Tutorial at KDD, August 14th 2021

From Deep Learning to Deep Reasoning

Part C: Memory | Data efficiency | Recursive reasoning

Truyen Tran, Vuong Le, Hung Le and Thao Le
{truyen.tran,vuong.le,thai.le,thao.le}@deakin.edu.au

https://bit.ly/37DYQn7

Agenda

- Reasoning with external memories
    - Memory of entities – memory-augmented neural networks
    - Memory of relations with tensors and graphs
    - Memory of programs & neural program construction.
- Learning to reason with less labels
    - Data augmentation with analogical and counterfactual examples
    - Question generation
    - Self-supervised learning for question answering
    - Learning with external knowledge graphs
- Recursive reasoning with neural theory of mind.

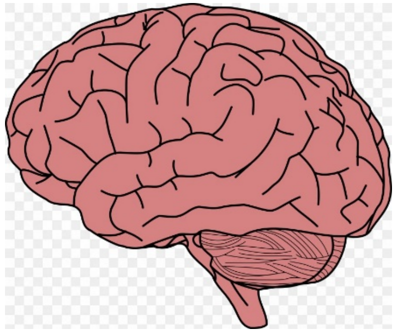
Agenda

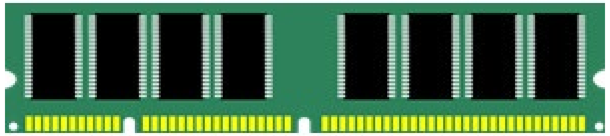
- **Reasoning with external memories**
    - Memory of entities – memory-augmented neural networks
    - Memory of relations with tensors and graphs
    - Memory of programs & neural program construction.
- Learning to reason with less labels:
    - Data augmentation with analogical and counterfactual examples
    - Question generation
    - Self-supervised learning for question answering
    - Learning with external knowledge graphs
- Recursive reasoning with neural theory of mind.

Introduction

Memory is part of intelligence

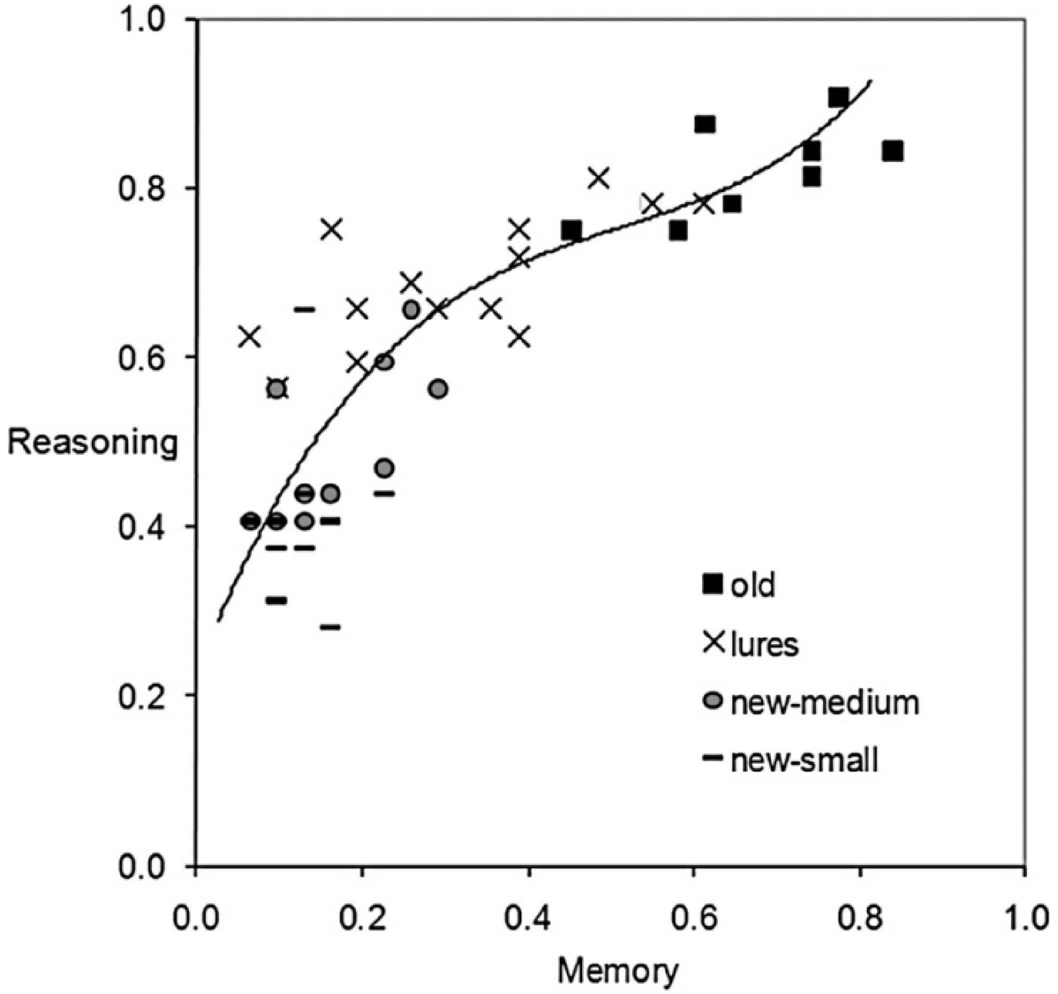
- Memory is the ability to store, retain and recall information
- Brain memory stores items, events and high-level structures
- Computer memory stores data and temporary variables





Memory-reasoning analogy

- 2 processes: fast-slow
  - Memory: familiarity-recollection
- Cognitive test:
  - Corresponding reasoning and memorization performance
  - Increasing # premises, inductive/deductive reasoning is affected



Heit, Evan, and Brett K. Hayes. "Predicting reasoning from memory." *Journal of Experimental Psychology: General* 140, no. 1 (2011): 76.

Common memory activities

- Encode: write information to the memory, often requiring compression capability
- Retain: keep the information overtime. This is often assumed in machinery memory
- Retrieve: read information from the memory to solve the task at hand

Encode

Retain

Retrieve

Memory taxonomy based on memory content

Item Memory	• Objects, events, items, variables, entities
Relational Memory	• Relationships, structures, graphs
Program Memory	• Programs, functions, procedures, how-to knowledge

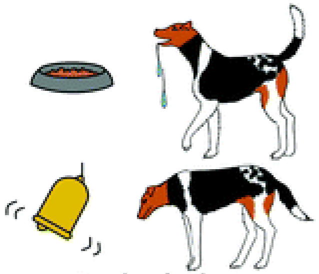
Item memory

Associative memory

RAM-like memory

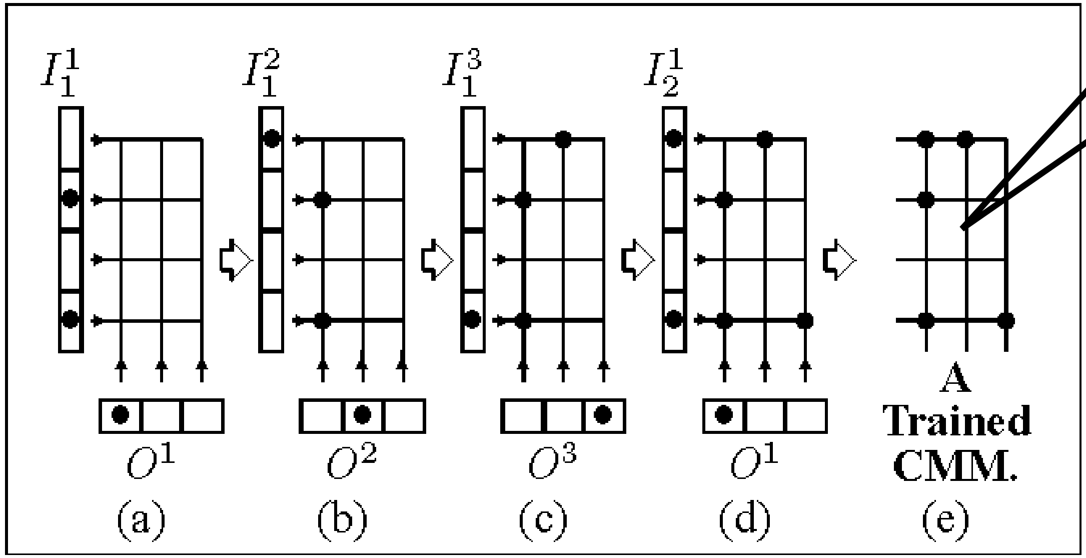
Independent memory

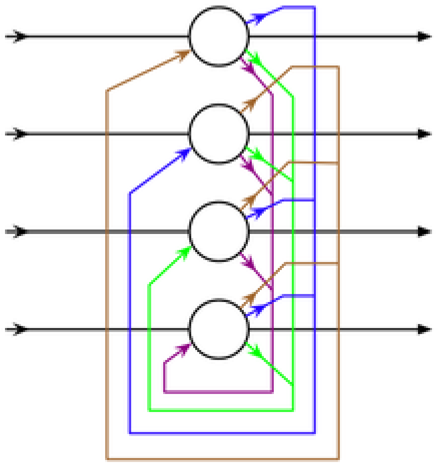
Distributed item memory as associative memory

Language	Time	Object	Behaviour
"Green" means "go," but what does "red" mean?	birthday party on 30^{th} Jan	Where is my pen? What is the password?	 Pavlov's dog
Semantic memory	Episodic memory	Working memory	Motor memory

Associate memory can be implemented as Hopfield network

Fast-weight
$\widehat{M}$





Correlation matrix memory

Hopfield network

Encode $$\hat{\mathbf{M}} = \sum_{k=1}^{q} \mathbf{b}_k \mathbf{a}_k^T$$

Retrieve $$\mathbf{b} = \hat{\mathbf{M}} \mathbf{a}_j$$

Feed-forward retrieval

Retrieve $$x_i(t+1) = \operatorname{sign}\left(\sum_{j=1}^{N} w_{ij} x_j(t)\right)$$

Recurrent retrieval

Rule-based reasoning with associative memory

- Encode a set of rules: "pre-conditions

→post-conditions"
- Support variable binding, rule-conflict handling and partial rule input
- Example of encoding rule "A:1,B:3,C:4→X"

Conversion of lexical items to binary patterns.

```
A = 100010      1 = 100001      X = 0011
B = 011000      2 = 010010      Y = 0110
C = 000110      3 = 001100      Z = 1100
                4 = 000101
                5 = 000011
                6 = 110000
```

Outer product for binding

Bind tokens to form tensors.

```
A:1=            B:3 =           C:4 =
    100001          000000          000000
    000000          001100          000000
    000000          001100          000000
    000000          000000          000101
    100001          000000          000101
    000000          000000          000000
```

Assign separators.

S_1 = 00001

S_2 = 10000

S_3 = 00100

Superimpose to form SIBs.

$A{:}1 \;.\; B{:}3 \;.\; C{:}4 = SIB_1 =$

```
100001
001100
001100
000101
100101
000000
```

Train into arity 3 network.

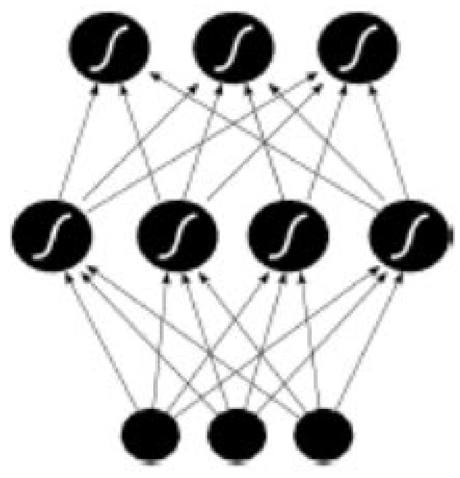
(too large to show).

$W_{new}{}^{in} = SIB_1 \times S_1$

$W_{new}{}^{out} = S_1 \times X$

Austin, Jim. "Distributed associative memories for high-speed symbolic reasoning." *Fuzzy Sets and Systems* 82, no. 2 (1996): 223-233.

Memory-augmented neural networks: computation-storage separation



'CPU'

+

Memory

RAM

=

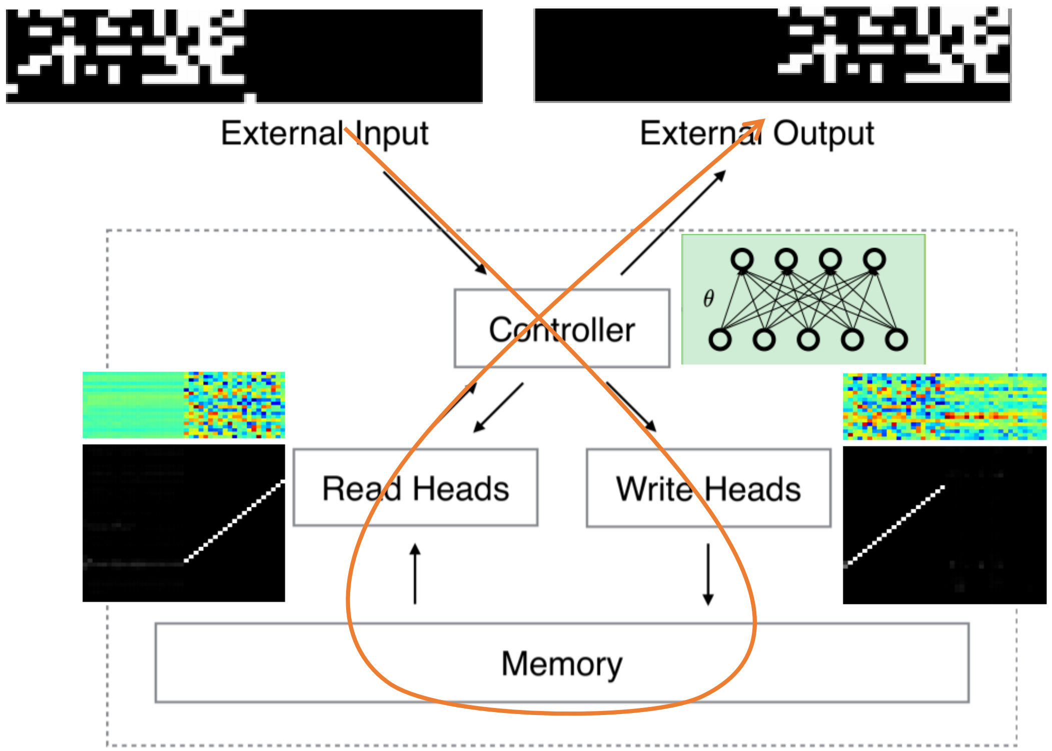
Computer that learns from examples

(*or* neural net that separates computation from memory)

Neural Turing Machine (NTM)

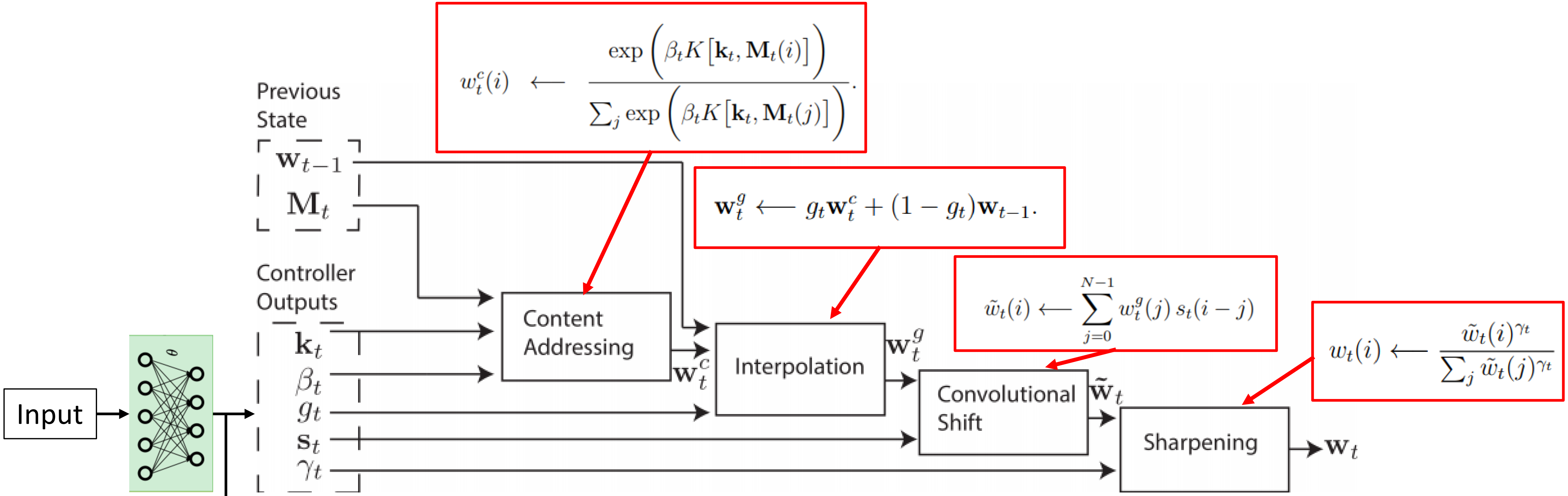
- Memory is a 2d matrix
- Controller is a neural network
- The controller read/writes to memory at certain addresses.
- Trained end-to-end, differentiable
- Simulate Turing Machine

→support symbolic reasoning, algorithm solving



Graves, Alex, Greg Wayne, and Ivo Danihelka. "Neural turing machines." *arXiv preprint arXiv:1410.5401* (2014).

