



CRIPAC

智能感知与计算研究中心
Center for Research on Intelligent Perception and Computing



中国科学院自动化研究所
Institute of Automation
Chinese Academy of Sciences

2023

“Deep Learning Lecture”

Lecture 10 : Deep Cognitive Networks

Yan Huang

Center for Research on Intelligent Perception and Computing (CRIPAC)
State Key Laboratory of Multimodal Artificial Intelligence Systems(MAIS)
Institute of Automation, Chinese Academy of Science (CASIA)

Outline

1 Course Review

2 Deep Cognitive Networks

3 Attention Models

4 Memory Models

Scenario of Reinforcement Learning



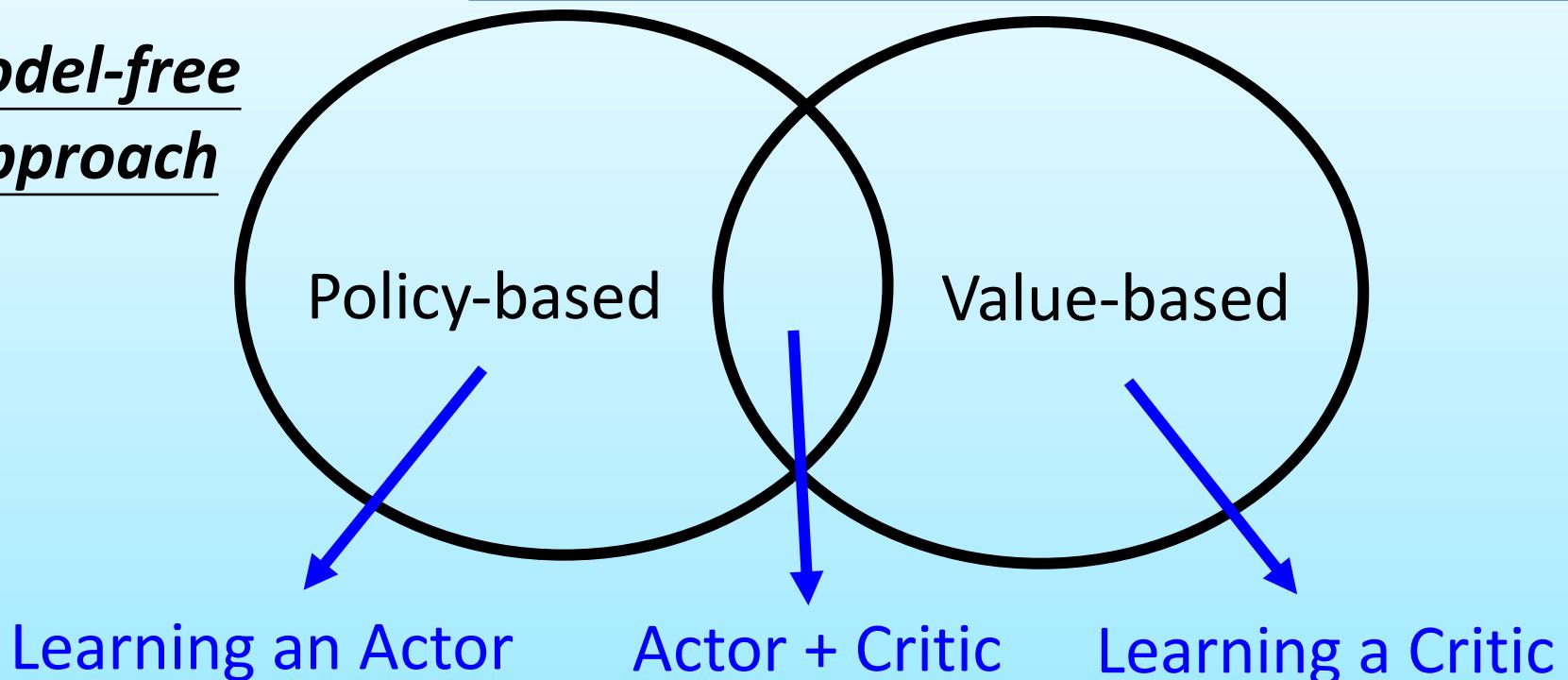
Scenario of Reinforcement Learning



Methods of Reinforcement Learning

Alpha Go: policy-based + value-based + model-based

Model-free Approach

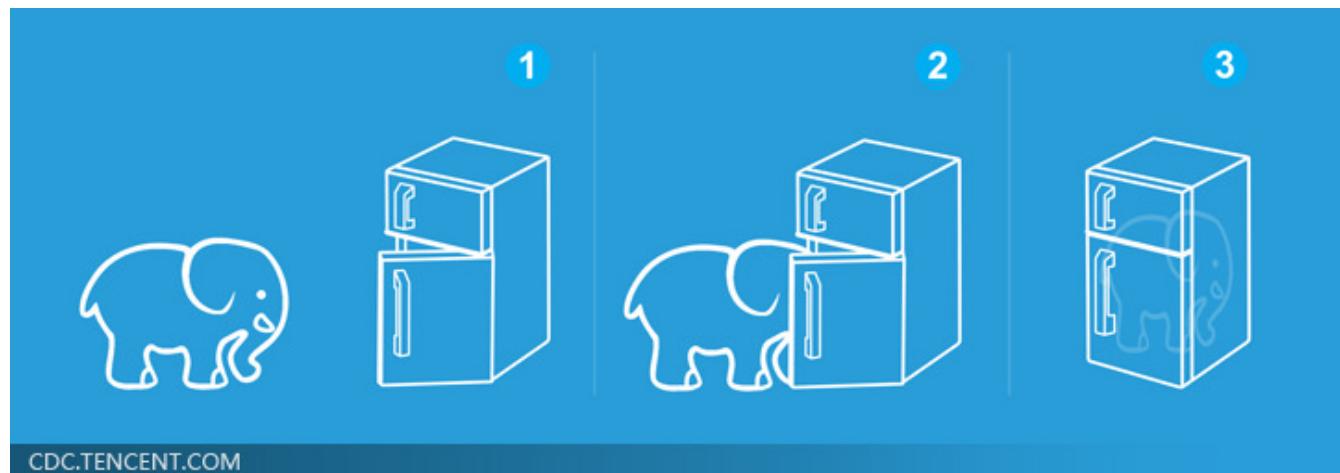


Model-based Approach

Three Steps for Deep Learning

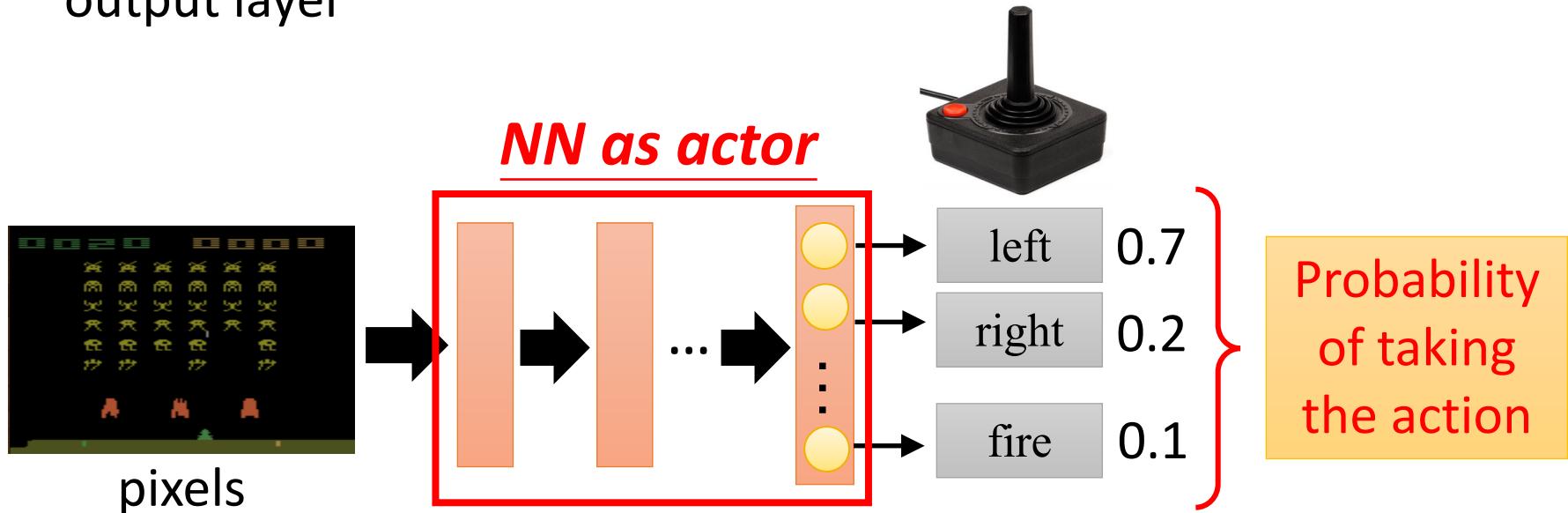


Deep Learning is so simple



Neural Network as Actor

- **Input** of neural network: the observation of machine represented as a vector or a matrix
- **Output** neural network: each action corresponds to a neuron in output layer



