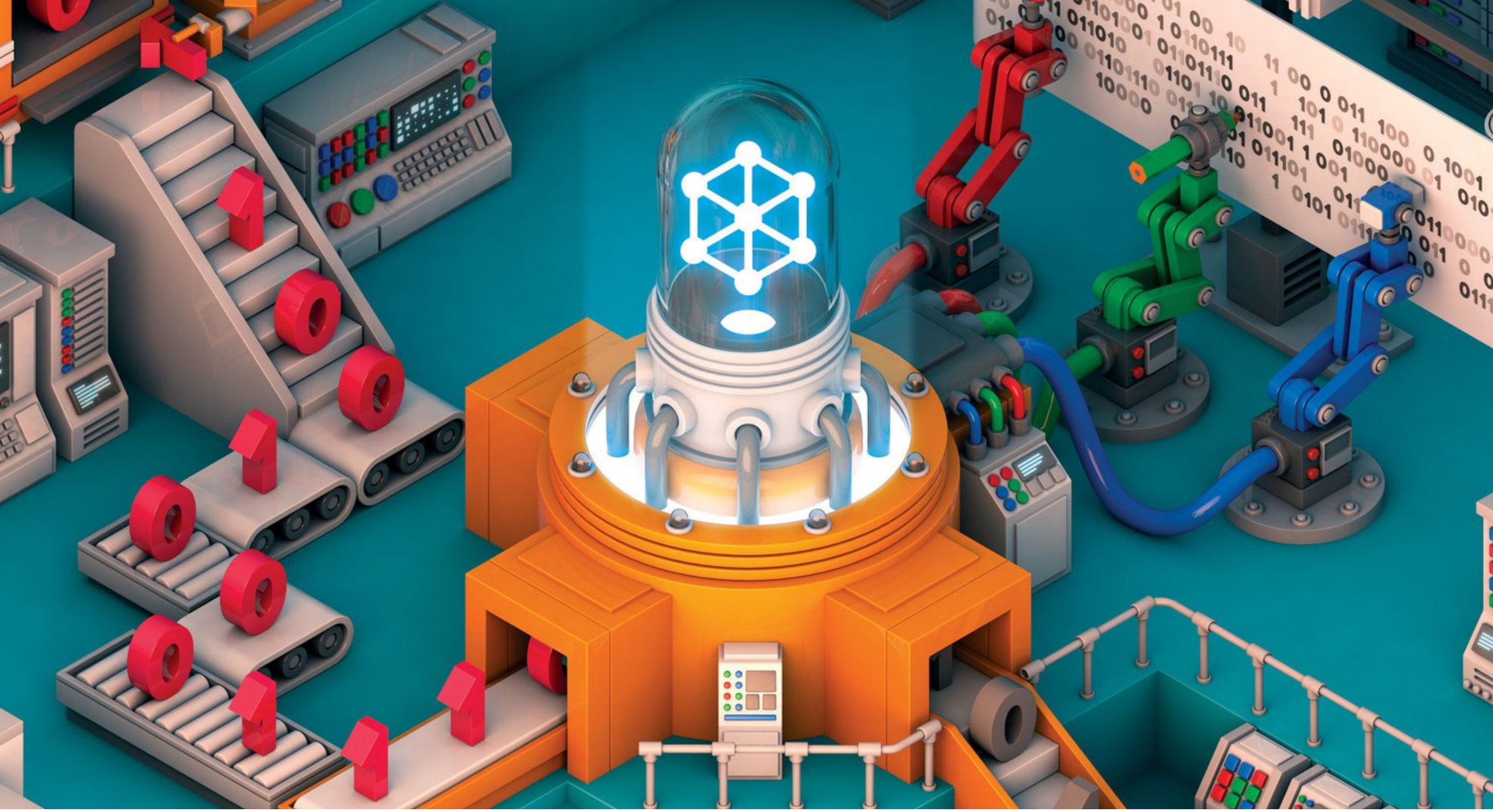


Memory augmented neural networks

Plan

- Neural Turing Machines
- Differentiable Neural Computer (DNC)
- Experiments with DNC

Neural Turing Machines

- Humans often use all sorts of memory
- Some tasks require storing information for long periods of time
- No way to scale RNN memory

Neural Turing Machines

- Controller

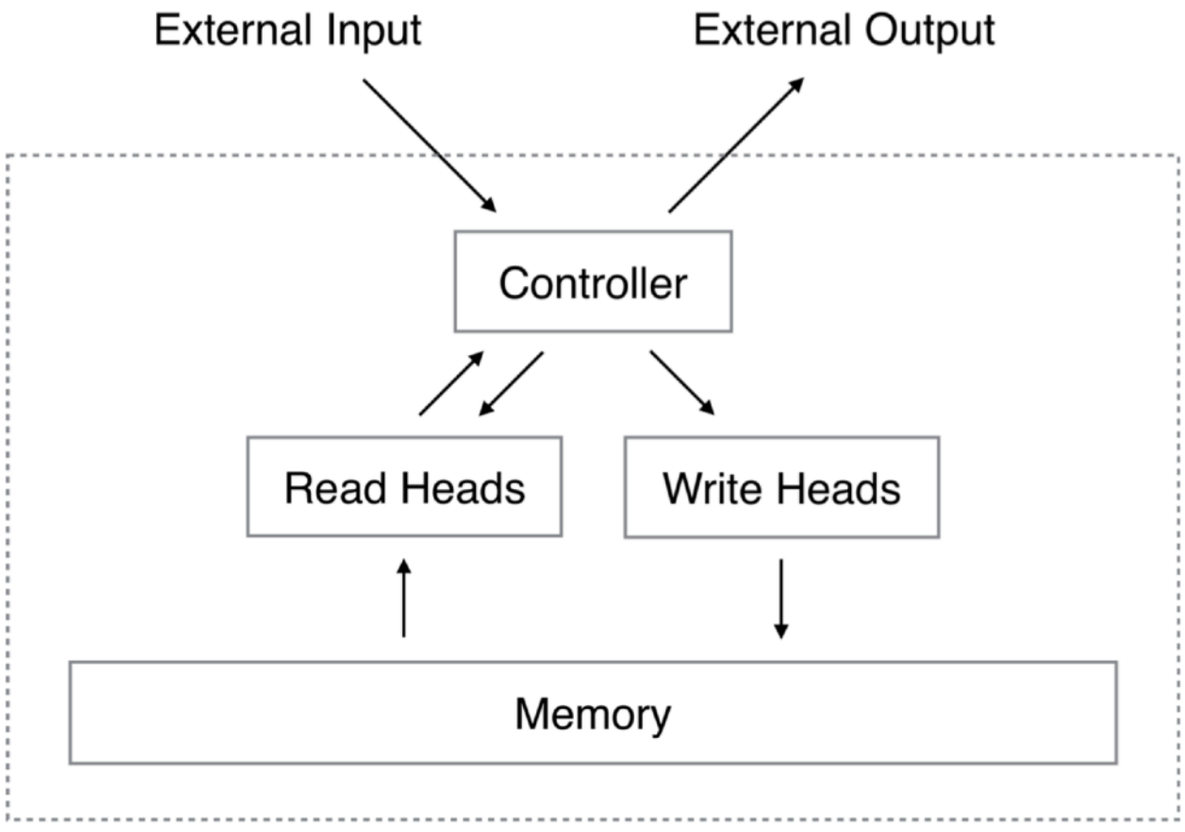
  Reads external inputs and controls read and write heads

