}^{(t)} + \beta y^{(t+1)} y^{(t)\top} \odot W_{AA}^{(t)} \end{aligned} \tag{1}$$

In the above set of equations, $W_{AS}$ and $W_{AA}$ refer to the weights of the afferent and recurrent synaptic connections respectively. $y^{(t)}$ refers to the pattern of activity in $A$ at time $t$, and at the end of $T$ rounds of firing, an assembly $y$ is created as the result of the operation $\texttt{project}(x, A)$. In each round, the synaptic weights are updated according to the rule of Hebbian plasticity, by a factor of $\beta$.

If we remove the recurrent connections, the update equations governing the project operation become:

$$\begin{aligned} y^{(t)} &= \text{cap}_k\left(W_{AS}^{(t)} x\right) \\ W_{AS}^{(t+1)} &= W_{AS}^{(t)} + \beta y^{(t)} x^\top \odot W_{AS}^{(t)} \end{aligned} \tag{2}$$

We performed experiments to show the impact of recurrent connections on the creation of an assembly $y$ through the projection of stimulus assembly $x$. We used a neural assembly network with assembly size $n = 1000$, cap size $k = 41$, and synaptic connection probability $p = 0.1$. We observed the total number of neurons activated in all rounds of firing, and declared that an assembly was formed when no new neurons were added to the set of activated neurons. We used different Hebbian plasticity values, $\beta = \{0.001, 0.005, 0.01, 0.05, 0.1\}$ and compared the creation of assemblies with and without recurrent connections (under update equations (1) and (2) respectively).

The results of the experiment with and without recurrent connections are both presented in Figure 1. As shown, though assemblies can be formed in models with recurrent connections, they are not necessary. Assemblies can also be formed simply using afferent connections and synaptic Hebbian plasticity. In fact, the creation of the assembly without recurrent connections is faster, since Hebbian plasticity ensures that the set of neurons selected in the cap during the first round will continue to be selected in the cap in all subsequent rounds of firing.

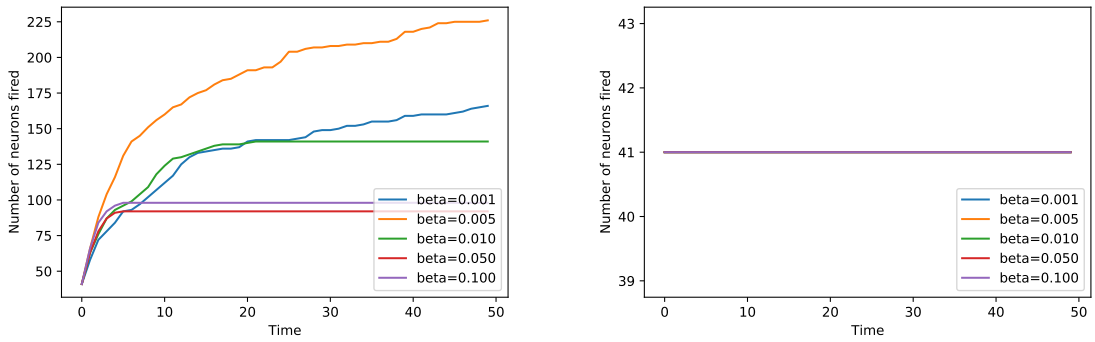


Figure 1: The size of the assembly $y$ being formed with (left) and without (right) recurrent connections under a single stimulus.

**Proposition 2.1.** *In a network with only afferent connections, the total number of neurons activated across all rounds of Random Projection and Cap is $k$.*

*Proof.* The synaptic input into each neuron in the brain area that receives input from the stimulus in the first round is a binomial random variable. Using the Normal approximation to the binomial distribution, and the tail bound on the Normal, we obtain that the activation threshold at the first time step is close to

$$C_1 = pk + \sqrt{2pk\ln\frac{n}{k}}$$

Now consider the second round of firing - due to Hebbian plasticity, each neuron in the cap after the first round will receive a synaptic input that is larger than the input it received in the previous round by a factor of $1+\beta$, and each neuron that is not in the cap, will only receive the same synaptic input that it received in the previous round. This means that no new neurons can make their way into the cap. Extending this argument over all rounds of firing proves the result.

□

## 2.2 Stream of stimuli from the same distribution

We have shown that it is possible to create assemblies that correspond to a single stimulus, but it is unlikely that the same pattern of neurons will fire for the same stimulus in a range of environments. In this subsection we also consider the case where over the course of formation of an assembly, the stimuli may vary from one timestep to another, but are all drawn from the same distribution of neural activation patterns. This is a probabilistic generalization of the single stimuli, as noted in [10]. We show that assemblies can be formed in this situation as well, both with and without recurrent connections in the area where the assembly is formed.

The projection operation is defined in a similar fashion as equations (1), with the only modification being - the stimuli at each timestep $x^{(t)}$, are all drawn from the same distribution, instead of being the same. The distribution from which stimuli are drawn is defined as follows: a fraction $p_r (= 0.9)$ of the $k$ neurons in the stimuli are drawn from a coreset $\mathcal{S}$ of neurons, while the remaining $(1-p_r)\times k$ neurons in each stimulus are drawn uniformly at random from the neurons not in the coreset $S$. We observed the creation of assemblies from stimuli drawn from this distribution, and report our results in Figure 2. The rest of the assembly network parameters $(n, k, p)$ were set to the same values as in the previous experiment. We can see that assemblies are created as long as the plasticity parameter $\beta$ is high enough, and that assembly creation occurs both in the presence and absence of recurrent connections.