What is the benefit of using network instead of lookup table?

generalization

Goodness of Actor

- Given an actor $\pi_\theta(s)$ with network parameter θ
- Use the actor $\pi_\theta(s)$ to play the video game
 - Start with observation s_1
 - Machine decides to take a_1
 - Machine obtains reward r_1
 - Machine sees observation s_2
 - Machine decides to take a_2
 - Machine obtains reward r_2
 - Machine sees observation s_3
 -
 - Machine decides to take a_T
 - Machine obtains reward r_T

END

Total reward: $R_\theta = \sum_{t=1}^T r_t$

Even with the same actor,
 R_θ is different each time

Randomness in the actor
and the game

We define \bar{R}_θ as the
expected value of R_θ , and
evaluate the goodness of
an actor $\pi_\theta(s)$

Gradient Ascent

- Problem statement

$$\theta^* = \arg \max_{\theta} \bar{R}_{\theta}, \quad \bar{R}_{\theta} = \sum_{\tau} R(\tau) P(\tau | \theta)$$

- Gradient ascent

- Start with θ^0

- $\theta^1 \leftarrow \theta^0 + \eta \nabla \bar{R}_{\theta^0}$

- $\theta^2 \leftarrow \theta^1 + \eta \nabla \bar{R}_{\theta^1}$

-

$$\theta = \{w_1, w_2, \dots, b_1, \dots\}$$

$$\nabla \bar{R}_{\theta} = \begin{bmatrix} \partial \bar{R}_{\theta} / \partial w_1 \\ \partial \bar{R}_{\theta} / \partial w_2 \\ \vdots \\ \partial \bar{R}_{\theta} / \partial b_1 \\ \vdots \end{bmatrix}$$

Policy Gradient

$$\bar{R}_\theta = \sum_\tau R(\tau)P(\tau|\theta) \quad \nabla \bar{R}_\theta = ?$$

$$\nabla \bar{R}_\theta = \sum_\tau R(\tau) \nabla P(\tau|\theta) = \sum_\tau R(\tau)P(\tau|\theta) \frac{\nabla P(\tau|\theta)}{P(\tau|\theta)}$$

$R(\tau)$ do not have to be differentiable
It can even be a black box.

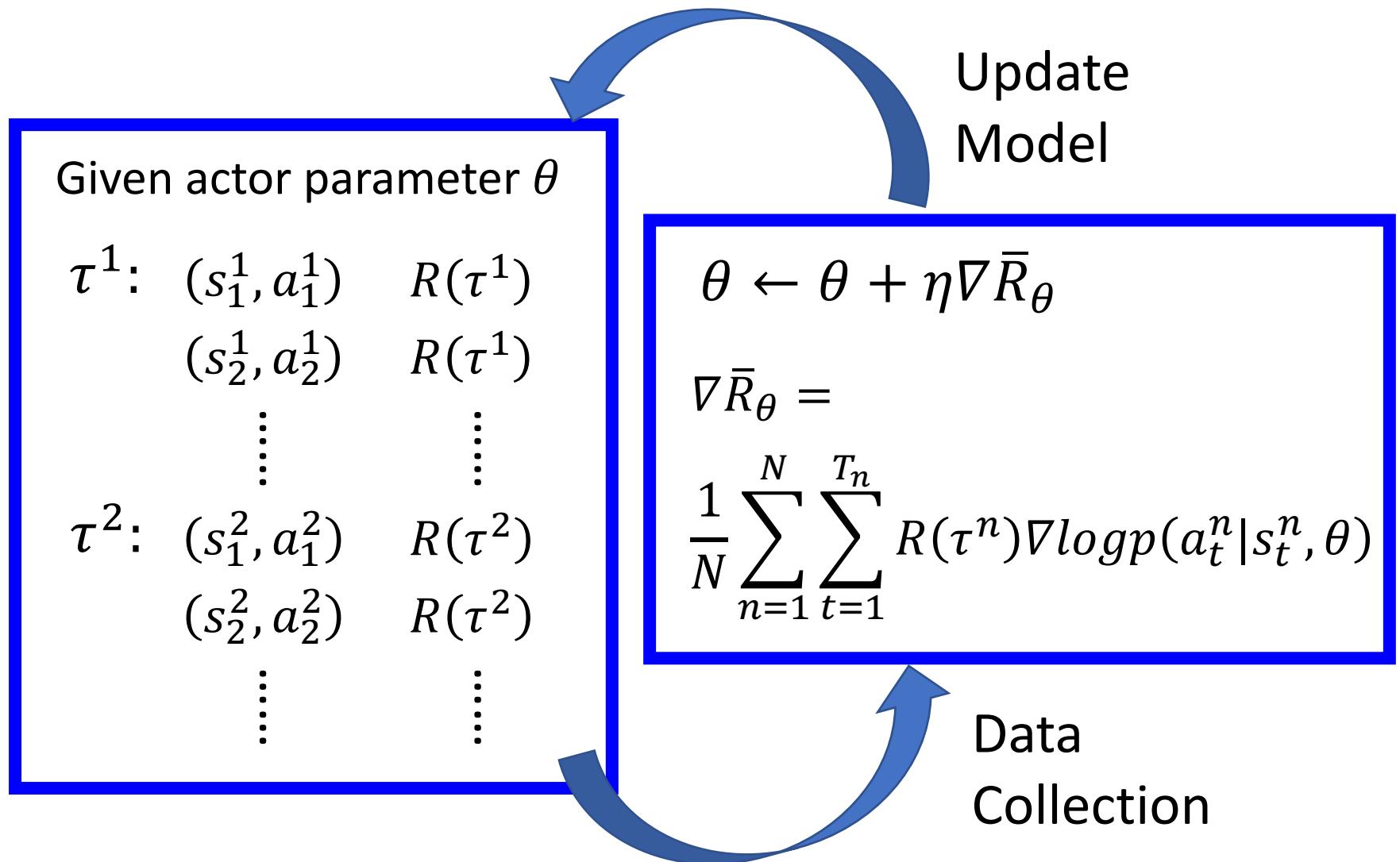
$$= \boxed{\sum_\tau} R(\tau) \boxed{P(\tau|\theta)} \nabla \log P(\tau|\theta)$$

$$\boxed{\frac{d \log(f(x))}{dx} = \frac{1}{f(x)} \frac{df(x)}{dx}}$$

$$\approx \frac{1}{N} \sum_{n=1}^N R(\tau^n) \underline{\nabla \log P(\tau^n|\theta)}$$

Use π_θ to play the game N times,
Obtain $\{\tau^1, \tau^2, \dots, \tau^N\}$

Policy Gradient



Critic

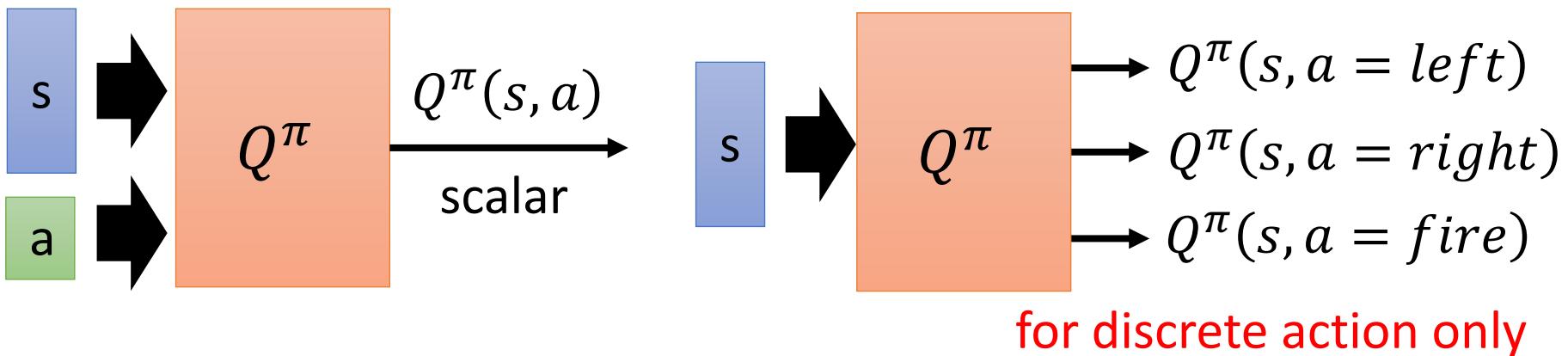
- A critic does not determine the action
- Given an actor π , it evaluates the how good the actor is



Critic

$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi \right]$$

- State-action value function $Q^\pi(s, a)$
 - When using actor π , the *cumulated* reward expects to be obtained after seeing observation s and taking a



Bellman Equation

The optimal Q-value function Q^* is the maximum expected cumulative reward achievable from a given (state, action) pair:

$$Q^*(s, a) = \max_{\pi} \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi \right]$$

Q^* satisfies the following Bellman equation:

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]$$

Intuition: if the optimal state-action values for the previous time-step $Q^*(s', a')$ are known, then the optimal strategy is to take the action that maximizes the expected value of $r + \gamma Q^*(s', a')$

The optimal policy π^* corresponds to taking the best action in any state as specified by Q^*

Bellman Equation

Value iteration algorithm: Use Bellman equation as an iterative update

$$Q_{i+1}(s, a) = \mathbb{E} \left[r + \gamma \max_{a'} Q_i(s', a') | s, a \right]$$

Q_i will converge to Q^* as $i \rightarrow \infty$

What's the problem with this?