- Read Heads

  Uses attention mechanisms to read from external memory

- Write Heads

  Uses attention mechanisms to write to external memory



https://arxiv.org/abs/1410.5401

Neural Turing Machines

Reading

$\mathbf{M}_t$ — $N \times M$ memory matrix at time step t

$\mathbf{w}_t$ — read weights, with elements $w_t(i)$

$$\sum_i w_t(i) = 1, \quad 0 \leqslant w_t(i) \leqslant 1, \forall i$$

Given memory and weights read vector is produced:

$$r_t \leftarrow \sum_i w_t(i)\ \mathbf{M}_t(i) = \mathbf{M}_t^T w_t$$

Neural Turing Machines

Writing

$\mathbf{M}_t$ — $N \times M$ memory matrix at time step t

$\mathbf{w}_t$ — read weights, with elements $w_t(i)$

$$\sum_i w_t(i) = 1, \quad 0 \leqslant w_t(i) \leqslant 1, \forall i$$

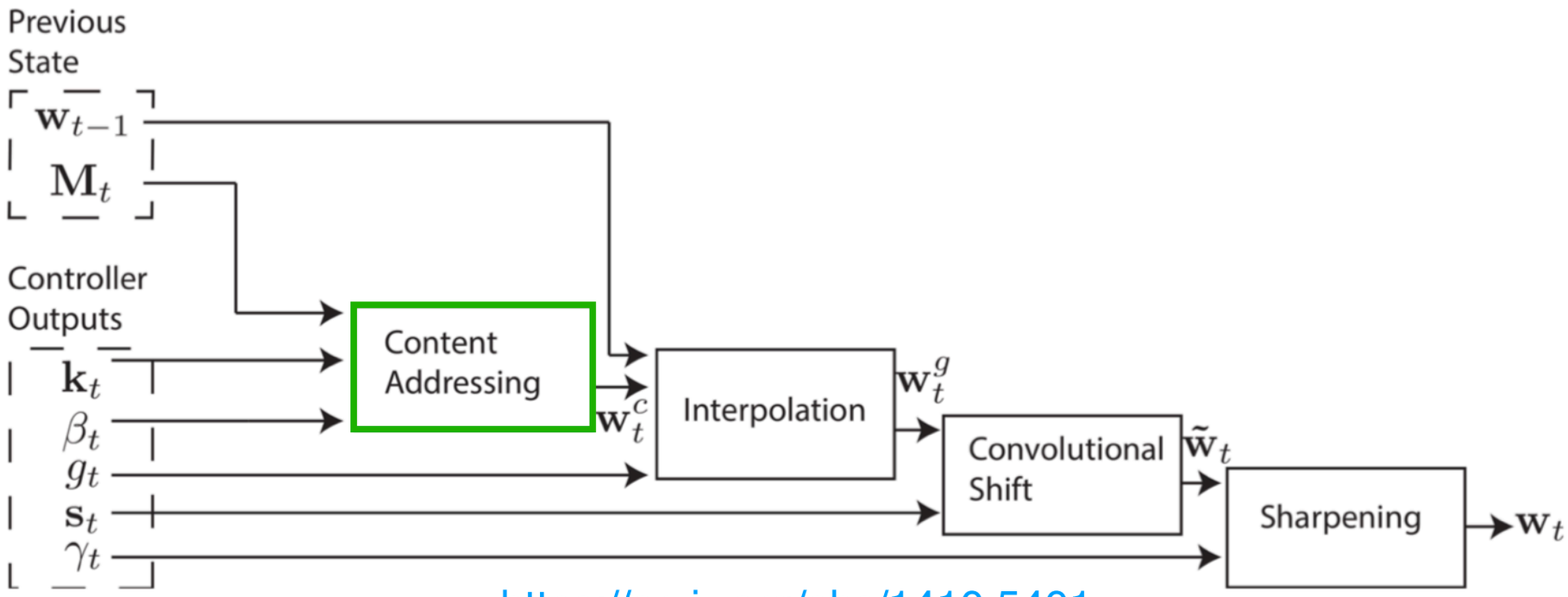
$\mathbf{e}_t$ — erase vector, with elements $e_t(i) \in (0,1)$

$\mathbf{a}_t$ — add vector

Given memory, weights, erase and add vectors memory is updated:

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

Neural Turing Machines

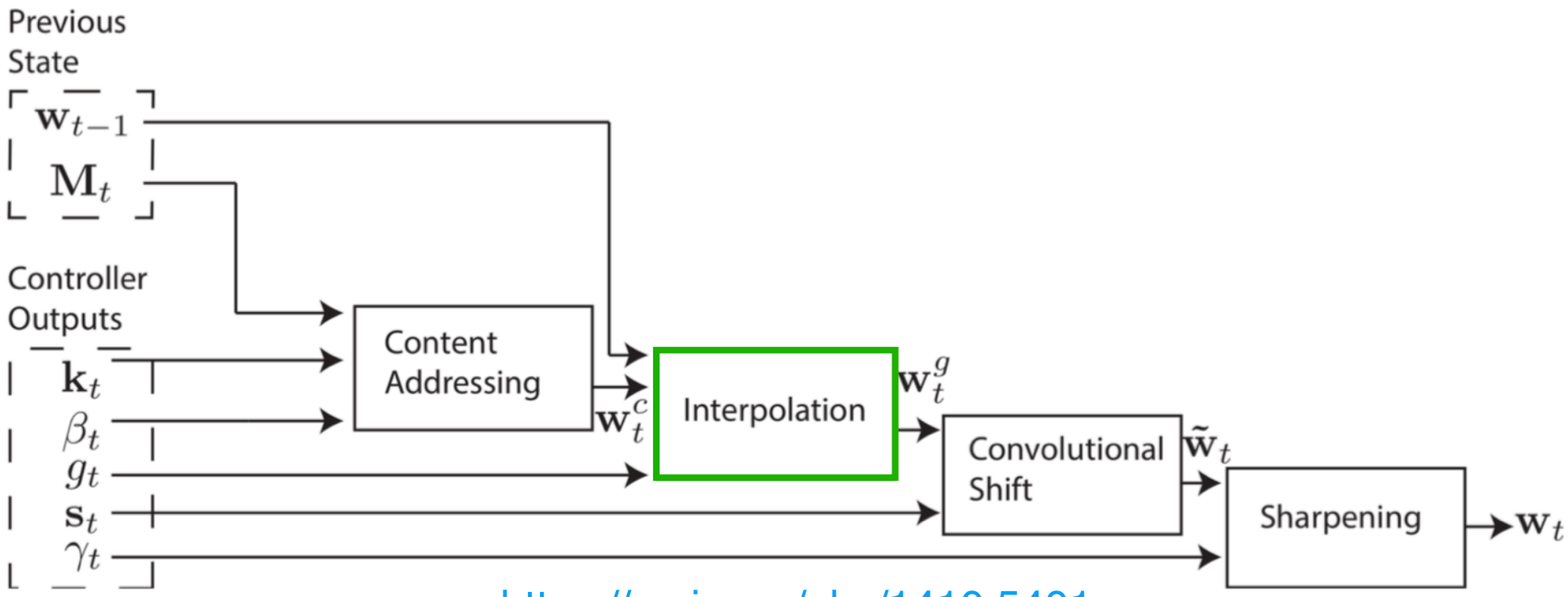


https://arxiv.org/abs/1410.5401

Content Addressing:

$$\mathbf{w}_t^c(i) \leftarrow \frac{\exp\left(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(i)]\right)}{\sum_j \exp\left(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(j)]\right)} \qquad K[u, v] = \frac{\langle u, v \rangle}{\|u\| \cdot \|v\|}$$

Neural Turing Machines

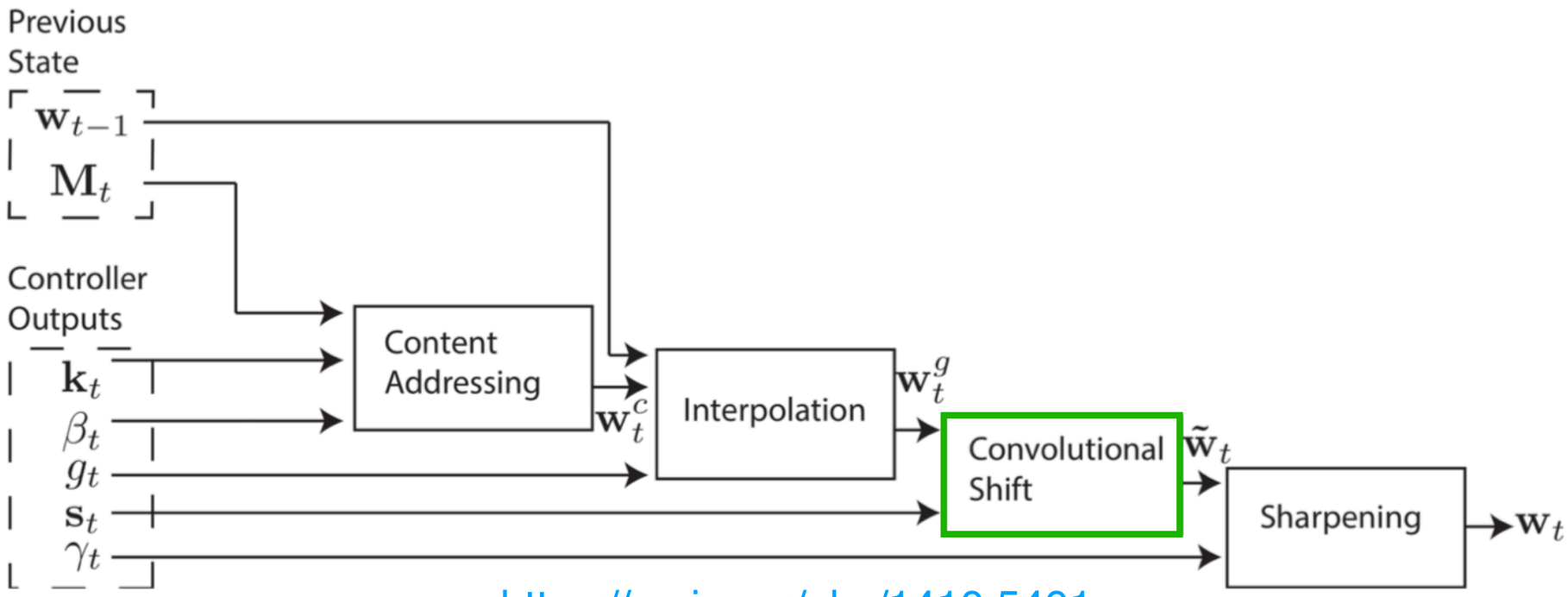


https://arxiv.org/abs/1410.5401

Interpolation:

$$\mathbf{w}_t^g = g_t \mathbf{w}_t^c + (1 - g_t)\mathbf{w}_{t-1}$$

Neural Turing Machines

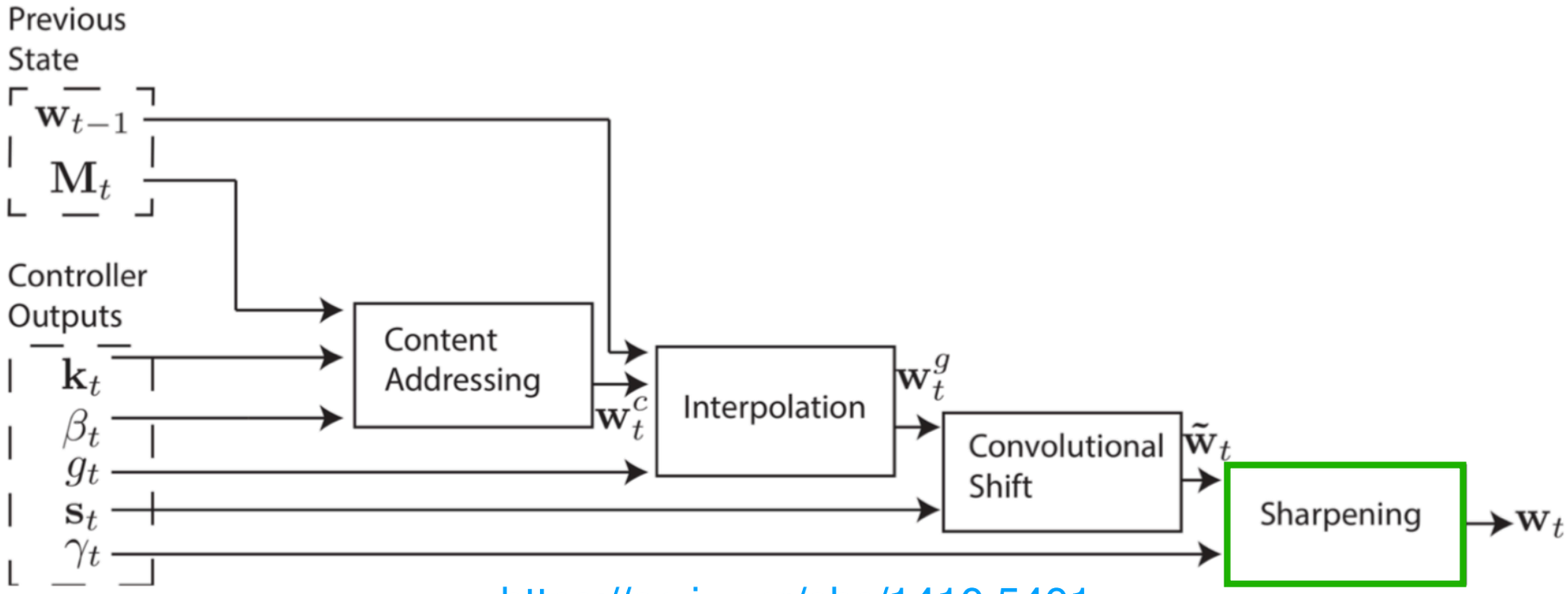


https://arxiv.org/abs/1410.5401

Convolutional Shift (location based addressing):