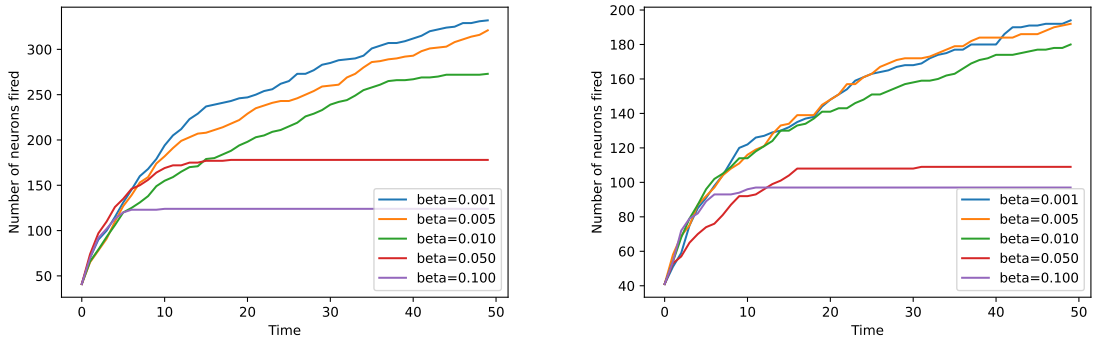


Figure 2: The size of the assembly $y$ being formed with (left) and without (right) recurrent connections under a stream of stimuli drawn from the same distribution.

# 3 Spectral Structure of Synaptic Connections

## 3.1 Single stimulus

In the previous section we saw that recurrent connections are not critical to the formation of assemblies. However, one could expect that the presence of recurrent connections could influence the structure of the synaptic connections between the stimuli and the brain area, and the rate at which assemblies are formed.

In this section we investigate the spectral structure of the synaptic connectivity matrices and show that in neural assemblies created from a single stimulus, the afferent connectivity matrix becomes a rank-$k$ matrix (where $k$ is the cap size), both with and without recurrent connections. Furthermore, the rate of convergence to a rank-$k$ matrix is not affected by the presence of recurrent connections.

We generated an assembly by running the projection operation using a stimulus $x$, and looked at the spectral structure of the afferent synaptic connectivity matrix $W_{AS}$. We use $R\left(W_{AS}\right) = \frac{\sum_{i=1}^{k} \sigma_i(W_{AS})}{\sum_{j=1}^{n} \sigma_j(W_{AS})}$ to measure how close $W_{AS}$ is to a rank-$k$ matrix, where $\sigma_i(W_{AS})$ denotes the $i^{th}$ singular value of $W_{AS}$. In Figure 4 we see that $W_{AS}$ is converging to a rank-$k$ matrix as $R\left(W_{AS}\right)$ converges to 1. This rank-$k$ structure emerges in models both with and without recurrent connections, and as Figure 4 shows, there is a negligible difference between the rate of convergence in the two cases.
We also look at the singular vectors $U, V$ of $W_{AS}$ and see that a curious relationship emerges between the singular vectors, the stimulus $x$ and the assembly $y$. For each right singular vector $v_i$, we computed the hamming distance between $\text{cap}_k(|v_i|)$ and the stimulus $x$, and for the left singular vectors $u_i$ we similarly computed the hamming distance between $\text{cap}_k(|u_i|)$ and the assembly $y$. Here $|\cdot|$ is an an entrywise absolute value operator when applied to a vector. As we see in Figure 3, the first $k$ right singular vectors are very close to $x$ while the first $k$ left singular vectors are very close to $y$. This structure appears in the afferent synaptic connectivity matrix $W_{AS}$, once again in both the presence and absence of recurrent connections. When we performed a similar analysis on the recurrent synaptic connectivity matrix $W_{AA}$, we observed that both the left and right singular vectors are close to the assembly $y$ in the sense described above. We present these observations in Figure 5.

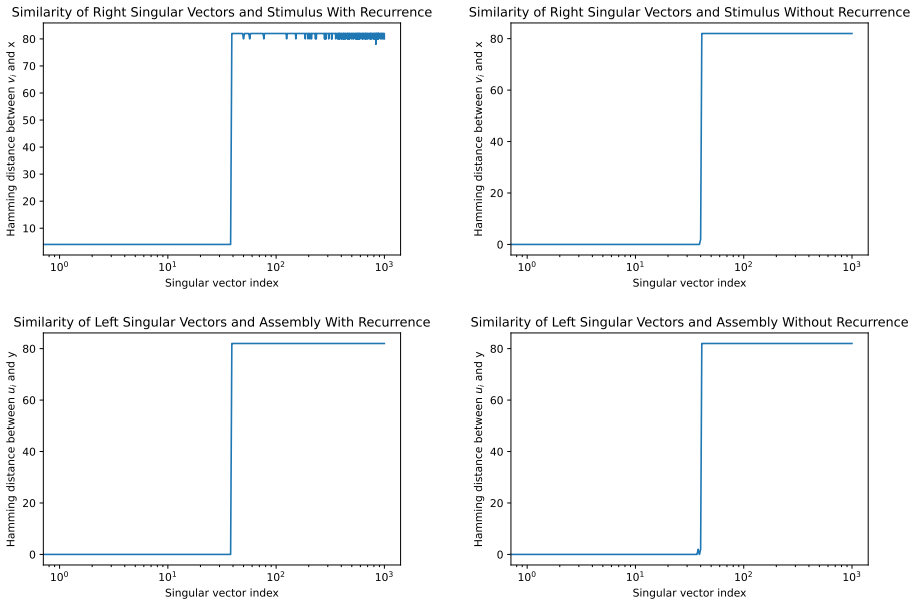


Figure 3: The hamming distance between the left singular vector $u_i$ of $W_{AS}$ and $y$ with recurrent connections (top left) and without (top right), and between the right singular vector $v_i$ of $W_{AS}$ and $x$ (top two) with recurrent connections (bottom left) and without (bottom right). As shown, the left singular vectors $u_1, \ldots, u_k$ converged to $y$ and the right singular vectors $v_1, \ldots, v_k$ converged to $x$. Further, the presence of recurrent connections showed little effect on the convergence result as almost no difference can be observed between two sides of the figures. Assemblies created using a single stimulus.

