A Sparse Distributed Model of Episodic and Semantic Spatiotemporal Memory

Gerard (Rod) Rinkus
Previously at Enkidu Research
June 25, 2004

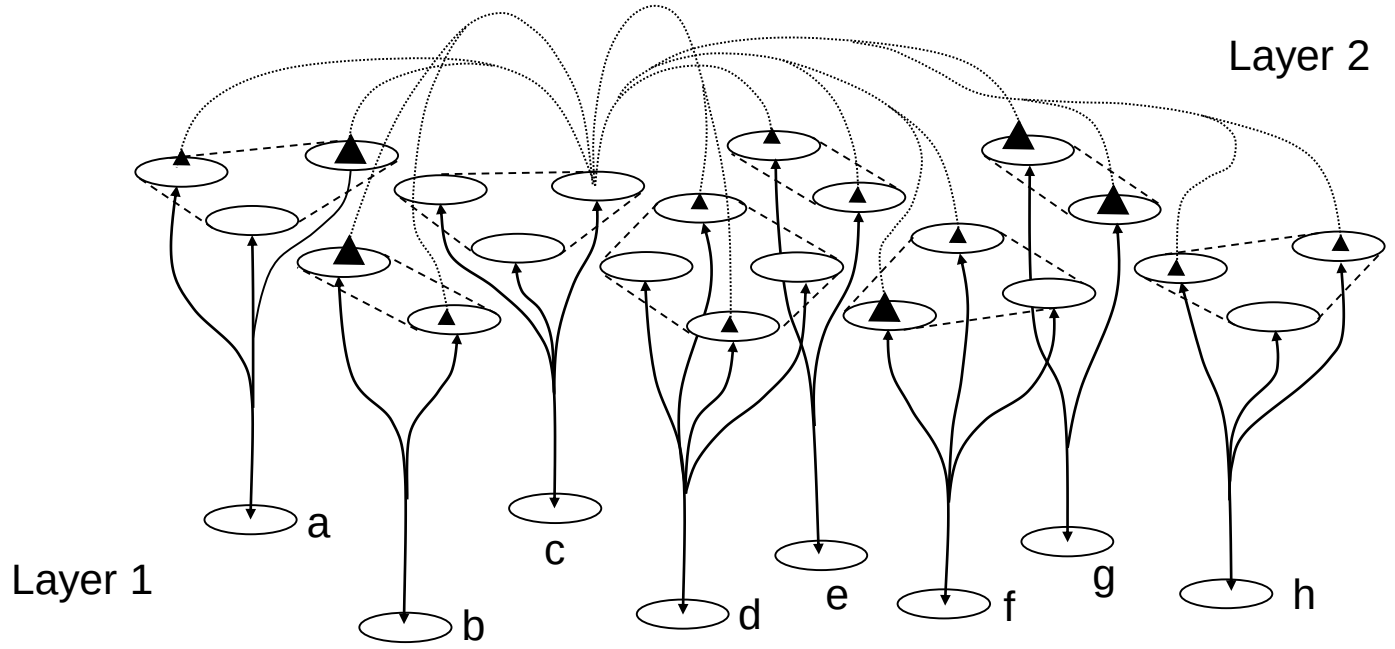
Outline

- Goals
- Describe TEMECOR
  - Temporal Episodic Memory using Combinatorial Representations
  - Sparse distributed binary model of spatiotemporal episodic memory
  - Uses associative chaining to explain sequence memory
  - Explain the capacity of episodic memory
  - Complex sequences [e.g., ABC, DBE]
  - Basic capacity result
  - Shortcomings
- Describe TESMECOR
  - Generalizes architecture of TEMECOR
  - Adds a general spatiotemporal matching algorithm that controls the choosing of internal representations
  - Example of spatiotemporal matching algorithm
  - Exhibits spatiotemporal pattern recognition and categorization
  - Allows simultaneous representation of multiple hypotheses
- Summary

Goals

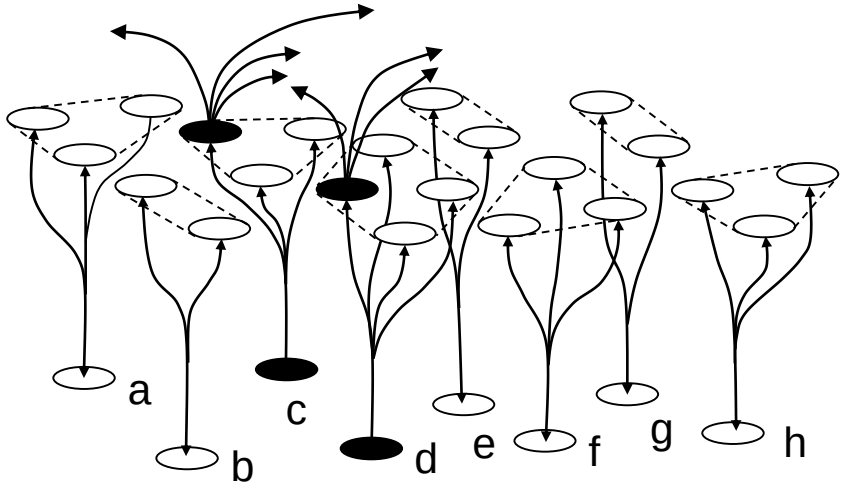
- Original
  - Explain the capacity of episodic memory
- Added Motivations
  - Explain memory for sequences (spatiotemporal patterns)
  - Explain spatiotemporal pattern recognition/categorization
  - Elucidate the relationship of episodic memory (specific instances) and semantic memory (general knowledge summarizing large numbers of instances)
  - All we experience is a stream of specific instances. How is general knowledge (lower-order statistics) represented in the brain? How are we able to use those representations?
    - Pre-computational theories
      - MLP-based models
    - Post-computational theories
      - Whittlesea, Dorken, Vokey, Brooks

TEMECOR Architecture

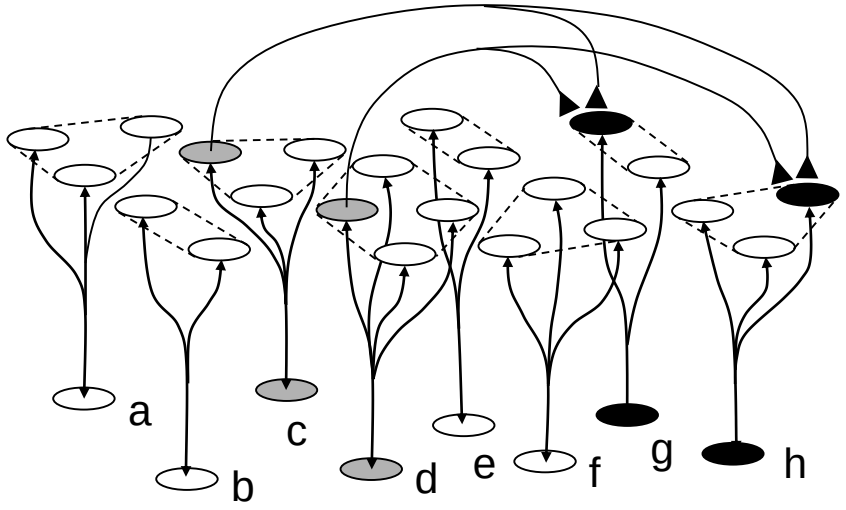


- Layer 1: Binary feature detectors
- Layer 2: Binary cells organized as winner-take-all competitive modules
- Bottom-up (forward) and top-down (reverse) connections between L1 and L2
- Modifiable horizontal (lateral) connections within L2
- Captures some general features of cortex

Embedding of Spatiotemporal Memory Trace

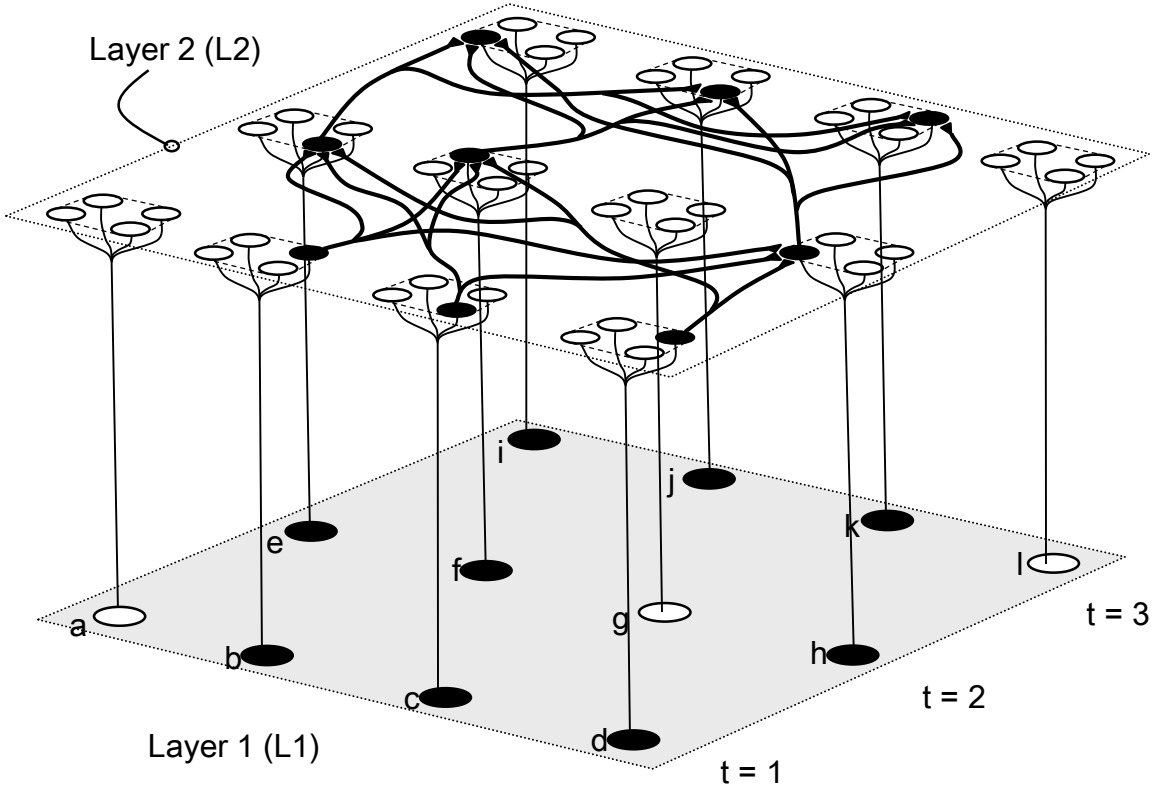


t = 1

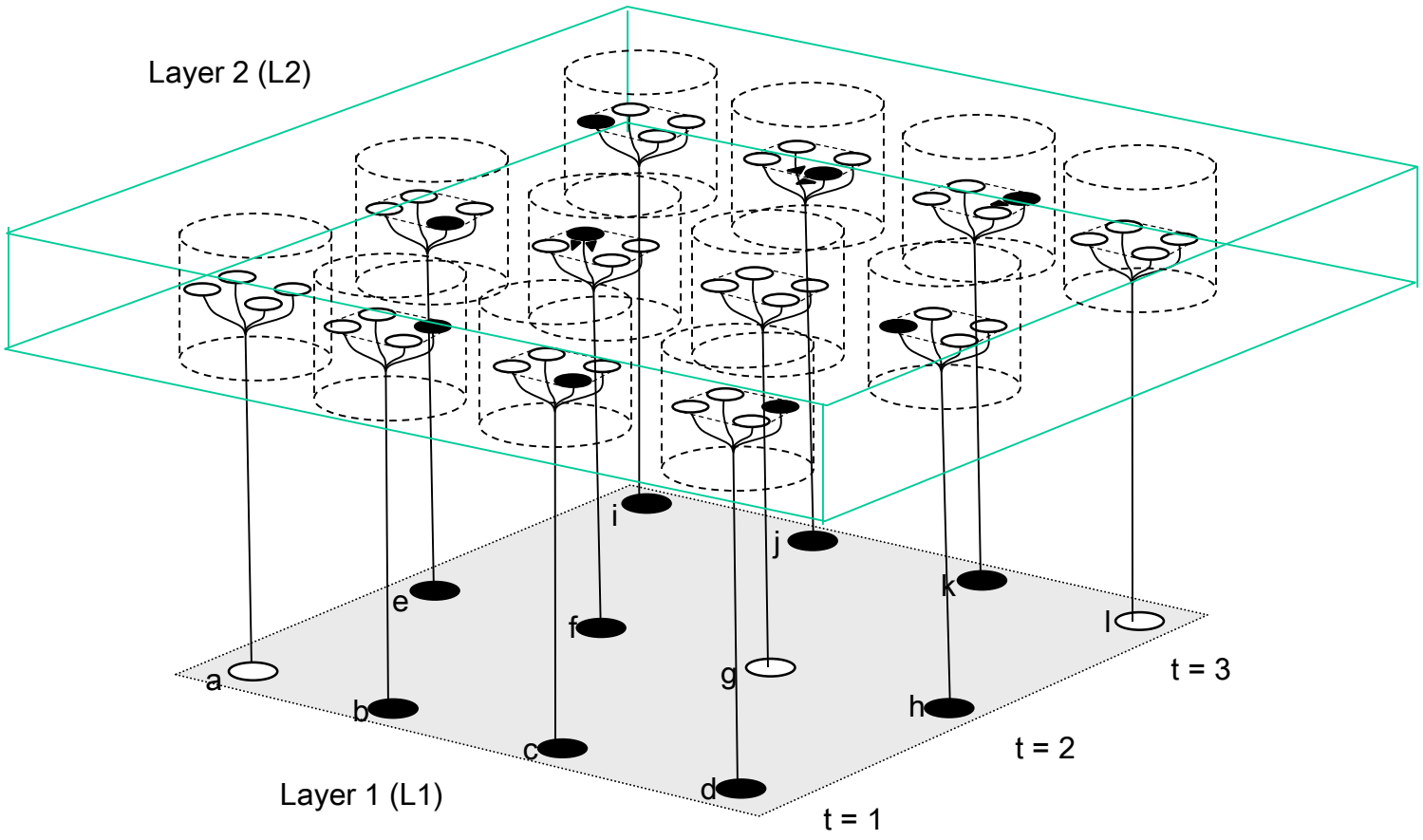


t = 2

TEMECOR architecture and swaths

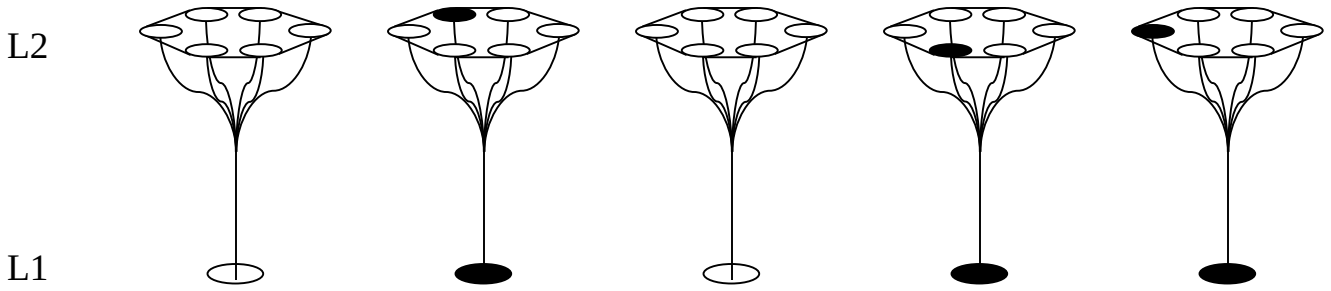


TEMECOR architecture and swaths (minicolumns)