Not scalable. Must compute $Q(s, a)$ for every state-action pair. If state is e.g. current game state pixels, computationally infeasible to compute for entire state space!

Solution: use a function approximator to estimate $Q(s, a)$. E.g. a neural network!

Q-learning

Q-learning: Use a function approximator to estimate the action-value function

$$Q(s, a; \theta) \approx Q^*(s, a)$$

function parameters (weights)

If the function approximator is a deep neural network => **deep q-learning!**

Remember: want to find a Q-function that satisfies the Bellman Equation:

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]$$

Forward Pass

Loss function: $L_i(\theta_i) = \mathbb{E}_{s, a \sim \rho(\cdot)} [(y_i - Q(s, a; \theta_i))^2]$

where $y_i = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a \right]$

Iteratively try to make the Q-value close to the target value (y_i) it should have, if Q-function corresponds to optimal Q^* (and optimal policy π^*)

Backward Pass

Gradient update (with respect to Q-function parameters θ):

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s, a \sim \rho(\cdot); s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i) \right] \nabla_{\theta_i} Q(s, a; \theta_i) \quad 16$$

Outline

1 Course Review

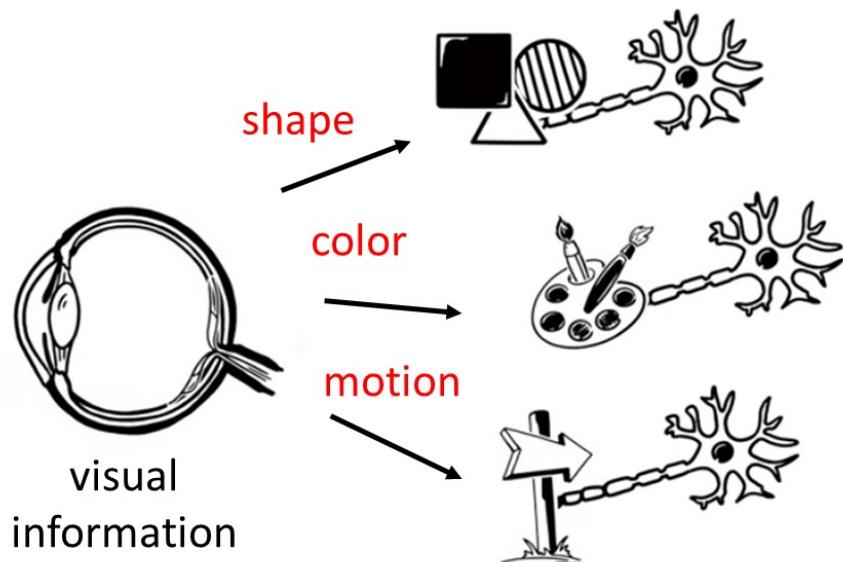
2 Deep Cognitive Networks

3 Attention Models

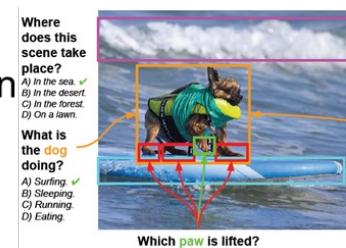
4 Memory Models

Deep Learning (Perception)

- Mainly **percept visual shape, color and motion information** based on neural networks
- Cannot well handle redundant content and complex relations in complex scenarios, bad performance (**accuracy < 80%**)



Visual question answering,
visual dialog



58% 2016
65% 2017
70% 2018
75% 2019

Visual
captioning,
cross-modal
retrieval

			Generated Sentence: SA: a man is riding a bike M ³ : a man is riding a motorcycle
			Generated Sentence: SA: a man is playing a guitar M ³ : a man is playing with a dog
			Generated Sentence: SA: men are playing soccer ball M ³ : people are playing basketball
			Generated Sentence: SA: a man is riding a horse M ³ : a woman is riding a horse

45% 2016
52% 2017
60% 2018
67% 2019

Performance on complex vision tasks

Challenges in Complex Scenarios

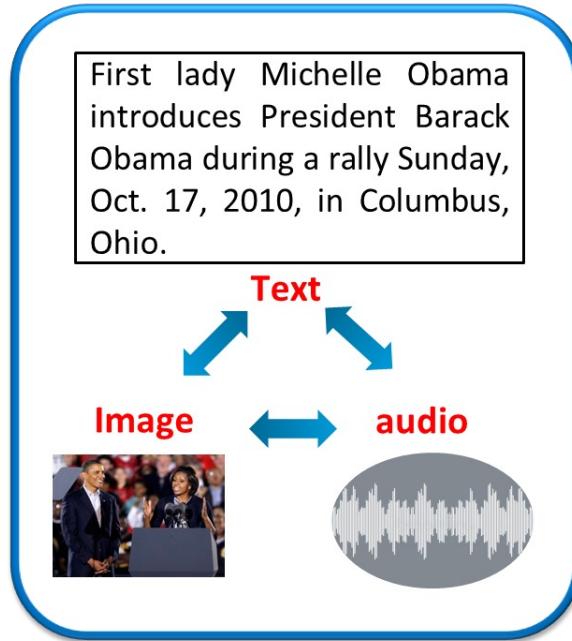
- Understand **high-level semantic** scenes, relations and motions
- Interact with other **abstract data modalities** such as text and audio



autonomous driving



Human-robot interaction

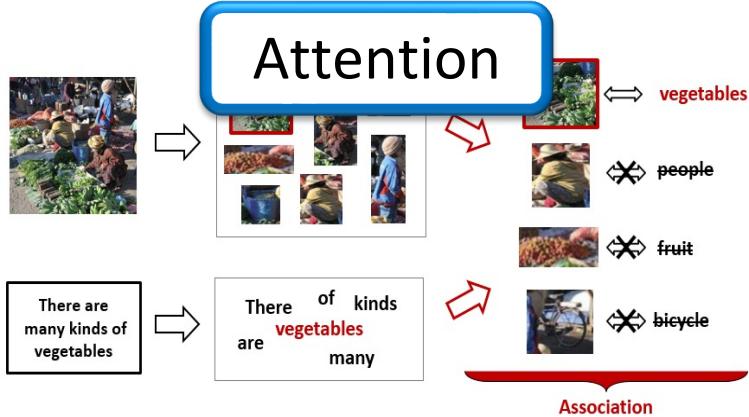


Multimodal surveillance



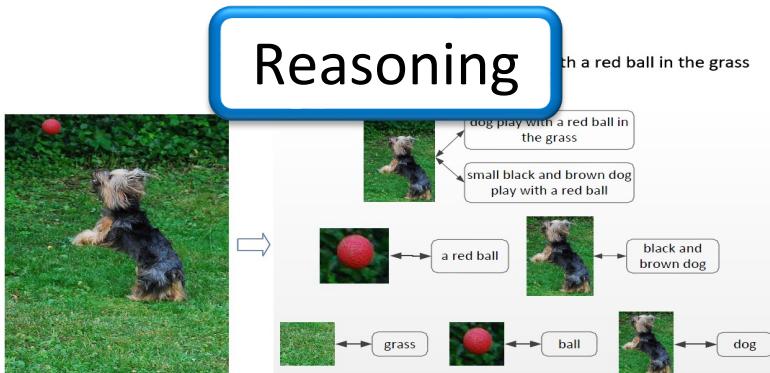
Smart home

Challenges in Complex Scenarios



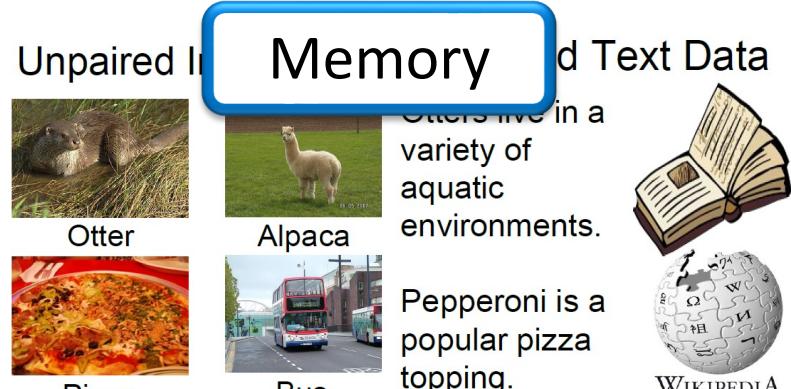
Redundant and task-irrelevant content

Content redundancy reduction



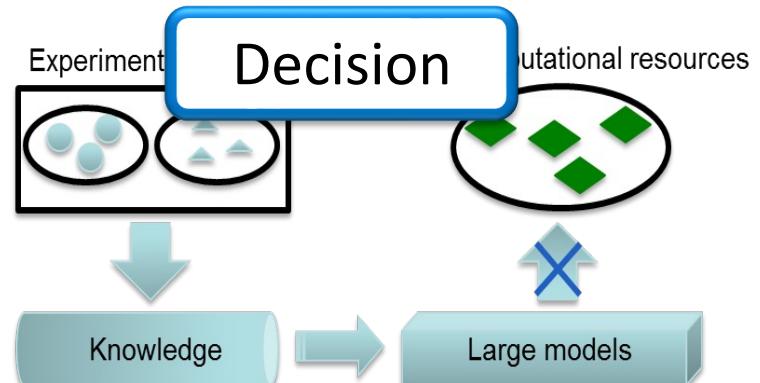
Relations exist in different granularities