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i-j)$$

Neural Turing Machines



https://arxiv.org/abs/1410.5401

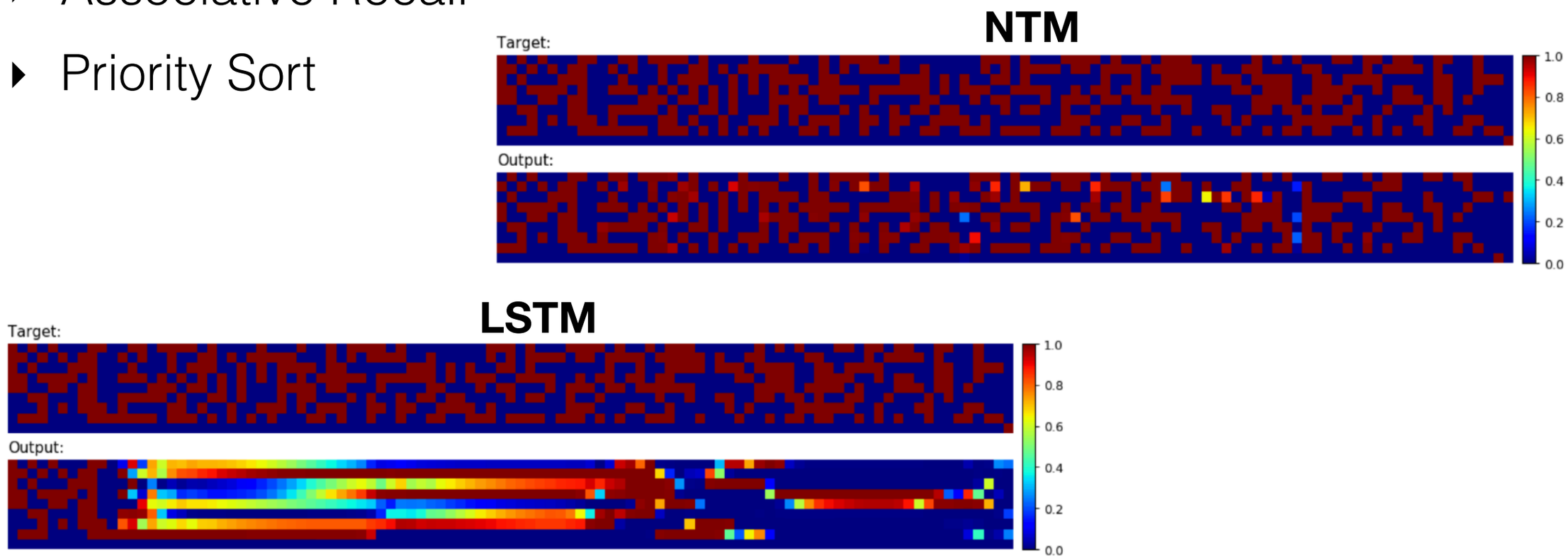
Sharpening:

$$w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}}$$

Neural Turing Machines

Authors achieved good (compared to LSTM) generalisation on algorithmic tasks:

- Copy
- Repeat Copy
- Associative Recall
- Priority Sort

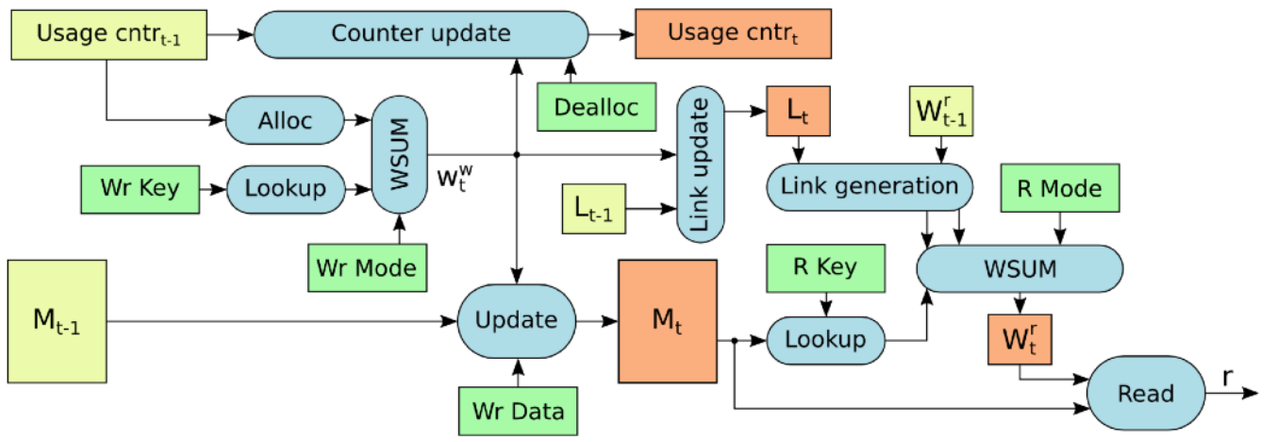


Differentiable Neural Computer

NTM had problems and limitations

- It stores information in continuous blocks:
  - There is no mechanism to ensure that blocks don't overlap
  - No relationship between blocks is saved
- No way of freeing memory cells*

DNC addressed them by changing attention schemes:



One step of DNC's memory access module

Differentiable Neural Computer

As in NTM controller outputs controls vector:

$$\boldsymbol{\xi}_t = \left[\boldsymbol{k}_t^{\mathrm{r},1}; \ldots; \boldsymbol{k}_t^{\mathrm{r},R}; \hat{\beta}_t^{\mathrm{r},1}; \ldots; \hat{\beta}_t^{\mathrm{r},R}; \boldsymbol{k}_t^{\mathrm{w}}; \hat{\beta}_t^{\mathrm{w}}; \hat{\boldsymbol{e}}_t; \boldsymbol{v}_t; \hat{f}_t^{1}; \ldots; \hat{f}_t^{R}; \hat{g}_t^{\mathrm{a}}; \hat{g}_t^{\mathrm{w}}; \hat{\boldsymbol{\pi}}_t^{1}; \ldots; \hat{\boldsymbol{\pi}}_t^{R} \right]$$

Which is used in 3 different attention schemes:

- Content based (as in NTM)
- Temporal links
  Stores order of writes. Used for reading in sequence
- Memory allocation
  Stores ‘free list’ of memory locations available for writing