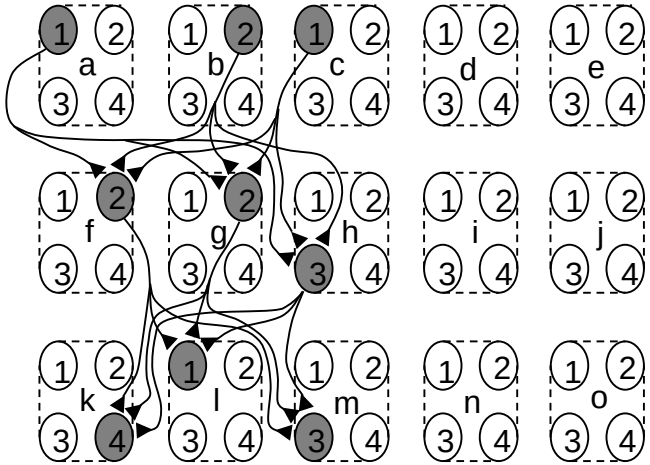
Achieving High Episodic Memory Capacity

- Maximizing capacity in a sparse distributed memory means minimizing overlap between traces.
- The Feedforward projection is a *sparsifying spatial* mapping
  - Q competitive modules (CMs) each with K cells.
    - How many unique codes are there? k^Q
    - How many codes are there such that no two intersect at more than J CMs? Unsolved, but a lot.
    - Generalizes Correlograph (Willshaw, Buneman, Longuet-Higgins, 1969)
      - k-ary as opposed binary (i.e., no CM organization)
      - Spatiotemporal as opposed to spatial
- A generalized Hamming distance



Learning Two Sequences with a Common State

Sequence (episode) 1

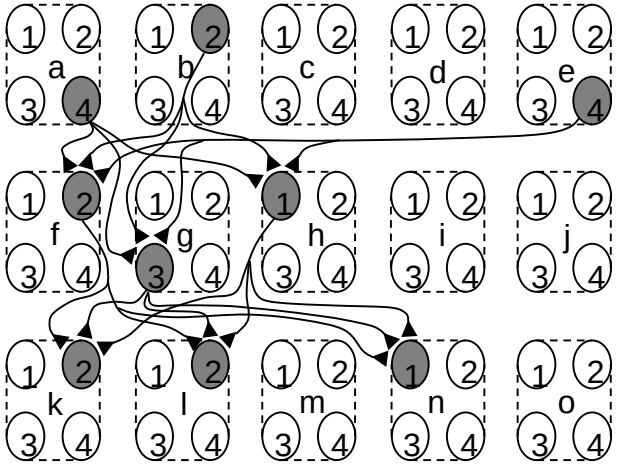


t = 1

t = 2

t = 3

Sequence (episode) 2



State A a b c d e

State **B** f g h i j

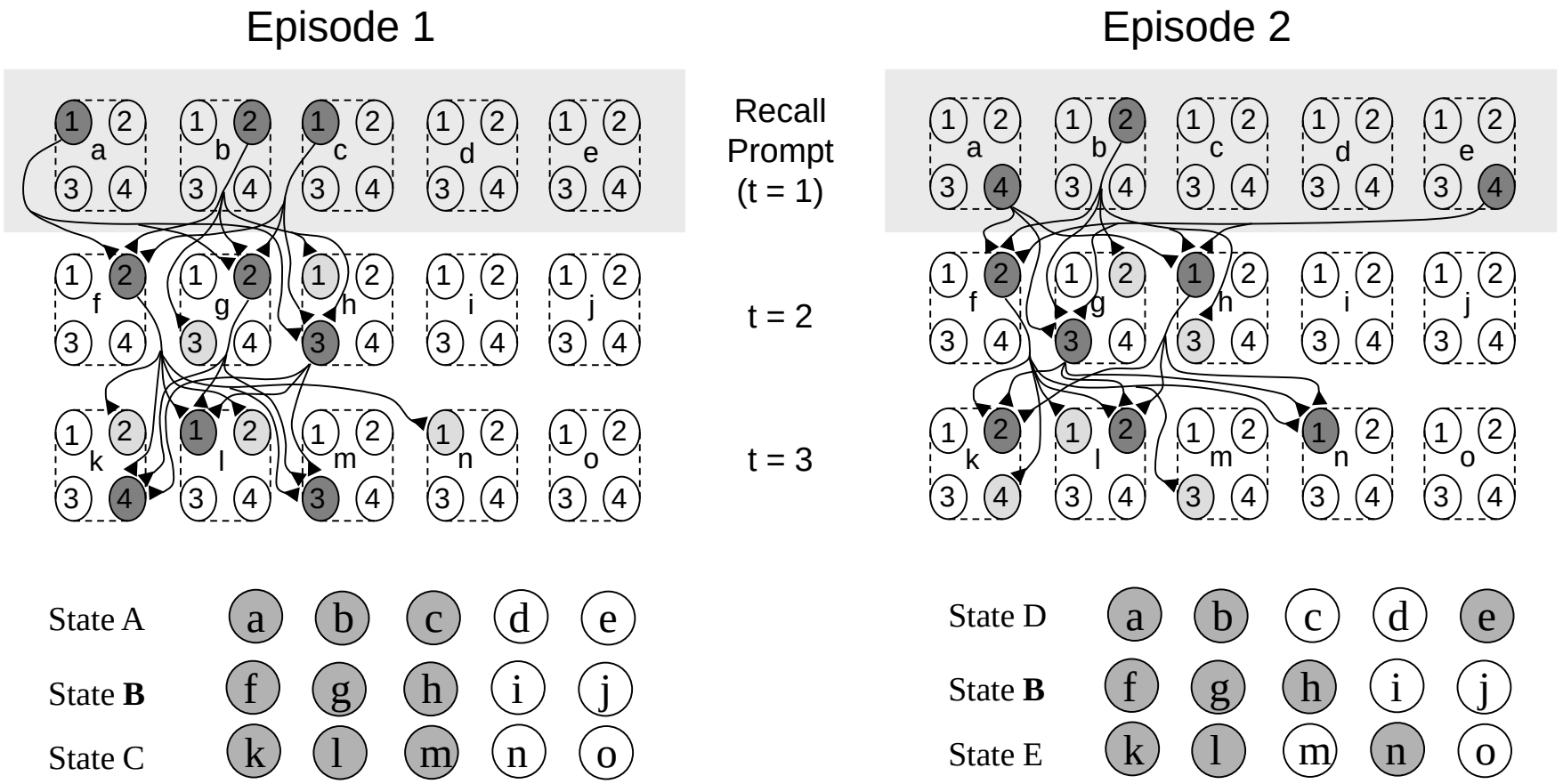
State C k l m n o

State D a b c d e

State **B** f g h i j

State E k l m n o

Recalling Two Sequences with a Common State



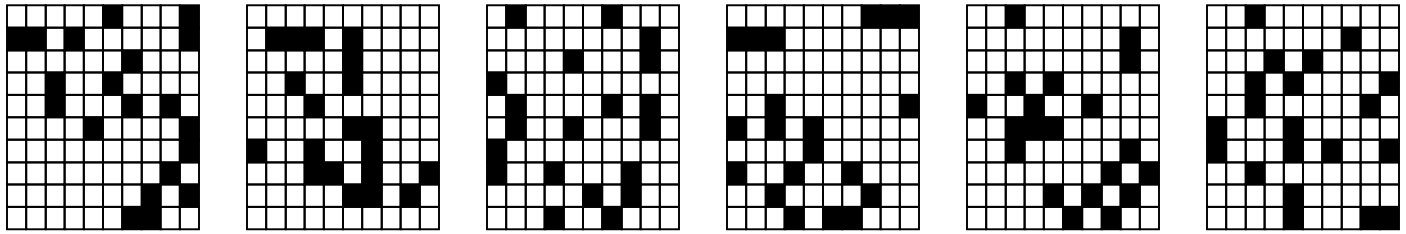
- Recall Threshold, Θ, equal to 2 or 3 prevents spurious L2 cells from firing (interference) during recall

TEMECOR Simulation Experiments

General Experimental Framework

- Generate a large set of binary spatiotemporal patterns (episodes)
- Present each episode to the model once
  - The model updates its horizontal weights in layer 2 to remember the episodes.
- Test the episodic recall of the model by presenting the first time step's pattern of each episode as a prompt and seeing how well the rest of the spatiotemporal memory trace in layer 2 reads out.
- In fact, we cheat by presenting—i.e., reinstating—the first time step's layer 2 pattern rather than the layer 1 (i.e., the actual input) pattern.
- Re-test recall performance after each new episode is learned.
- Stop when the average recall accuracy dips below some set criterion, e.g., 97%.

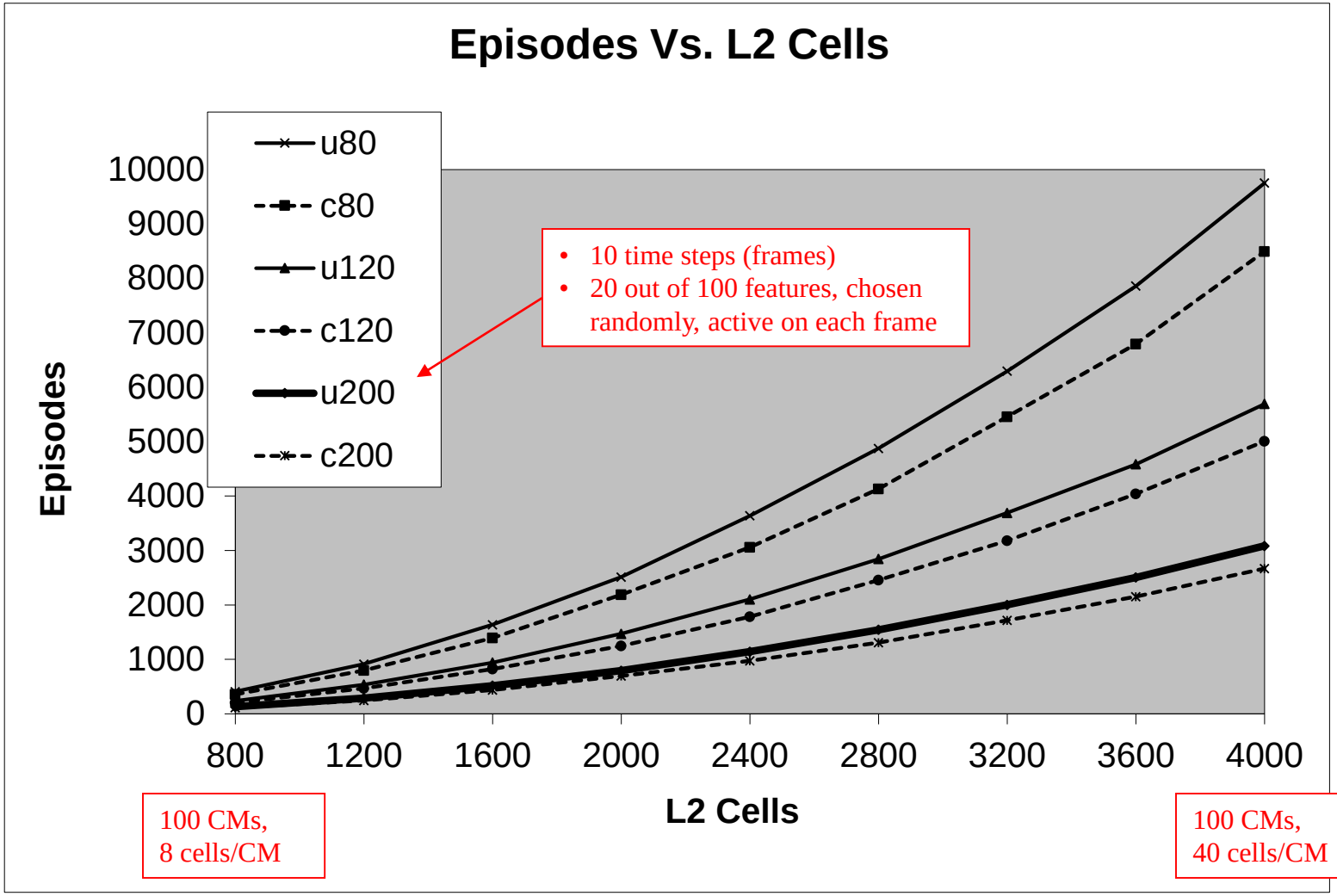
Episode Details



- Each frame has 20 out of 100 binary features active
- Experiments involved episodes with from four to ten frames.
- Model tested with patterns generated:
  - completely at random (uncorrelated case)
  - by choosing with replacement from pregenerated “alphabet” of 100 states (correlated case)
- Recall accuracy measure:

$$accuracy(e) = \frac{correct(e) - deletions(e)}{correct(e) + intrusions(e)}$$

Storage Capacity Results



- Each episode presented only once
- Faster-than-linear scaling

Example Complex Sequence Set Learned by Model

Γ^{1}	C C D A D B D C A B A C A B D B B B A A	Γ^{11}	D C D B D A A D A A B D A A A D B A A A
Γ^{2}	A A A A B C C C C C A A A D B A A A C A	Γ^{12}	A D D A C C C D A A D A C C B C C C B D
Γ^{3}	B C C B D B C B D C B D C B D D A B C C	Γ^{13}	C C C C C D C D D C D D B C D D A B C B
Γ^{4}	A C D A B A A C A C C B A B C A C A B B	Γ^{14}	B B B A D C C C A D B C B D B D C D D B
Γ^{5}	C C A B A D A A B C C B A B B C B C A B	Γ^{15}	B C A C B D B B A D C C B D C A C A C C
Γ^{6}	A C C B D A B C A C D D A A A A D A A A	Γ^{16}	D D B A D C D B B C D C B D A C D B D D
Γ^{7}	D D B D A D B C B B D B A C C D C D B D	Γ^{17}	A C B B B D C D D A C C A D C D B C A C
Γ^{8}	B B C C B C C A C D B B C C B C B C A C	Γ^{18}	A D A B D A D C B D B B B D D C C B C C
Γ^{9}	A A A A C C D A C B D D C B B D D A D C	Γ^{19}	B A A B D D A A B B C B A C D D C C B D
Γ^{10}	D A D B D A D A D A D D A D D C B C C D	Γ^{20}	B D A C C D B A A D C C D D D C C A C D