Complex cross-modal relation



Annotating paired multimodal data

Cross-modal few-shot data

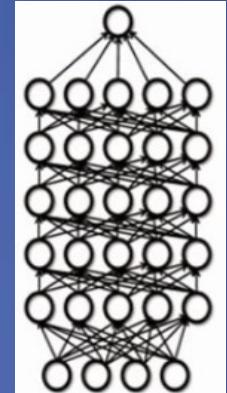
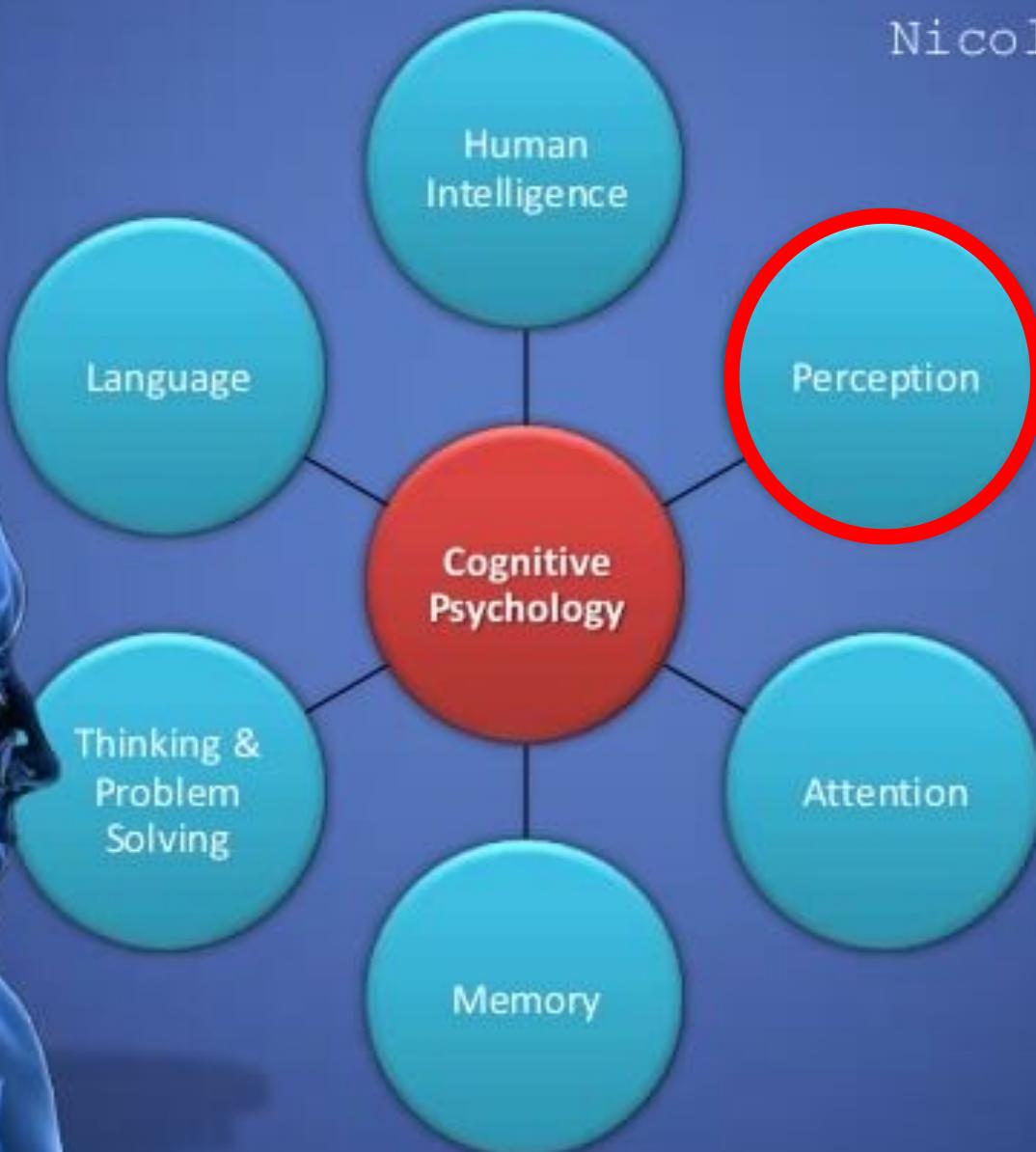