Addressing mechanism in NTM



e_t, a_t

Memory writing

$$\tilde{\mathbf{M}}_t(i) \longleftarrow \mathbf{M}_{t-1}(i)\left[\mathbf{1} - w_t(i)\mathbf{e}_t\right]$$

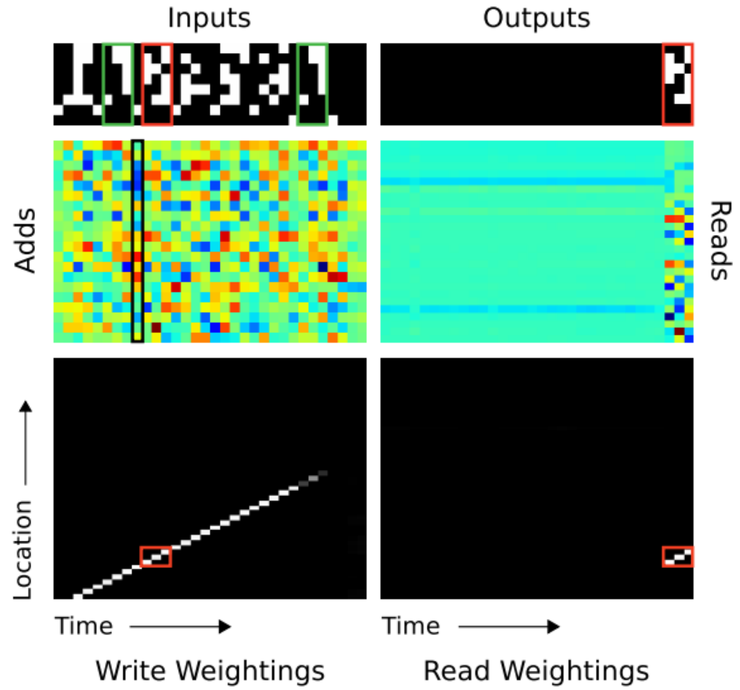
$$\mathbf{M}_t(i) \longleftarrow \tilde{\mathbf{M}}_t(i) + w_t(i)\,\mathbf{a}_t.$$

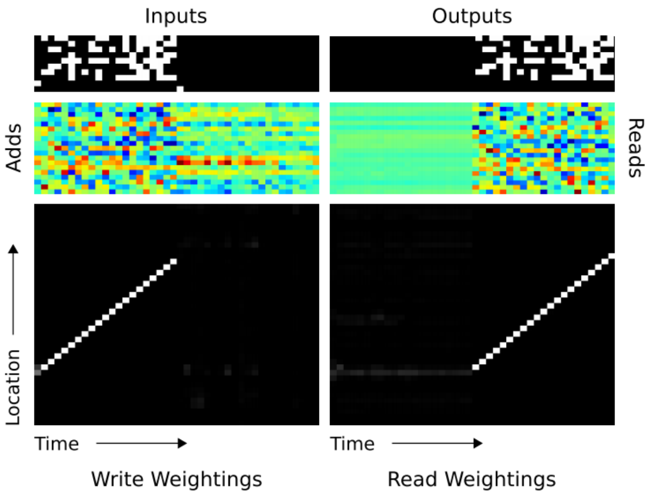
Memory reading

$$\mathbf{r}_t \longleftarrow \sum_i w_t(i)\mathbf{M}_t(i),$$

Algorithmic reasoning

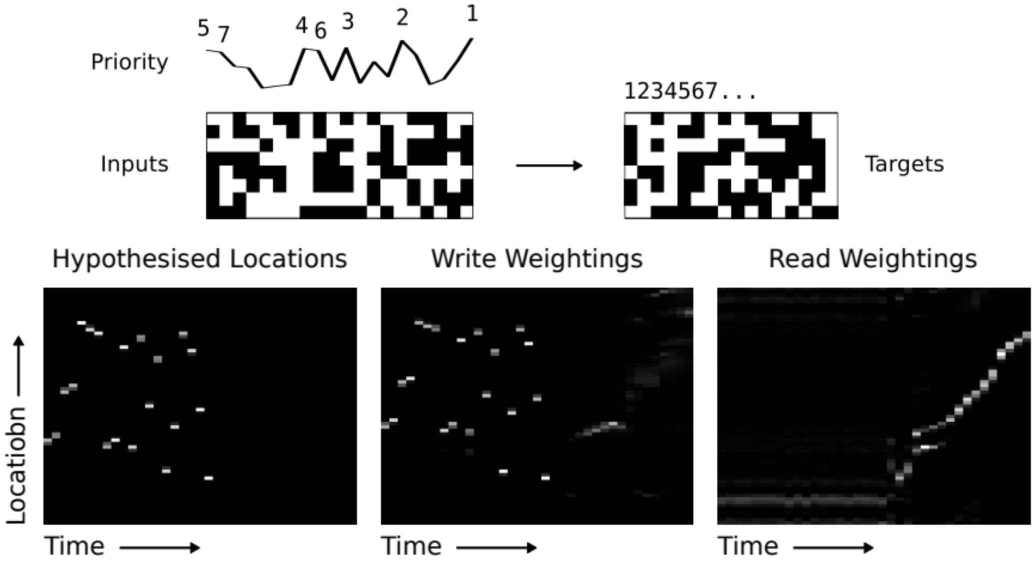
Associative recall





Copy

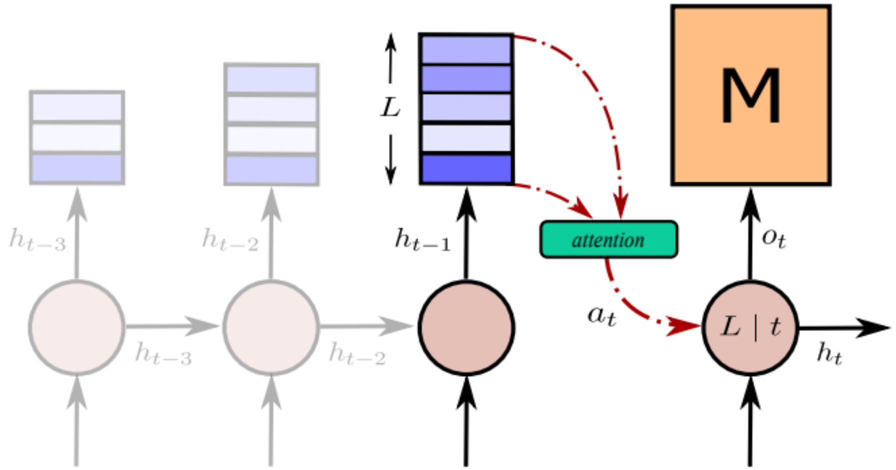
initialise: move head to start location
while input delimiter not seen **do**
    receive input vector
    write input to head location
    increment head location by 1
end while
return head to start location
while true **do**
    read output vector from head location
    emit output
    increment head location by 1
end while



Priority sort

Optimal memory writing for memorization

- Simple finding: writing too often deteriorates memory content (not retainable)
- Given input sequence of length T and only D writes, *when should we write to the memory?*



Theorem 3. *Given D memory slots, a sequence with length T, a decay rate $0 < \lambda \leq 1$, then the optimal intervals $\{l_i \in \mathbb{R}^+\}_{i=1}^{D+1}$ satisfying $T = \sum_{i=1}^{D+1} l_i$ such that the lower bound on the average contribution $I_\lambda = \frac{C}{T} \sum_{i=1}^{D+1} f_\lambda(l_i)$ is maximized are the following:*

$$l_1 = l_2 = ... = l_{D+1} = \frac{T}{D+1} \tag{7}$$

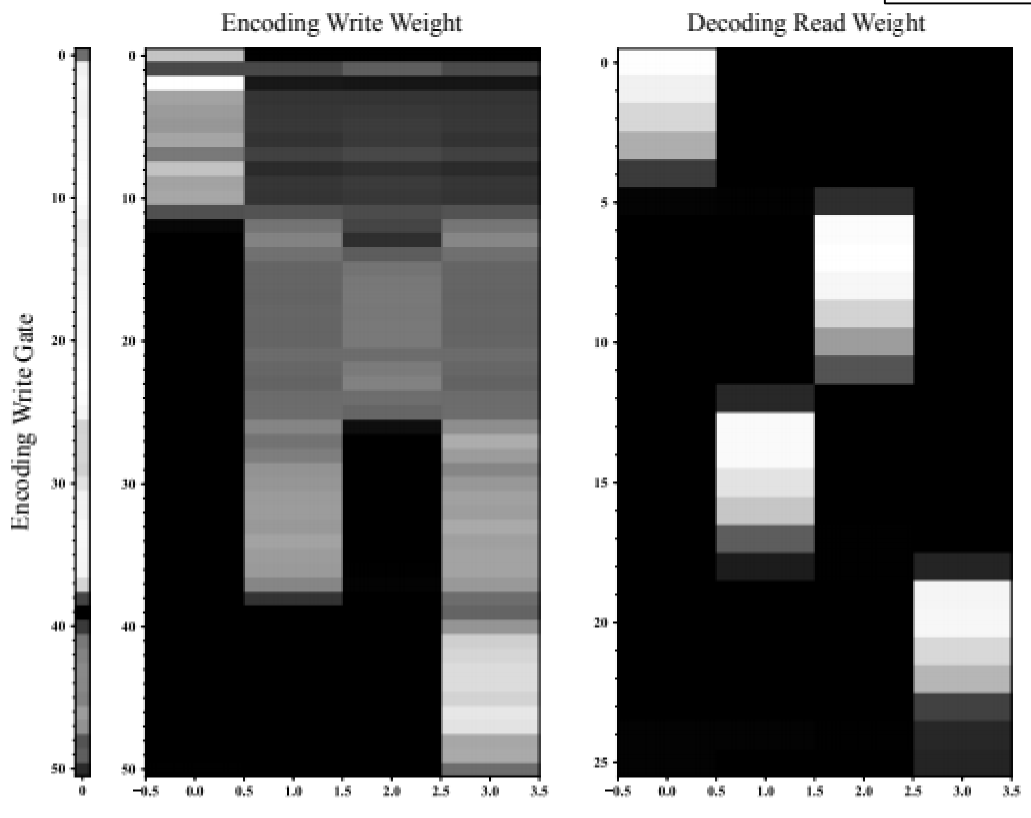
Uniform writing is optimal for memorization

Le, Hung, Truyen Tran, and Svetha Venkatesh. "Learning to Remember More with Less Memorization." In *International Conference on Learning Representations*. 2018.

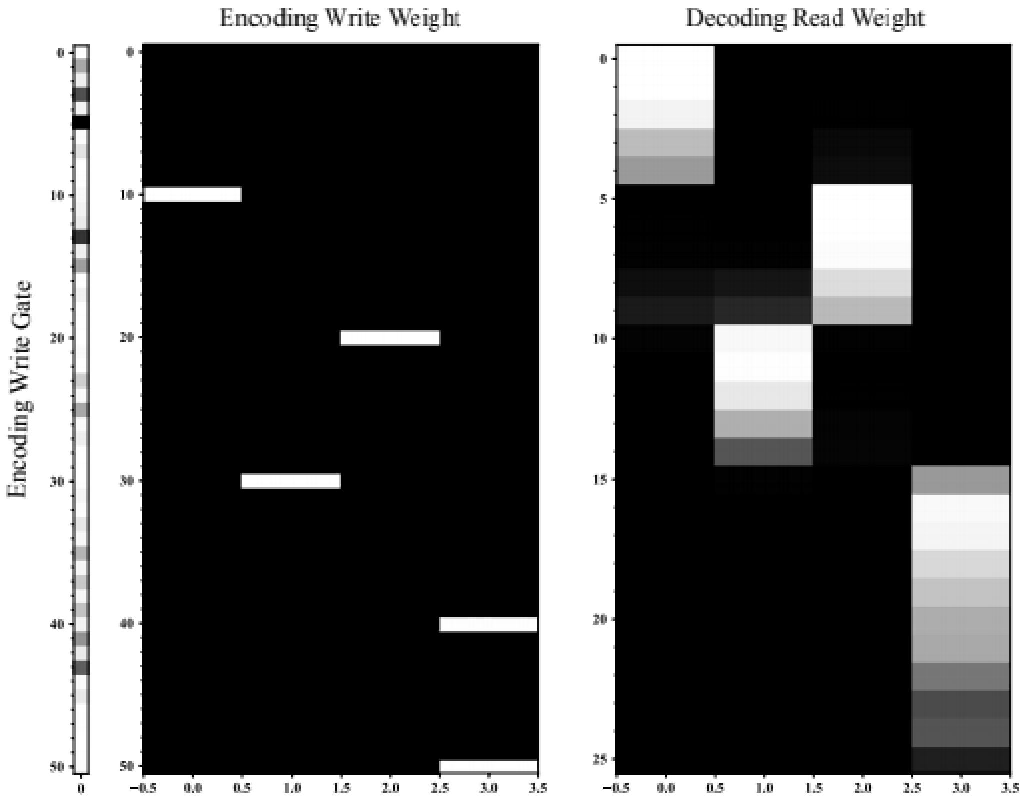
Better memorization means better algorithmic reasoning

Max	$x_1 x_2 \ldots x_T$	$\max(x_1, x_2) \max(x_3, x_4) \ldots \max(x_{T-1}, x_T)$

T=50, D=5



Regular



Uniform (cached)

Memory of independent entities

Weston, Jason, Bordes, Antoine, Chopra, Sumit, and Mikolov, Tomas.
Towards ai-complete question answering: A set of prerequisite toy tasks. CoRR, abs/1502.05698, 2015.

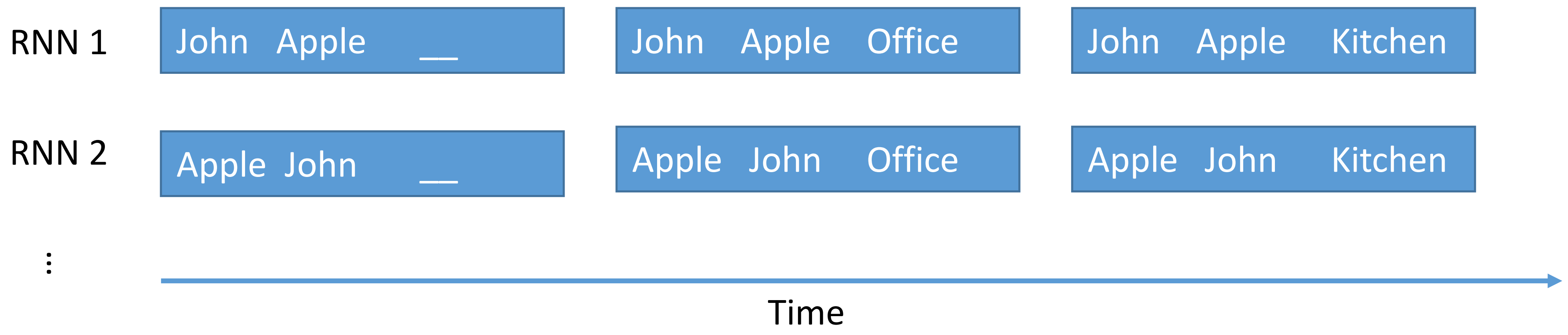
- Each slot store one or some entities
- Memory writing is done separately for each memory slot

→each slot maintains the life of one or more entities

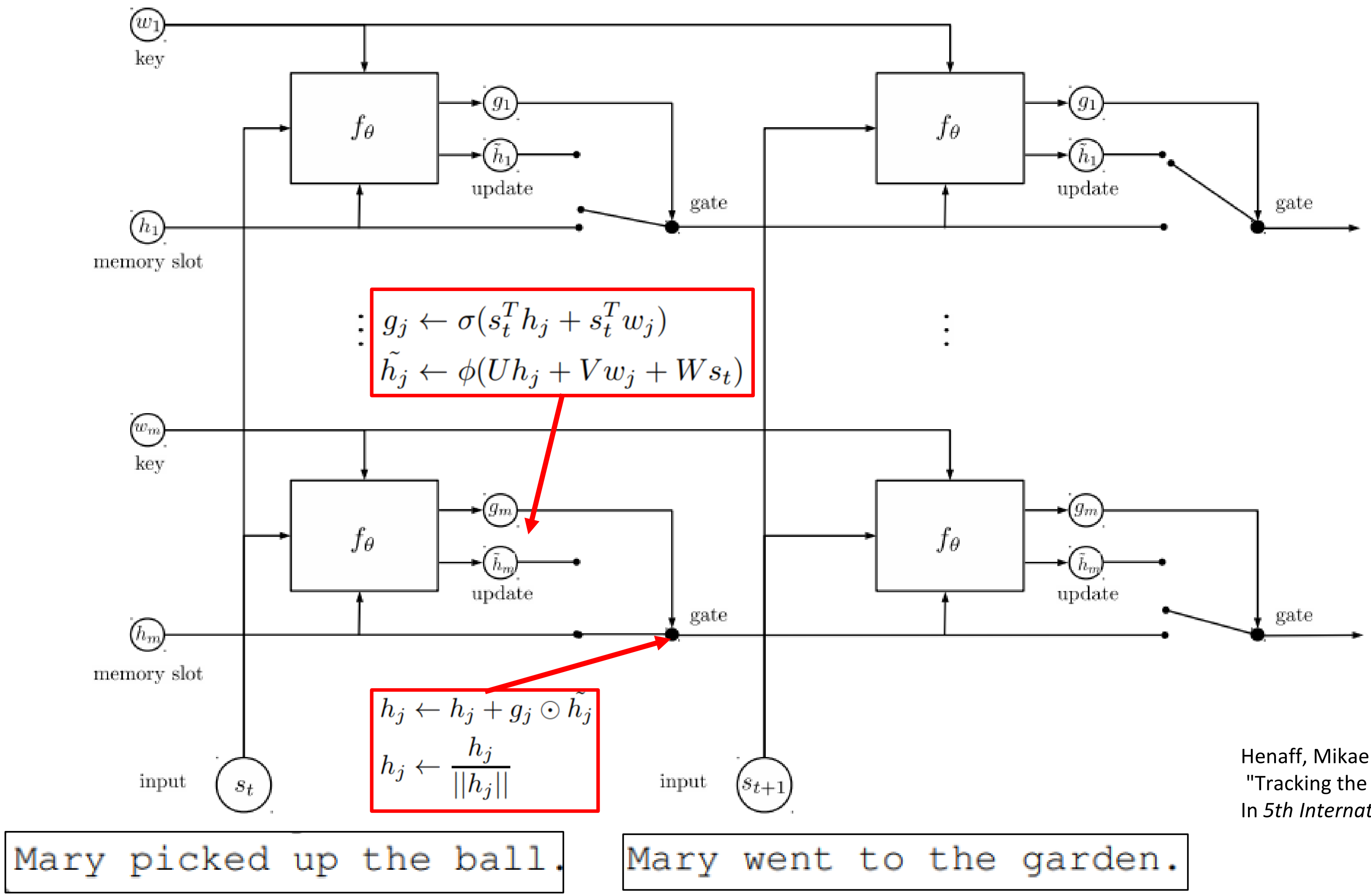
- The memory is a set of N parallel RNNs

Task 3: Three Supporting Facts

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office



Recurrent entity network



"Where is the ball?"

$$p_j = \text{Softmax}(q^T h_j)$$

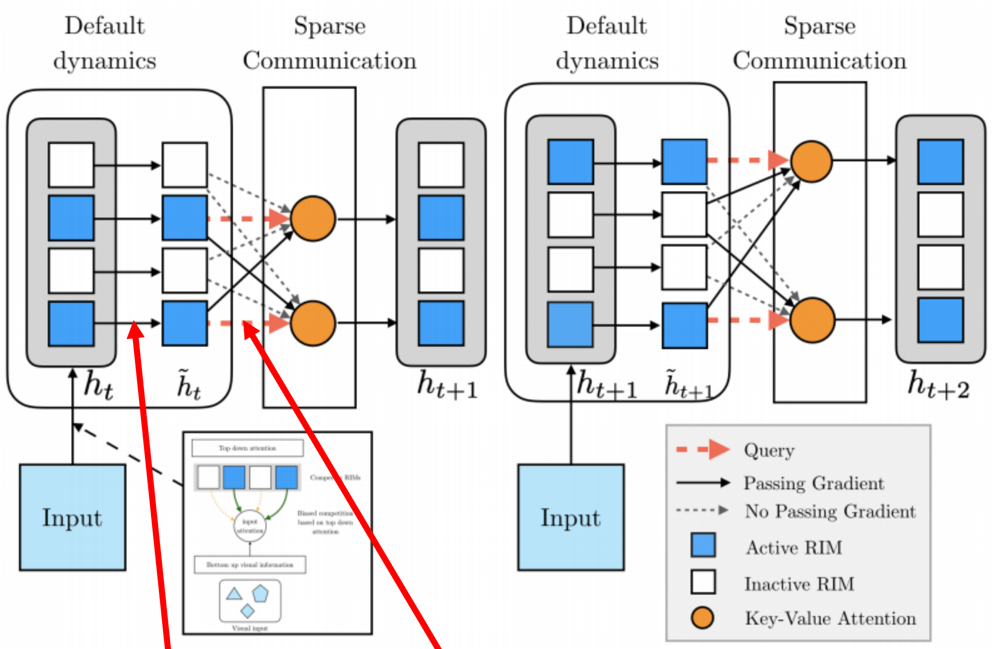
$$u = \sum_j p_j h_j$$

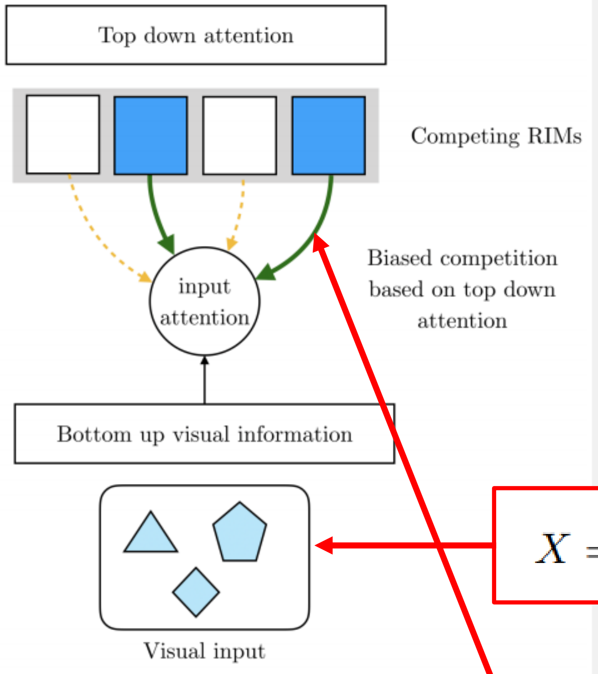
$$y = R\phi(q + Hu)$$

Garden

Henaff, Mikael, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. "Tracking the world state with recurrent entity networks." In *5th International Conference on Learning Representations, ICLR 2017*. 2017.