- Model had 100 CMs with 16 cells each
- Four possible states, A, B, C, D
- States had 25 out of 100 features active
- Sequences had 20 states
- Total of 20 sequences
- Total of 10,000 featural occurrences
- Recall Accuracy ≈ 100%

Related Models

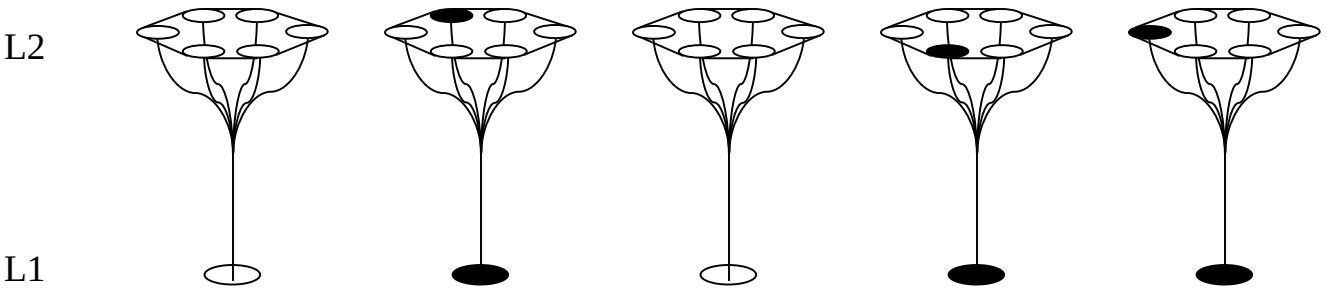
- Willshaw, Buneman & Longuet-Higgins (1969)
- Marr (1969)
- Albus (1971)
- Moll & Miikkulainen (1993, 1995)
- Palm (1980)

- All these models utilized same idea of distributed, overlapped patterns and a threshold for avoiding interference between stored memories
- However, all were developed only for the spatial pattern domain rather than the spatio-temporal domain

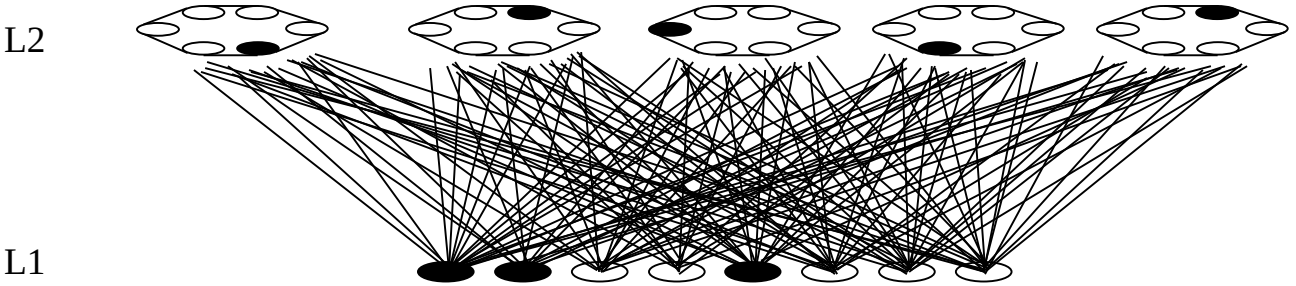
Shortcomings of TEMECOR

- For recall testing, requires prompts at L2 (internal representation layer)
  - because vertical weights not modifiable
  - therefore no way for L1 pattern to become associated with a particular L2 code
- Does not exhibit similarity-based generalization and categorization
  - because L2 codes chosen at random (high noise) rather than as a function of inputs, via either horizontal or bottom-up weight matrices
- Captures several key properties of episodic memory but not of semantic memory, nor of the interaction between episodic and semantic memory

Two Wiring Schemes



TEMECOR



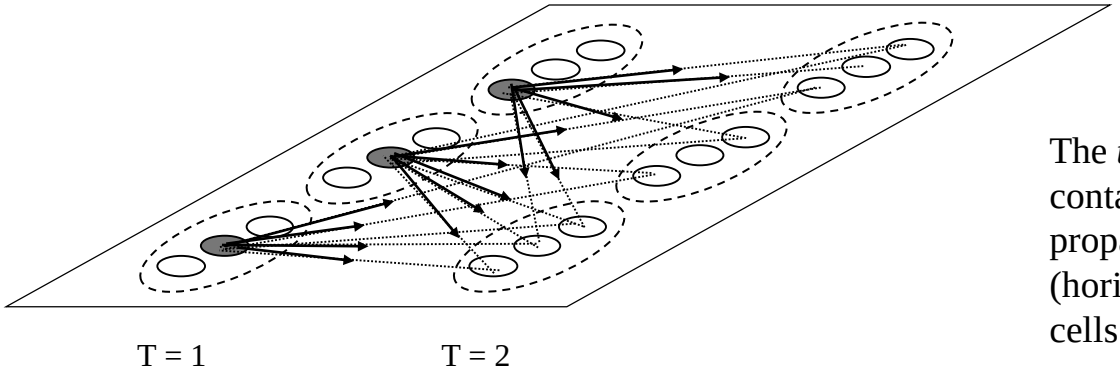
TESMECOR

Question: How do you combine the two vectors, the feedforward (bottom-up) vector and the horizontal vector?

- This led to the development of the spatiotemporal similarity computation
  - The horizontal weight matrix carries the *expectation*, based on the *spatiotemporal context*, as to what the current input should be
  - The bottom-up weight matrix carries the *expectation*, based on the current input, as to what the current input should be
  - See if the two vectors agree at one or a small number (i.e., 2 or 3) of cells (in each CM), and additionally, measure how certain each distribution is.
  - This computation is computed for each cell in each CM on each time step

Spatiotemporal Matching Algorithm - Overview

Step 1: See what the *temporal context* information is telling us

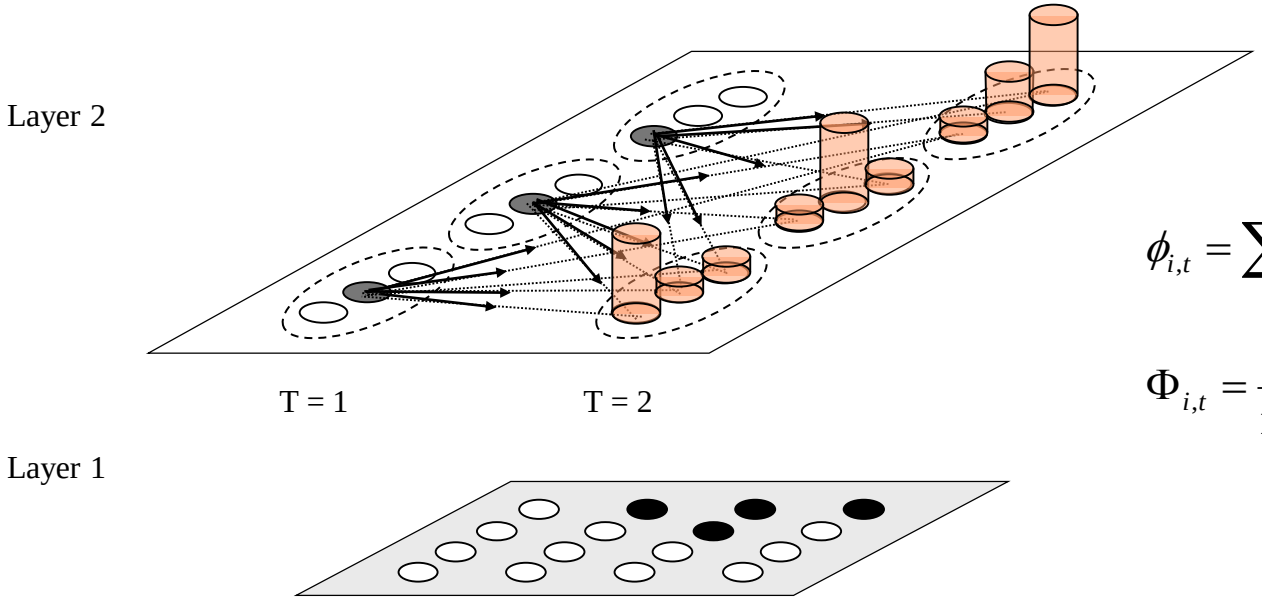


The *temporal context* information is contained in the vector of signals propagating, via the intrinsic (horizontal) projection, from the L2 cells active on the prior time step

In this and the next few slides, the model must figure out what internal representation should be become active at T = 2.

Spatiotemporal Matching Algorithm - Overview

See what the *temporal context* information is telling us



The heights of the cylinders represents $\Phi_{i,t}$ values.

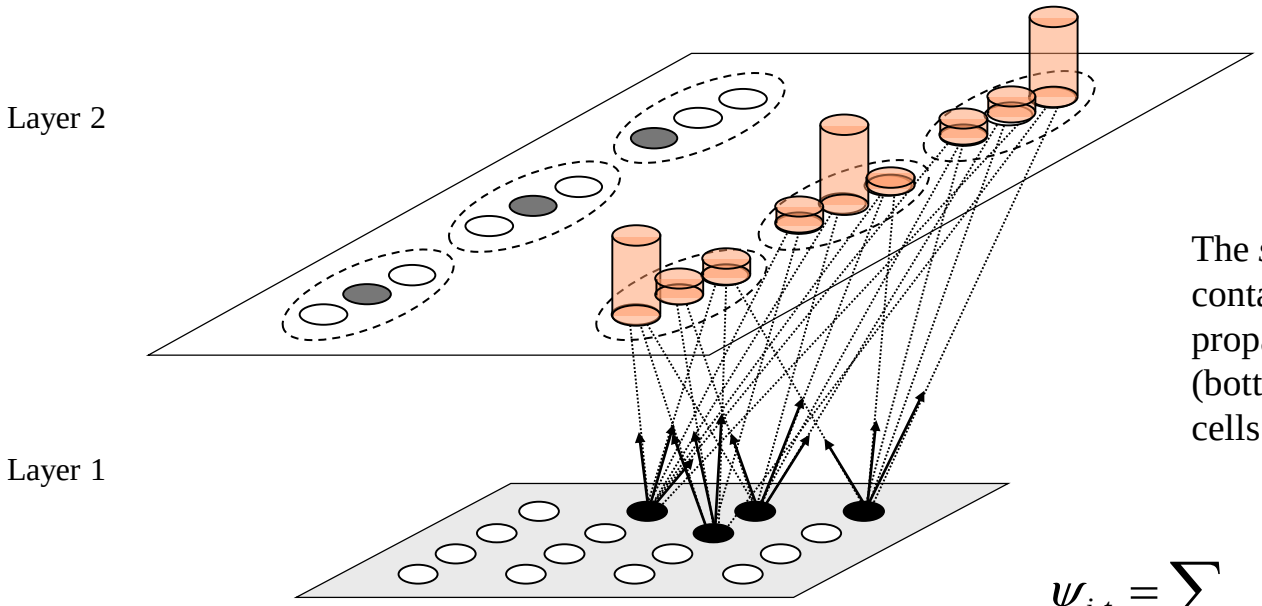
$$\phi_{i,t} = \sum_{j \in \Delta_{t-1}} w_{ji}$$

$$\Phi_{i,t} = \frac{\phi_{i,t}}{\max(\max_{j \in CM}(\phi_{j,t}), {}^{H}\Theta_t)}$$

The outer 'max' is needed to prevent the system from concluding that there is a high degree of temporal support for one or more cells during the early period of its existence.

Spatiotemporal Matching Algorithm - Overview

Step 2: See what the *spatial context* information is telling us



The heights of the cylinders represents $\Psi_{i,t}$ values.

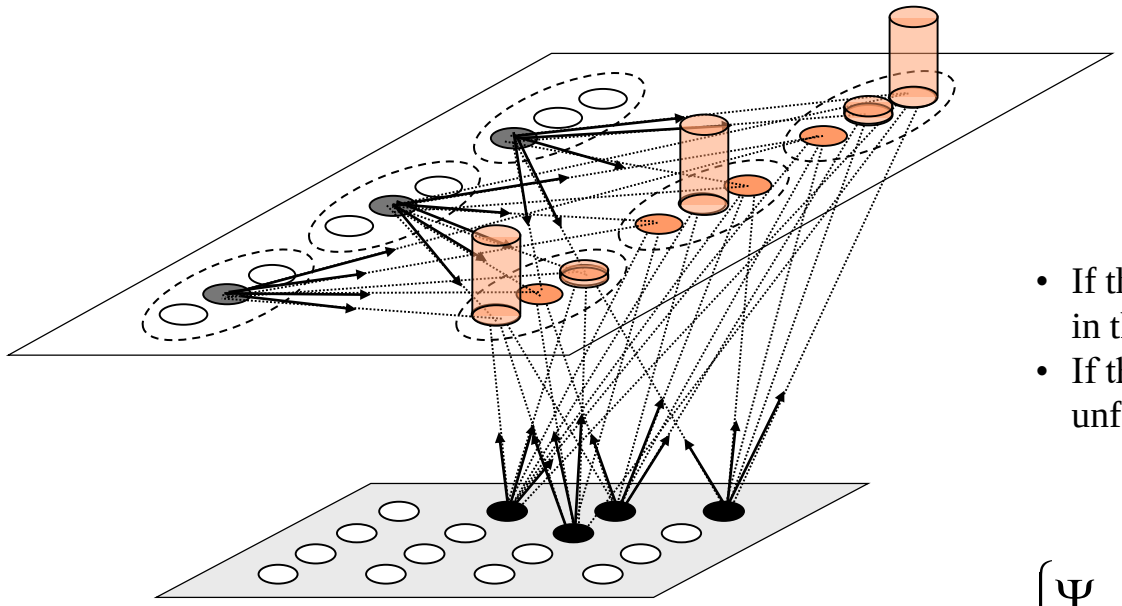
The *spatial context* information is contained in the vector of signals propagating, via the forward (bottom-up) projection, from the L1 cells active on the current time step.

$$\psi_{i,t} = \sum_{j \in \Gamma_t} w_{ji}$$

$$\Psi_{i,t} = \frac{\psi_{i,t}}{\max(\max_{j \in CM}(\psi_{j,t}), {}^{F}\Theta_t)}$$

Spatiotemporal Matching Algorithm - Overview

Two sources of evidence about the current situation are combined.