Hard to adapt to new scenarios

Poor model generalization

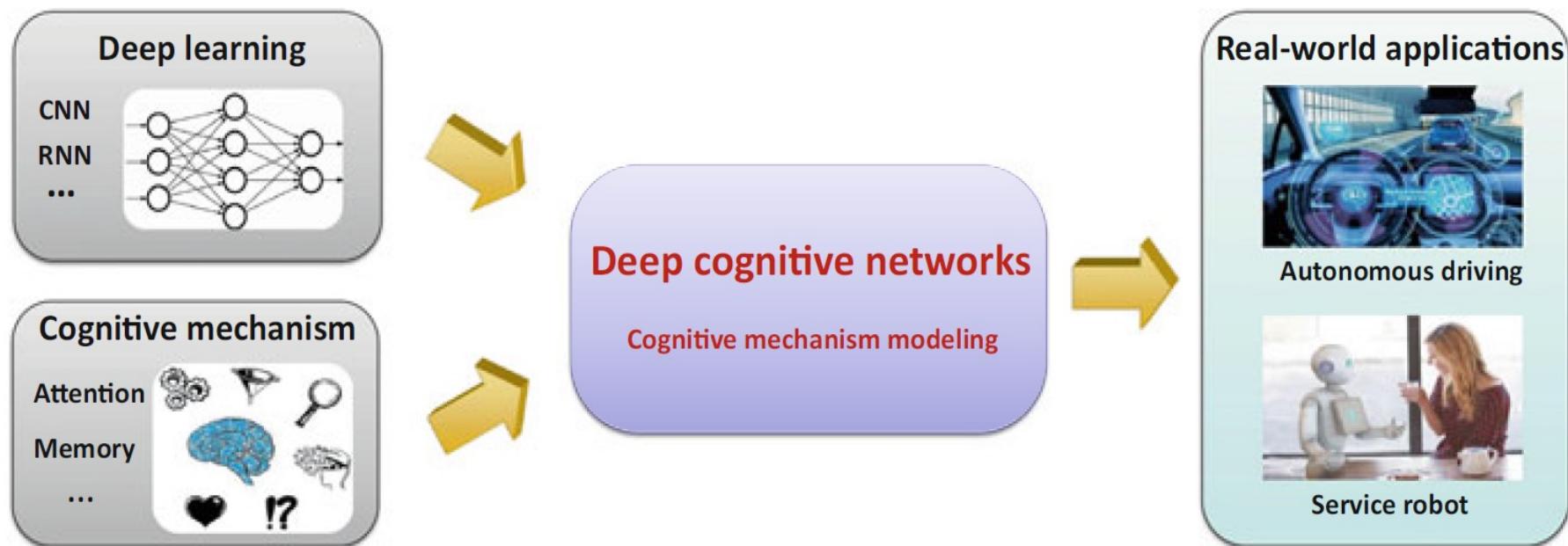
Cognitive Psychology

Nicole Conner



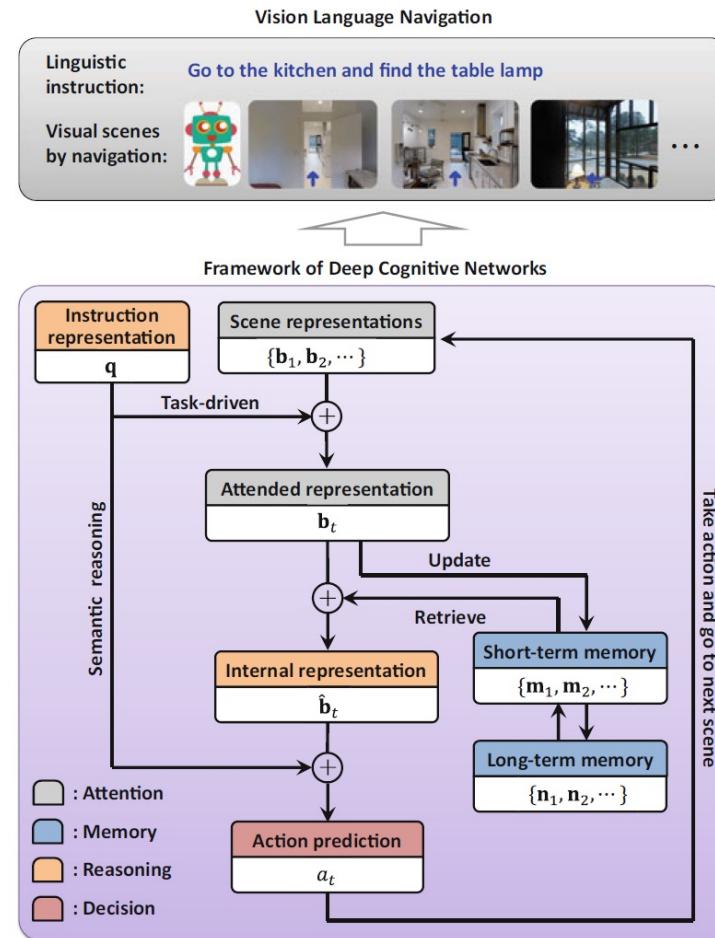
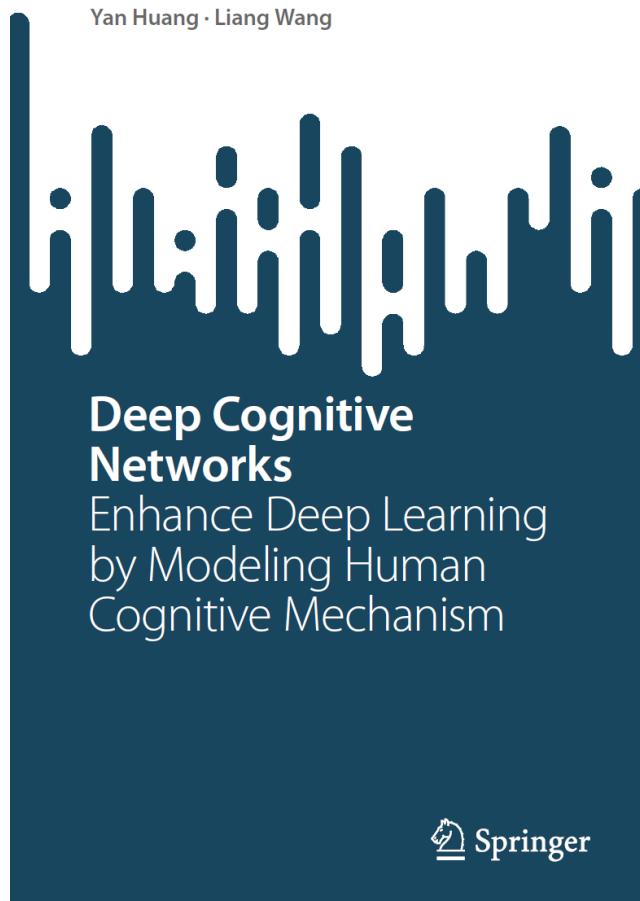
Deep Cognitive Networks

- Model **cognitive mechanisms** such as attention and memory, and be capable of information filtering, reuse, reasoning, etc



General Framework

- Summarize important principles when jointly modeling multiple cognitive mechanisms from the viewpoint of cognitive psychology

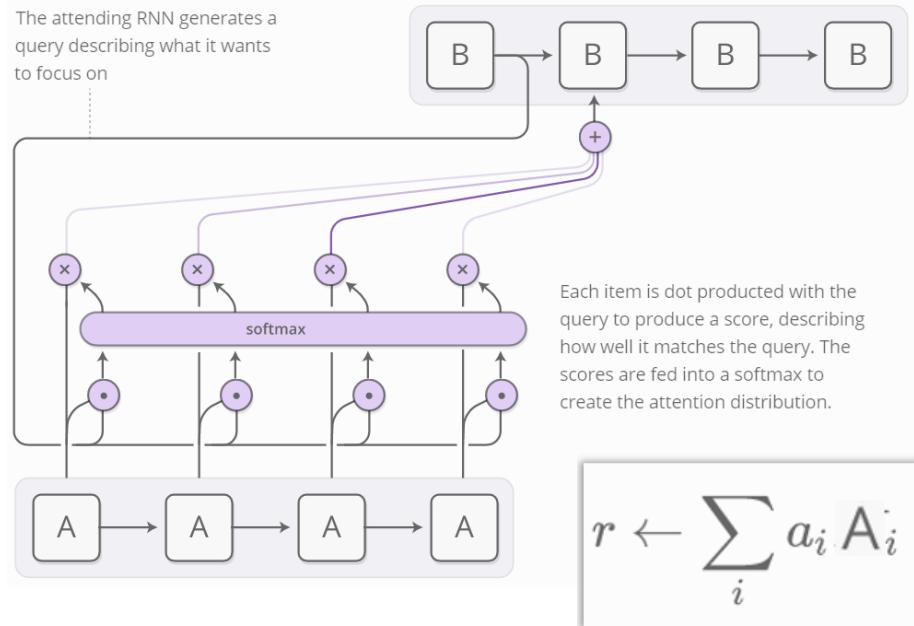


Attention-based DCN

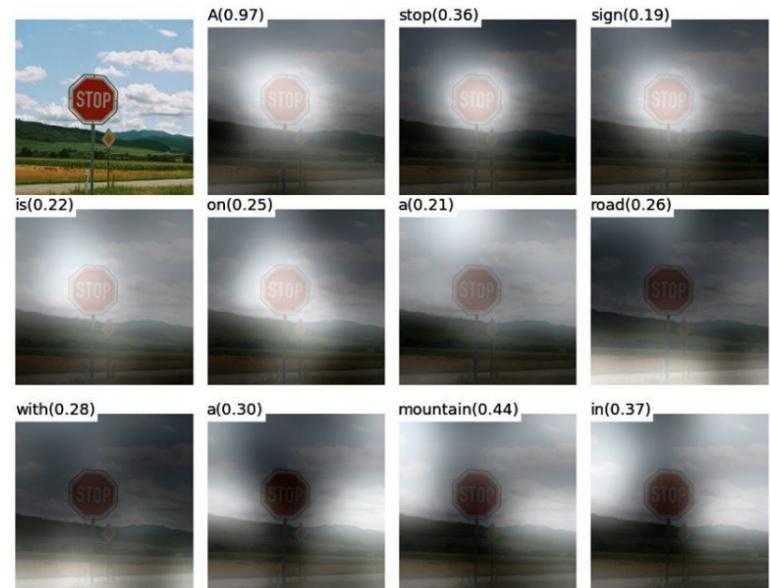
Class	Sub-class	Representative DCN	Theory, model and evidence
Hard attention	Sequential attention	Sequential attention model [2], third-order Boltzmann machine [9]	Filter theory [10], spotlight theory [11], shifter circuits model [12]
	Transformable attention	Spatial transformer network [5], deep recurrent attentive writer [13]	
Soft attention	Recurrent attention	Attention-based RNN [3], show, attend and tell model [4]	Attenuation theory [14], cross-channel priming effect [15], feature integration theory [16], biased competition theory [17]
	Cross-modal attention	Hierarchical question-image co-attention [6]	
	Channel attention	Squeeze-and-excitation network [7]	
	Self attention	transformer [8]	

Attention-based DCN

- First **predict weights** for all components and then **weighted sum** them
- Very useful in various CV and NLP tasks