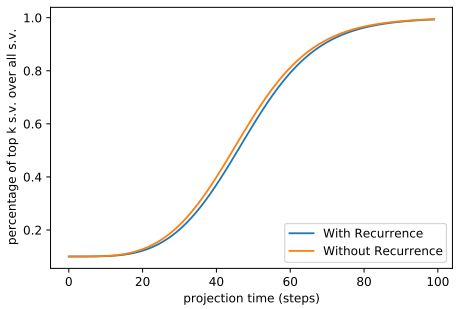


Figure 4: The trajectory of $W_{AS}$ converging to a rank-$k$ matrix by measuring the ratio of the sum of top $k$ singular values to the sum of all $n$ singular values of $W_{AS}$. Assemblies created using a single stimulus.

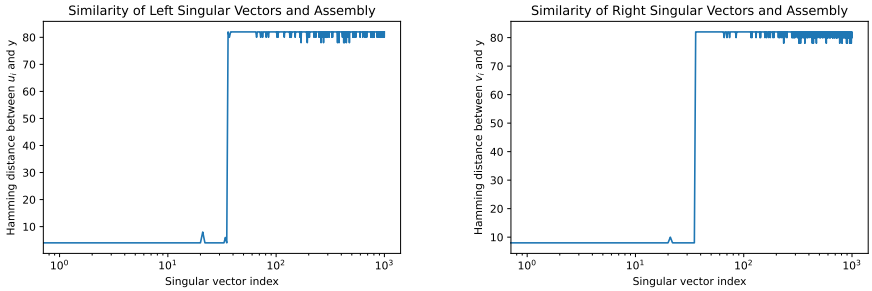


Figure 5: The hamming distance between the left singular vectors $U$ of $W_{AA}$ and $y$ (left), and between the right singular vectors $V$ of $W_{AA}$ and $y$ (right). Assemblies created using a single stimulus.

## 3.2 Stream of stimuli from the same distribution

Having seen the special spectral structure that emerges in the assembly network during the creation of an assembly from a single stimulus, we now study the same phenomena when assemblies are created by a stream of stimuli drawn from the same distribution. The distribution of stimuli that we consider is the same as the one described in the previous section. We see from Figures 6, 7, and 8 that the results are similar in essence to those seen in the case of a single stimulus. We note that comparisons to the left singular vectors are made against the neurons in the coreset $\mathcal{S}$ instead of any particular $x^{(t)}$ drawn from the distribution.

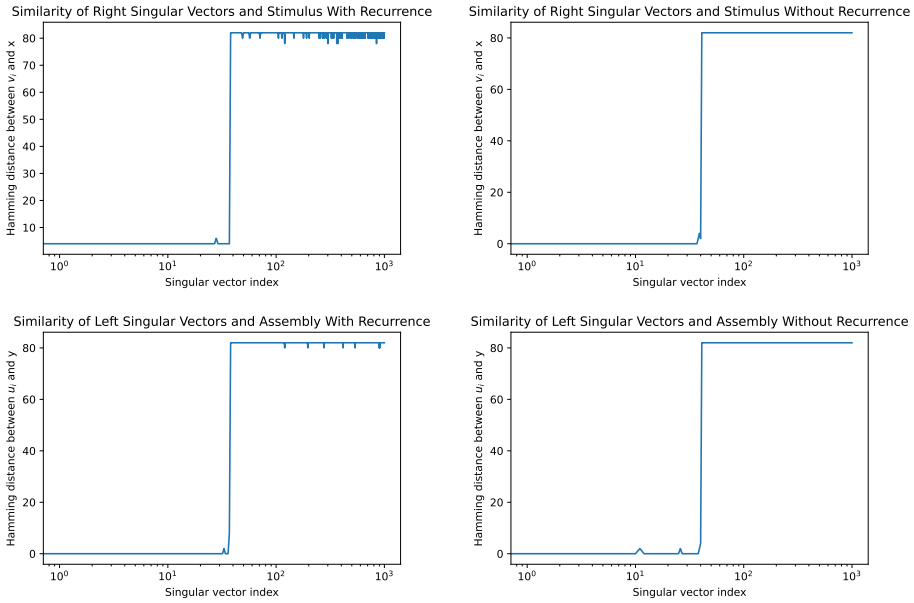


Figure 6: The hamming distance between the left singular vector $u_i$ of $W_{AS}$ and $y$ with recurrent connections (top left) and without (top right), and between the right singular vector $v_i$ of $W_{AS}$ and $\mathcal{S}$ (top two) with recurrent connections (bottom left) and without (bottom right). As shown, the left singular vectors $u_1, \ldots, u_k$ converged to $y$ and the right singular vectors $v_1, \ldots, v_k$ converged to $\mathcal{S}$. Further, the presence of recurrent connections showed little effect on the convergence result as almost no difference can be observed between two sides of the figures. Assemblies created using a stream of stimuli from the same distribution

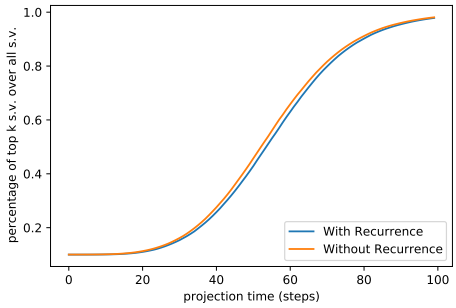


Figure 7: The trajectory of $W_{AS}$ converging to a rank-$k$ matrix by measuring the ratio of the sum of top $k$ singular values to the sum of all $n$ singular values of $W_{AS}$. Assemblies created using a stream of stimuli from the same distribution