The heights of the cylinders represents $X_{i,t}$ values.

- If they agree, then the organism is in the midst of a familiar situation.
- If they disagree, then it's in an unfamiliar situation.

$$\chi_{i,t} = \begin{cases} \Psi_{i,t}{}^{u}\Phi_{i,t}{}^{v} & ,t > 0 \\ \Psi_{i,t}{}^{w} & ,t = 0 \end{cases}$$

$$\mathrm{X}_{i,t} = \frac{\chi_{i,t}}{\max(\max_{j \in CM}(\chi_{j,t}), {}^{\chi}\Theta)}$$

Spatiotemporal Matching Algorithm - Overview

Step 3: Find maximum *X* values in each CM.

$$\pi_{k,t} = \max_{j \in CM_k} \mathrm{X}_{j,t} \quad ,1 \le k \le Q$$

Step 4: Compute *global degree* of match (average over all *Q* CMs).

$$G_t = \sum_{k=1}^{Q} \pi_{k,t} \Big/ Q$$

Step 5: In each CM, Convert *X* values to probabilities.

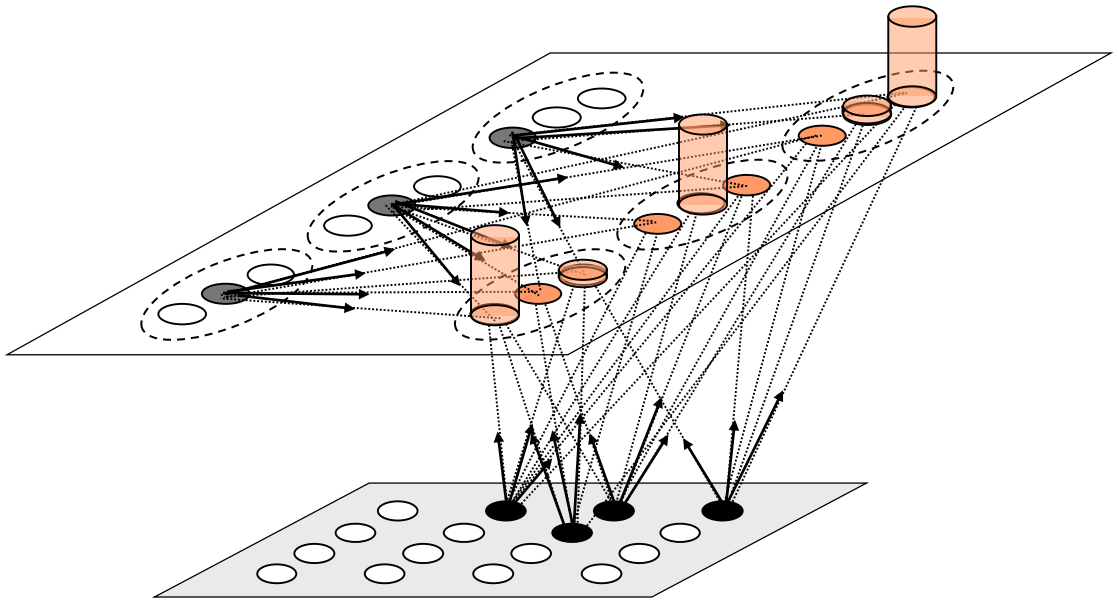
$$p_{i,t} = \frac{f(\mathrm{X}_{i,t}, G_t)}{\sum_{j \in CM} f(\mathrm{X}_{j,t}, G_t)}$$

Step 6: Determine the number of multiple competing hypotheses (MCHs).

Step 7: Pick winners according to the probability distribution.

Spatiotemporal Matching Algorithm - Overview

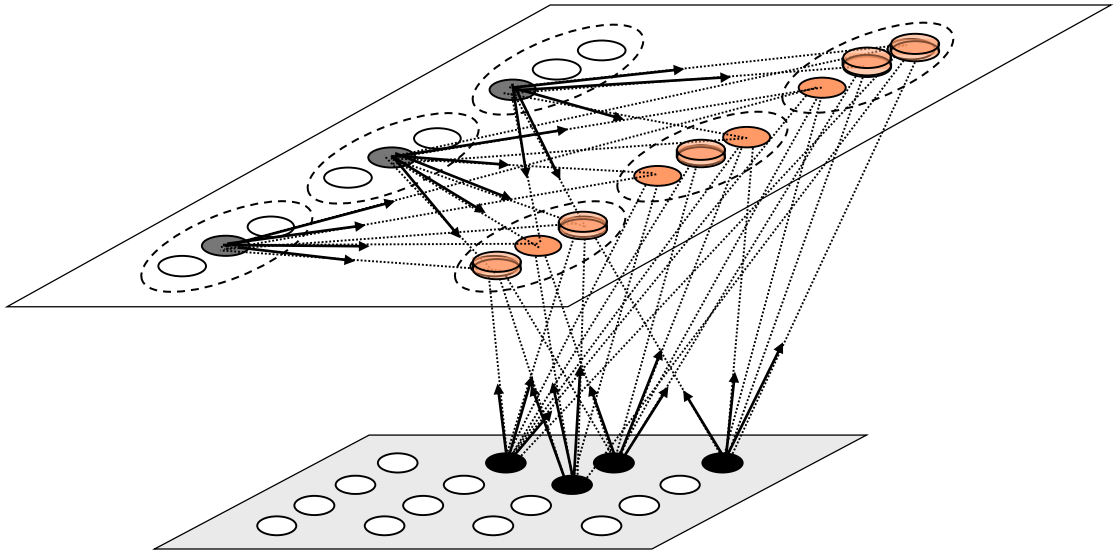
- Strong case of agreement between the two vectors



In this case ($G \approx 1$), we want to add none (or very little) noise into the winner selection process and let prior learning determine the final winners in each CM.

Spatiotemporal Matching Algorithm - Overview

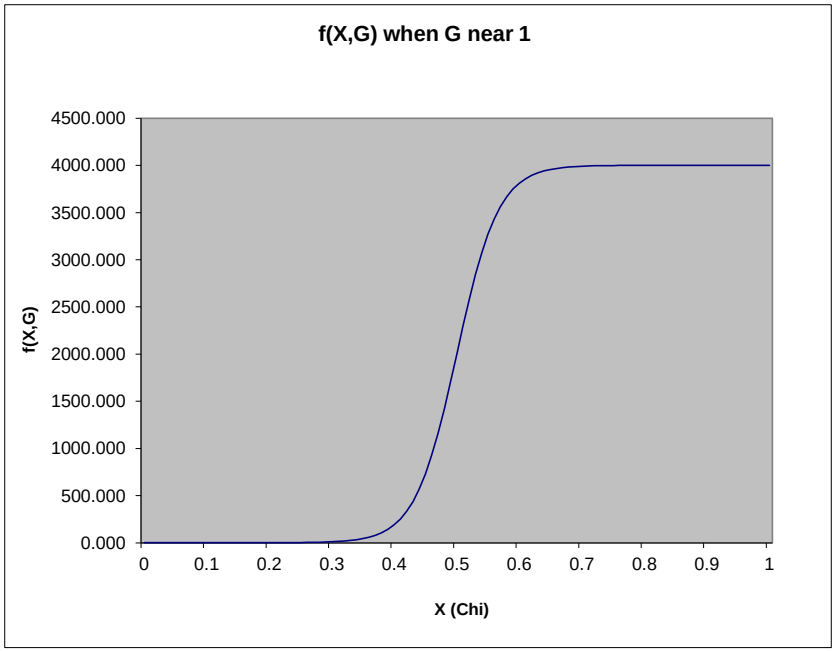
• Strong disagreement between the two vectors



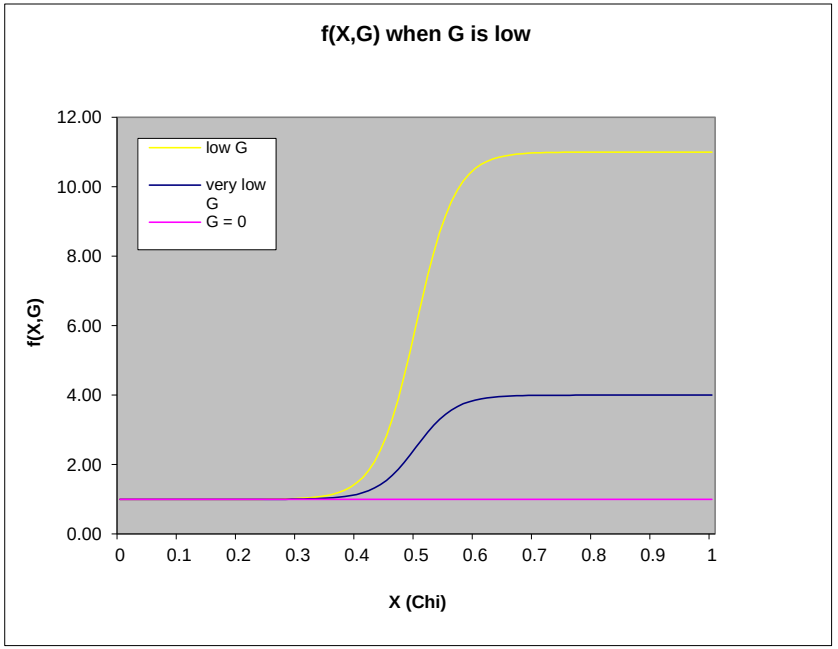
In this case ($G \approx 0$), we want to add a lot of noise into the winner selection process so as to *drown out* the influence of prior learning and approximate a uniform distribution for selecting a winner in each CM.

Spatiotemporal Matching Algorithm - Overview

$$p_{i,t} = \frac{f(\mathrm{X}_{i,t}, G_t)}{\sum_{j \in CM} f(\mathrm{X}_{j,t}, G_t)}$$