Soft attention

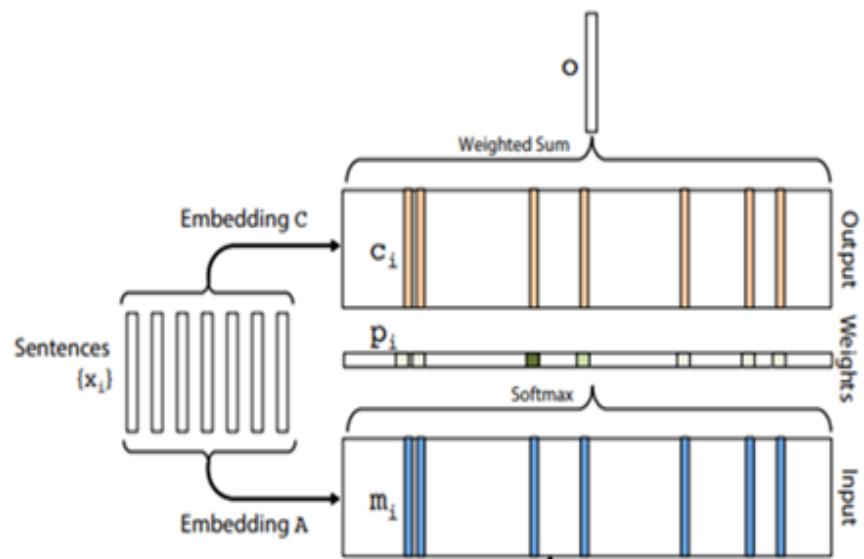


Memory-based DCNs

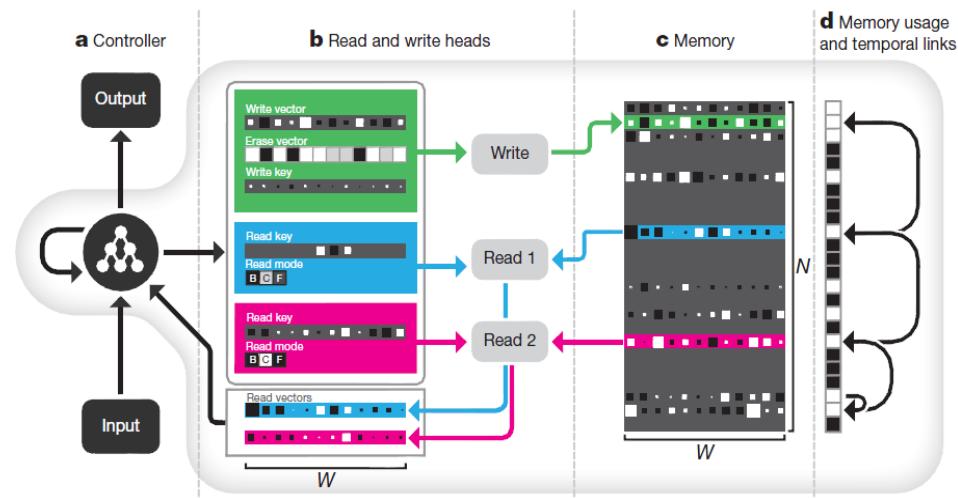
Class	Sub-class	Representative DCN	Theory, model and evidence
Short-term memory	Working memory	Neural turing machine [1], Differential neural computer [6]	Baddeley-Hitch model [7], Atkinson-Shiffrin model [8]
	Short-term and long-term memory	Long short-term memory [4], gated recurrent unit [9]	
Long-term memory	Episodic memory	Memory network [2], end-to-end memory network [10]	Episodic memory and semantic memory [11], dual coding of knowledge [12]
	Conceptual memory	Prototypical network [3], multimodal aligned conceptual knowledge [13]	
	Semantic memory	Multimodal knowledge graph [5]	

Memory-based DCNs

- Use **external memory items** to store long-term information for later reuse, and selectively **read and update** memory items



End-to-end memory
(memory items can only be addressed)



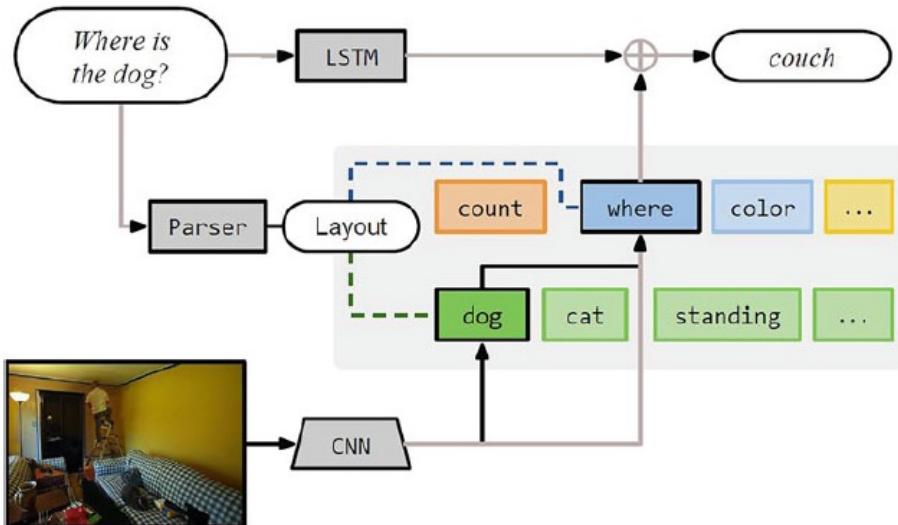
Neural Turing memory
(memory items can be updated)

Reasoning-based DCNs

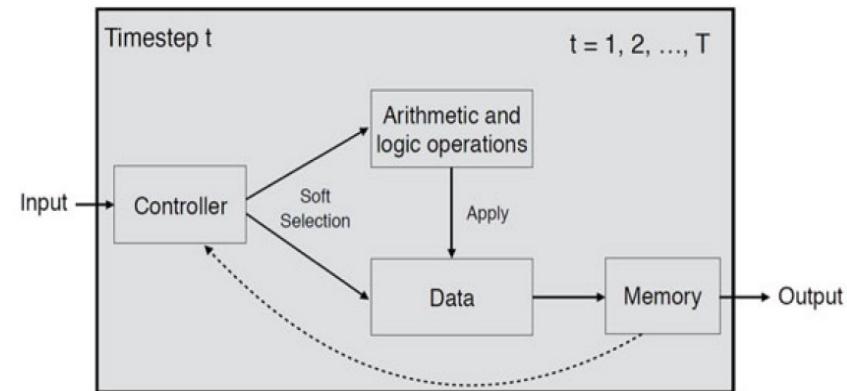
Class	Sub-class	Representative DCN	Theory, model and evidence
Analogical reasoning	Memory reasoning	Memory network [1], End-to-end memory network [5]	Structure mapping theory [6, 7], learning and inference with schemas and analogies [8, 9]
	Abstract reasoning	IQ of deep learning [2], relation network [10]	
Deductive reasoning	Compositional reasoning	Neural module network [3], routing network [11]	Modular organization and structure [12, 13], symbolic numerical process [14, 15], mental logic theory [16, 17]
	Programmed reasoning	Neural programmer [4], neural programmer-interpreter[18]	

Reasoning-based DCNs

- Model various **relations** among different objects, attributes, actions, etc.
- Need **multiple hops** to finally get targets/predictions



Neural module network



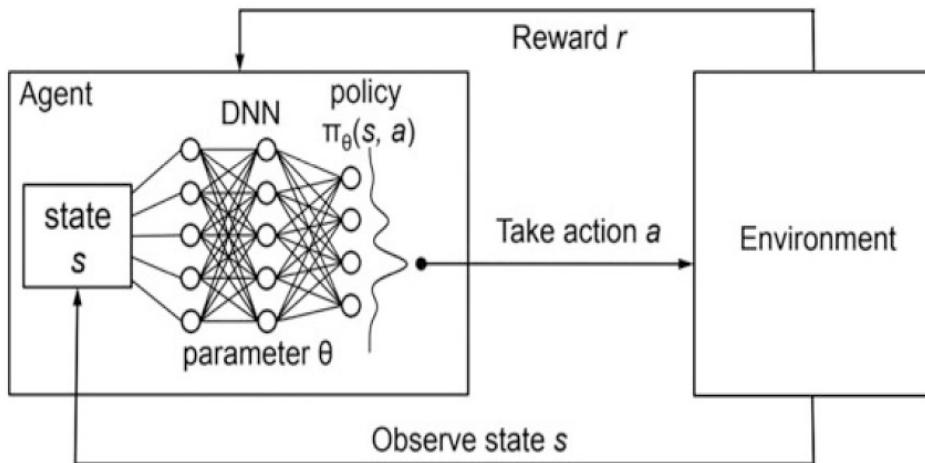
Neural programmer

Decision-based DCNs

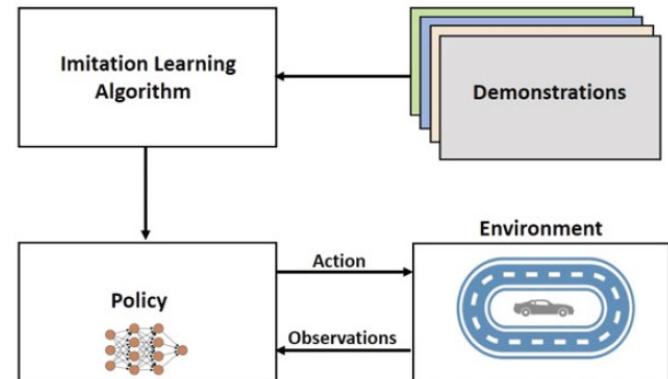
Class	Sub-class	Representative DCN	Theory, model and evidence
Normative decision	Sequential decision	Deep Q-network [1, 2], deep neural decision trees [20]	Expected utility theory [9], multi-attribute utility theory [10], normative model of decision making [25]
	Group decision	multi-agent deep reinforcement learning [3, 4]	
Descriptive decision	Emotional decision	Deep counterfactual regret minimization [5, 6]	Image theory [39, 40], bounded rationality theory [53], elimination by aspects [54]
	Imitative decision	Deep imitation learning [7], generative adversarial imitation learning [61]	

Decision-based DCNs

- An agent chooses how **to interact with external environment**
- Very important for many real-world applications such as service robot and autonomous driving