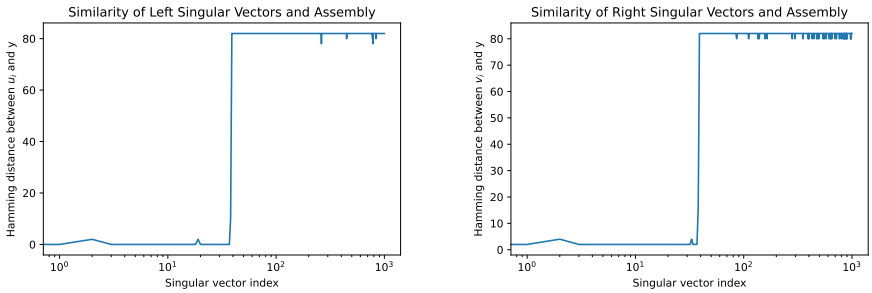


Figure 8: The hamming distance between the left singular vectors $U$ of $W_{AA}$ and $y$ (left), and between the right singular vectors $V$ of $W_{AA}$ and $y$ (right). Assemblies created using a stream of stimuli from the same distribution

# 4 Assemblies created *without* recurrent connections are well separated

Previously, Dabagia, Papadimitriou, and Vempala [10] rigorously showed that Assembly Calculus provides a learning mechanism for classifying samples from reasonably separated classes. We highlight here in particular that AC can, *without* recurrent connectivity, reliably form an assembly that represents each concept class in response to (a) a few stimuli when they are clusters of similar assemblies, or more generally (b) divided by a halfspace with margin, both with experimental evidences. Without recurrent connectivity, this learning mechanism still exhibits all key attributes of brain-like learning mechanism: it is entirely online and generalized from very few samples with only mild supervision.

We followed the methodology purposed in [10]. Both experiments for (a) stimulus classes and (b) halfspace-separated shared the basic setup of presenting a few examples from the same class and allowing plasticity to train synaptic weights. Additionally, the incoming weights of each neuron are re-normalized to sum to one after the conclusion of presenting each class, based on the assumption that synapses are subject to homeostasis in between training and evaluation [11]. We performed classification on top of the learned assemblies by predicting which class corresponds to the assembly with the most neurons on. We had success with large enough plasticity parameter $\beta = 0.1$ for stimulus classes and $\beta = 1.0$ for halfspaces.

## 4.1 Classification in stimulus classes without recurrent connections

A *stimulus* is an assembly-like representation, defined as a set of $k$ firing neurons $x$, that encodes training or testing data in a special area $S$ called the *sensory area*. The learning process happens in another brain area $A$, of which only one area is required, through the formation of assemblies in response to the projection of sequences of stimuli from a stimulus class $C$ in the sensory area. The projection evokes a *response* $R$, a distribution over assemblies in the brain area $A$. Due to plasticity and inhibition, $R$ tends to be highly concentrated. We call the intersection $R^* = \bigcap_{x \in \mathcal{S}} x$ the *core* of $R$ for set $S_R$ of all assemblies x that have positive probability in $R$. In particular, neurons in $R^*$ fire significantly more often on average than other neurons [12]. Note that a *stimulus* class $C$ is a distribution over stimuli defined by two parameters $r, q \in [0, 1], r > qk/n$ and the set of k neurons $S_A$ in the sensory area. To generate a stimulus $x \in \{0, 1\}^n$ within the distribution $C$, each neuron $i \in S_A$ is chosen independently with probability $r$, while each neuron $i \notin S_A$ is chosen independently with probability $qk/n$ [10].

Here, we demonstrate that Assembly Calculus can learn to classify reasonably separated stimulus classes *without* recurrent connectivity as follows: we chose a stimulus class $C$, randomly sample and present a new stimulus from $C$ at each time step for a fixed number of rounds. We repeated this procedure for different stimulus classes to solely train afferent synaptic weights. Finally, we presented new stimuli in a random order of different stimulus classes trained to test the extent of learning.

In figure 9, we demonstrate learning of classifying four stimulus classes, with and without recurrence connections, both reach perfect accuracy. Most importantly, we highlight Assembly Calculus can learn to classify stimulus classes perfectly *without* recurrent connectivity. Figure 10 also reinforces this message by visualizing the learned assemblies and the extent of their overlap when learned with stimuli from the same class.