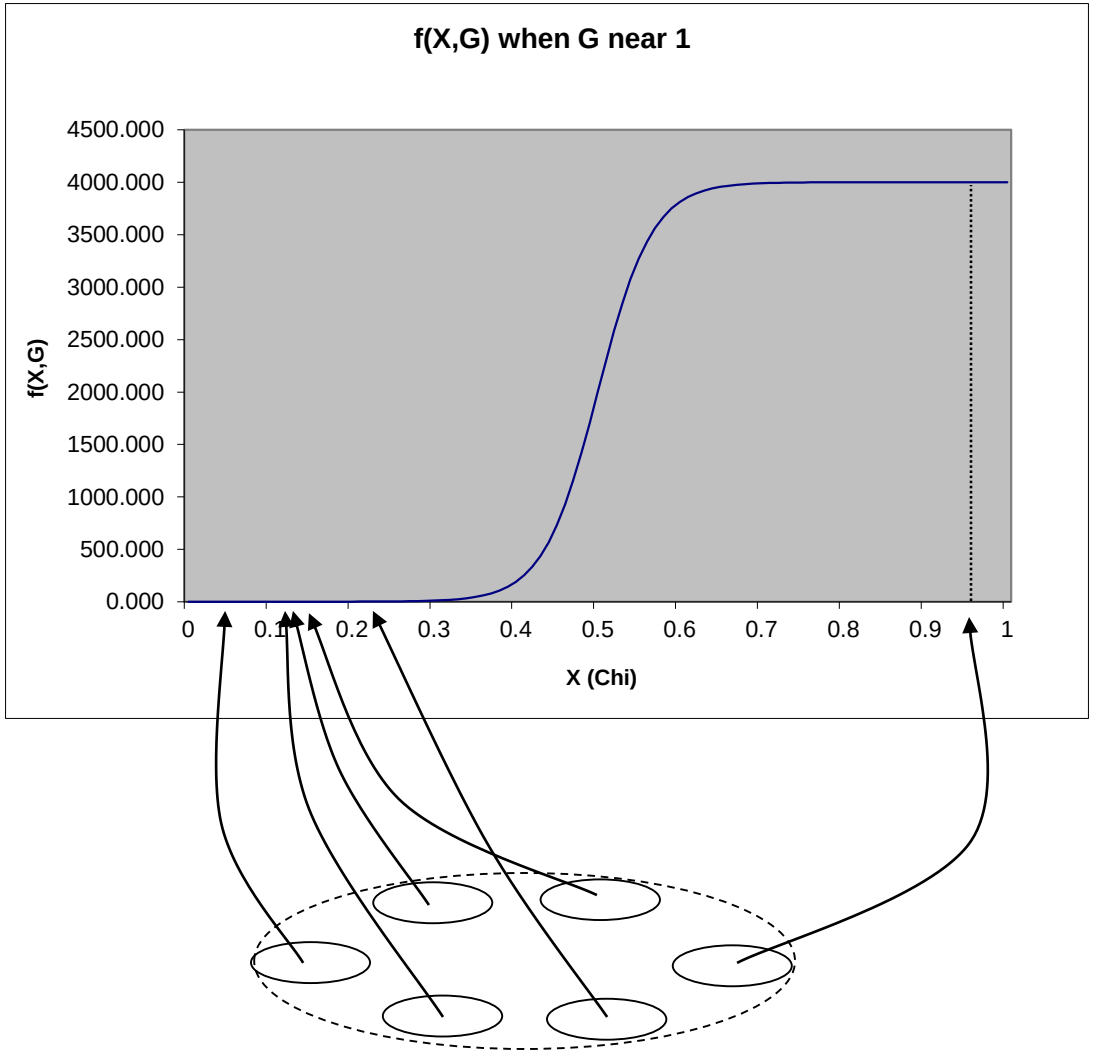


Expansive nonlinearity



Compressive nonlinearity

Spatiotemporal Matching Algorithm - Overview



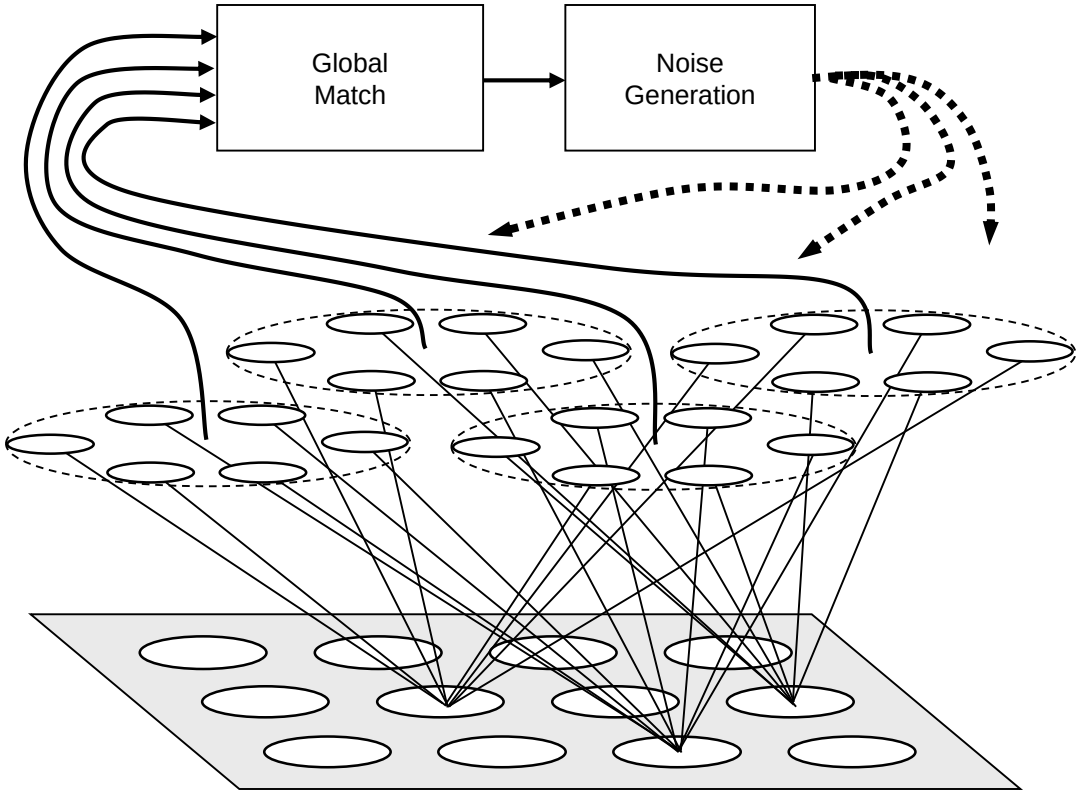
$$p_{i,t} = \frac{f(\mathrm{X}_{i,t}, G_t)}{\sum_{j \in CM} f(\mathrm{X}_{j,t}, G_t)}$$

When one cell, c, in the CM has a very high *X* value and the rest have very low values, then c will be chosen with very high probability.

TESMECOR Architecture

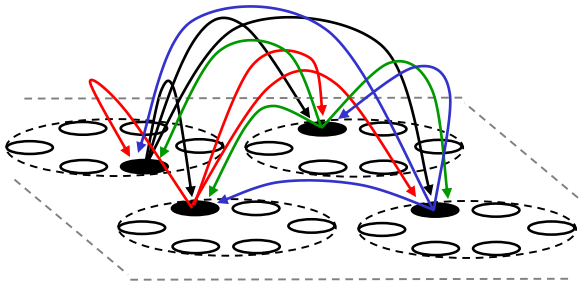
Note that TESMECOR has several operational modes. This figure pertains to what might classically be thought of as a ‘recognition mode’. In the classical recognition test, the entire input pattern is actually presented to the human (system). Thus, in such a mode, information flows from the system’s input surface (layer 1) and it utilized during processing.

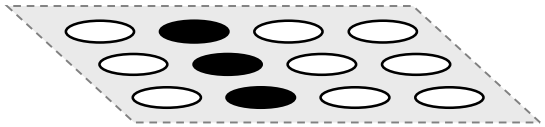
This contrasts with the classical ‘recall’ test in which only a portion of the pattern to be recalled (or even some more general type of prompt) is presented to the system whose task is then to output the rest of the pattern (or, at least, reinstate the full internal representation of the pattern and output some function of that full representation). Thus, in a ‘recall’ mode, information from the input surface is necessarily turned off (or ignored) following the prompt and for the remainder of the recall trial.

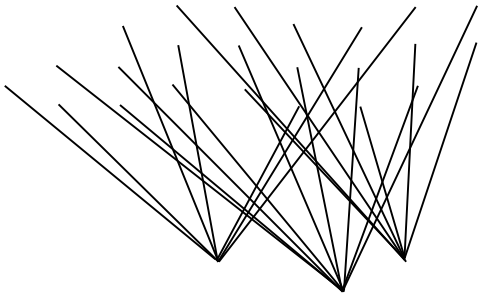


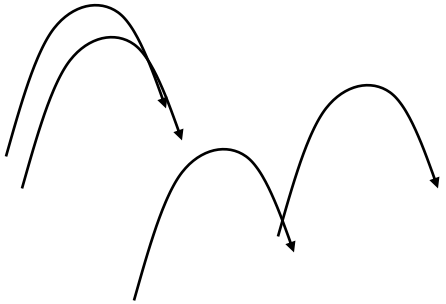
Thus, when in a pure recall mode, the matching process and concomitant noise generation process is not operative. Rather, during successive time slices of pure recall mode, the internal representations that become active depend only on propagation of signals in the horizontal matrix (not shown above), the weights of which reflect prior learning. Of course, there is also a top-down flow of information from layer 2 to layer 1 in this mode (which is formally necessary in order to verify that system truly recalls the correct input.

TESMECOR Architecture (Recall Mode)

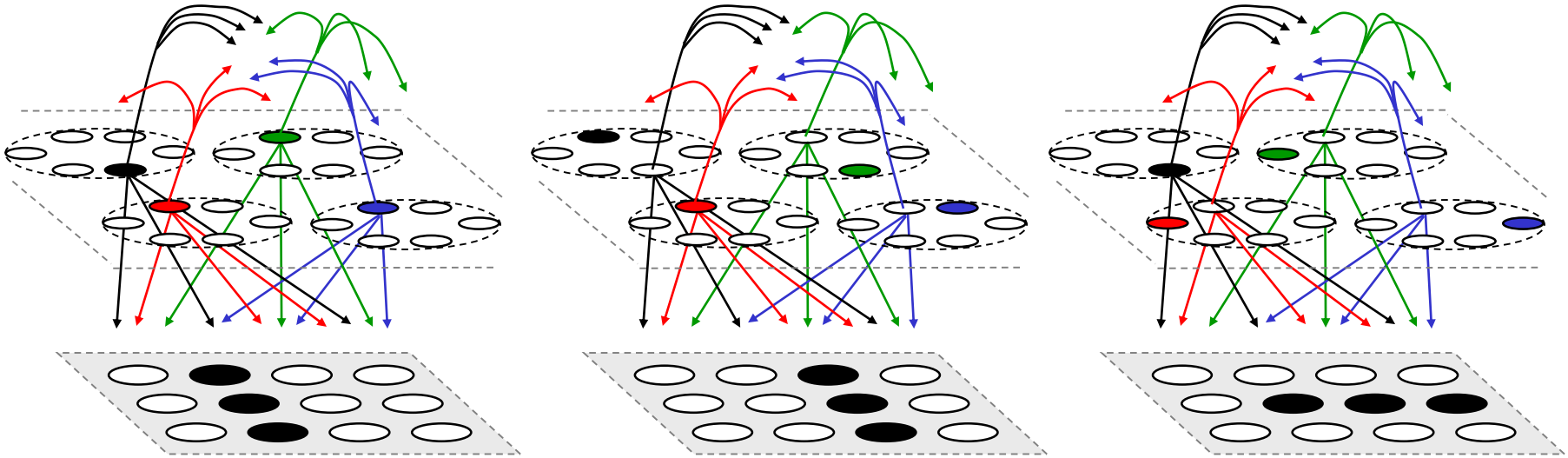








TESMECOR Architecture (Recall Mode)



Spatiotemporal Matching Algorithm - 2

$$\psi_{i,t} = \sum_{j \in \Gamma_t} w_{ji}$$

$$\Psi_{i,t} = \frac{\psi_{i,t}}{\max(\max_{j \in CM}(\psi_{j,t}), {}^{F}\Theta_t)}$$

$$\phi_{i,t} = \sum_{j \in \Delta_{t-1}} w_{ji} \quad , t > 0$$

$$\Phi_{i,t} = \frac{\phi_{i,t}}{\max(\max_{j \in CM}(\phi_{j,t}), {}^{H}\Theta_t)} \quad , t > 0$$

$$\chi_{i,t} = \begin{cases} \Psi_{i,t}{}^{u} \Phi_{i,t}{}^{v} & , t > 0 \\ \Psi_{i,t}{}^{w} & , t = 0 \end{cases}$$

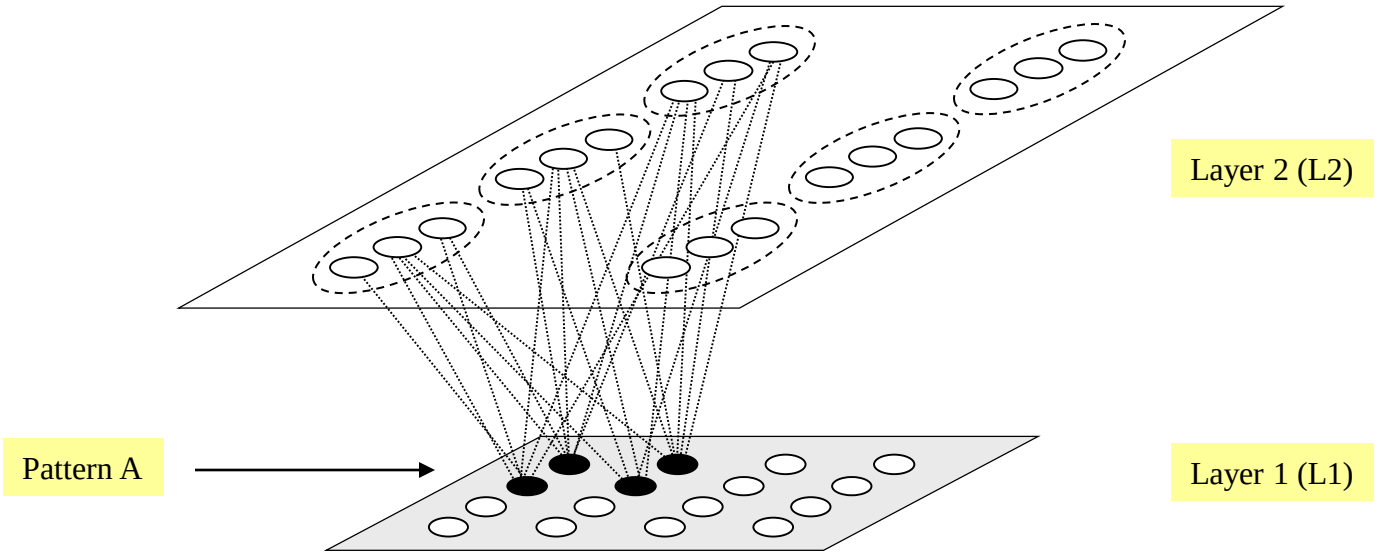
$$\mathrm{X}_{i,t} = \frac{\chi_{i,t}}{\max(\max_{j \in CM}(\chi_{j,t}), {}^{\chi}\Theta)}$$

$$\pi_{k,t} = \max_{j \in CM_k} \mathrm{X}_{j,t} \quad , 1 \le k \le Q$$

$$G_t = \sum_{k=1}^{Q} \pi_{k,t} \Big/ Q$$

$$p_{i,t} = \frac{f(\mathrm{X}_{i,t}, G_t)}{\sum_{j \in CM} f(\mathrm{X}_{j,t}, G_t)}$$