Recurrent Independent Mechanisms





$$X = \emptyset \oplus x_t.$$

$$\tilde{h}_{t,k} = D_k(h_{t,k}) = LSTM(h_{t,k}, A_k^{(in)}; \theta_k^{(D)}) \quad \forall k \in \mathcal{S}_t$$

$$A_k^{(in)} = \text{softmax}\left(\frac{h_t W_k^q (X W^e)^T}{\sqrt{d_e}}\right) X W^v,$$

$$Q_{t,k} = \tilde{W}_k^q \tilde{h}_{t,k}, \forall k \in \mathcal{S}_t \qquad K_{t,k} = \tilde{W}_k^e \tilde{h}_{t,k}, \forall k \qquad V_{t,k} = \tilde{W}_k^v \tilde{h}_{t,k}, \forall k$$

$$h_{t+1,k} = \text{softmax}\left(\frac{Q_{t,k}(K_{t,:})^T}{\sqrt{d_e}}\right) V_{t,:} + \tilde{h}_{t,k} \forall k \in \mathcal{S}_t.$$

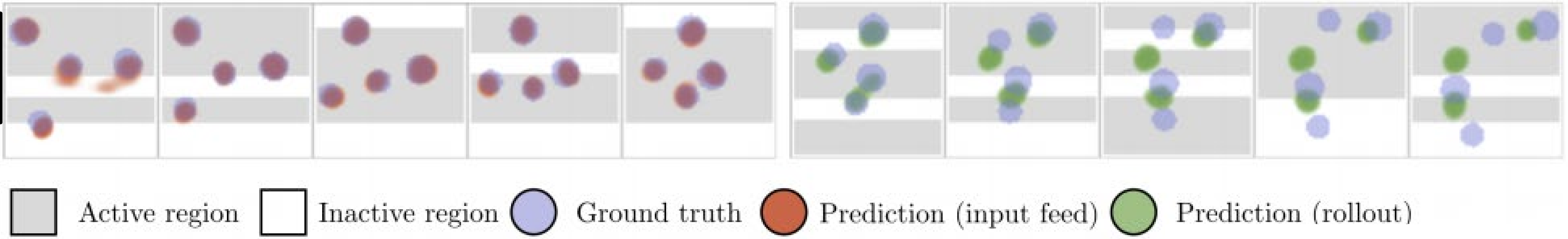
Goyal, Anirudh, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. "Recurrent independent mechanisms." ICLR21.

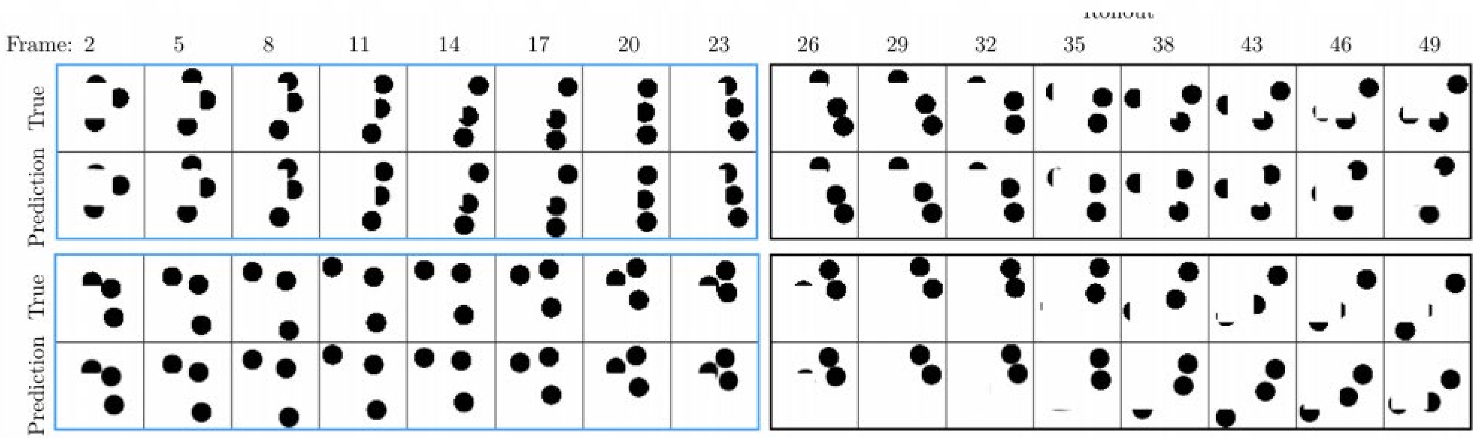
Reasoning with independent dynamics

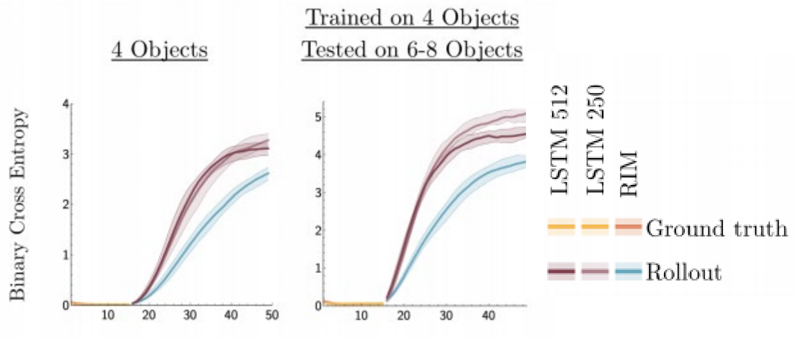
Copying	k_T	k_A	h_{size}	Train(50) CE	Test(200) CE
RIMs	6	4	600	**0.00**	**0.00**
	6	3	600	**0.00**	**0.00**
	6	2	600	**0.00**	**0.00**
	5	2	500	**0.00**	**0.00**
LSTM	-	-	300	0.00	4.32
	-	-	600	0.00	3.56
NTM	-	-	-	0.00	2.54
RMC	-	-	-	0.00	0.13
Transformers		-	-	0.00	0.54

Copy

Ball dynamics







Relational memory

Graph memory

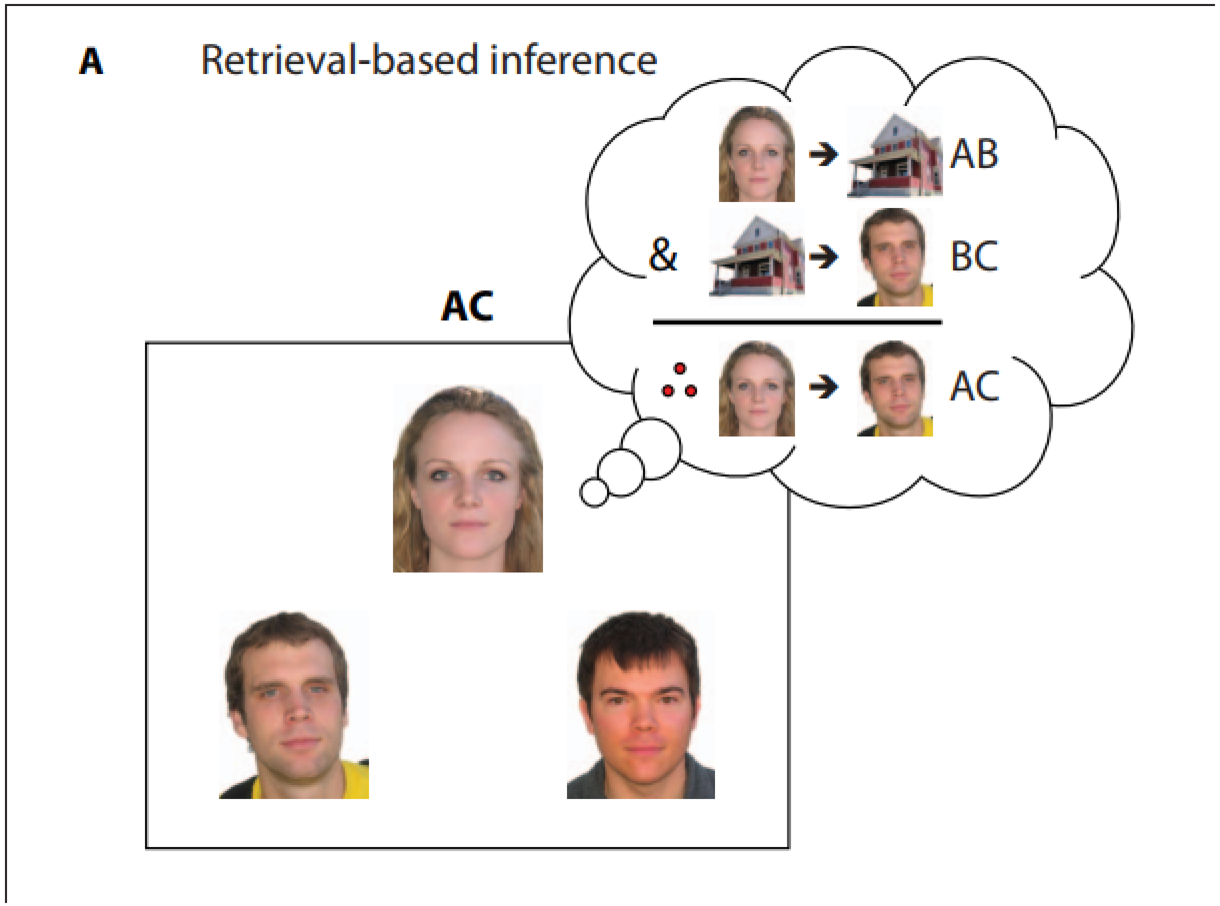
Tensor memory

Why relational memory? Item memory is weak at recognizing relationships

Item Memory

- Store and retrieve individual items
- Relate pair of items of the same time step
- Fail to relate temporally distant items

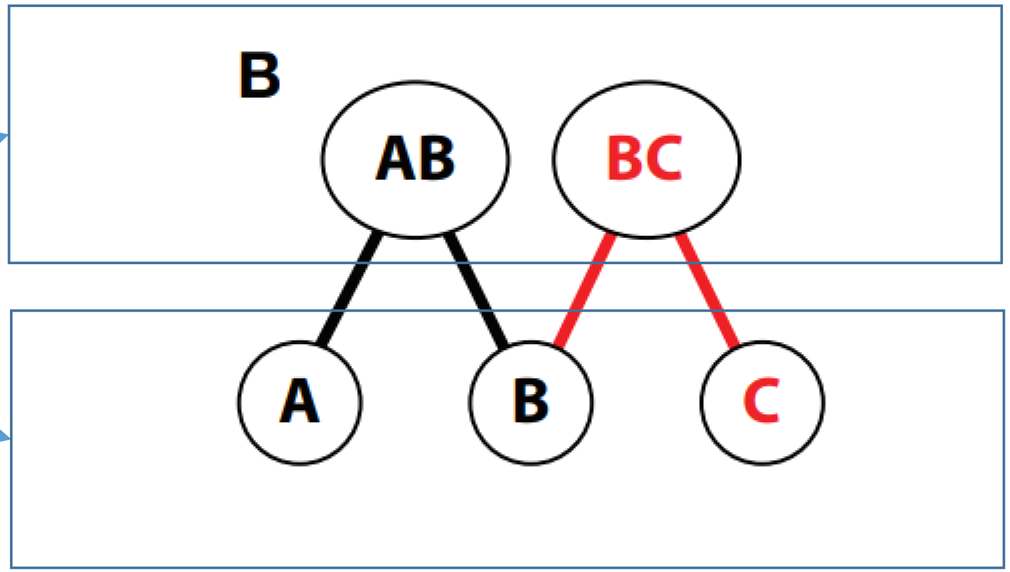
$$\hat{\mathbf{M}} = \sum_{k=1}^{q} \mathbf{b}_k \mathbf{a}_k^T$$



Dual process in memory

Item Memory

- Store items
- Simple, low-order
- System 1



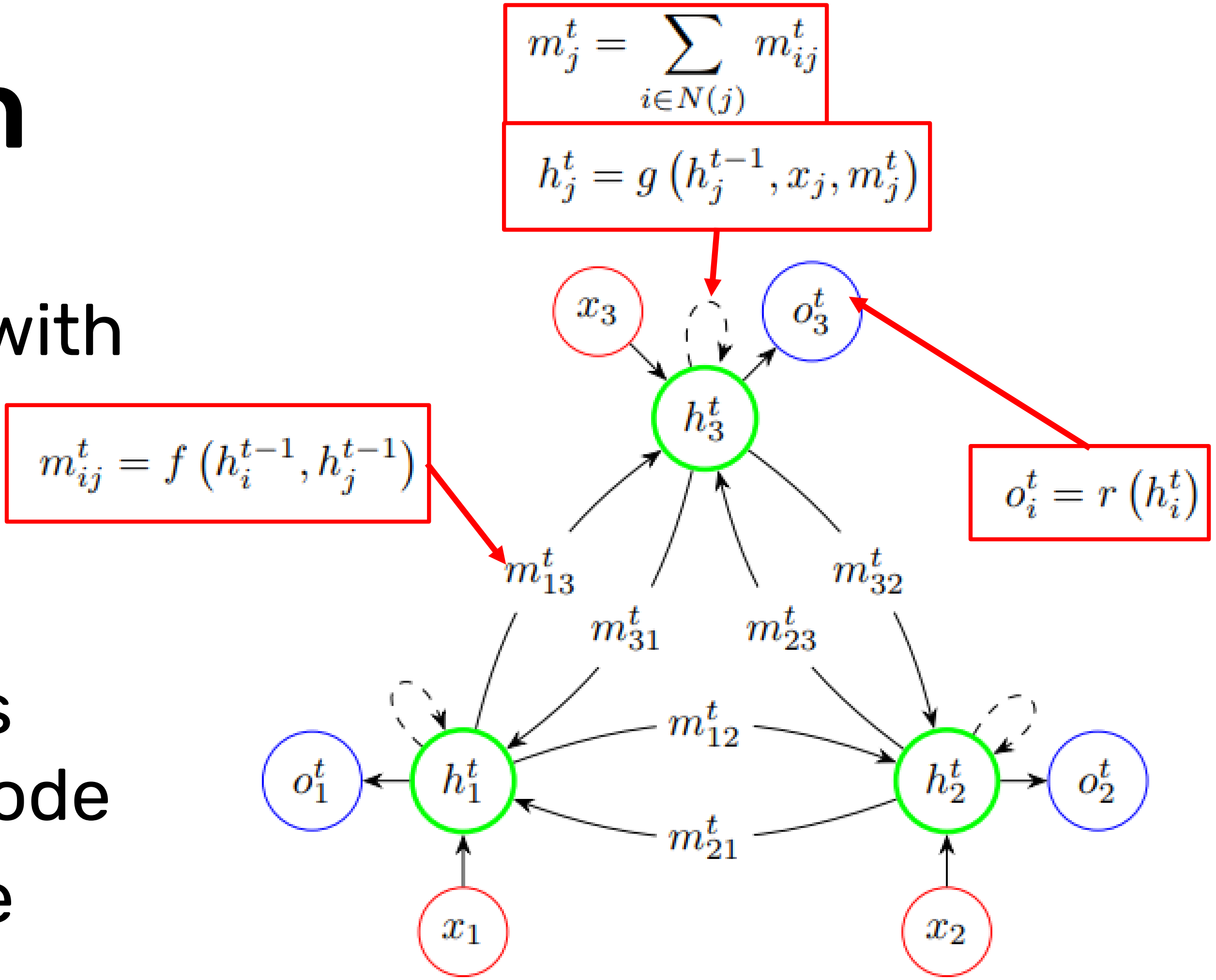
Relational Memory

- Store relationships between items
- Complicated, high-order
- System 2

Howard Eichenbaum, *Memory, amnesia, and the hippocampal system* (MIT press, 1993).
Alex Konkel and Neal J Cohen, "Relational memory and the hippocampus: representations and methods", *Frontiers in neuroscience* 3 (2009).