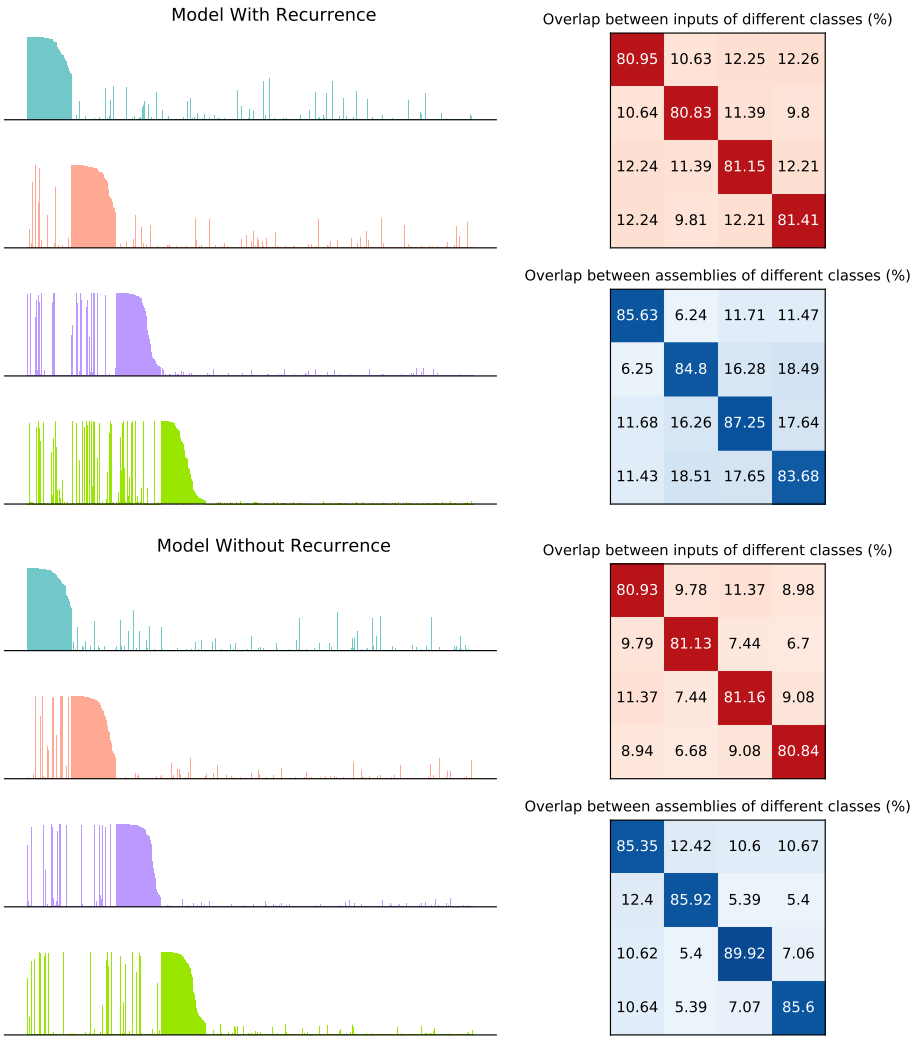


Figure 9: Assemblies learned for four stimulus classes with (left) and without (right) recurrent connections. For each of the two subfigures, the distributions of firing probabilities over neurons are on the left, and the average overlap of the assemblies are on the right. Despite each additional class slightly overlap with previous ones, a simple readout over assembly of neurons suffices perfect classification accuracy in both cases. Little to no difference can be observed between the two cases. Here, $n = 10^3, k = 10^2, p = 0.1, r = 0.9, q = 0.1, \beta = 0.1$, with 5 samples per class. [10]

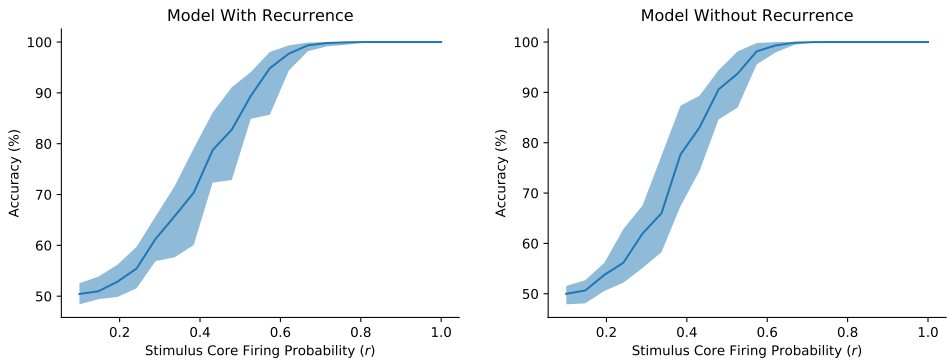


Figure 10: Accuracy of classifying stimulus classes as a function of stimulus core firing probability $r$ with (left) and without (right) recurrence. [10]

## 4.2 Classification in halfspaces without recurrent connections

The halfspace classification experiment shares the same setup as what is described in section 4.1 [10]: it is accomplished by presenting classes of stimuli, along with plasticity and inhibition mechanisms. Here, we consider a linear threshold function as the labeling function, parameterized by an arbitrary non-negative vector $v$ and margin $\Delta$. We create a single assembly to represent examples on one side of the threshold and denote their distribution by $\mathcal{D}_+$, where each coordinate is an independent Bernoulli variable with mean $\mathbb{E}(X_i) = (\frac{k}{n} + \Delta v_i)$. We denote the distribution of negative examples $\mathcal{D}_-$, in which each coordinate is an independent, yet identically distributed Bernoulli variable with mean $k/n$. Note that $\mathcal{D}_+$ and $\mathcal{D}_-$ share the same support. For the purpose of classifying, a fraction $1 - \epsilon_+$ of neurons is guaranteed to fire for a positive example and a fraction $\epsilon_- < 1 - \epsilon_+$ of neurons is guaranteed *not* to fire for a negative example. We classify a test example as positive if at least a $1 - \epsilon$ fraction of neurons fire and as negative otherwise for $\epsilon \in [\epsilon_-, 1 - \epsilon_+]$.

In contrast to the experiment on stimulus classes, we only present positive examples in halfspace experiment, and hence only one assembly is formed for one class. We classify by comparing against half the cap size. Using a simple sum readout over assembly neurons, halfspaces are classified with perfect accuracy both with and without recurrent connections as shown in figure 11. We can however see that the overlap between the assemblies of the positive examples is much larger in the case of the model with recurrent connections when compared to the one without. Figure 12 also shows us that the classification accuracy is similar between the two models and finds similar trends as the margin $\Delta$ increases.