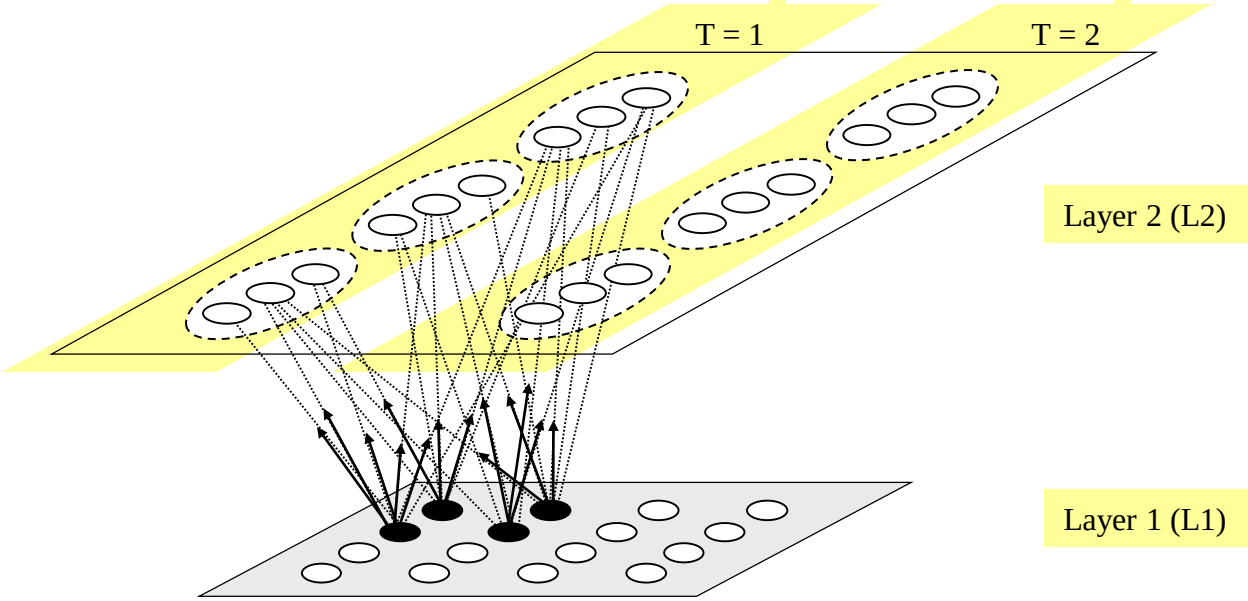
Pattern A presents at Layer 1



F-vector starts

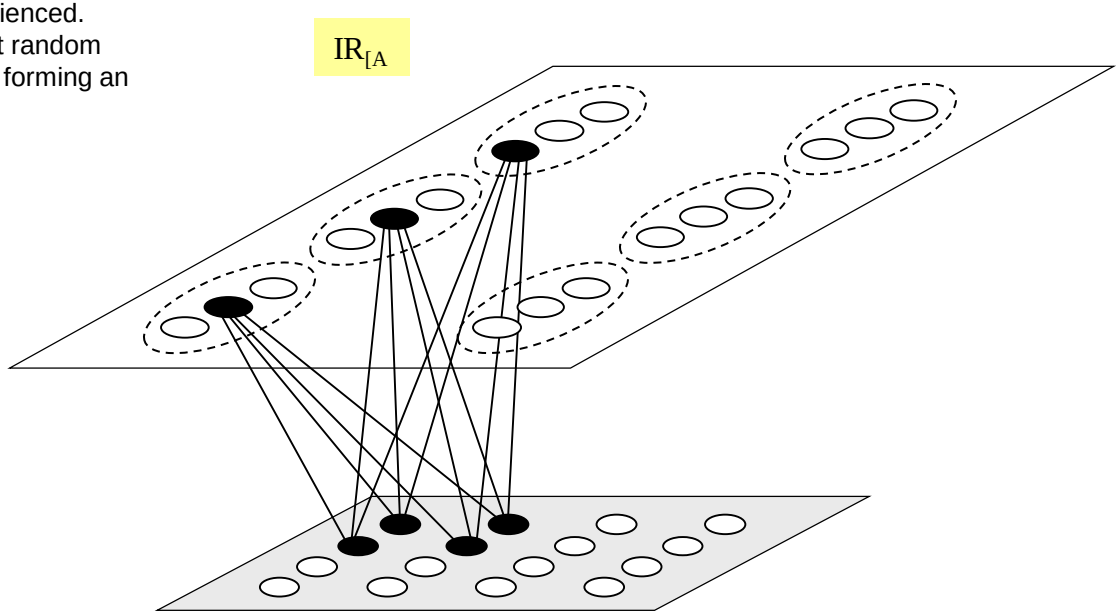
- L1 cells send activation up (forward) to L2 cells.
- We're assuming that this is a brand new network, i.e., all weights = 0

NOTE: For purposes of keeping the figures readable we divide L2 into two zones, left right. The three CMs on the left will hold the internal representation of the L1 pattern at T = 1. The three on the right will hold the representation for T = 2. In the model's actual operation, ALL CMs are used on EVERY time slice.



$IR_{[A}$ forms

- There is complete mismatch between expected and actual input (really, there is no expected input on this very first time slice of the very first episode experienced.
- Winners are selected at random and weights increased, forming an IR, $IR_{[A}$, for pattern A.

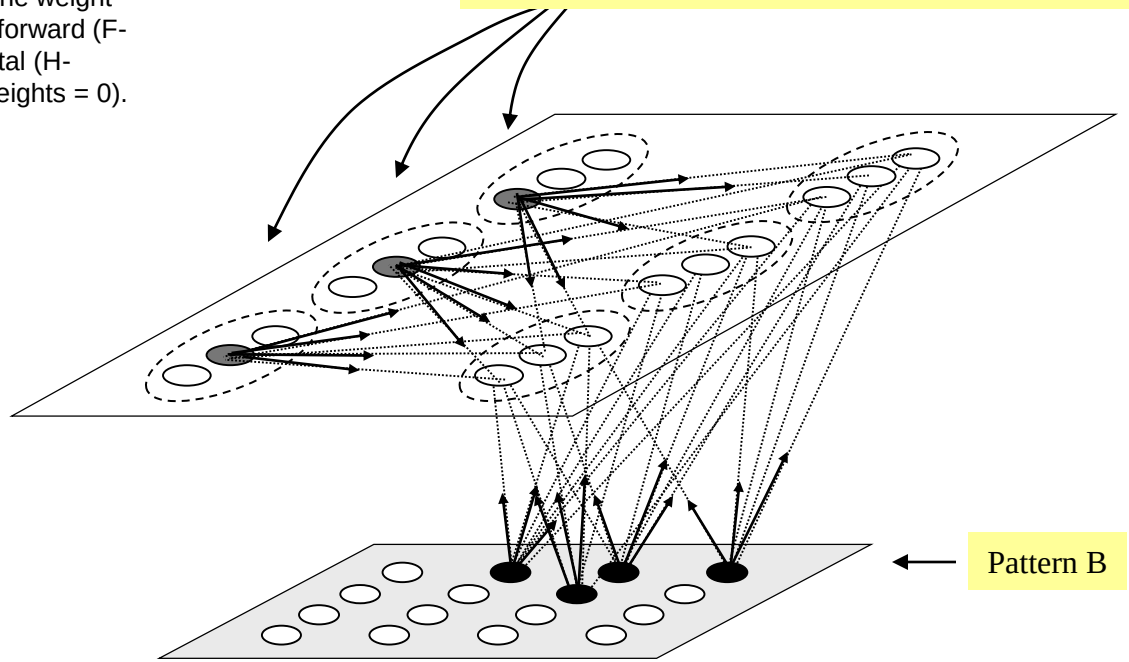


NOTE: The notation, "[A", as in $IR_{[A}$, means that this is an episode-initial instance of state A, i.e., it is state A occurring on the first time slice of an episode.

B at L1

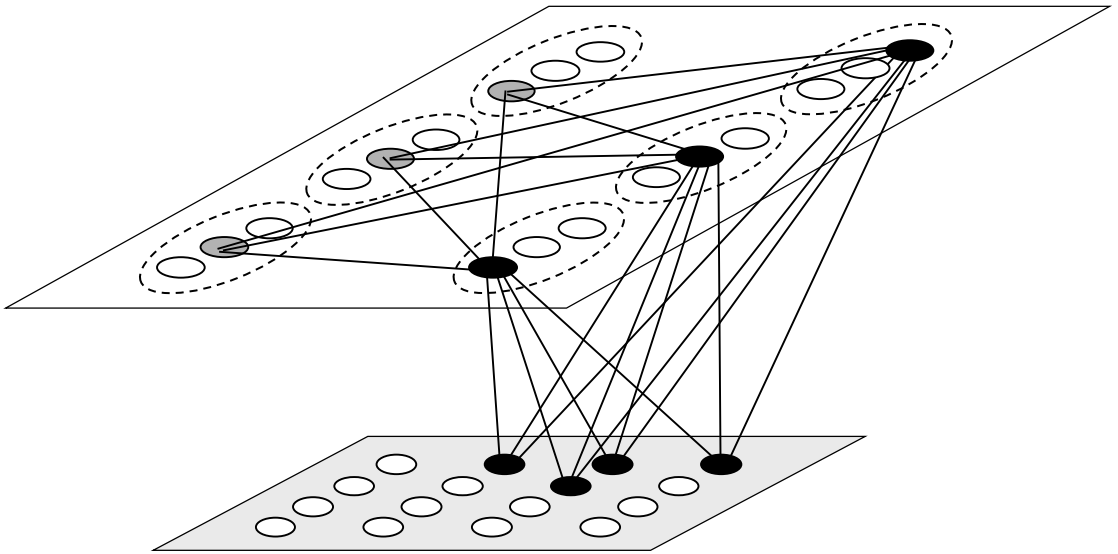
- At T = 2, there is again complete mismatch at the CMs for T = 2.
- Again, this is because the weight matrices, both the feedforward (F-matrix) and the horizontal (H-matrix) are naïve (all weights = 0).

NOTE: The gray shading of the cells of $IR_{[A}$ indicates that their activation level is fading since they were active at T = 1 and now we are at T = 2



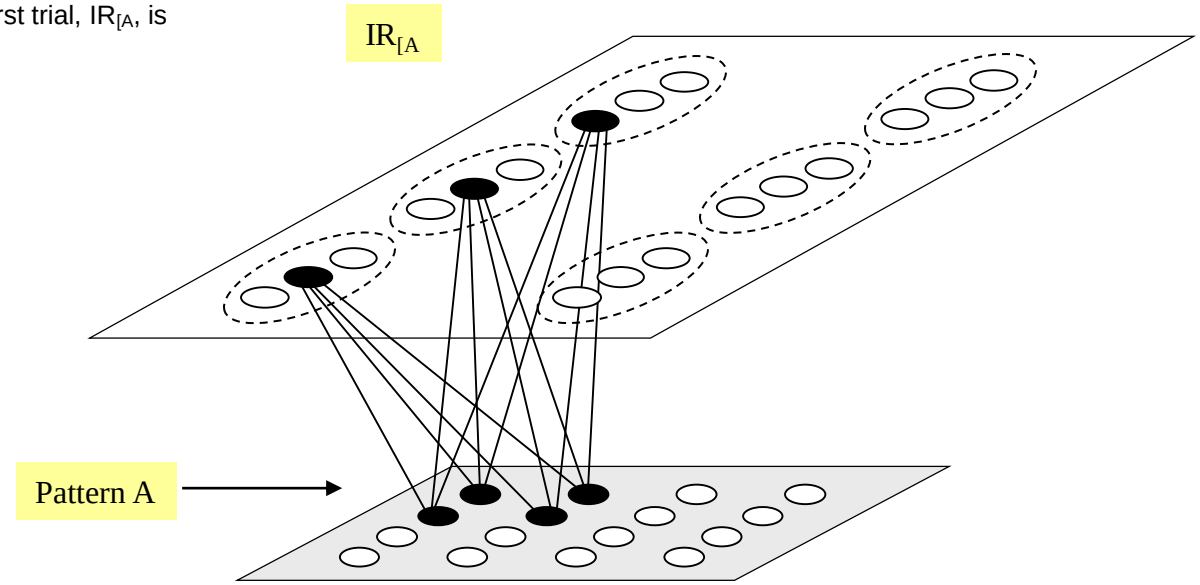
IR_{AB} forms

- Winners are chosen at random, forming a new IR, IR_{AB}, for pattern B. The horizontal and forward weights shown (depicted with solid lines) are increased.



Trial 2, A at L1

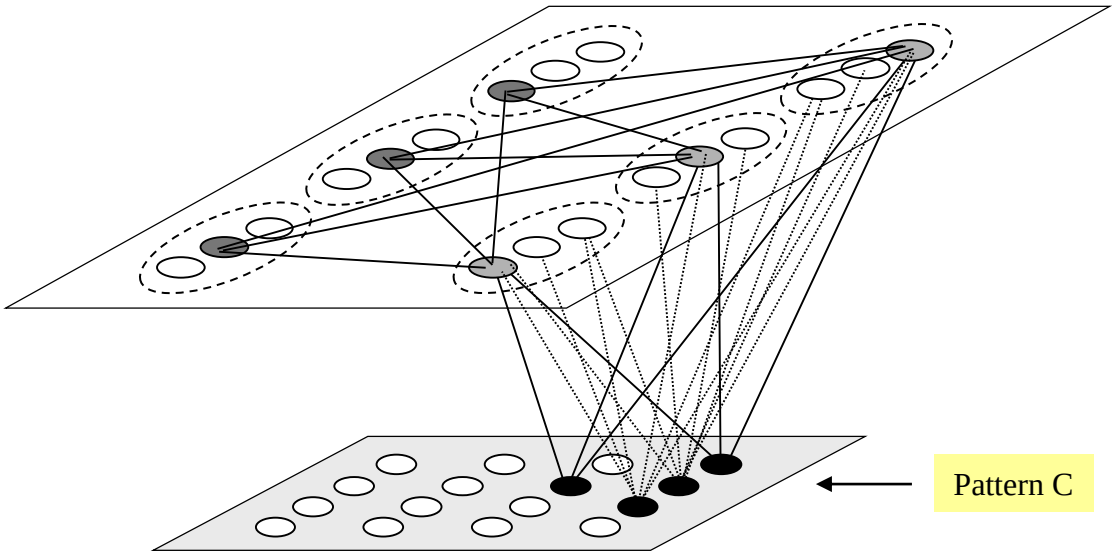
- Now on another trial we present pattern A again at T = 1.
- This time, there is a perfect match at Layer 2 (see note below). Therefore the same IR that was learned on the first trial, $IR_{[A}$, is reactivated.



Note: The model can detect that it is a perfect match at Layer 2 in this case because we assume that the amount of activity (the total number of cells active at L1 on any time slice) is normalized. So, in this case for example, if an L2 cell knows it is receiving four large inputs and if it knows that the model's parameters are set so that L1 patterns will always have exactly four active cells, it then knows that it is in a perfect match condition.