Deep Q-network



Deep imitation learning

Outline

1 Course Review

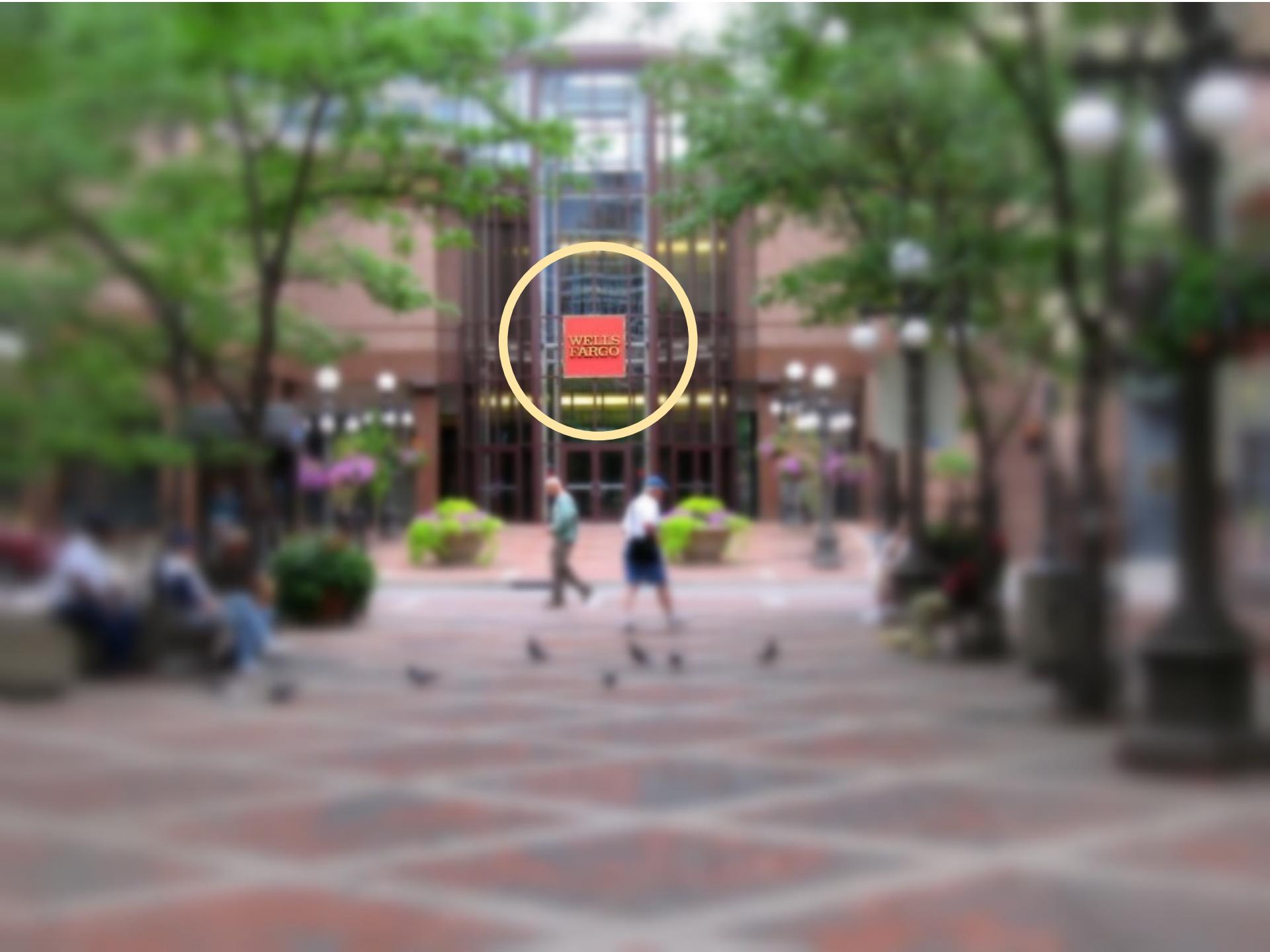
2 Deep Cognitive Networks

3 Attention Models

4 Memory Models







WELLS
FARGO



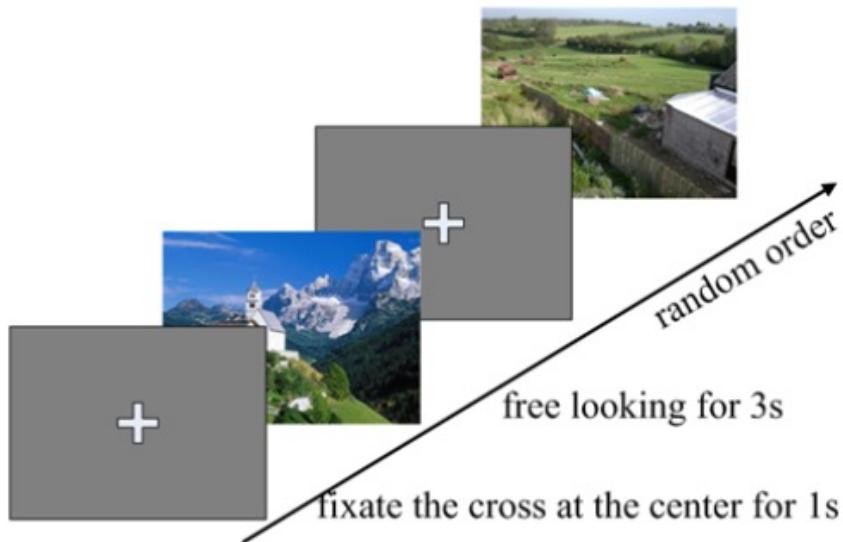




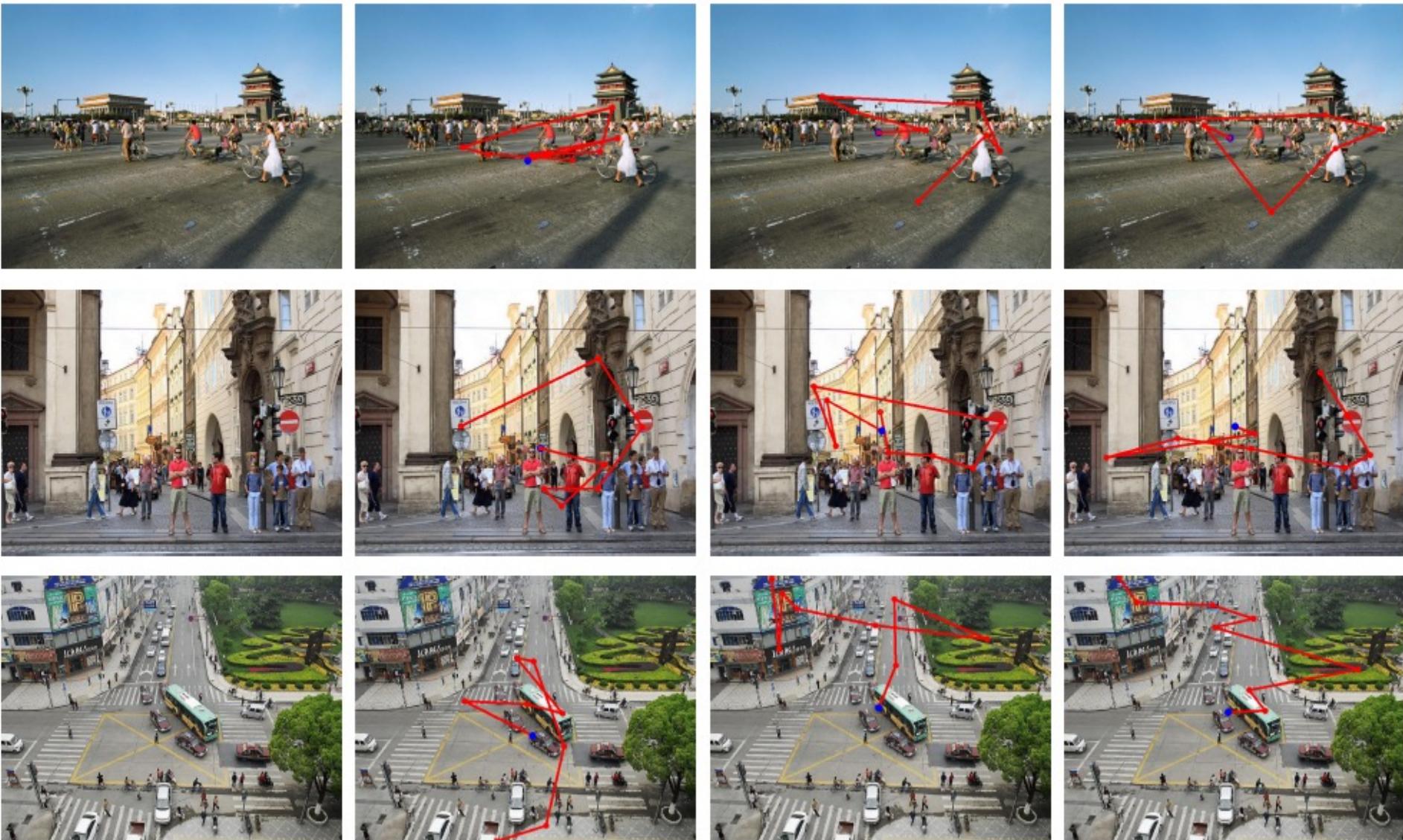
Eye Tracking



任务：让被试看完一副图片后，用文字对
图片内容进行描述



Task-Free Saccadic Scanpath



Given Image

No. 1

No. 2

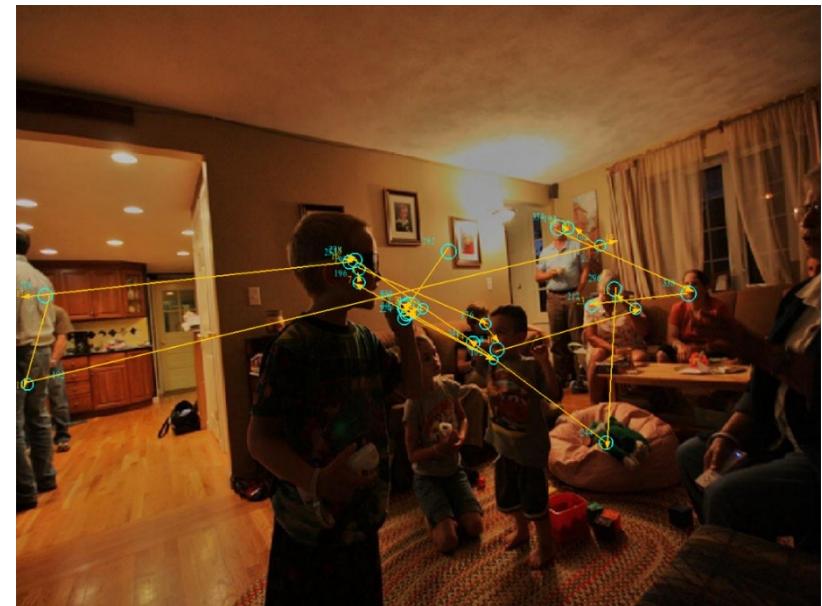
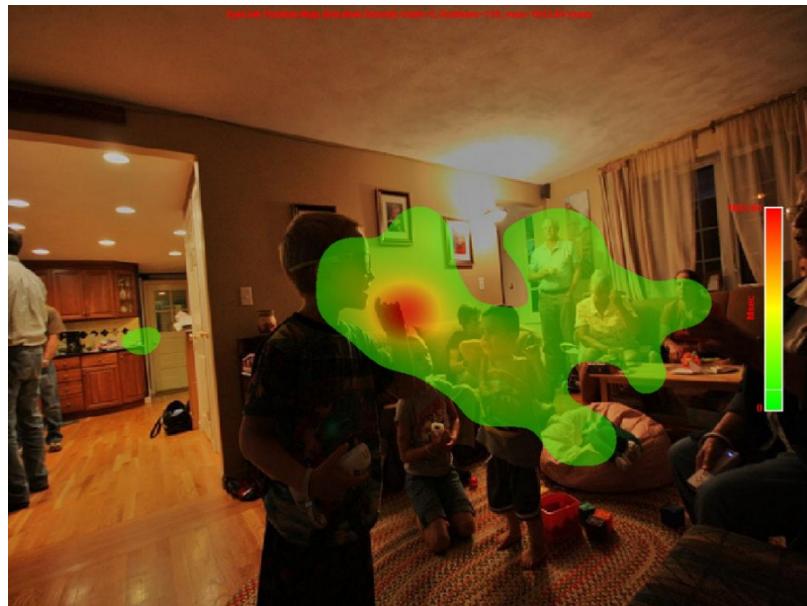
No. 3

Task-Driven Saccadic Scanpath

Tasks:

- 房间里有一群人在嬉闹
- 晚饭过后一家人在客厅里享受悠闲的时光
- 在一个灯光昏暗的房间里有许多小孩在玩耍

- 屋子里一群孩子围在一起玩耍
- 许多人在房间里聚会



Attention

Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence. It implies withdrawal from some things in order to deal effectively with others.

--- Williams James