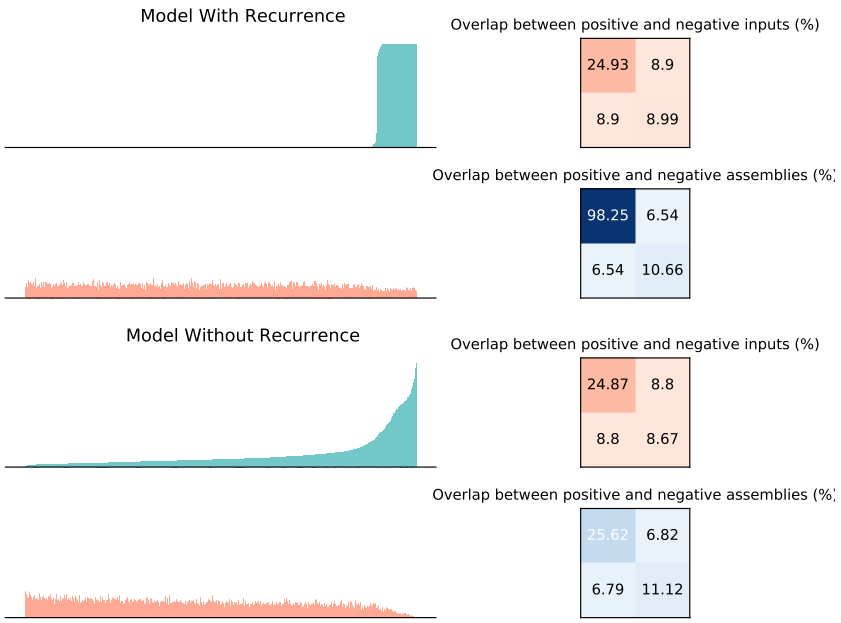


Figure 11: Assemblies learned for the positive and negative classes with (left) and without (right) recurrent connections. For each of the two subfigures, the distributions of firing probabilities over neurons are on the left, and the average overlap of the assemblies are on the right. Despite each additional class slightly overlap with previous ones, a simple readout over assembly of neurons suffices to achieve perfect classification accuracy in both cases.

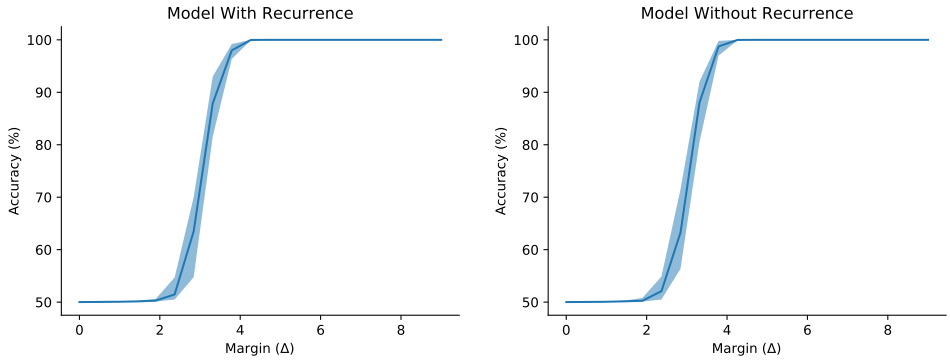


Figure 12: Accuracy of classifying stimulus classes as a function of the margin $\Delta$ between the positive and negative classes with (left) and without (right) recurrent connections. [10]

# 5 Assemblies created *with* recurrent connection are more robust in assembly recall

A natural question immediately follows the experimental results shown in section 2, 4 - what *useful* role do recurrent connections hold in Assembly Calculus, if they had little to no impact on the convergence of afferent connections and classification performance?

In this section we identify a key role for the recurrent connections in Assembly Calculus - providing robustness in *assembly recall*. We show that assemblies created with recurrent connections are able to recover the created assembly from a corrupted input more robustly than those created with just afferent connections.

## 5.1 Single Stimulus

In order to investigate the robustness of assembly recall, we first trained the assembly network with a single stimulus $x_0 \in \{0,1\}^n$ to form an assembly $y_0$ in the brain area $A$ by firing for $T_{train}$ steps. We then probed the assembly network with a corrupted input $x$ that is a certain hamming distance away from the input stimulus $x_0$. We recovered an assembly $y$ by projecting the corrupted input $x$ for $T_{recover}$ steps. The robustness of assembly recall was measured using the hamming distance between $y$ and $y_0$.
Denote the hamming distance between input $x$ and $x_0$ as $d_H(x, x_0)$. Experimental evidence in figure 13 shows that, despite recalling in a network trained *without* recurrent connections displayed slightly better performance in $d_H(x, x_0) \lessapprox k/2$, a network trained *with* recurrent connections exhibited dominantly better performance and provided more robustness under larger $d_H(x, x_0)$.

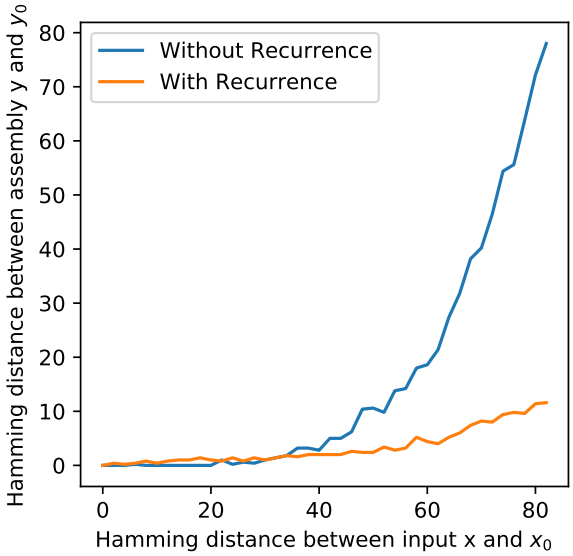


Figure 13: Assembly performance in recalling the original formed assembly $y_0$ when probed by a corrupted input $x$ that is $d_H(x, x_0)$ away from the original input stimulus $x_0$. Here, we used $n = 1000, k = 41, p = 0.03$, $\beta = 0.05$, $T_{train} = 40$, $T_{recover} = 10$. Results are averaged over 10 trials. Results are shown for models with and without recurrent connections, and the models with recurrent connections are more robust.

## 5.2 Stream of stimuli from the same distribution

We repeated our experiment testing the robustness of assembly recall in the situation where the input stimuli are not fixed through all timesteps, but are instead drawn from the same distribution. We used the same setup as in sections 2.2 and 3.2 where the stimuli are drawn from a distribution where a $p_r$ fraction are drawn from a coreset $\mathcal{S}$, and the rest are drawn uniformly at random. We measured the robustness of assembly recall to two perturbations from this distribution. In the first experiment (results in Figure 14) we kept the coreset $\mathcal{S}$ the same for the corrupted stimuli, but changed the fraction $p_r$ that was drawn from the coreset. In the second experiment (results in Figure 15) the hamming distance between the coreset $\mathcal{S}'$ for the corrupted stimuli and the coreset $\mathcal{S}$ for the original stimuli was varied from 0 to $2k$.
In both experiments, as shown in the figures, we see that the models with recurrent connections are more robust to pertubations, even in the case where the assemblies are created by stimuli drawn from a distribution rather than a single stimulus.