Memory as graph

- Memory is a static graph with fixed nodes and edges
- Relationship is somehow known
- Each memory node stores the state of the graph's node
- Write to node via message passing
- Read from node via MLP



Palm, Rasmus Berg, Ulrich Paquet, and Ole Winther. "Recurrent Relational Networks." In *NeurIPS*. 2018.

bAbI

$$x_i = \mathrm{MLP}(\mathrm{concat}(\mathrm{last}(\mathrm{LSTM}_S(s_i)), \mathrm{last}(\mathrm{LSTM}_Q(q)), \mathrm{onehot}(p_i + o)))$$

Node

Edge

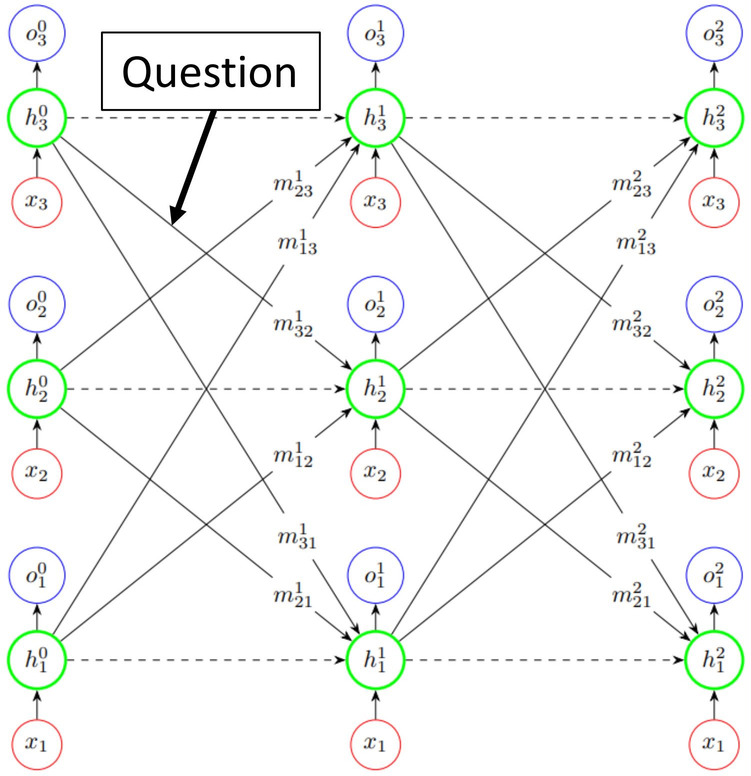
Fact 1

Fact 2

Fact 3

Question

Answer



$$o^t = r\left(\sum_i h_i^t\right)$$

$$m_{ij}^t = f\left(h_i^{t-1}, h_j^{t-1}, e_{ij}\right)$$

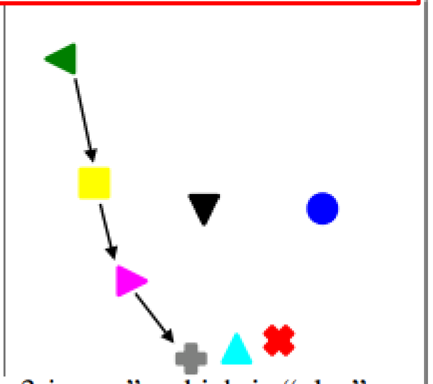
Method	N	Mean Error (%)	Failed tasks (err. >5%)
*RRN** (this work)	15	**0.46 ± 0.77**	**0.13 ± 0.35**

CLEVER

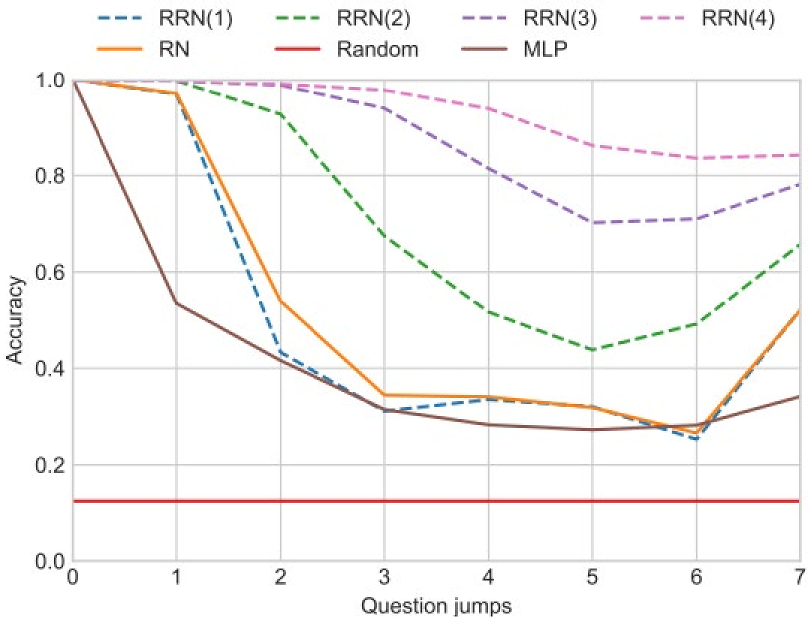
$$o_i = \mathrm{concat}(p_i, \mathrm{onehot}(c_i), \mathrm{onehot}(m_i))$$
$$q = \mathrm{concat}(\mathrm{onehot}(s), \mathrm{onehot}(n))$$
$$x_i = \mathrm{MLP}(\mathrm{concat}(o_i, q))$$

Node
(colour, shape. position)

Edge
(distance)

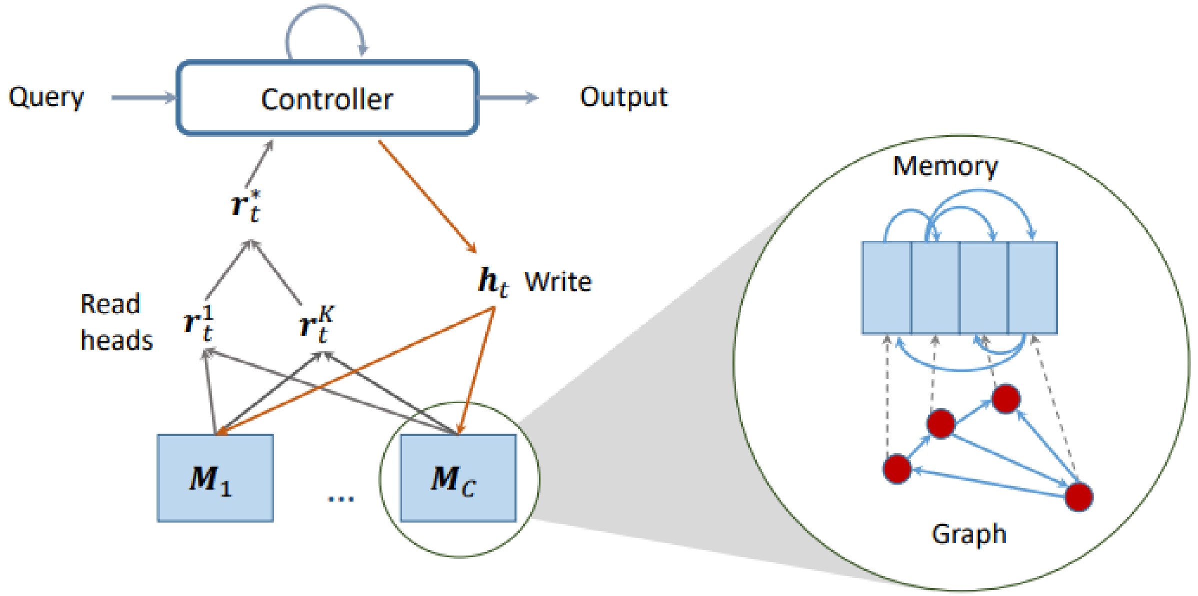


solution to the question: "green, 3 jumps", which is "plus",



Memory of graphs access conditioned on query

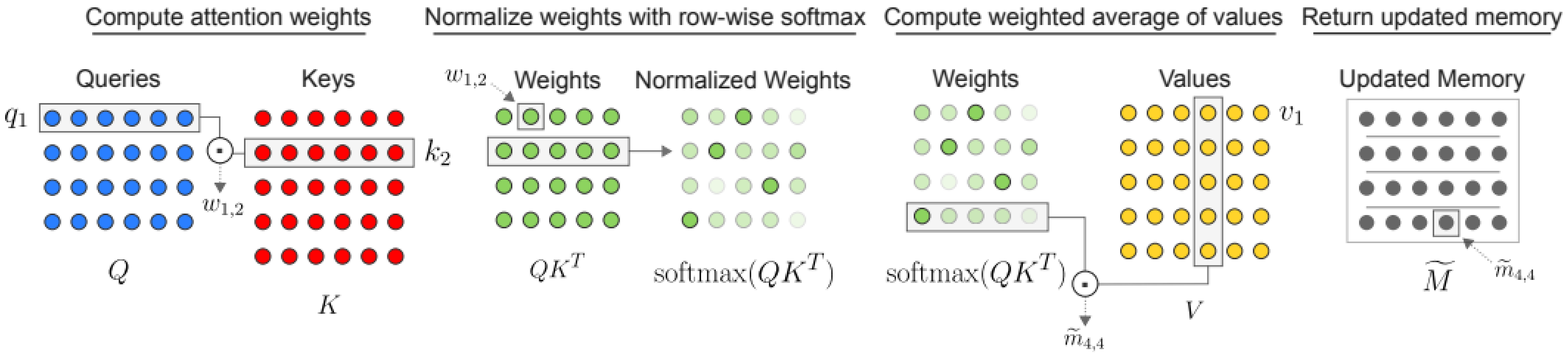
- Encode multiple graphs, each graph is stored in a set of memory row
- For each graph, the controller read/write to the memory:
  - Read uses content-based attention
  - Write use message passing
- Aggregate read vectors from all graphs to create output



Pham, Trang, Truyen Tran, and Svetha Venkatesh. "Relational dynamic memory networks." *arXiv preprint arXiv:1808.04247* (2018).

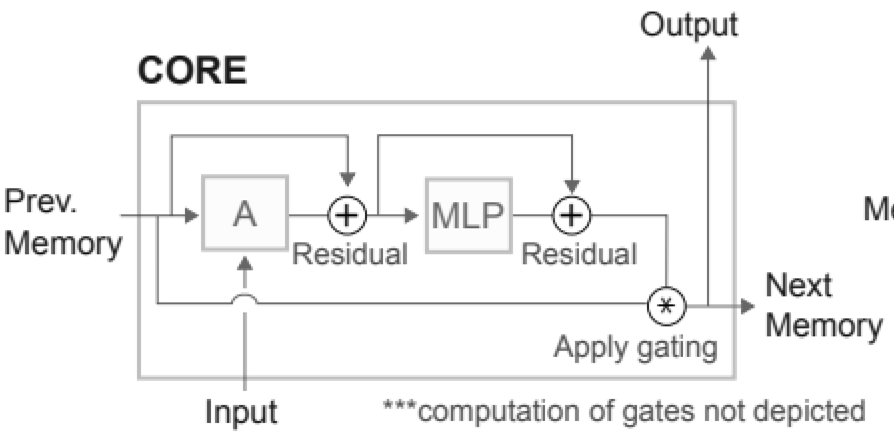
Capturing relationship can be done via memory slot interactions using attention

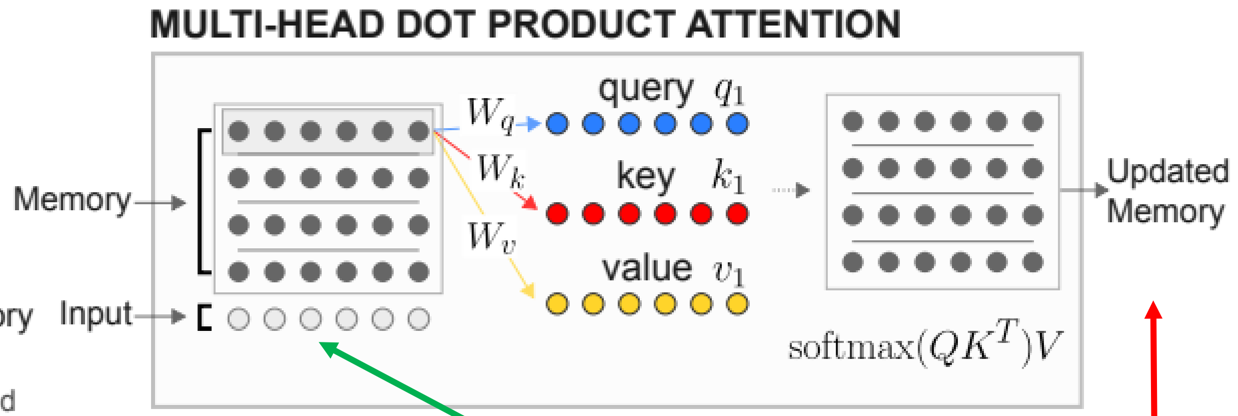
- Graph memory needs customization to an explicit design of nodes and edges
- Can we automatically learns structure with a 2d tensor memory?
- Capture relationship: each slot interacts with all other slots (self-attention)



Santoro, Adam, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Théophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. "Relational recurrent neural networks." In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7310-7321. 2018.

Relational Memory Core (RMC) operation



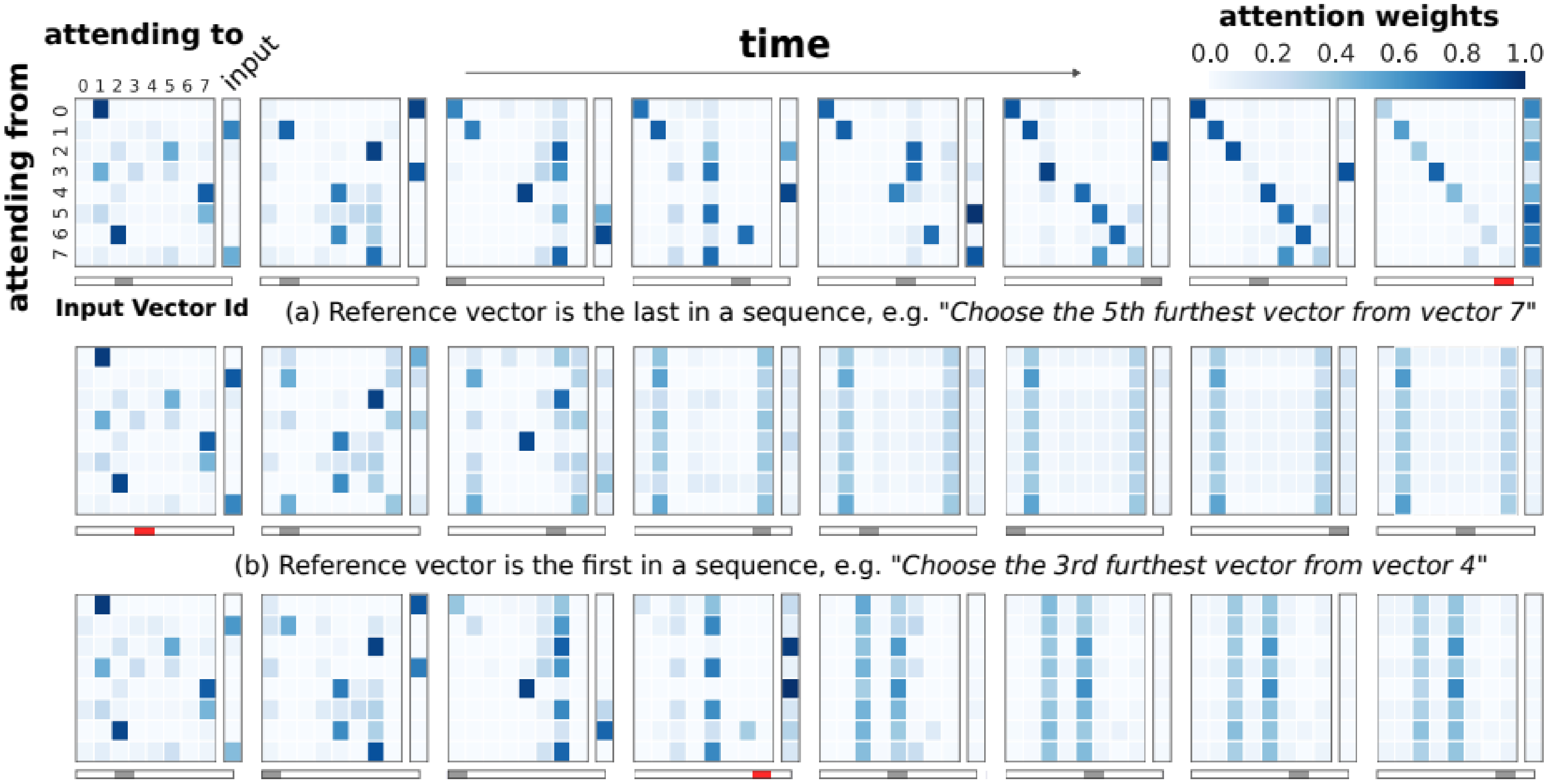


RNN-like
Interface

$$
\begin{aligned}
s_{i,t} &= (h_{i,t-1}, m_{i,t-1}) \\
f_{i,t} &= W^f x_t + U^f h_{i,t-1} + b^f \\
i_{i,t} &= W^i x_t + U^i h_{i,t-1} + b^i \\
o_{i,t} &= W^o x_t + U^o h_{i,t-1} + b^o \\
m_{i,t} &= \sigma(f_{i,t} + \tilde{b}^f) \circ m_{i,t-1} + \sigma(i_{i,t}) \circ \underbrace{g_\psi(\tilde{m}_{i,t})}_{\text{row/memory-wise MLP with layer normalisation}} \\
h_{i,t} &= \sigma(o_{i,t}) \circ \tanh(m_{i,t}) \\
s_{i,t+1} &= (m_{i,t}, h_{i,t})
\end{aligned}
$$

$$
\widetilde{M} = \mathrm{softmax}\left(\frac{MW^q(\boxed{[M;x]}W^k)^T}{\sqrt{d^k}}\right)[M;x]W^v,
$$

Allowing pair-wise interactions can answer questions on temporal relationship

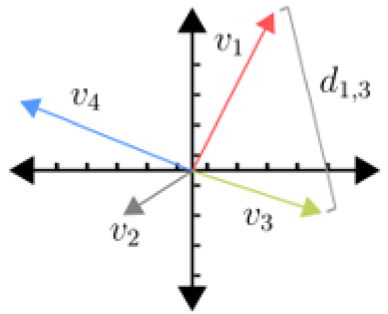


(a) Reference vector is the last in a sequence, e.g. "*Choose the 5th furthest vector from vector 7*"

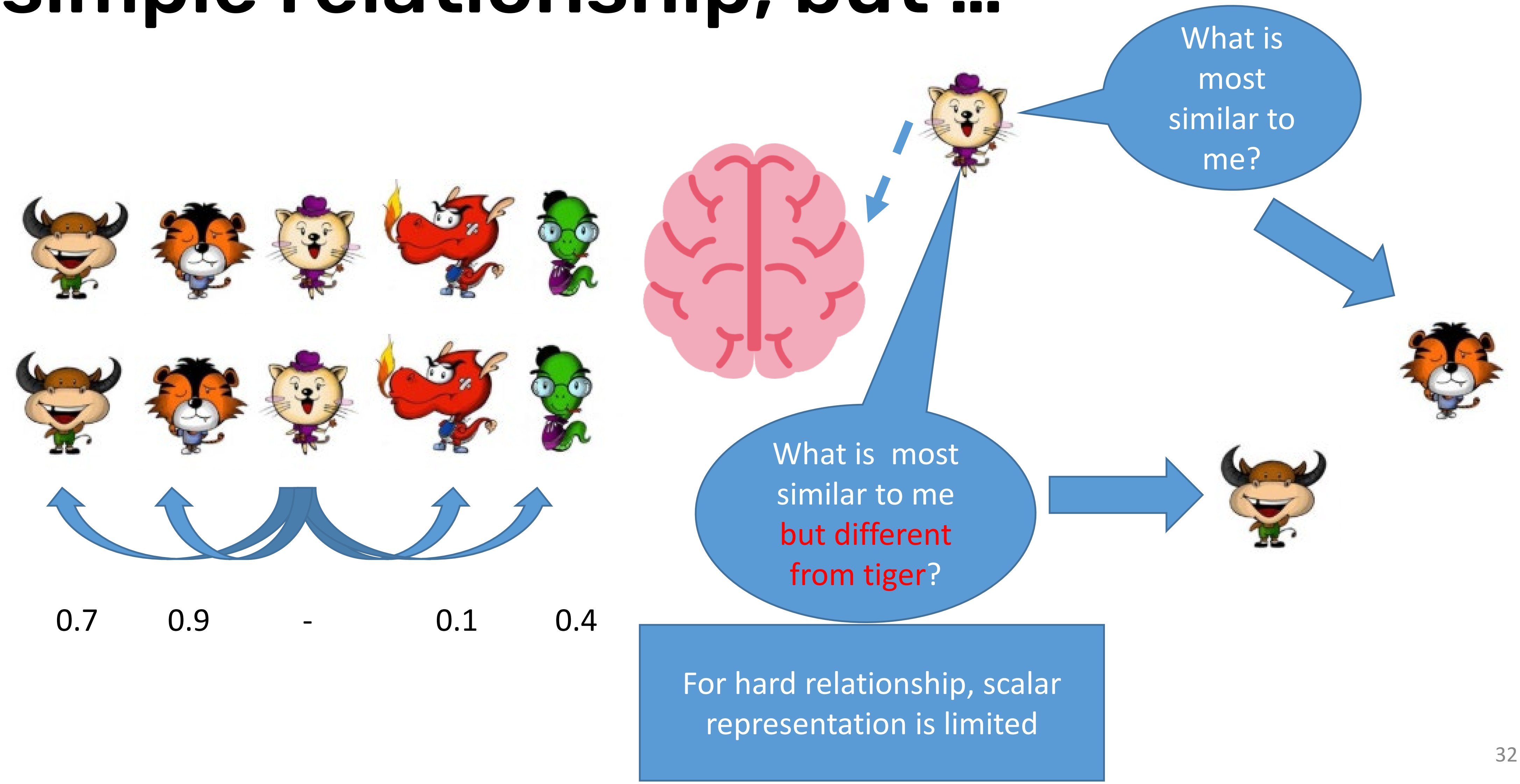
(b) Reference vector is the first in a sequence, e.g. "*Choose the 3rd furthest vector from vector 4*"