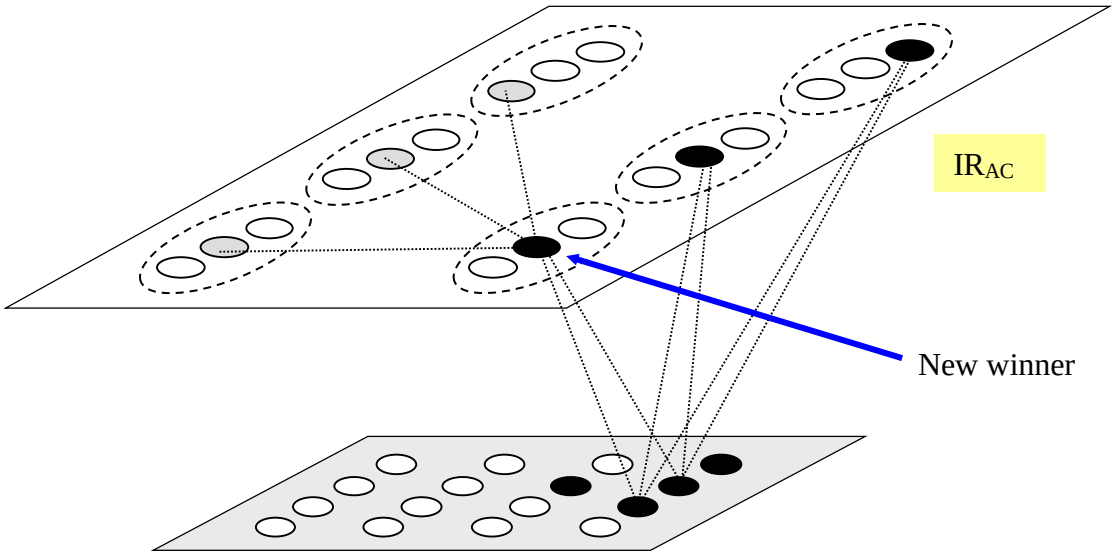
Pattern C at L1

- But now, a unexpected pattern, C, occurs at T = 2.
- The cells of IR_{AB} receive the maximum possible support (three out of three), via the H-matrix, from from L2 cells that were active at T = 1.
- But they only receive 50% (two out of four) of the maximum, via the F-matrix, from currently active L1 cells.
- Note that the grayness of the L2 cells at T = 2 is used to connote partial activation due to partial match. As mentioned earlier, we also use grayness to connote decaying activation, e.g., in the L2 cells for T = 1.



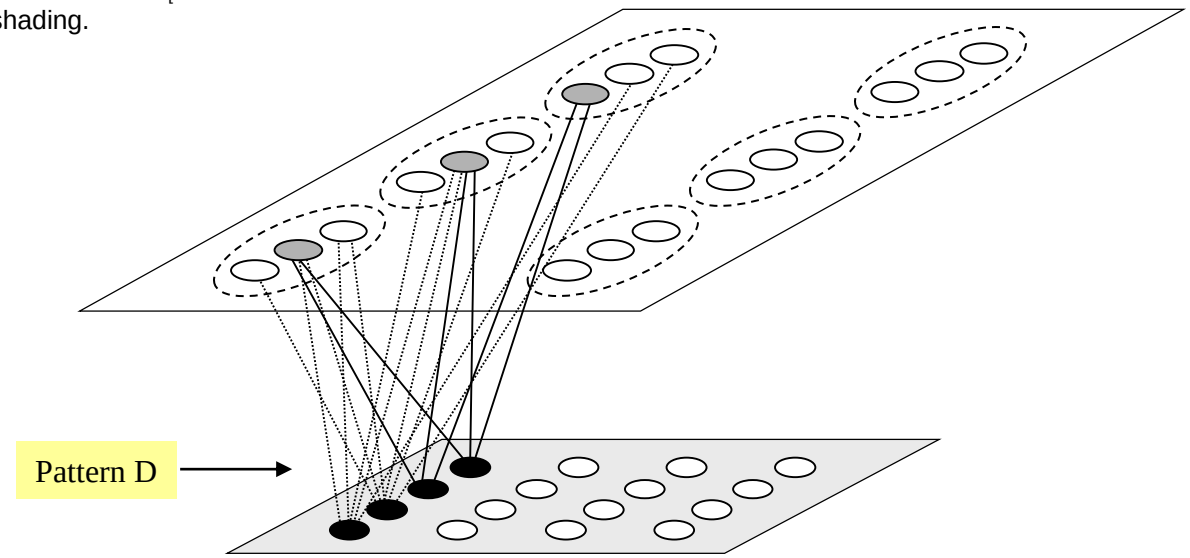
IR_{AC} forms

- This amount of mismatch leads to a significant amount of noise being added to the winner selection process. Let's assume that this amount of noise leads, probabilistically, to a new winner being chosen in one out of the three CMs for T = 2. We've arbitrarily chosen the lower right-hand CM to have the new winner.
- Thus we have a new IR, IR_{AC}, for the novel pattern, C. The newly increased weights are shown as dotted lines.
- Even though the new IR, IR_{AC}, is unique, it does have overlap with IR_{AB}. This overlap reflects the overall spatiotemporal similarity between this event, i.e., 'state C following state A', and the previously experienced (and learned) event, 'state B following state A'. In both cases, state A was the predecessor and state C has two out of four features in common with state B.



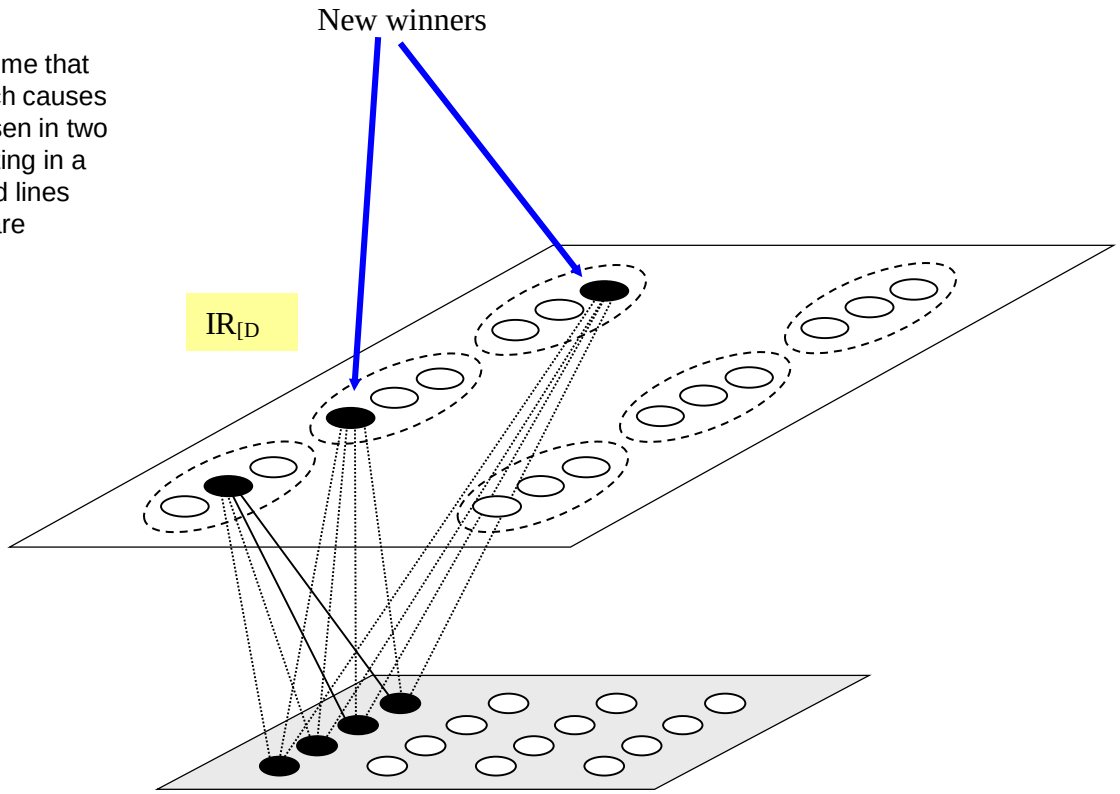
Trial 3, pattern D at L1

- Now, on another trial, a new pattern, D, having only 2 out of 4 features (i.e., active L1 cells) in common with A, presents at L1.
- So only 50% of the total possible input from L1 occurs at the cells of $IR_{[A}$. Hence, the gray shading.



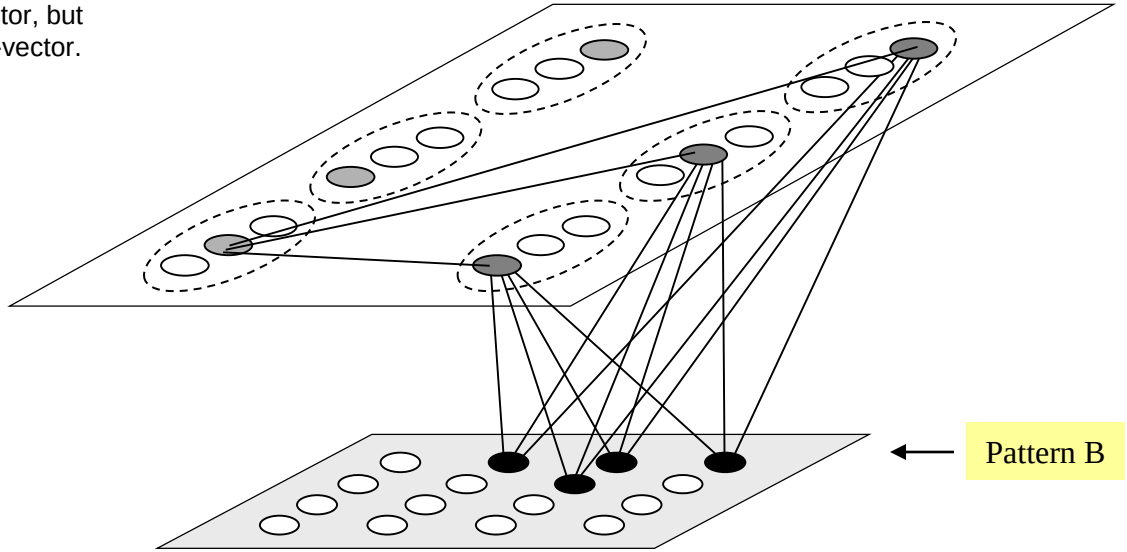
$IR_{[D}$ forms

- For concreteness, assume that that amount of mismatch causes new winners to be chosen in two out of three CMs, resulting in a new IR, $IR_{[D}$. The dotted lines denote F-weights that are increased.



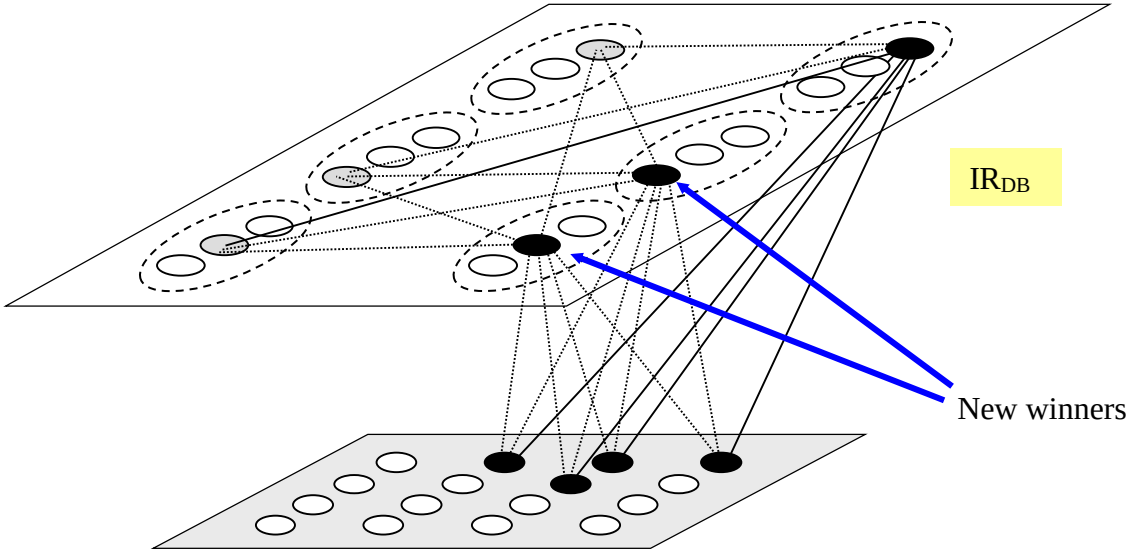
Pattern B at L1

- Pattern B, a familiar spatial pattern, presents again at T = 2. But it is in a novel temporal context; it has never occurred following pattern D before.
- IR_{AB} receives full support (i.e., 4 out of 4) from the F-vector, but only 1 out 3 from the H-vector.



IR_{DB} forms

- For concreteness, assume that this level of mismatch leads to new winners in 2 out of 3 CMs, resulting in a new IR, IR_{DB}, for state B. New learning shown with dotted lines.



Complex Sequence Set Learned

The model has learned the complex sequence set consisting of the three 2-state sequences:

- [AB]
- [AC]
- [DB]

It has different internal representations, IR_{AB} and IR_{DB}, for the two instances of state B.

Because of this, it can successfully recall either sequence given the corresponding prompt, i.e, A or D.

Furthermore, it would have no trouble uniquely associating these instances of B with different successor states. That is, it could just as easily have learned the three sequences:

- [ABE]
- [ACD]
- [DBG]

Indeed, the model has no problem with huge sets of complex sequences.