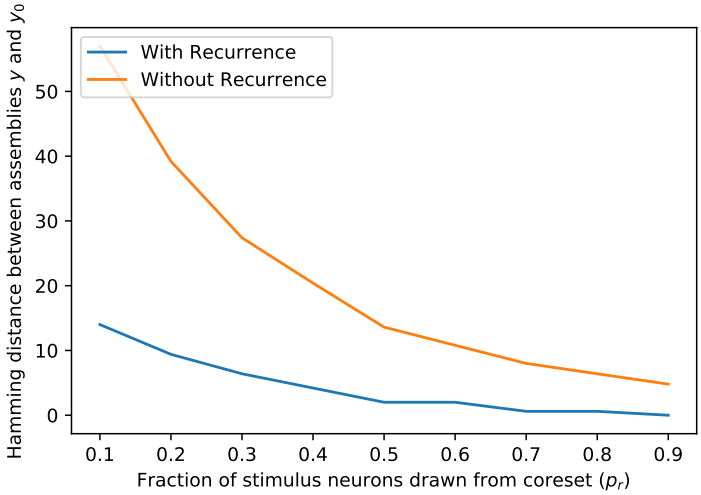


Figure 14: Assembly performance in recalling the original formed assembly $y_0$ when probed by a stream of corrupted inputs $x$ that contain a different fraction $p_r$ of the coreset than the stream $x_0$ that was used to create the assembly. Here, we used $n = 1000, k = 41, p = 0.03$, $\beta = 0.05$, $T_{train} = 40$, $T_{recover} = 10$. Results are averaged over 10 trials. Results are shown for models with and without recurrent connections, and the models with recurrent connections are more robust.

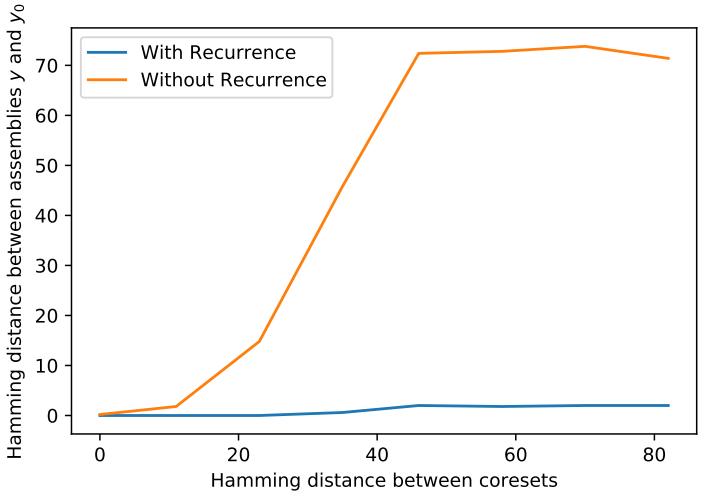


Figure 15: Assembly performance in recalling the original formed assembly $y_0$ when probed by a stream of corrupted inputs $x$ that are drawn from a coreset $\mathcal{S}'$ that differs by a certain hamming distance from the coreset $\mathcal{S}$ of the original stream $x_0$ that was used to create the assembly. Here, we used $n = 1000, k = 41, p = 0.03$, $\beta = 0.05$, $T_{train} = 40$, $T_{recover} = 10$. Results are averaged over 10 trials. Results are shown for models with and without recurrent connections, and the models with recurrent connections are more robust.

## 6 Recurrent connections alone can complete patterns

We highlight the significant role recurrent connections hold in *pattern completion*. Pattern completion is an important and well-studied phenomenon involving assemblies: the firing of the whole assembly $y$ in response to the firing of a small subset of its cells [13]. We emphasize recurrent connections are crucial to pattern completion, in which it alone suffices to complete the whole assembly very accurately by its

subset with enough reinforcement of the original assembly [1]. Here, we demonstrate an experiment that trains a standard assembly network with both afferent connection $W_{AS}$ and recurrent synaptic connection $W_{AA}$ for $T_{train}$ steps to form an assembly $y_0 \in \{0,1\}^n$, in which $y_0$ consists of k firing neurons emerged from projecting with Hebbian multiplicative plasticity and inhibition. Then, we perturb $y_0$ by leaving only $\alpha$ fraction of k firing neurons on for $\alpha \in [0,1]$ to form $y$: among $k$ original winners, $\alpha k$ of them are randomly sampled to be left on without replacement and consequently the rest are turned off. We attempt to recover $y_0$ by setting $y^{(0)} = y$ and performing random projection and cap with the recurrent weights $W_{AA}$ for $T_{recover}$ steps. Mathematically, the neurons in brain area $A$ evolve as:

$$y^{(t+1)} = \text{cap}_k \left( W_{AA}^{(t)} y^{(t)} \right)$$

After, we examined the performance of pattern completion by measuring the percentage of assembly recovered by measuring $\frac{|S_{y_0} \cap S_y|}{|S_{y_0}|}$, where $S_y$ denotes the set of $k$ winners in assembly $y$. The model showed promising results in figure 16 that recurrent terms alone can complete the assembly. With $\alpha = 0.4$, $\beta = 0.05$, $p = 0.03$, more than half of the assembly was recovered with $T_{train} = 25$; as the extent of reinforcement increases, ie. when $T_{train} = 45$, more than 85% of the assembly was recovered. Similarly, with same $\alpha$ and $\beta$, and $p = 0.08$, the recurrent terms completed pattern near perfectly.