(c) Reference vector comes in the middle of a sequence, e.g. "*Choose the 6th furthest vector from vector 6*"

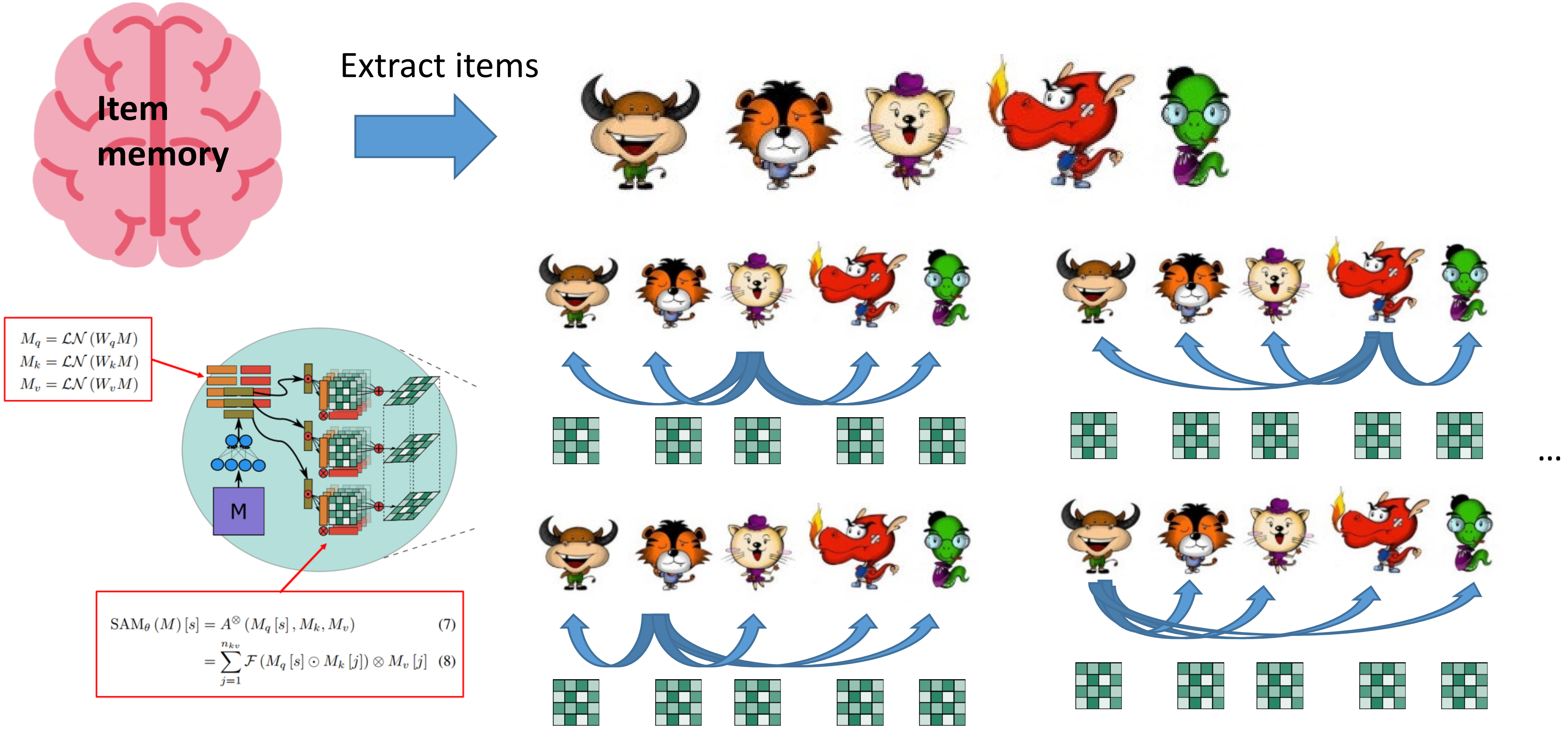
What is the N^{th} farthest from vector m?

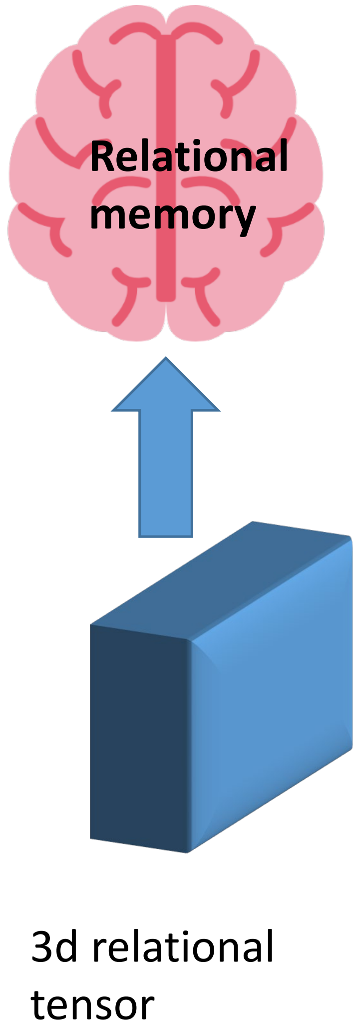


Dot product attention works for simple relationship, but …



Complicated relationship needs high-order relational memory





Le, Hung, Truyen Tran, and Svetha Venkatesh. "Self-attentive associative memory." In *International Conference on Machine Learning*, pp. 5682-5691. PMLR, 2020.

Associate every pairs of them

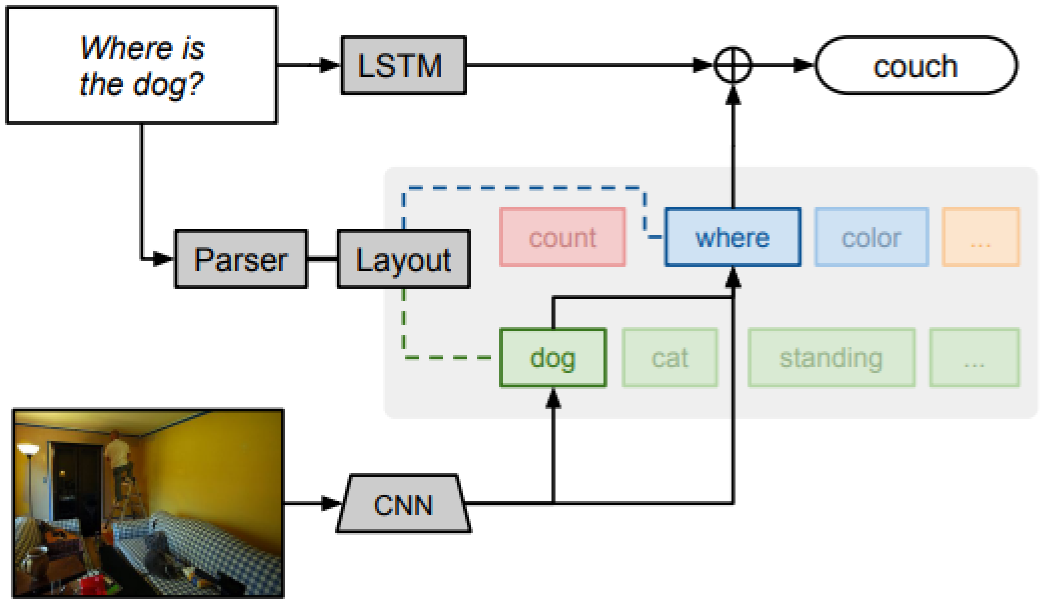
Program memory

Module memory

Stored-program memory

Predefining program for subtask

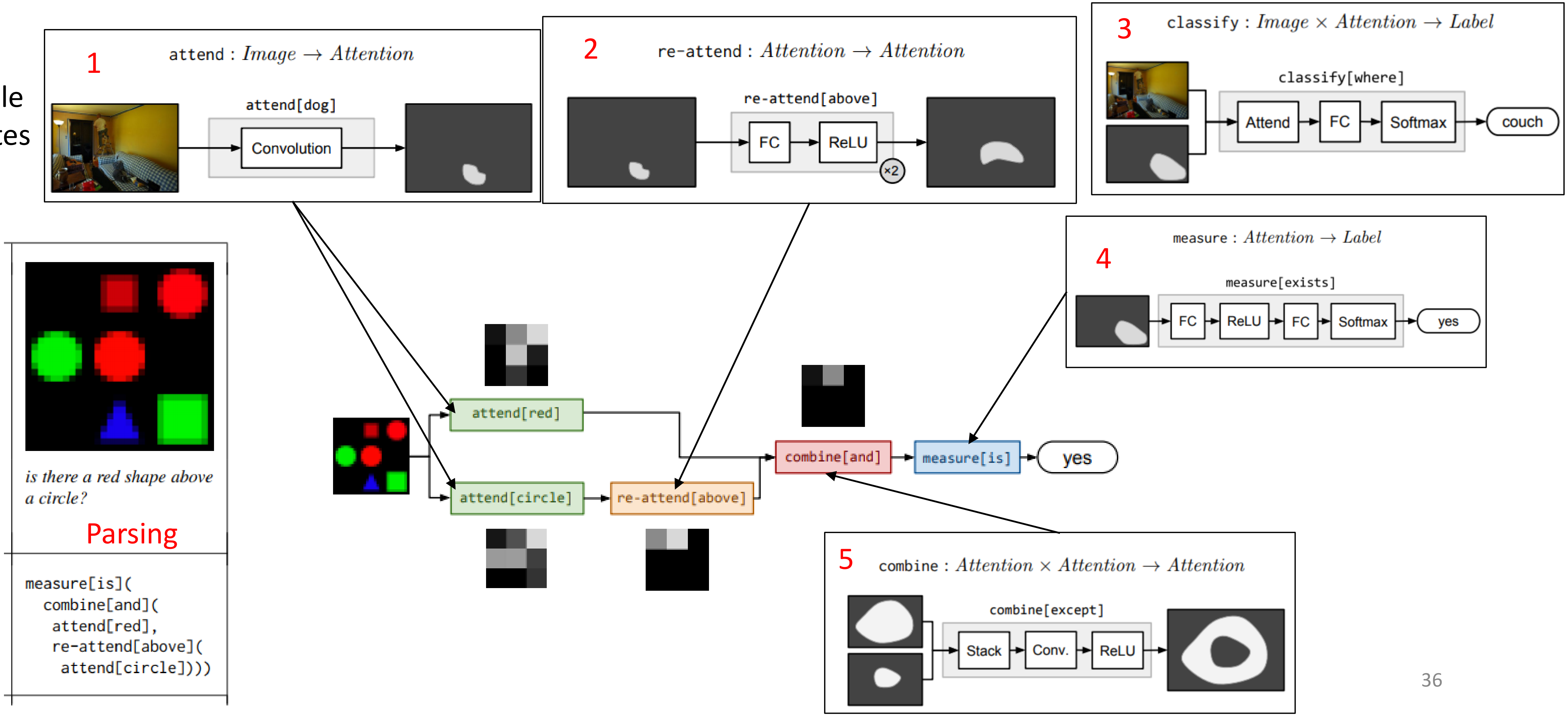
- A program designed for a task becomes a module
- Parse a question to module layout (order of program execution)
- Learn the weight of each module to master the task



Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Neural module networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 39-48. 2016.

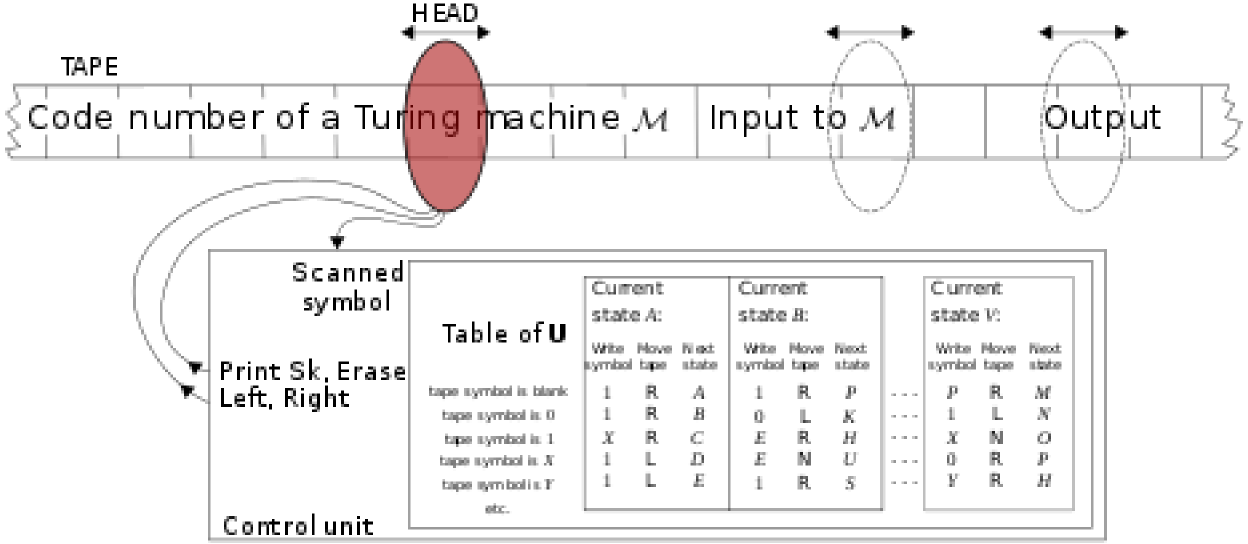
Program selection is based on parser, others are end2end trained

5 module templates



The most powerful memory is one that stores both program and data

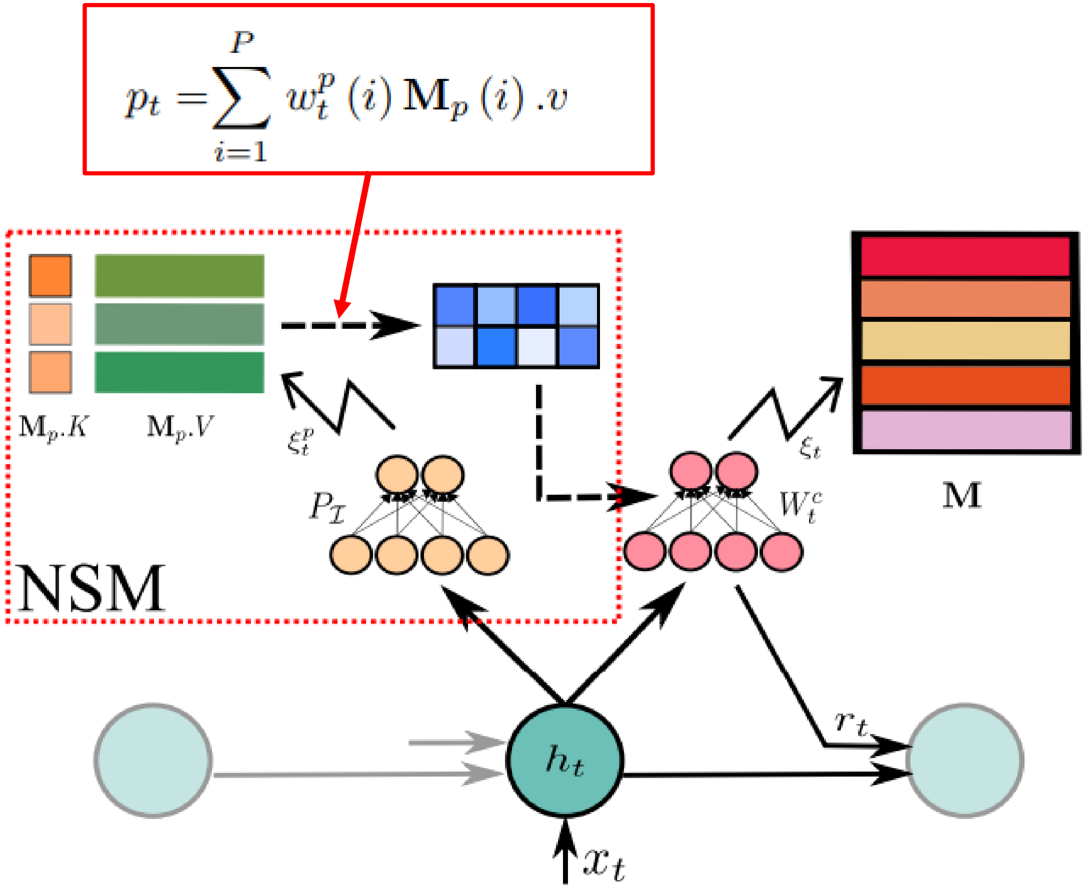
- Computer architecture: Universal Turing Machines/Harvard/VNM
- Stored-program principle
- Break a big task into subtasks, each can be handled by a TM/single purposed program stored in a program memory



https://en.wikipedia.org/

NUTM: Learn to select program (neural weight) via program attention

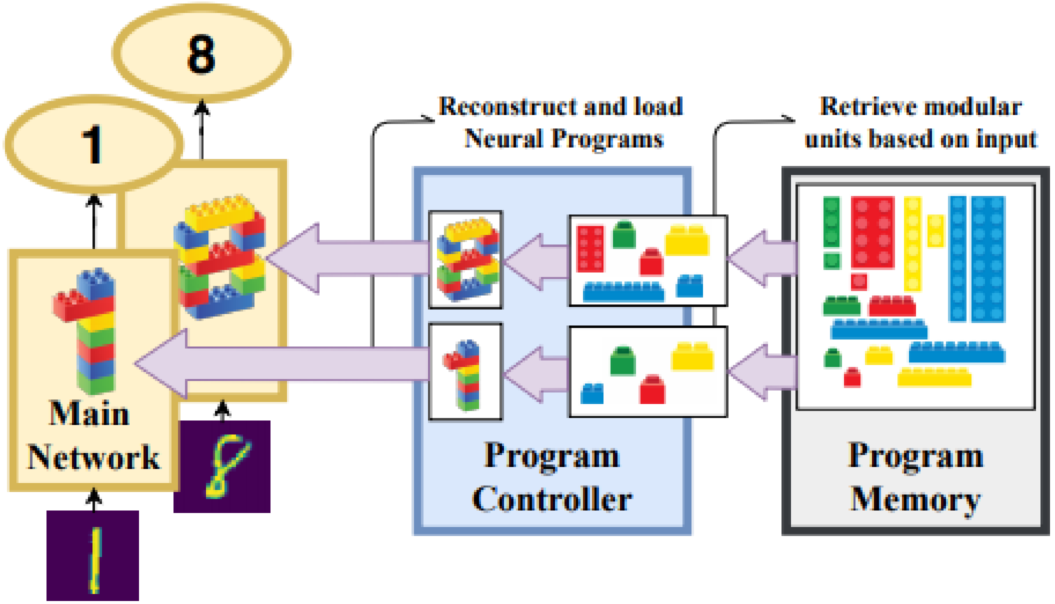
- Neural stored-program memory (NSM) stores key (the address) and values (the weight)
- The weight is selected and loaded to the controller of NTM
- The stored NTM weights and the weight of the NUTM is learnt end-to-end by backpropagation



Le, Hung, Truyen Tran, and Svetha Venkatesh. "Neural Stored-program Memory." In *International Conference on Learning Representations*. 2019.