This basic scheme of comparing two distributions and using it to control the selection of the internal representation can be extended to as many distributions as you want

- All you have to do is multiply the evidence vectors together, elementwise. There can be as many factors as you want. There can be a term for the top-down vector, or terms for multiple top-down vectors, for the hippocampus, for multiple bottom-up vectors.

$$\chi_{i,t} = \begin{cases} \Psi_{i,t}^{u} \Phi_{i,t}^{v} \Omega_{i,t}^{y} & , t > 0 \\ \Psi_{i,t}^{w} \Omega_{i,t}^{z} & , t = 0 \end{cases}$$

Representing Multiple Competing Hypotheses (MCHs)

Sequence 1: B → C → D

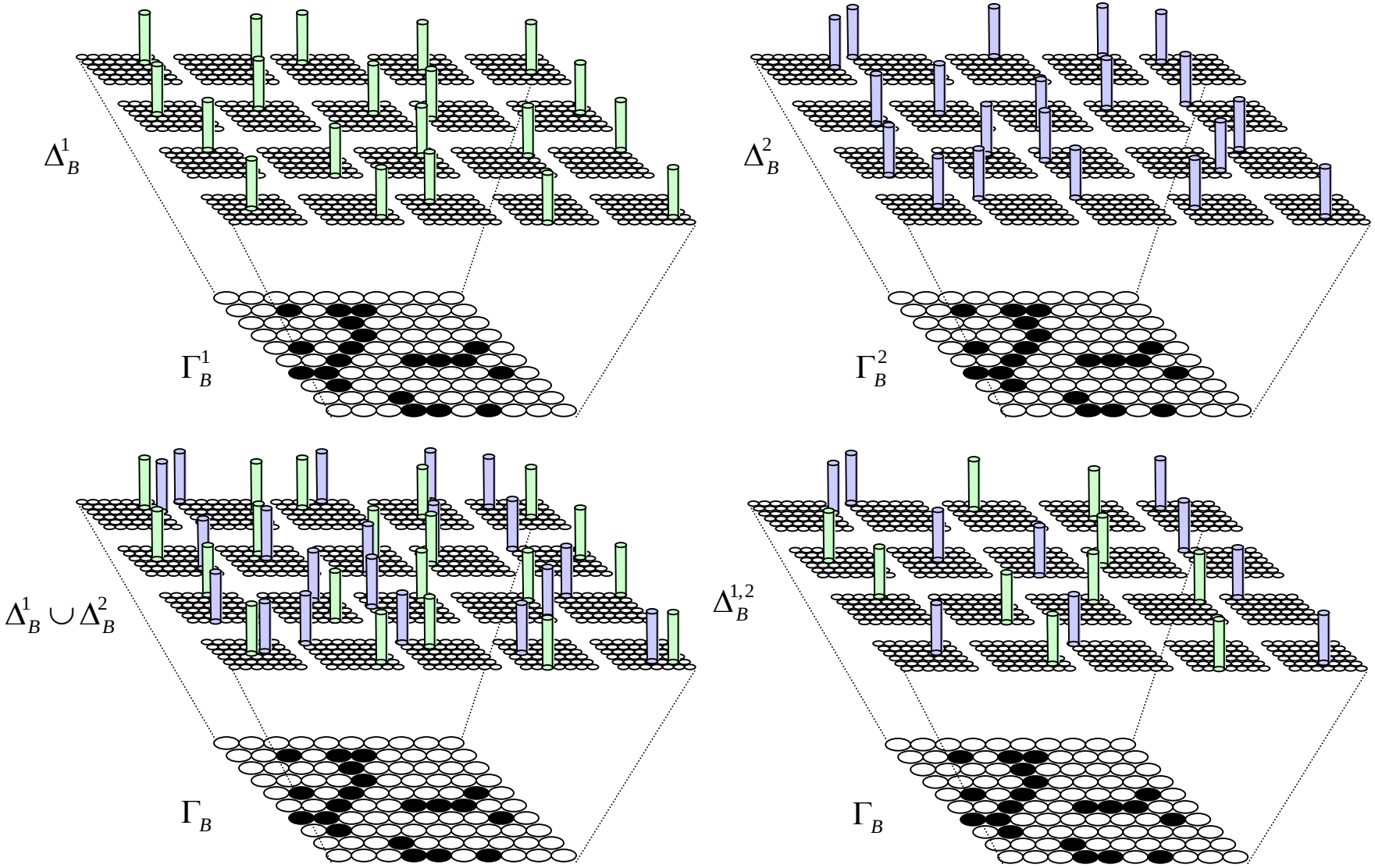
Sequence 2: D → C → B

Then presenting B as a first time slice is ambiguous.

Presenting C as a first time slice would also be ambiguous.

	L2 Code	G		${}^{H}\theta$
Δ^1_B	12 20 **9** **24** 25 **16** 10 18 **21** 19 **4** **39** **12** **13** 6 17 **33** 9 **37** 33	0	0	16
Δ^1_C	31 4 22 14 12 13 17 0 4 14 14 17 34 24 2 19 0 12 38 22	0	0	16
Δ^1_D	5 25 20 34 31 4 14 2 38 3 29 20 10 15 33 31 2 3 6 1	0	0	16
Δ^2_D	5 25 20 34 31 4 14 2 38 3 29 20 10 15 33 31 2 3 6 1	1	1	16
Δ^2_C	32 25 11 23 29 0 11 25 7 35 17 32 27 12 11 23 1 38 35 36	0	0	16
Δ^2_B	27 0 4 5 8 33 17 37 8 0 37 11 14 34 1 18 0 0 15 31	0	0	16
Δ^1_B	27 0 **9** **24** 8 **16** 17 37 **21** 0 **4** **39** **12** **13** 1 18 **33** 0 **37** 31	1	2	16
Δ^1_C	31 4 22 14 12 13 17 0 4 14 14 17 34 24 2 19 0 12 38 22	1	1	8
Δ^1_D	5 25 20 34 31 4 14 2 38 3 29 20 10 15 33 31 2 3 6 1	1	1	16
Δ^2_D	5 25 20 34 31 4 14 2 38 3 29 20 10 15 33 31 2 3 6 1	1	1	16
Δ^2_C	32 25 11 23 29 0 11 25 7 35 17 32 27 12 11 23 1 38 35 36	1	1	16
Δ^2_B	27 0 4 5 8 33 17 37 8 0 37 11 14 34 1 18 0 0 15 31	1	1	16

MCH Illustration



- Layer 2 (20 CMs, each with 50 cells)
- Layer 1 (100 cells)

Generalization Experiments Parameters

Parameter	*1*	*2*	*3*	*4*
M	100	100	100	100
Q	20	20	20	20
K	50	50	50	50
S	20	20	20	20
T	5	5	5	5
${}^{H}\theta$	16.0	13.6	16.0	16.0
${}^{F}\theta$	20.0	10.0	10.0	10.0
${}^{\chi}\theta$	0.9	0.8	0.75	0.75
b	100	11	5	5
u	10	5	3	4
v	10	5	3	3
w	10	7	4	4
n	2	2	2	2

Generalization Example – L1 Details

Original Episode

Γ_1^4 : 12 14 15 24 34 41 43 48 **52** 55 **56** 57 60 61 68 71 83 93 **94** **96**

Γ_2^4 : 7 **12** 17 19 20 23 26 27 **29** 32 **33** 34 44 46 54 65 68 **80** 85 89

Γ_3^4 : 2 3 7 9 10 13 18 20 22 47 51 54 58 59 74 83 88 89 91 99

Γ_4^4 : 1 4 5 6 33 44 50 53 54 58 60 64 65 68 70 74 75 76 96 98

Γ_5^4 : 3 6 7 8 12 13 24 26 46 47 67 72 74 77 79 82 83 95 96 97

Perturbed Episode

Γ_1^{4*} : 12 14 15 24 **30** 34 41 43 **45** 48 55 57 60 61 **62** 68 71 **76** 83 93

Γ_2^{4*} : 7 17 19 20 23 26 27 32 34 44 46 **53** 54 **56** 65 68 **79** **82** 85 89

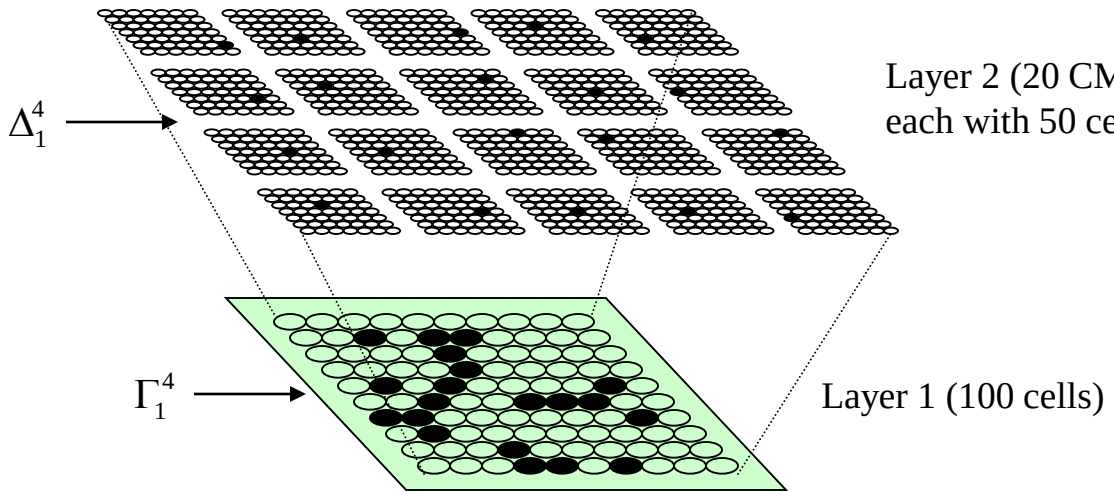
Prompt

Γ_3^{4*} : 2 3 7 9 10 13 18 20 22 47 51 54 58 59 74 83 88 89 91 99

Γ_4^{4*} : 1 4 5 6 33 44 50 53 54 58 60 64 65 68 70 74 75 76 96 98

Γ_5^{4*} : 3 6 7 8 12 13 24 26 46 47 67 72 74 77 79 82 83 95 96 97

Recalled Portion



Generalization Example – L2 Details

Original Trace

Δ_1^4 : 41 32 27 **17** 29 **33** 16 12 24 **21** 25 23 4 9 **5** 17 26 24 23 28

Δ_2^4 : 44 41 24 37 **21** 13 30 29 41 41 32 33 23 9 0 2 43 **17** 14 17

Δ_3^4 : 16 2 29 8 32 11 45 2 42 5 25 12 23 29 28 8 9 13 8 33

Δ_4^4 : 33 24 17 45 11 43 42 36 29 42 27 17 8 47 9 45 34 35 31 32

Δ_5^4 : 7 2 4 7 13 8 17 48 16 5 32 49 7 14 36 49 27 48 30 19

Recalled Trace

Δ_1^{4*} : 41 32 27 **45** 29 **2** 16 12 24 **5** 25 23 4 9 **40** 17 26 24 23 28

Δ_2^{4*} : 44 41 24 37 **23** 13 30 29 41 41 32 33 23 9 0 2 43 **0** 14 17

Δ_3^{4*} : 16 2 29 8 32 11 45 2 42 5 25 12 23 29 28 8 9 13 8 33

Δ_4^{4*} : 33 24 17 45 11 43 42 36 29 42 27 17 8 47 9 45 34 35 31 32

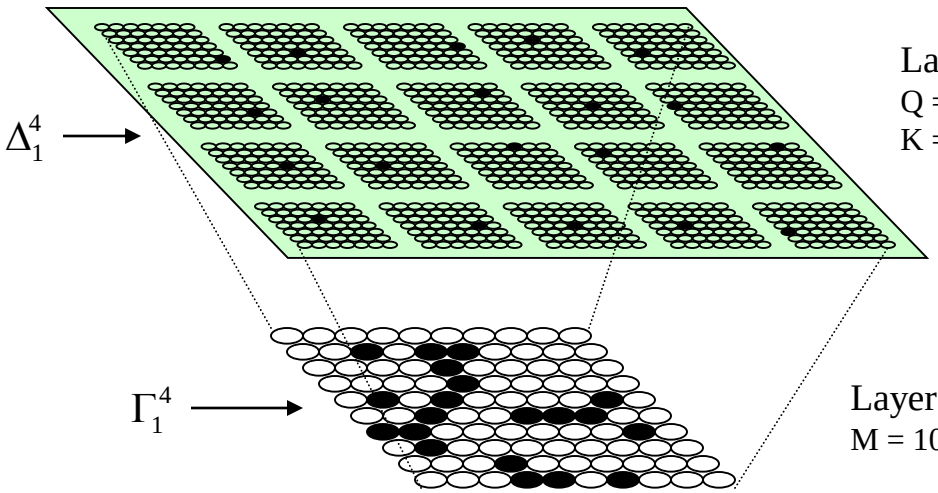
Δ_5^{4*} : 7 2 4 7 13 8 17 48 16 5 32 49 7 14 36 49 27 48 30 19

L2 recall Accuracy

0.67	0.8
0.82	0.9
1.0	1.0
1.0	1.0
1.0	1.0

$$R_t^i = \frac{\left|\Delta_t^i \cap \Delta_t^{i*}\right|}{\left|\Delta_t^i \cup \Delta_t^{i*}\right|}$$

$$R_t^i = \frac{\left|\Delta_t^i \cap \Delta_t^{i*}\right|}{Q}$$



Layer 2
Q = 20 CMs
K = 50 cells/CM

Layer 1
M = 100 cells

Generalization Experiments – L2 Accuracy by Slices

Simulation	E	*d*	φ	R_{set}
1	27	2	1	92.3%
2	13	4	1	98.0%
3	7	6	1	98.3%
4	13	8	2	82.7%

Prompt Length

Main Point

L2 trace for the perturbed episode *locks into* the L2 trace for the original episode as time progresses.

$\Theta^h = 16$

L2 Accuracy of Individual Time Slices

← Interactive Tracking → ← Solipsistic Recall →

Episode	T = 1	T = 2	T = 3	T = 4	T = 5
0	0.9	0.9	1.0	1.0	1.0
1	0.82	1.0	1.0	1.0	1.0
2	0.67	1.0	1.0	1.0	1.0
3	0.82	0.9	1.0	1.0	1.0
4	0.67	0.82	1.0	1.0	1.0
5	0.67	0.9	1.0	1.0	1.0
6	0.9	1.0	1.0	1.0	1.0
7	0.74	1.0	1.0	1.0	1.0
8	0.74	1.0	1.0	1.0	1.0
9	0.67	0.82	1.0	1.0	1.0
10	0.54	0.67	0.22	0.0	0.0
11	0.48	0.21	0.0	0.0	0.0
12	0.82	0.9	1.0	1.0	1.0

14 (out of 20) correct cells. $\frac{14}{26} \cong .54$

16 (out of 20) correct cells. $\frac{16}{24} \cong .67$

13 (out of 20) correct cells. $\frac{13}{27} \cong .48$

Layer 1 recall accuracy

$$R_t^i = \frac{\left|\Gamma_t^i \cap \Gamma_t^{i*}\right|}{\left|\Gamma_t^i \cup \Gamma_t^{i*}\right|}$$

C_i	:	number of features correctly recalled for episode i
D_i	:	number of features deleted when recalling episode i
I_i	:	number of intruding features when recalling episode i

$$R_t^i = \frac{C_i - D_i}{C_i + I_i}$$

Layer 2 recall accuracy

$$R_t^i = \frac{\left|\Delta_t^i \cap \Delta_t^{i*}\right|}{\left|\Delta_t^i \cup \Delta_t^{i*}\right|}$$

Harsher measure

$$R_t^i = \frac{\left|\Delta_t^i \cap \Delta_t^{i*}\right|}{Q}$$

Probably should have used this