Differentiable Neural Computer

Focusing on content

Identical to NTM

Used in both read and write weightings

$$\boldsymbol{w}_t^c(i) \leftarrow \frac{\exp\left(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(i)]\right)}{\sum_j \exp\left(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(j)]\right)} \qquad K[u, v] = \frac{\langle u, v \rangle}{\|u\| \cdot \|v\|}$$

Differentiable Neural Computer

Memory allocation

Used in write weightings to free and allocate memory

- Updates memory usage $\boldsymbol{u}_t \in [0,1]^N, \quad \boldsymbol{u}_0 = \mathbf{0}$
- Uses free gates from controller $f_t^i \in [0,1]$
- Memory retention vector:

$$\boldsymbol{\psi}_t = \prod_{i=1}^{R} \left(\mathbf{1} - f_t^i \mathbf{w}_{t-1}^{\mathrm{r},i} \right)$$

- Usage is updated:

$$\boldsymbol{u}_t = (\boldsymbol{u}_{t-1} + \boldsymbol{w}_{t-1}^{\mathrm{w}} - \boldsymbol{u}_{t-1} \circ \boldsymbol{w}_{t-1}^{\mathrm{w}}) \circ \boldsymbol{\psi}_t$$

Differentiable Neural Computer

Memory allocation

Used in write weightings to free and allocate memory

- Allocation weighting:

$$\boldsymbol{a}_t[\phi_t[j]] = (1 - \boldsymbol{u}_t[\phi_t[j]]) \prod_{i=1}^{j-1} \boldsymbol{u}_t[\phi_t[i]]$$

$\phi_t[j]$ argsort in descending order

- Write weightings:

$$\boldsymbol{w}_t^w = g_t^w \left[\hat{g}_t^a \boldsymbol{a}_t + (1 - \hat{g}_t^a)\boldsymbol{w}_t^{c,w}\right]$$

Gates from controls vector

Differentiable Neural Computer

Temporal linkage

Used in read weightings to take the order of writes into account

- Temporal link matrix $L_t \in [0,1]^{N \times N}$
- Each element $L_t[i,j]$
  represents degree to which location i
  was the location written to after location j

$$\sum_i L_t[i,\cdot\,] \leqslant 1, \quad \sum_j L_t[\,\cdot\,,j] \leqslant 1$$

- Precedence weighting:

$$\boldsymbol{p}_0 = \mathbf{0}$$

$$\boldsymbol{p}_t = \left(1 - \sum_i \boldsymbol{w}_t^{\mathrm{w}}[i]\right)\boldsymbol{p}_{t-1} + \boldsymbol{w}_t^{\mathrm{w}}$$

Differentiable Neural Computer

Temporal linkage

Used in read weightings to take the order of writes into account

- Link matrix update rule:

$$L_0[i,j]=0 \quad \forall\, i,j$$

$$L_t[i,i]=0 \quad \forall\, i$$

$$L_t[i,j] = (1 - \boldsymbol{w}_t^{\mathrm{w}}[i] - \boldsymbol{w}_t^{\mathrm{w}}[j])L_{t-1}[i,j] + \boldsymbol{w}_t^{\mathrm{w}}[i]\boldsymbol{p}_{t-1}[j]$$

- Forward and backward weightings:

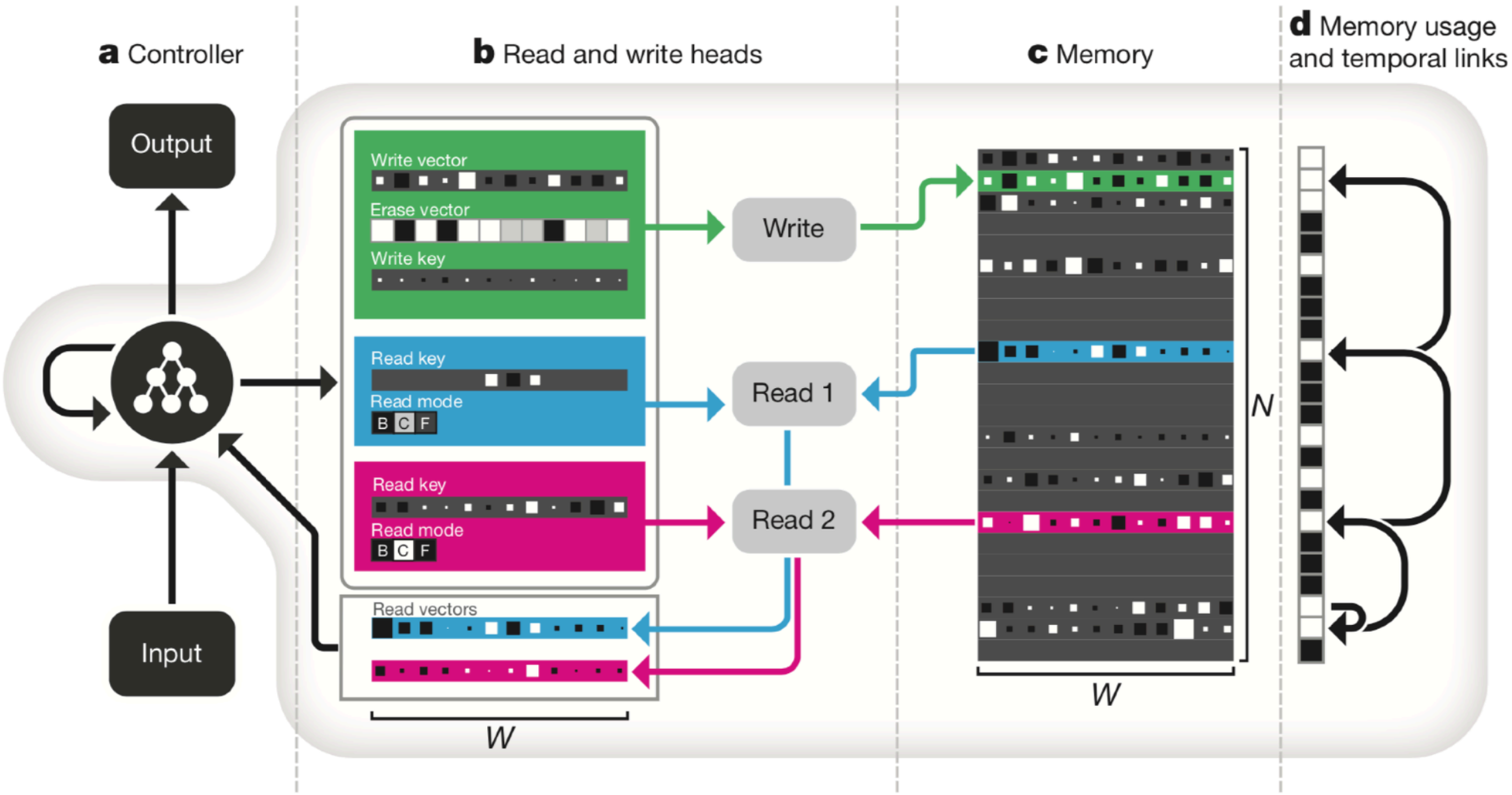
$$\boldsymbol{f}_t^i = L_t \boldsymbol{w}_{t-1}^{\mathrm{r},i}$$

$$\boldsymbol{b}_t^i = L_t^\top \boldsymbol{w}_{t-1}^{\mathrm{r},i}$$

$\sum_{i=1}^{3} \boldsymbol{\pi}[i] = 1$ from controls vector

- Read weighting: $\boldsymbol{w}_t^r = \boldsymbol{\pi}_t[1]\boldsymbol{b}_t + \boldsymbol{\pi}_t[2]\boldsymbol{w}_t^{c,r} + \boldsymbol{\pi}_t[3]\boldsymbol{f}_t$

Differentiable Neural Computer



https://www.nature.com/articles/nature20101

Sparse reads and writes

https://arxiv.org/pdf/1610.09027

Memory is not scalable because of content based attention.