Scaling with memory of mini-programs

- Prior, 1 program = 1 neural network (millions of parameters)
- Parameter inefficiency since the programs do not share common parameters
- Solution: store sharable mini-programs to compose infinite number of programs



it is analogous to building Lego structures corresponding to inputs from basic Lego bricks.

Recurrent program attention to retrieve singular components of a program

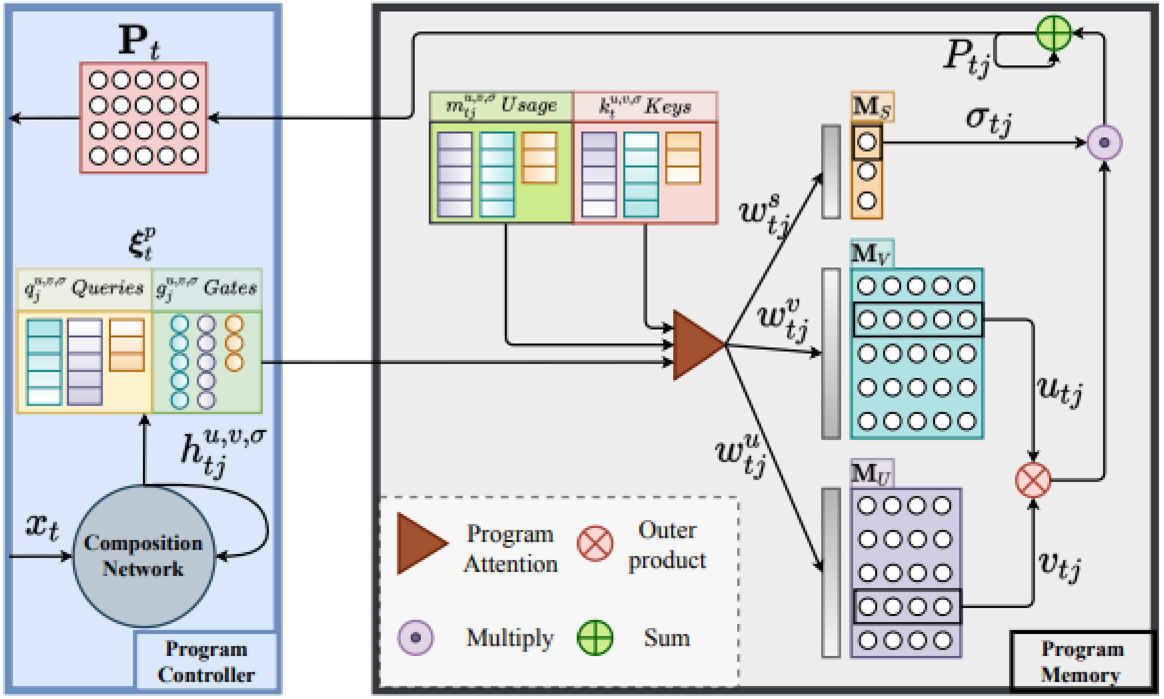
$$\mathbf{P}_t = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

$$= \sum_{n}^{r_m} \sigma_{tn} u_{tn} v_{tn}^{\top}$$

$$u_{tn} = \sum_{i=1}^{P_u} w_{tin}^{u} \mathbf{M}_U(i)$$

$$v_{tn} = \sum_{i=1}^{P_v} w_{tin}^{v} \mathbf{M}_V(i)$$

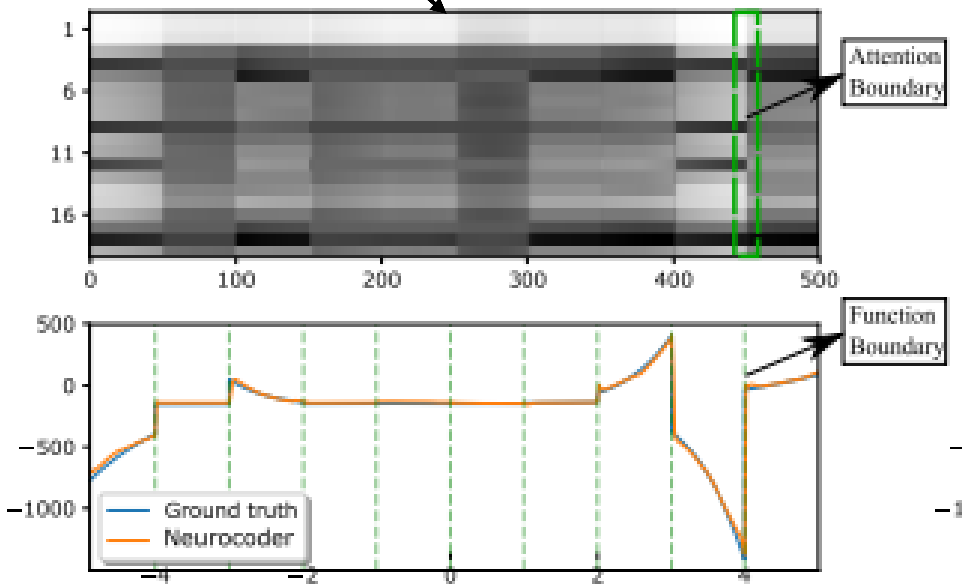
$$\sigma_{tn} = \begin{cases} \mathrm{softplus}\left(\sum_{i=1}^{P_s} w_{tin}^{\sigma} \mathbf{M}_S(i)\right) & n = r_m \\ \sigma_{tn+1} + \mathrm{softplus}\left(\sum_{i=1}^{P_s} w_{tin}^{\sigma} \mathbf{M}_S(i)\right) & n < r_m \end{cases}$$

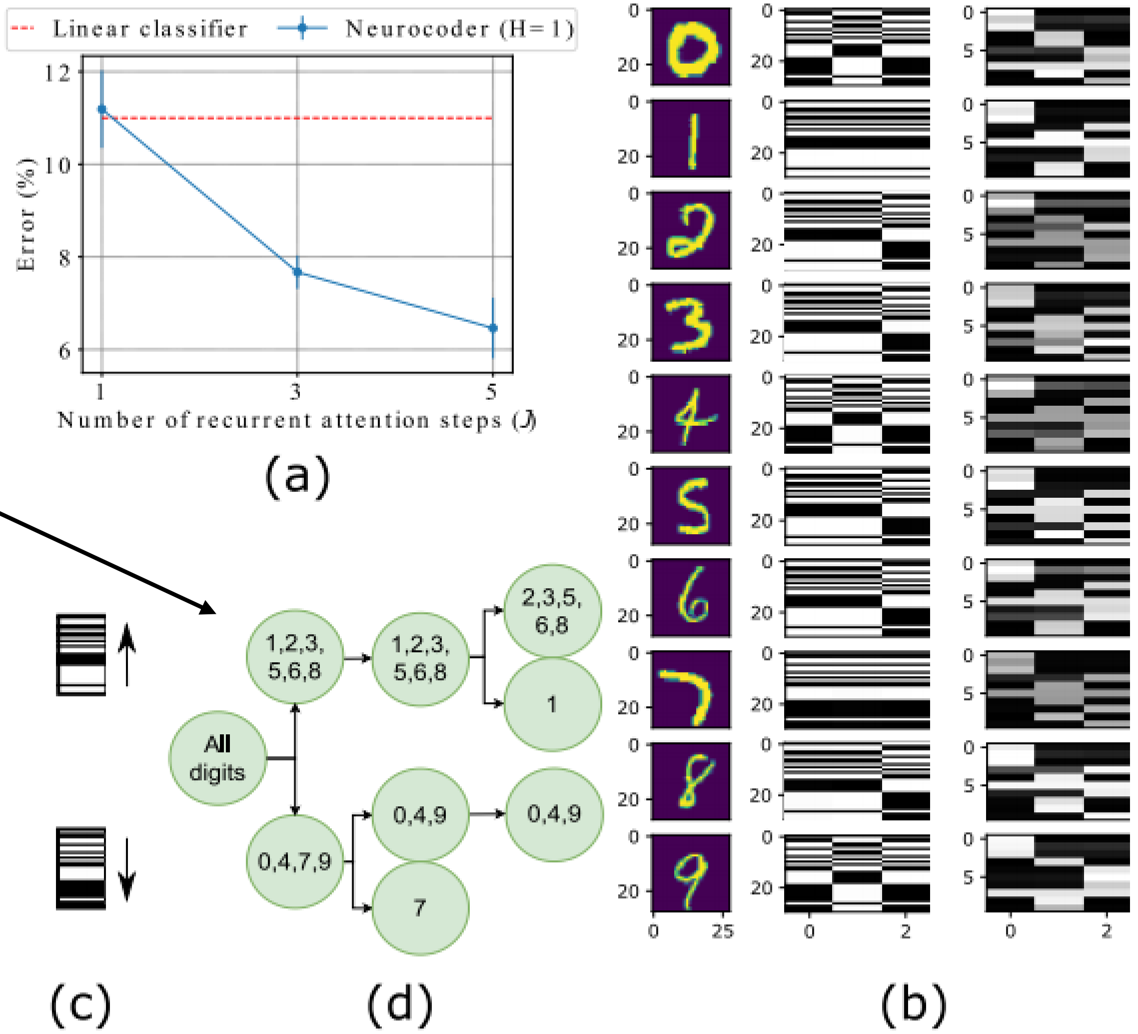


Le, Hung, and Svetha Venkatesh. "Neurocoder: Learning General-Purpose Computation Using Stored Neural Programs." *arXiv preprint arXiv:2009.11443* (2020).

Program attention is equivalent to binary decision tree reasoning

Recurrent program attention auto detects task boundary

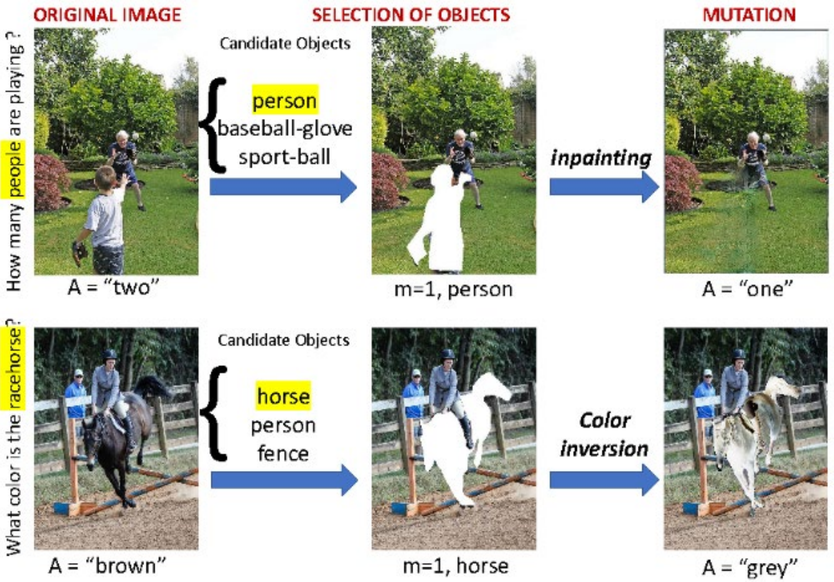




Agenda

- Reasoning with external memories
  - Memory of entities – memory-augmented neural networks
  - Memory of relations with tensors and graphs
  - Memory of programs & neural program construction.
- **Learning to reason with less labels:**
  - Data augmentation with analogical and counterfactual examples
  - Question generation
  - Self-supervised learning for question answering
  - Learning with external knowledge graphs
- Recursive reasoning with neural theory of mind.

Data Augmentation with Analogical and Counterfactual Examples



Visual counterfactual example

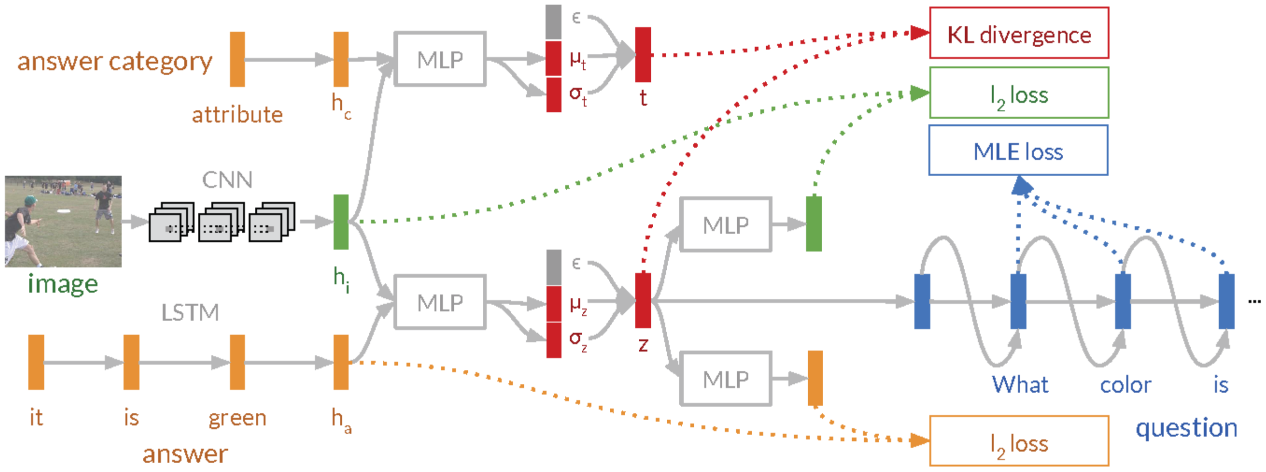
- **Poor generalization** when training under independent and identically distributed assumption.
- **Intuition**: augmenting counterfactual samples to allow machines to understand the critical changes in the input that lead to changes in the answer space.
  - Perceptually similar, yet
  - Semantically dissimilar realistic samples

Mutation Type	Question	Answer
Original	Is the lady holding the baby?	Yes
Substitution (Negation)	Is the lady not holding the baby?	No
Substitution (Adversarial)	Is the cat holding the baby?	No
Original	How many people are there?	Three
Deletion (Masking)	How many [MASK] are there?	"Number"
Original	What is the color of the man's shirt?	Blue
Substitution (Negation)	What is not the color of the man's shirt?	Magenta
Deletion (Masking)	Is the [MASK] holding the baby?	Can't say
Original	What color is the umbrella ?	Pink
Deletion (Masking)	What color is the [MASK]?	"color"

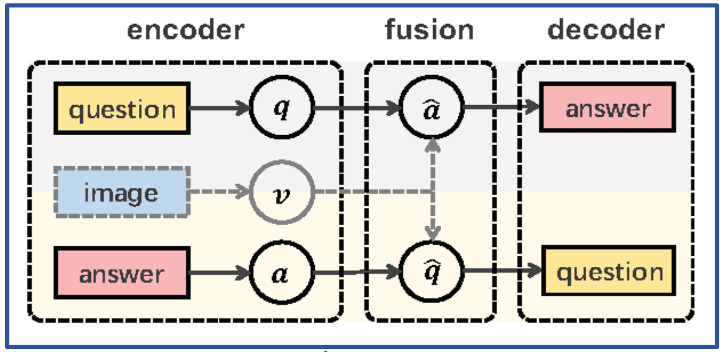
Language counterfactual examples

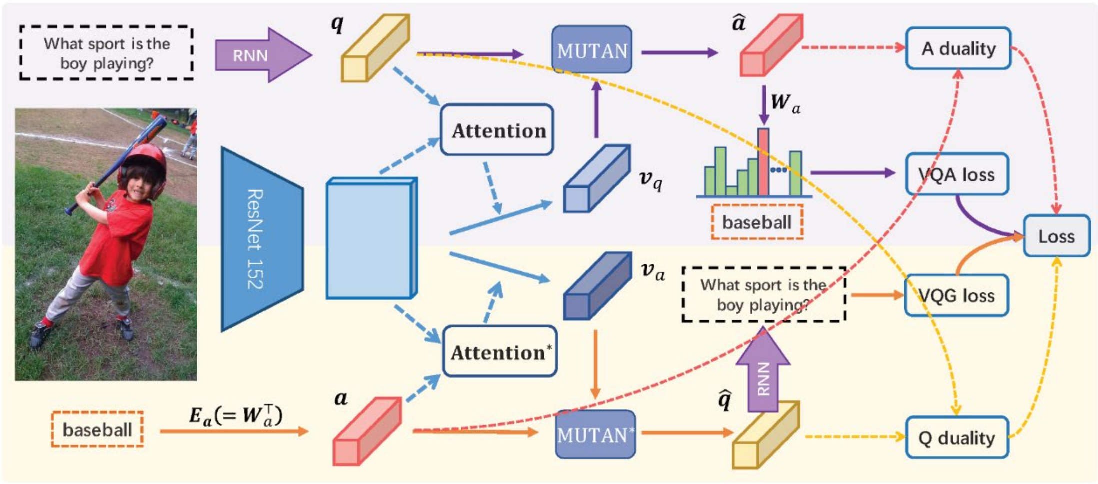
Gokhale, Tejas, et al. "Mutant: A training paradigm for out-of-distribution generalization in visual question answering." *EMNLP'20*.

Question Generations



Krishna, Ranjay, Michael Bernstein, and Li Fei-Fei. "Information maximizing visual question generation." *CVPR*'19.