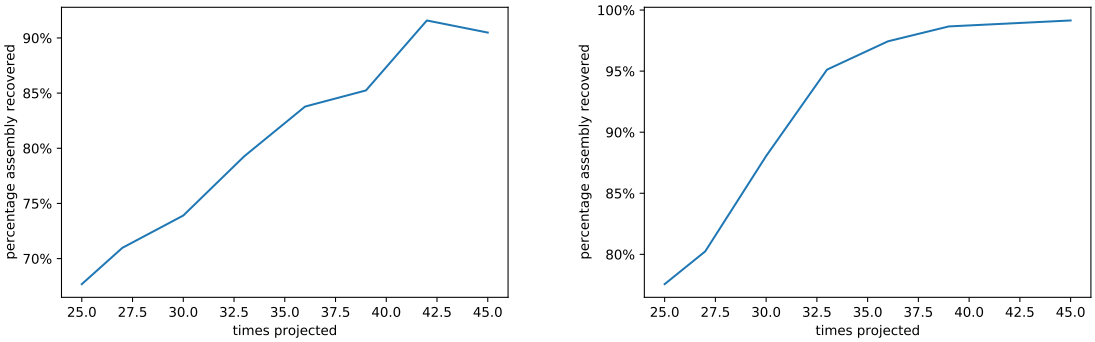


Figure 16: Pattern completion with $n = 1000, k = 41, T_{train} = \{25, 27, 30, 33, 36, 39, 42, 45\}, T_{recover} = 2, \alpha = 0.4$, and $\beta = 0.05$. Here, we used $p = 0.03$ on the left, and $p = 0.08$ on the right. We replicated the experiment on each $T_{train}$ values for 20 trials and took the average to distinguish the result from random events.

# 7 Discussion

In this note, we have conducted various experiments on the Assembly Calculus (AC) model to explore the role of recurrent synaptic connections. We see that the absence of recurrent connections does not significantly alter assembly formation (Section 2) and the spectral structure of the synaptic connections (Section 3). These observations were observed both in assemblies created from single stimuli, as well as those created from a stream of stimuli all drawn from the same distribution. In section 4, we also saw that models without recurrent connections are able to classify stimuli from different distributions as well as those with recurrent connections, with a notable difference. The assemblies for stimuli in the same class that are created with recurrent connections have a larger overlap than those with created without recurrent connections.

This observation is also consistent with our experiments in Section 5, where we test the robustness of assembly recall in models with and without recurrent connections. We observe that models with recurrent connections are more robust to noisy inputs and shifting distributions. The models with recurrent connections are able to recall assemblies from stimuli that are much more noisier than models without recurrent connections.

An initial hypothesis that can explain our observations on the role of recurrent connections in AC is suggested by our observations in Section 3 that neural assembly models are akin to associative memories [14]. We were able to make this observation because the feedforward and recurrent synaptic weight matrices take on the following forms: $W_{AS} \approx \sum^k yx^\top, W_{AA} \approx \sum^k yy^\top$. We can analogize the difference between models with and without recurrent connections to the difference between single-shot and iterative associative memories. Drawing on basic results in linear systems theory [15], we can understand how models with recurrent connections, which are globally stable systems, converge to their top singular vector. This can explain how models with recurrent connections are more robust. We can also draw on linear systems theory to explain how pattern completion can be performed by models with recurrent connections. Models without recurrent connections are, of course, unable to recall assemblies from corrupted versions due to architectural limitations. This argument will be made more formal in a new version of the note. In the updated version we also hope to make a stronger connection between associative memories and AC.

# References


[1] C. H. Papadimitriou, S. S. Vempala, D. Mitropolsky, M. Collins, and W. Maass, "Brain computation by assemblies of neurons," *Proceedings of the National Academy of Sciences*, vol. 117, no. 25, pp. 14464–14472, 2020.

[2] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[4] D. O. Hebb, *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949.

[5] K. D. Harris, "Neural signatures of cell assembly organization," *Nature Reviews Neuroscience*, vol. 6, no. 5, pp. 399–407, 2005.

[6] K. D. Harris, J. Csicsvari, H. Hirase, G. Dragoi, and G. Buzsáki, "Organization of cell assemblies in the hippocampus," *Nature*, vol. 424, no. 6948, pp. 552–556, 2003.

[7] E. Pastalkova, V. Itskov, A. Amarasingham, and G. Buzsáki, "Internally generated cell assembly sequences in the rat hippocampus," *Science*, vol. 321, no. 5894, pp. 1322–1327, 2008.

[8] L. Carrillo-Reid, W. Yang, Y. Bando, D. S. Peterka, and R. Yuste, "Imprinting and recalling cortical ensembles," *Science*, vol. 353, no. 6300, pp. 691–694, 2016.

[9] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[10] M. Dabagia, C. H. Papadimitriou, and S. S. Vempala, "Assemblies of neurons can learn to classify well-separated distributions," *CoRR*, vol. abs/2110.03171, 2021.

[11] G. Turrigiano, "Too many cooks? intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement," *Annual Review of Neuroscience*, vol. 34, no. 1, pp. 89–103, 2011. PMID: 21438687.

[12] C. H. Papadimitriou and S. S. Vempala, "Random projection in the brain and computation with assemblies of neurons," in *ITCS*, 2019.

[13] J. eun Kang Miller, I. Ayzenshtat, L. Carrillo-Reid, and R. Yuste, "Visual stimuli recruit intrinsically generated cortical ensembles," *Proceedings of the National Academy of Sciences*, vol. 111, no. 38, pp. E4053–E4061, 2014.

[14] T. Kohonen, *Self-organization and associative memory*, vol. 8. Springer Science & Business Media, 2012.

[15] R. E. Kalman and J. E. Bertram, "Control system analysis and design via the "second method" of lyapunov: I—continuous-time systems," 1960.