Proposed solution:

Use only k nearest memory cells.
Do it fast with approximate nearest neighbours

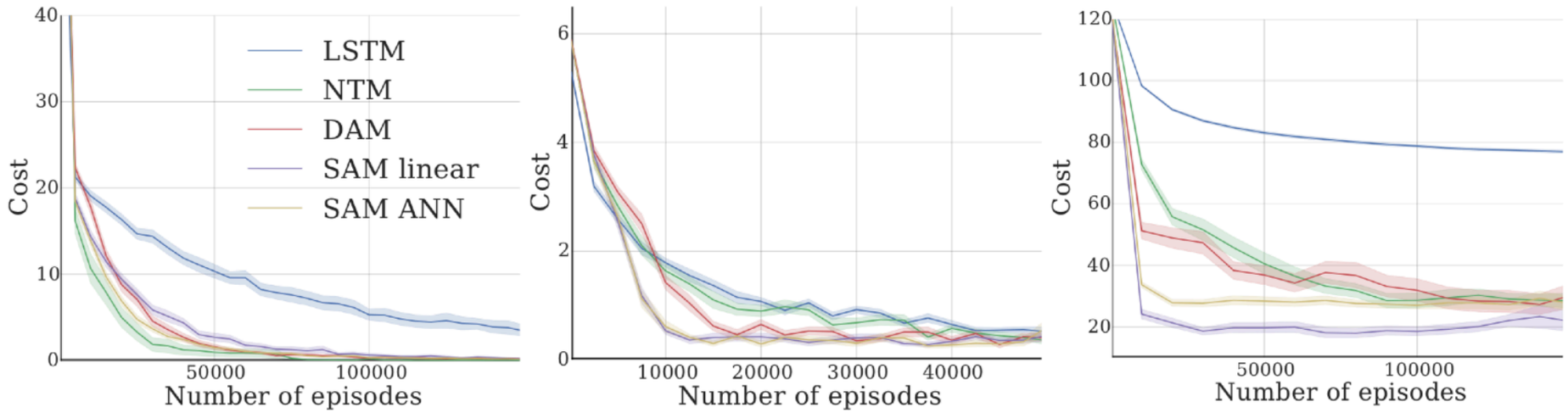


Figure 2: Training curves for sparse (SAM) and dense (DAM, NTM) models. SAM trains comparably for the Copy task, and reaches asymptotic error significantly faster for Associative Recall and Priority Sort.Light colors indicate one standard deviation over 30 random seeds.

Sparse reads and writes

https://arxiv.org/pdf/1610.09027

Memory is not scalable because of content based attention.

Proposed solution:

Use only k nearest memory cells.
Do it fast with approximate nearest neighbours

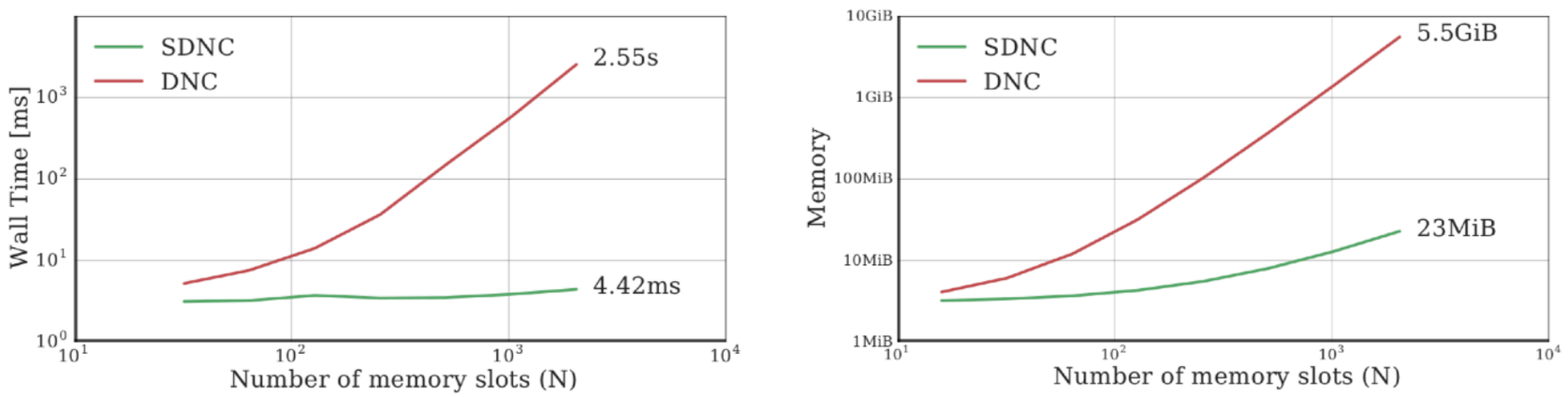


Figure 7: Performance benchmarks between the DNC and SDNC for small to medium memory sizes. Here the SDNC uses a linear KNN. **(a)** Wall-clock time of a single forward and backward pass. **(b)** Total memory usage (including initialization) when trained over sequence of 10 time steps.

DNC Experiments

bAbI

Dataset of 20 types of synthetically generated questions:

```
1 Wolves are afraid of mice.
2 Sheep are afraid of mice.
3 Winona is a sheep.
4 Mice are afraid of cats.
5 Cats are afraid of wolves.
6 Jessica is a mouse.
7 Emily is a cat.
8 Gertrude is a wolf.
9 What is emily afraid of?      wolf    7 5
10 What is winona afraid of?    mouse   3 2
11 What is gertrude afraid of?  mouse   8 1
12 What is jessica afraid of?   cat     6 4
```

```
1 Mary got the milk there.
2 John moved to the bedroom.
3 Sandra went back to the kitchen.
4 Mary travelled to the hallway.
5 Where is the milk?    hallway 1 4
```

```
1 Sumit is tired.
2 Where will sumit go?  bedroom 1
3 Sumit went back to the bedroom.
4 Why did sumit go to the bedroom?      tired   1
5 Sumit grabbed the pajamas there.
6 Why did sumit get the pajamas?        tired   1
7 Yann is bored.
8 Where will yann go?   garden  7
9 Jason is thirsty.
10 Where will jason go? kitchen 9
11 Yann travelled to the garden.
12 Why did yann go to the garden?       bored   7
13 Yann got the football there.
14 Why did yann get the football?       bored   7
15 Jason went back to the kitchen.
16 Why did jason go to the kitchen?     thirsty 9
17 Antoine is thirsty.
18 Where will antoine go?       kitchen 17
```