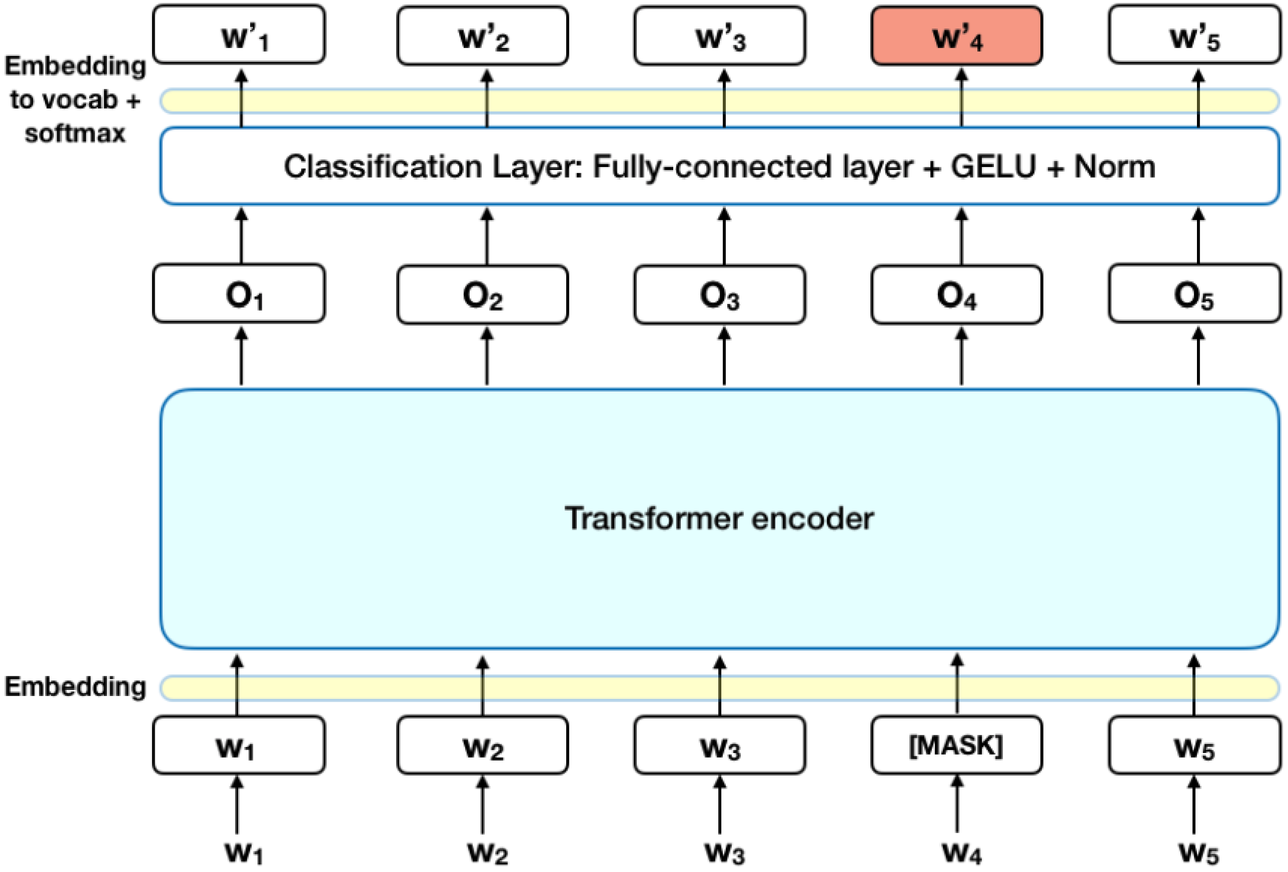
- Question answering is a zero-shot learning problem. Question generation helps cover a wider range of concepts.
- Question generation can be done with either supervised and unsupervised learning.

Li, Yikang, et al. "Visual question generation as dual task of visual question answering." *CVPR*'18.

BERT: Transformer That Predicts Its Own Masked Parts

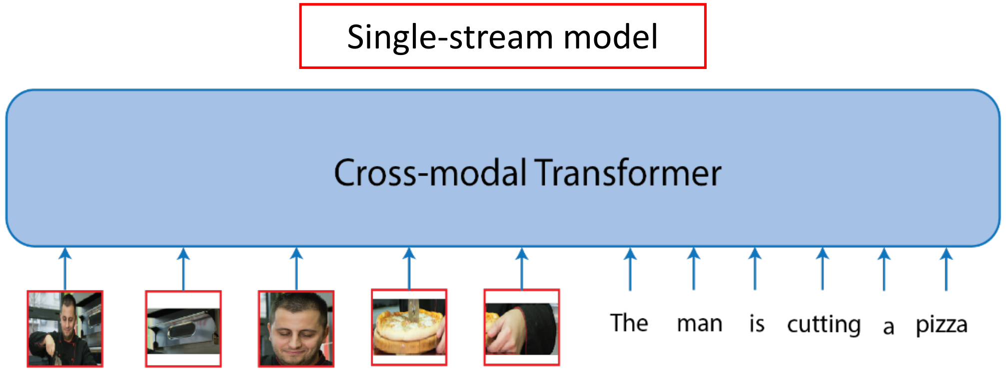
BERT is like parallel approximate pseudo-likelihood

- ~ Maximizing the conditional likelihood of some variables given the rest.
- When the number of variables is large, this converses to MLE (maximum likelihood estimate).

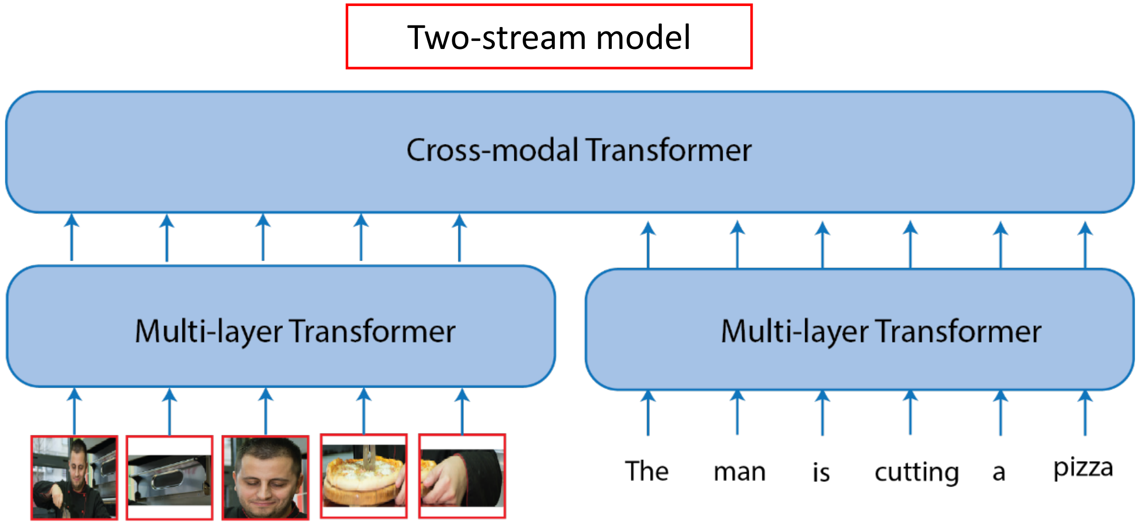


Visual QA as a Down-stream Task of Visual-Language BERT Pre-trained Models

Numerous pre-trained visual language models during 2019-2021.



VisualBERT (Li, Liunian Harold, et al., 2019)
VL-BERT (Su, Weijie, et al., 2019)
UNITER (Chen, Yen-Chun, et al., 2019)
12-in-1 (Lu, Jiasen, et al., 2020)
Pixel-BERT (Huang, Zhicheng, et al., 2019)
OSCAR (Li, Xiujun, et al., 2020)



ViLBERT (Lu, Jiasen, et al. , 2019)
LXMERT (Tan, Hao, and Mohit Bansal, 2019)

[Slide credit: Licheng Yu et al.]

Learning with External Knowledge

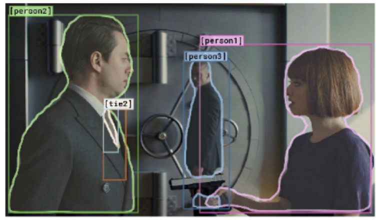
Why external knowledge for reasoning?

- Questions can be beyond visual recognition (e.g. firetrucks usually use a fire hydrant).
- Human's prior knowledge for cognition-level reasoning (e.g. human's goals, intents etc.)

Q: What sort of vehicle uses this item?
A: firetruck

Q: What is the sports position of the man in the orange shirt?
A: goalie/goalkeeper

Marino, Kenneth, et al. "Ok-vqa: A visual question answering benchmark requiring external knowledge." *CVPR'19*.



Why is [person1] pointing a gun at [person2]?

a) [person1] wants to kill [person2].(1%)

b) [person1] and [person3] are robbing the bank and [person2] is the bank manager. (71%)

c) [person2] has done something to upset [person1]. (18%)

d) Because [person2] is [person1]'s daughter. [person1] wants to protect [person2]. (8%)

b) is right because...

a) [person1] is chasing [person1] and [person3] because they just robbed a bank. (33%)

b) Robbers will sometimes hold their gun in the air to get everyone's attention. (5%)

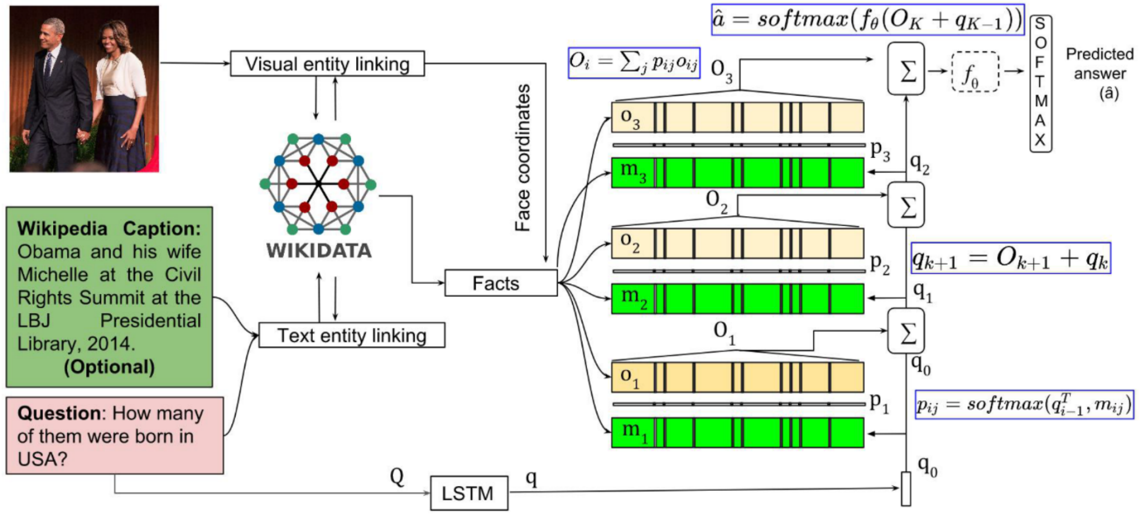
c) The vault in the background is similar to a bank vault. [person3] is waiting by the vault for someone to open it. (49%)

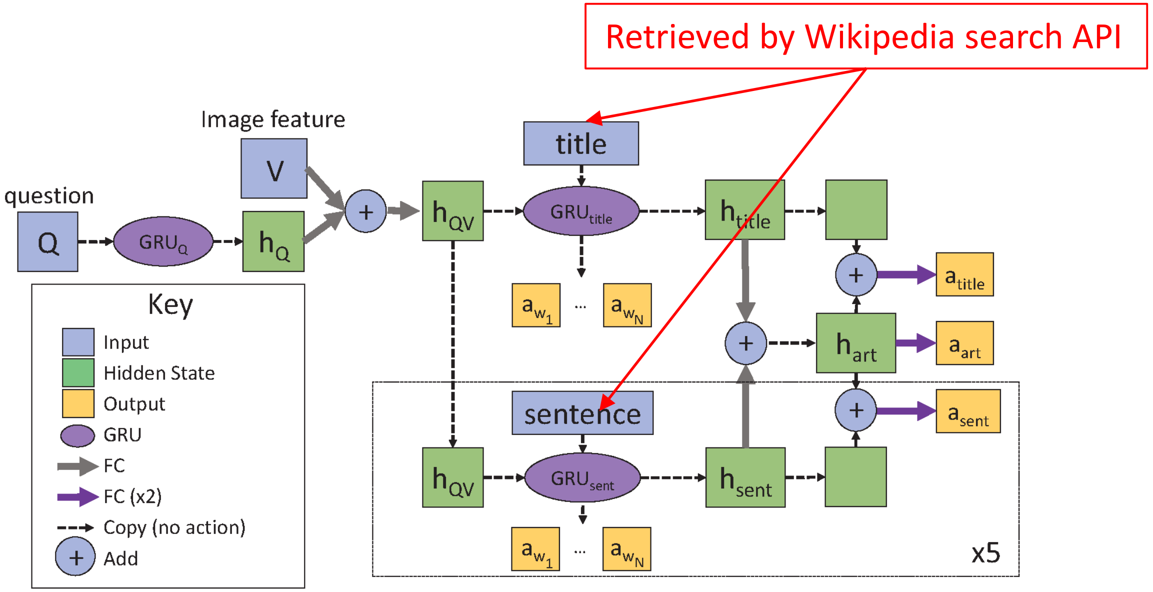
d) A room with barred windows and a counter usually resembles a bank. (11%)

Zellers, Rowan, et al. "From recognition to cognition: Visual commonsense reasoning." *CVPR'19*.

Learning with External Knowledge

Shah, Sanket, et al. "Kvqa: Knowledge-aware visual question answering." *AAAI'19*.

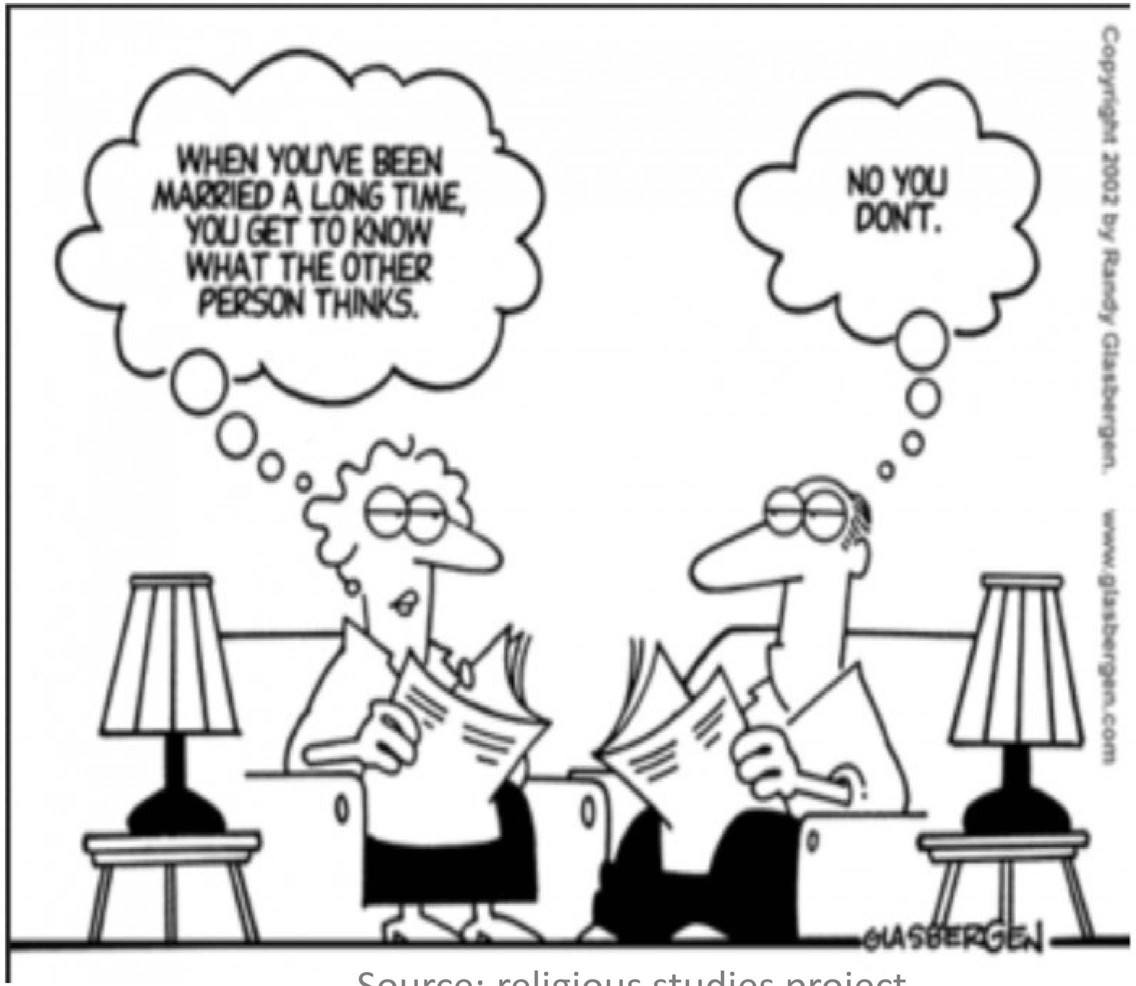




Marino, Kenneth, et al. "Ok-vqa: A visual question answering benchmark requiring external knowledge." *CVPR'19*.

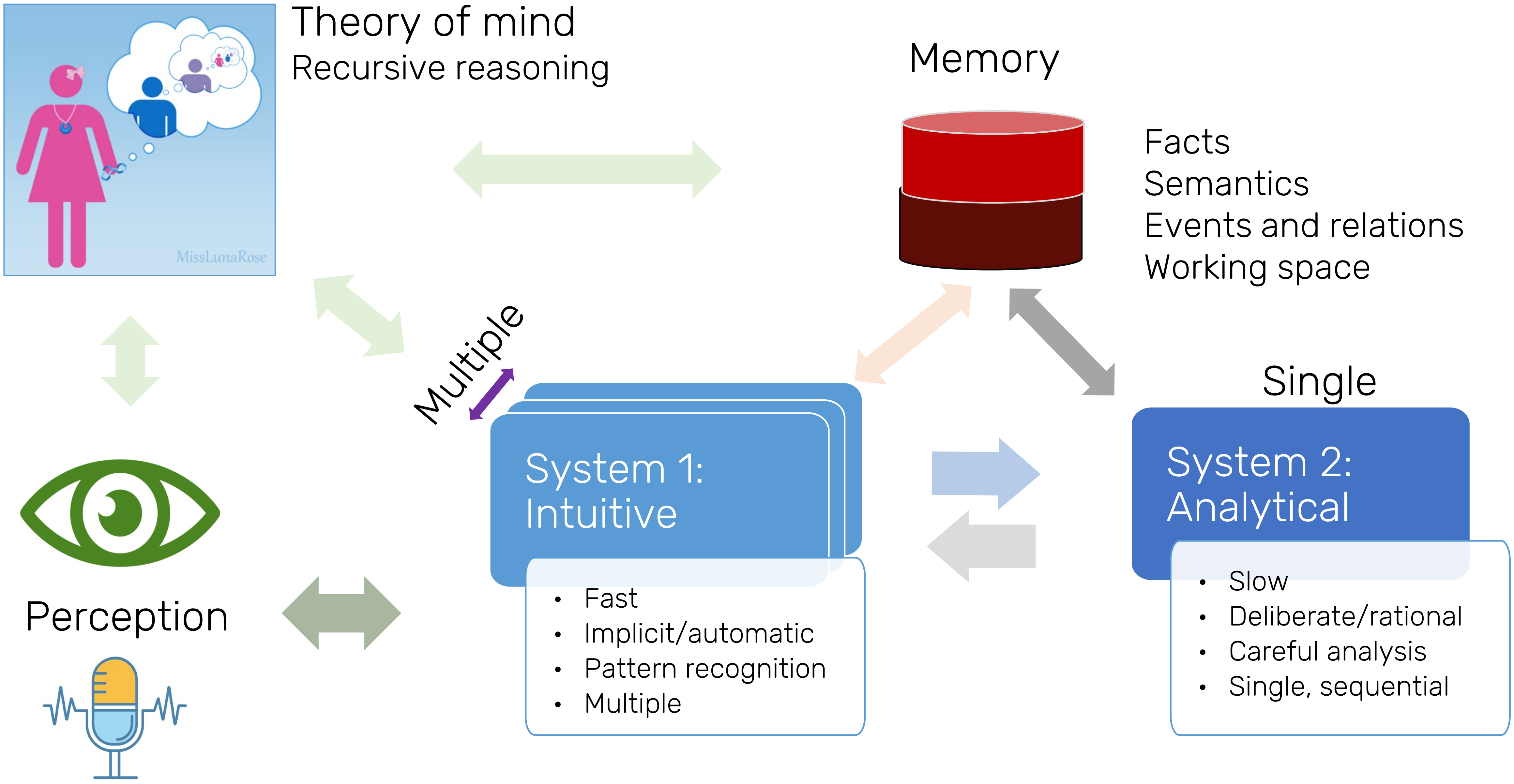
Agenda

- Reasoning with external memories
  - Memory of entities – memory-augmented neural networks
  - Memory of relations with tensors and graphs
  - Memory of programs & neural program construction.
- Learning to reason with less labels:
  - Data augmentation with analogical and counterfactual examples
  - Question generation
  - Self-supervised learning for question answering
  - Learning with external knowledge graphs
- **Recursive reasoning with neural theory of mind.**



Source: religious studies project

Core AI faculty: Theory of mind



Where would ToM fit in?

Image credit: VectorStock | Wikimedia

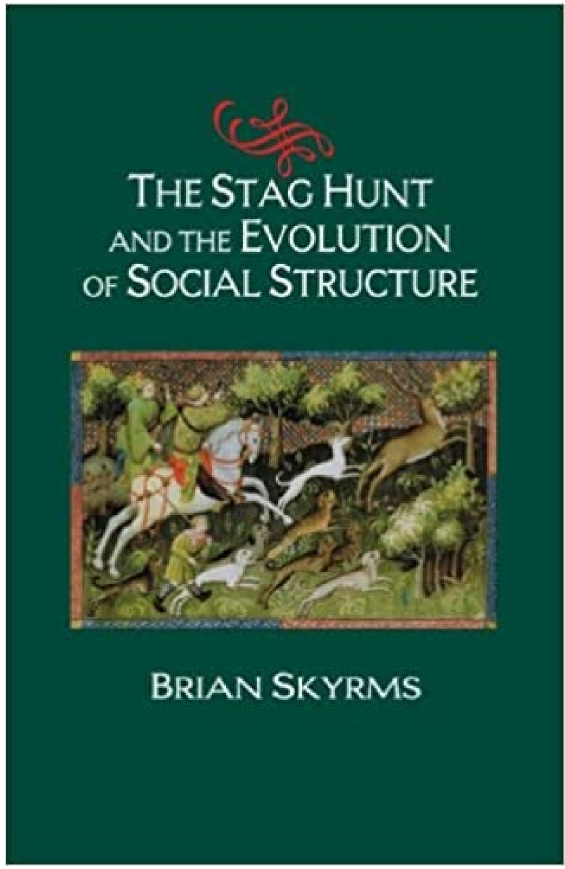
Contextualized recursive reasoning

- Thus far, QA tasks are straightforward and objective:
  - Questioner: I will ask about what I don't know.
  - Answerer: I will answer what I know.
- Real life can be tricky, more subjective:
  - Questioner: I will ask only questions I think they can answer.
  - Answerer 1: This is what I think they want from an answer.
  - Answerer 2: I will answer only what I think they think I can.

→ **We need Theory of Mind to function socially.**

Social dilemma: Stag Hunt games

- **Difficult decision**: individual outcomes (selfish) or group outcomes (cooperative).
  - Together hunt Stag (both are cooperative): Both have more meat.
  - Solely hunt Hare (both are selfish): Both have less meat.
  - One hunts Stag (cooperative), other hunts Hare (selfish): Only one hunts hare has meat.
- **Human evidence**: Self-interested but considerate of others (cultures vary).
- **Idea**: Belief-based guilt-aversion
  - One experiences loss if it lets other down.
  - Necessitates Theory of Mind: reasoning about other's mind.



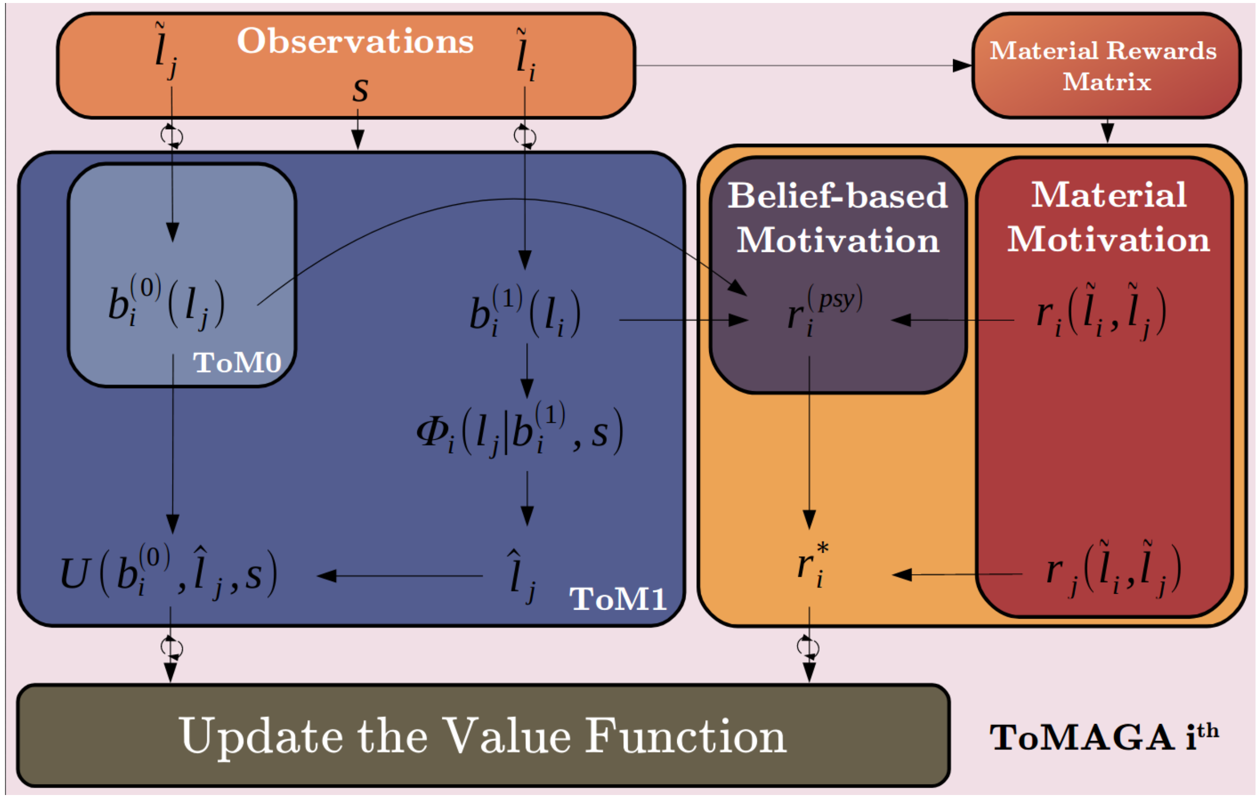
Theory of Mind Agent with Guilt Aversion (ToMAGA)

Update Theory of Mind

- Predict whether other's behaviour are cooperative or uncooperative
- Updated the zero-order belief (what other will do)
- Update the first-order belief (what other think about me)

Guilt Aversion

- Compute *the expected material reward* of other based on Theory of Mind
- Compute *the psychological rewards*, i.e. "feeling guilty"
- Reward shaping: subtract the expected loss of the other.

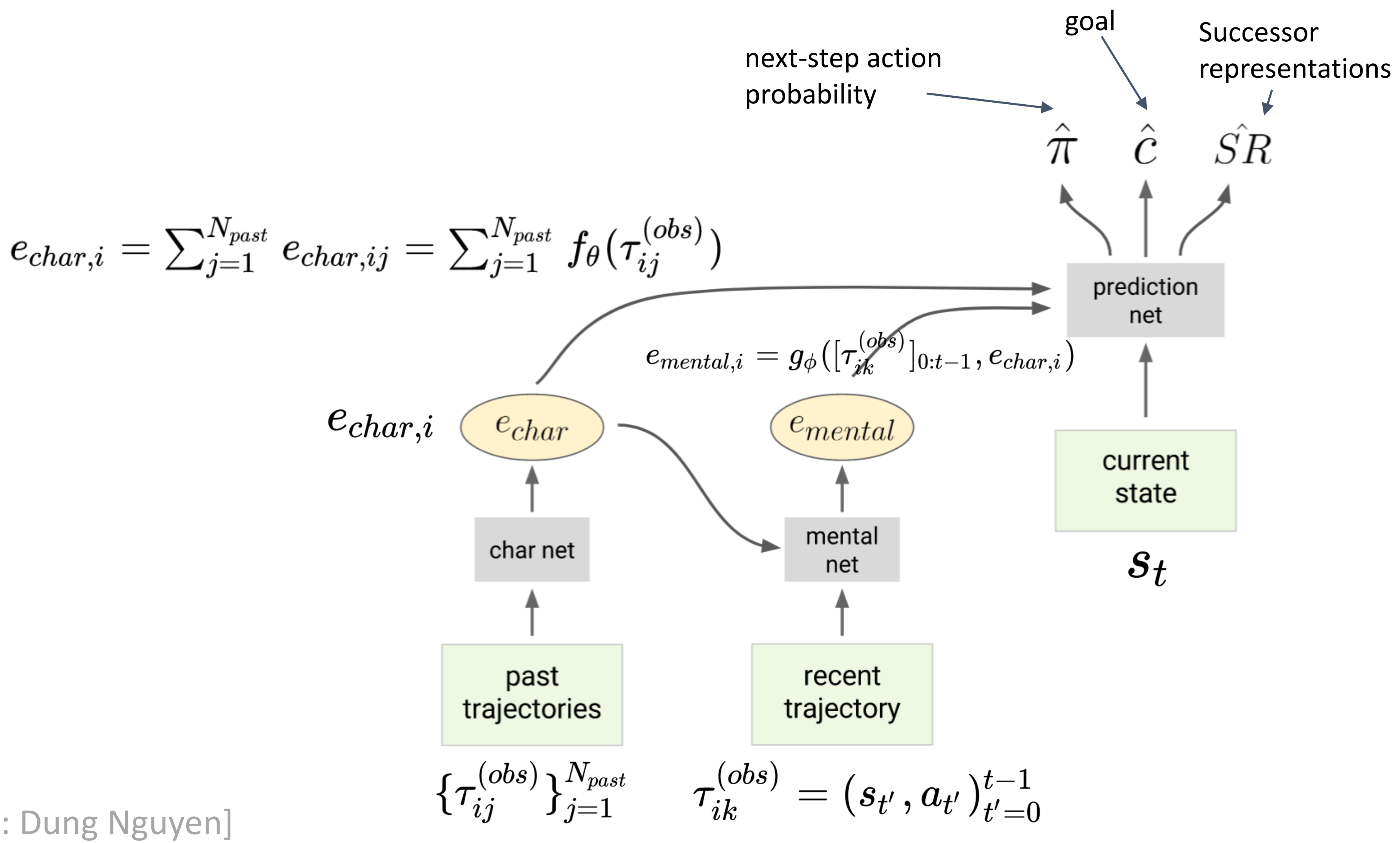


Nguyen, Dung, et al. "Theory of Mind with Guilt Aversion Facilitates Cooperative Reinforcement Learning." *Asian Conference on Machine Learning*. PMLR, 2020.

[Slide credit: Dung Nguyen]

Rabinowitz, Neil, et al. "Machine theory of mind." *International conference on machine learning*. PMLR, 2018.

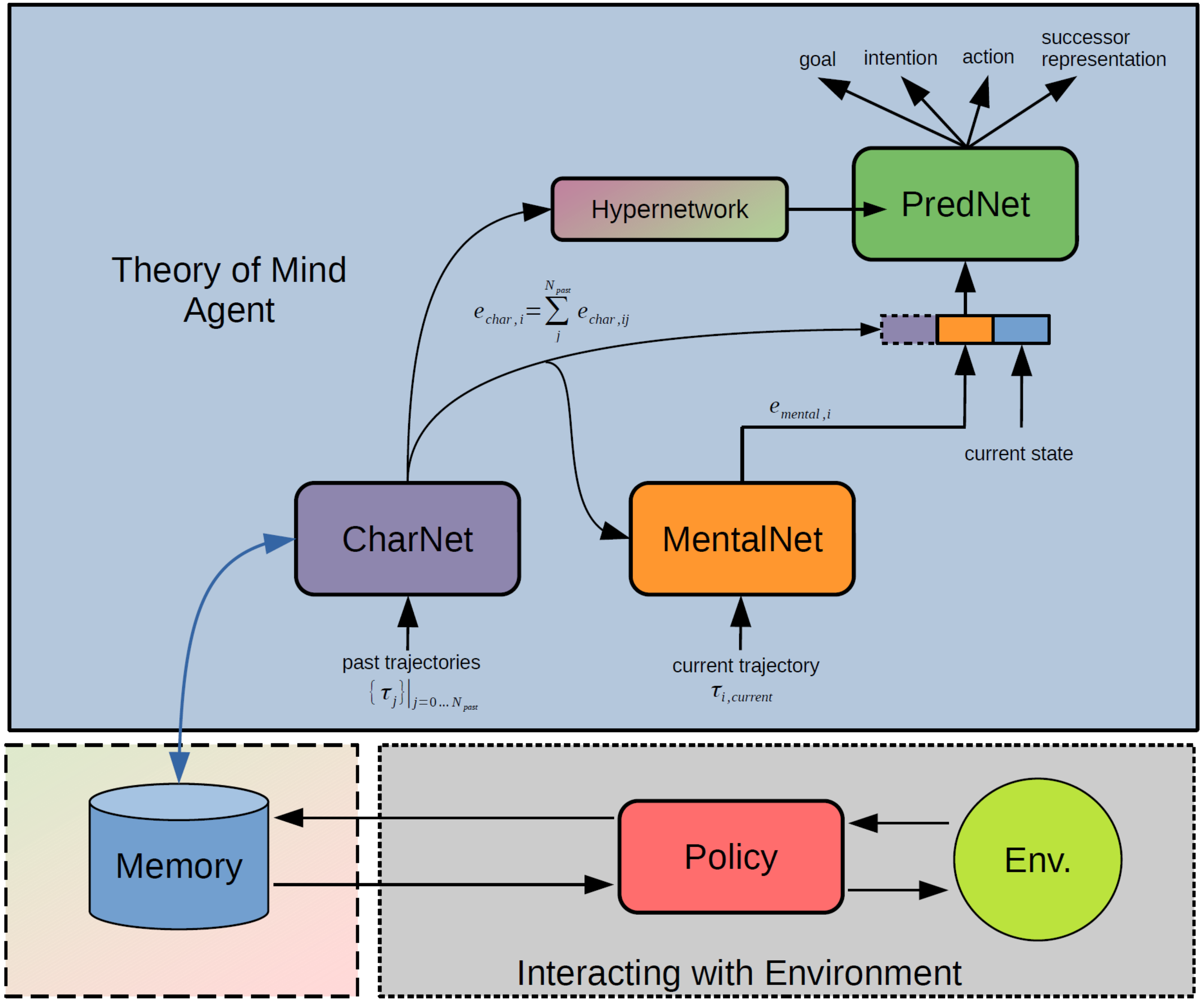
Machine Theory of Mind Architecture (inside the Observer)



[Slide credit: Dung Nguyen]

A ToM architecture

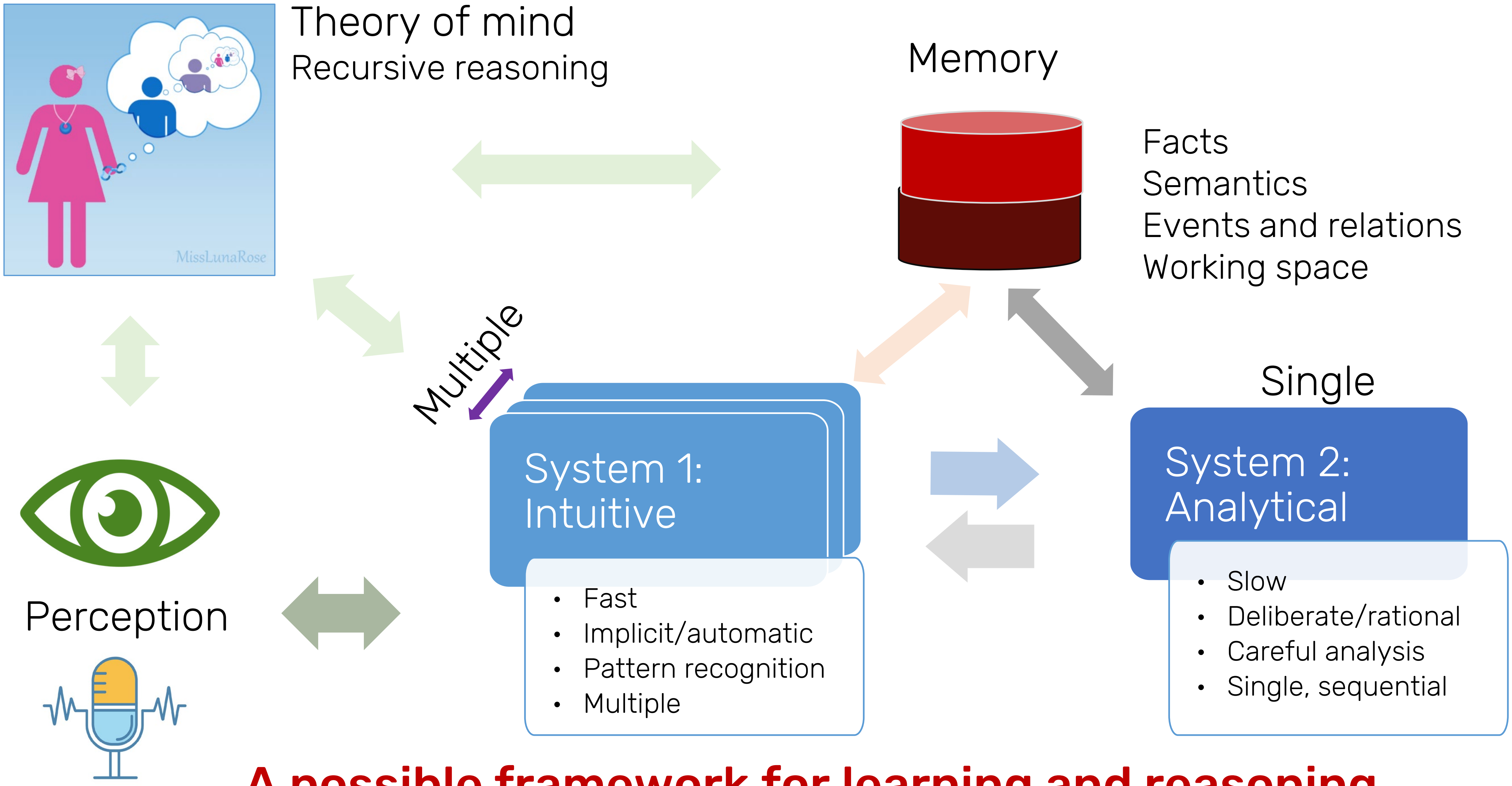
- Observer maintains memory of previous episodes of the agent.
- It theorizes the "traits" of the agent.
  - Implemented as Hyper Networks.
- Given the current episode, the observer tries to infer goal, intention, action, etc of the agent.
  - Implemented as memory retrieval through attention mechanisms.



Wrapping up

Wrapping up

- Reasoning as the next challenge for deep neural networks
- Part A: Learning-to-reason framework
    - Reasoning as a prediction skill that can be learnt from data
    - Dynamic neural networks are capable
    - Combinatorics reasoning
- Part B: Reasoning over unstructured and structured data
    - Reasoning over unstructured sets
    - Relational reasoning over structured data
- Part C: Memory | Data efficiency | Recursive reasoning
    - Memories of items, relations and programs
    - Learning with less labels
    - Theory of mind



A possible framework for learning and reasoning with deep neural networks

Image credit: VectorStock | Wikimedia

QA

https://bit.ly/37DYQn7