DNC Experiments

bAbI

Results:

	bAbI Best Results							bAbI Mean Results			
Task	LSTM (Joint)	NTM (Joint)	DNC1 (Joint)	DNC2 (Joint)	MemN2N (Joint) [21]	MemN2N (Single) [21]	DMN (Single) [20]	LSTM	NTM	DNC1	DNC2
1: 1 supporting fact	24.5	31.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	28.4 ± 1.5	40.6 ± 6.7	**9.0 ± 12.6**	16.2 ± 13.7
2: 2 supporting facts	53.2	54.5	1.3	0.4	1.0	**0.3**	1.8	56.0 ± 1.5	56.3 ± 1.5	**39.2 ± 20.5**	47.5 ± 17.3
3: 3 supporting facts	48.3	43.9	2.4	**1.8**	6.8	2.1	4.8	51.3 ± 1.4	47.8 ± 1.7	**39.6 ± 16.4**	44.3 ± 14.5
4: 2 argument rels.	0.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.8 ± 0.5	0.9 ± 0.7	**0.4 ± 0.7**	**0.4 ± 0.3**
5: 3 argument rels.	3.5	0.8	**0.5**	0.8	6.1	0.8	0.7	3.2 ± 0.5	1.9 ± 0.8	**1.5 ± 1.0**	1.9 ± 0.6
6: yes/no questions	11.5	17.1	**0.0**	**0.0**	0.1	0.1	**0.0**	15.2 ± 1.5	18.4 ± 1.6	**6.9 ± 7.5**	11.1 ± 7.1
7: counting	15.0	17.8	**0.2**	0.6	6.6	2.0	3.1	16.4 ± 1.4	19.9 ± 2.5	**9.8 ± 7.0**	15.4 ± 7.1
8: lists/sets	16.5	13.8	**0.1**	0.3	2.7	0.9	3.5	17.7 ± 1.2	18.5 ± 4.9	**5.5 ± 5.9**	10.0 ± 6.6
9: simple negation	10.5	16.4	**0.0**	0.2	**0.0**	0.3	**0.0**	15.4 ± 1.5	17.9 ± 2.0	**7.7 ± 8.3**	11.7 ± 7.4
10: indefinite knowl.	22.9	16.6	0.2	0.2	0.5	**0.0**	**0.0**	28.7 ± 1.7	25.7 ± 7.3	**9.6 ± 11.4**	14.7 ± 10.8
11: basic coreference	6.1	15.2	**0.0**	**0.0**	**0.0**	0.1	0.1	12.2 ± 3.5	24.4 ± 7.0	**3.3 ± 5.7**	7.2 ± 8.1
12: conjunction	3.8	8.9	0.1	**0.0**	0.1	**0.0**	**0.0**	5.4 ± 0.6	21.9 ± 6.6	**5.0 ± 6.3**	10.1 ± 8.1
13: compound coref.	0.5	7.4	**0.0**	0.1	**0.0**	**0.0**	0.2	7.2 ± 2.3	8.2 ± 0.8	**3.1 ± 3.6**	5.5 ± 3.4
14: time reasoning	55.3	24.2	0.3	0.4	**0.0**	0.1	**0.0**	55.9 ± 1.2	44.9 ± 13.0	**11.0 ± 7.5**	15.0 ± 7.4
15: basic deduction	44.7	47.0	**0.0**	**0.0**	0.2	**0.0**	**0.0**	47.0 ± 1.7	46.5 ± 1.6	**27.2 ± 20.1**	40.2 ± 11.1
16: basic induction	52.6	53.6	52.4	55.1	**0.2**	51.8	0.6	**53.3 ± 1.3**	53.8 ± 1.4	53.6 ± 1.9	54.7 ± 1.3
17: positional reas.	39.2	25.5	24.1	**12.0**	41.8	18.6	40.4	34.8 ± 4.1	**29.9 ± 5.2**	32.4 ± 8.0	30.9 ± 10.1
18: size reasoning	4.8	2.2	4.0	**0.8**	8.0	5.3	4.7	5.0 ± 1.4	4.5 ± 1.3	**4.2 ± 1.8**	4.3 ± 2.1
19: path finding	89.5	4.3	**0.1**	3.9	75.7	2.3	65.5	90.9 ± 1.1	86.5 ± 19.4	**64.6 ± 37.4**	75.8 ± 30.4
20: agent motiv.	1.3	1.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	1.3 ± 0.4	1.4 ± 0.6	**0.0 ± 0.1**	**0.0 ± 0.0**
Mean Err. (%)	25.2	20.1	4.3	**3.8**	7.5	4.2	6.4	27.3 ± 0.8	28.5 ± 2.9	**16.7 ± 7.6**	20.8 ± 7.1
Failed (err. > 5%)	15	16	**2**	**2**	6	3	**2**	17.1 ± 1.0	17.3 ± 0.7	**11.2 ± 5.4**	14.0 ± 5.0

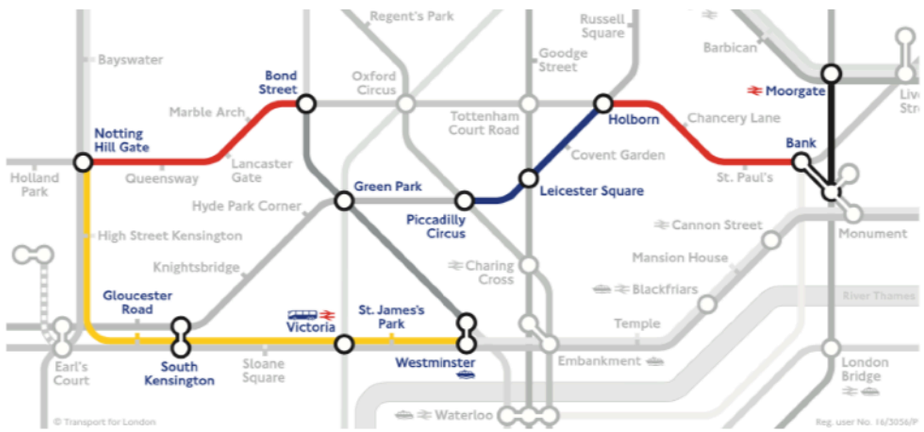
https://www.nature.com/articles/nature20101

DNC Experiments

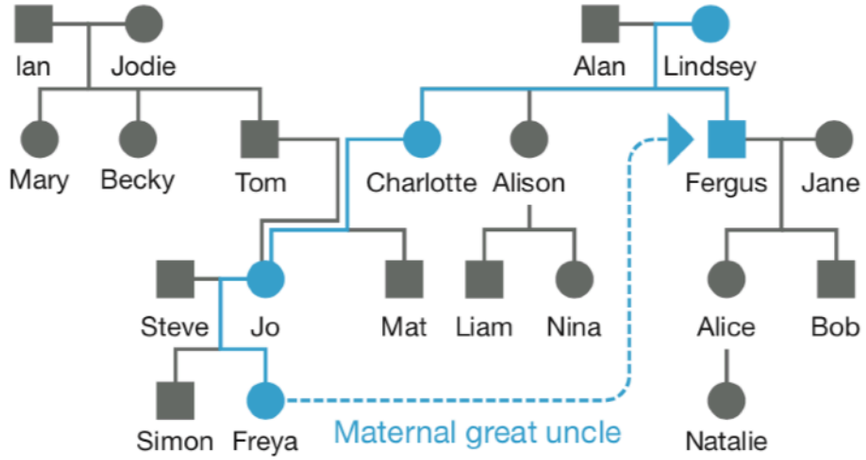
Graphs

a Random graph

b London Underground



c Family tree



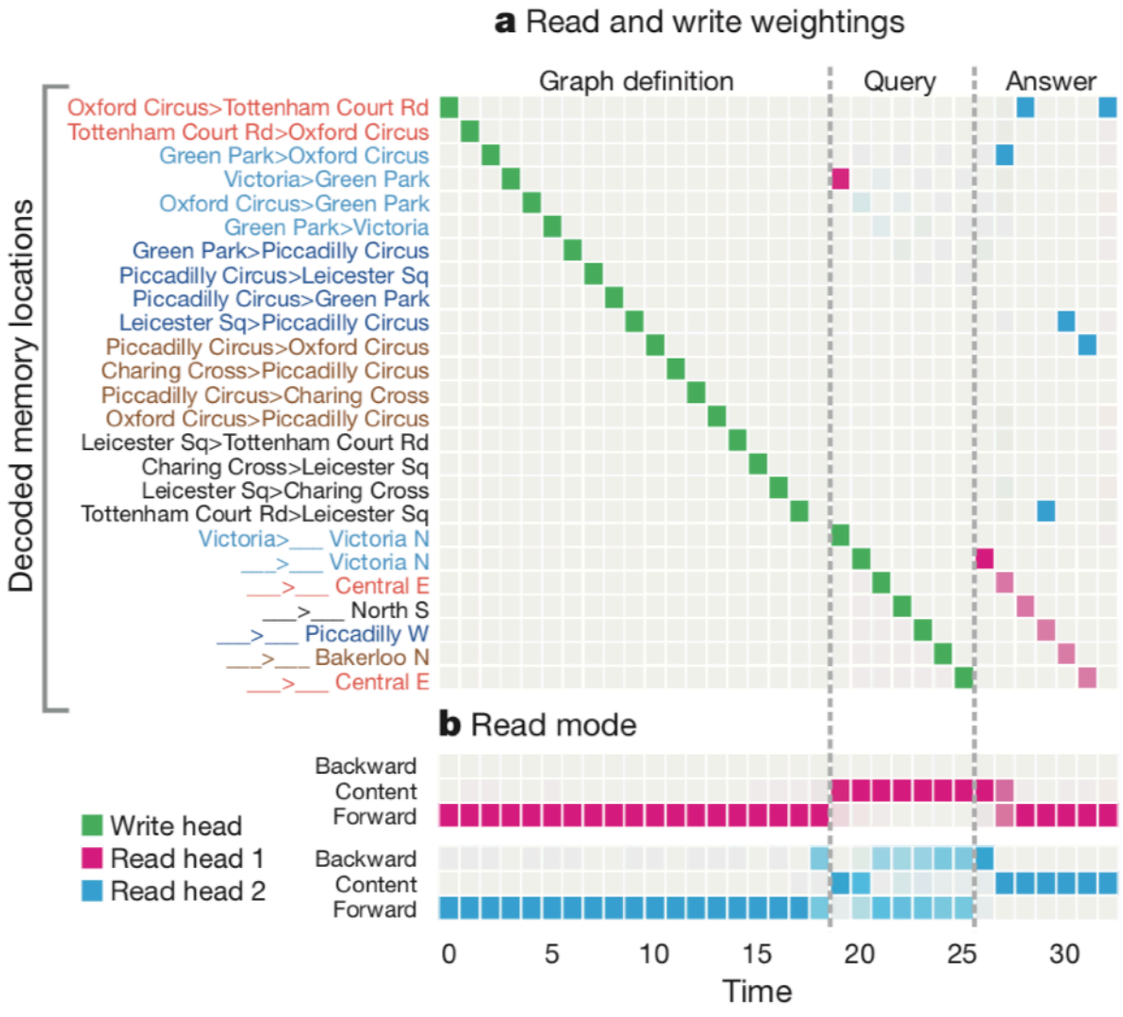
	Traversal	Shortest-path
Underground input: (OxfordCircus, TottenhamCtRd, Central) (TottenhamCtRd, OxfordCircus, Central) (BakerSt, Marylebone, Circle) (BakerSt, Marylebone, Bakerloo) (BakerSt, OxfordCircus, Bakerloo) ⋮ (LeicesterSq, CharingCross, Northern) (TottenhamCtRd, LeicesterSq, Northern) (OxfordCircus, PiccadillyCircus, Bakerloo) (OxfordCircus, NottingHillGate, Central) (OxfordCircus, Euston, Victoria) 84 edges in total	Traversal question: (BondSt, _, Central), (_, _, Circle), (_, _, Circle), (_, _, Circle), (_, _, Circle), (_, _, Jubilee), (_, _, Jubilee), Answer: (BondSt, NottingHillGate, Central) (NottingHillGate, GloucesterRd, Circle) ⋮ (Westminster, GreenPark, Jubilee) (GreenPark, BondSt, Jubilee)	Shortest-path question: (Moorgate, PiccadillyCircus, _) Answer: (Moorgate, Bank, Northern) (Bank, Holborn, Central) (Holborn, LeicesterSq, Piccadilly) (LeicesterSq, PiccadillyCircus, Piccadilly)

Family tree input: (Charlotte, Alan, Father) (Simon, Steve, Father) (Steve , Simon, Son1) (Nina, Alison, Mother) (Lindsey, Fergus, Son1) ⋮ (Bob, Jane, Mother) (Natalie, Alice, Mother) (Mary, Ian, Father) (Jane, Alice, Daughter1) (Mat, Charlotte, Mother) 54 edges in total	Inference question: (Freya, _, MaternalGreatUncle) Answer: (Freya, Fergus, MaternalGreatUncle)

https://www.nature.com/articles/nature20101

DNC Experiments

Graphs



https://www.nature.com/articles/nature20101

Summary

- We can train a neural network with external memory end-to-end with gradient descent
- These models outperform LSTM's on tasks, which require memory
- MANNs are not *yet* applied in practice

Resources

https://www.nature.com/articles/nature20101
https://arxiv.org/pdf/1410.5401.pdf
https://arxiv.org/pdf/1610.09027.pdf