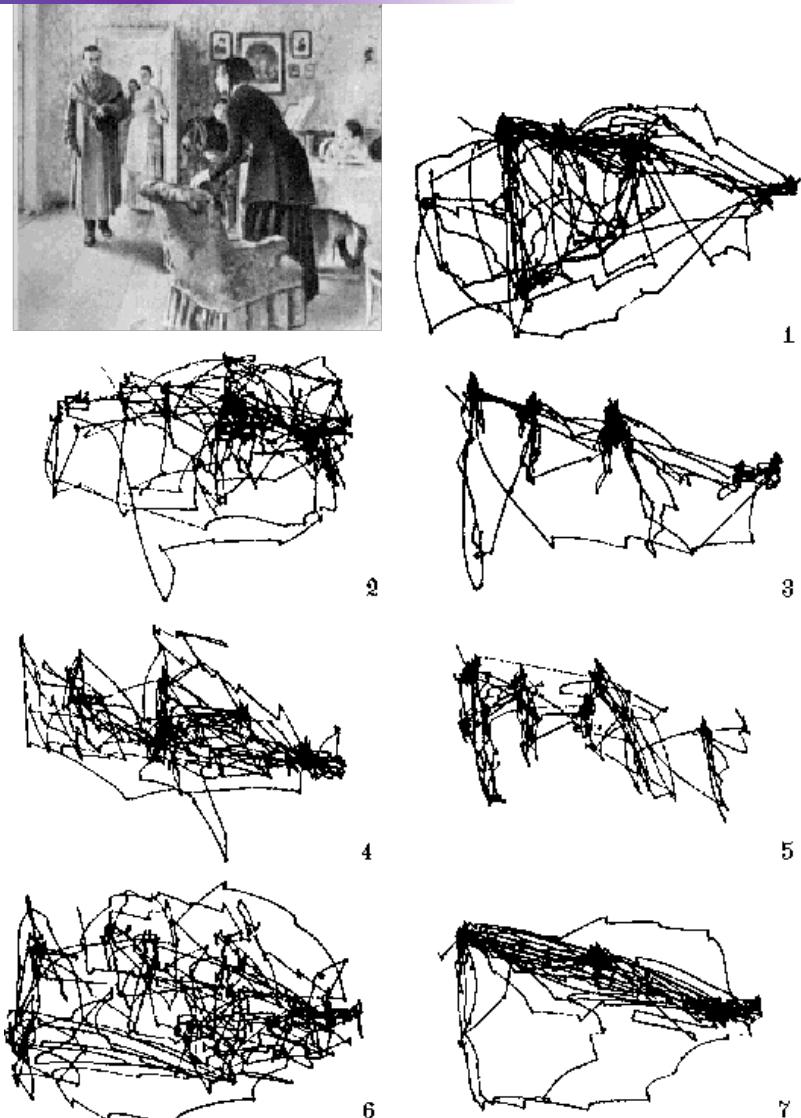
注意在人眼视觉信息处理过程中体现为信息的**选择与过滤**，从而减少向高级皮层传递的信息，避免冗余或者噪声信息对高级视觉任务的干扰，这种选择过程发生在视觉通道的**多个阶段**，而且这种选择过程既有各阶段**视觉特征的刺激**，也有高级皮层向下的**反馈调节**，对选择性注意的研究不仅有理论上的意义，而且有现实的迫切需求

Attention and Task

Yarbus demonstrated how eye movements changed depending on the question asked of the subject:

1. No question asked
2. Judge economic status
3. "What were they doing before the visitor arrived?"
4. "What clothes are they wearing?"
5. "Where are they?"
6. "How long is it since the visitor has seen the family?"
7. Estimate how long the "unexpected visitor" had been away from the family

Each recording lasted
3 minutes

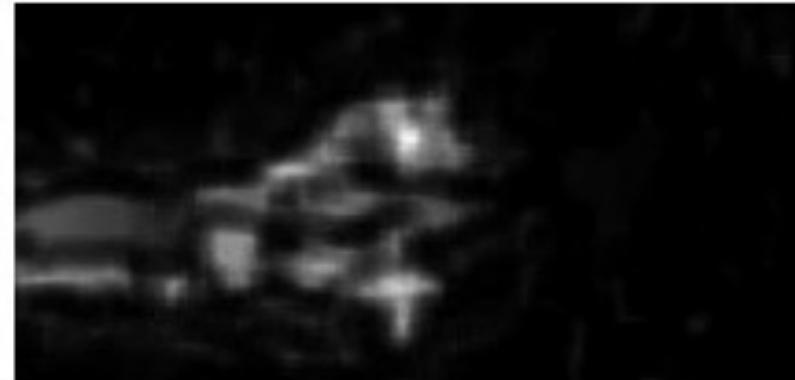
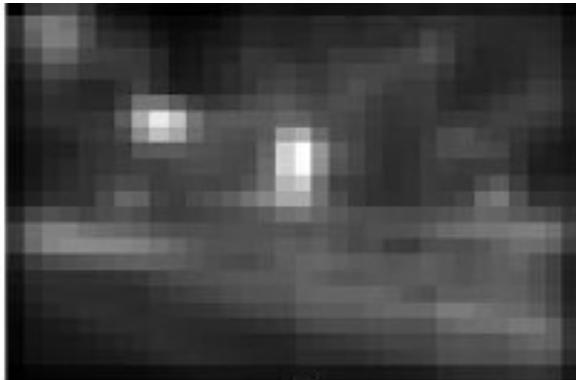


Early Computational Models of Visual Attention

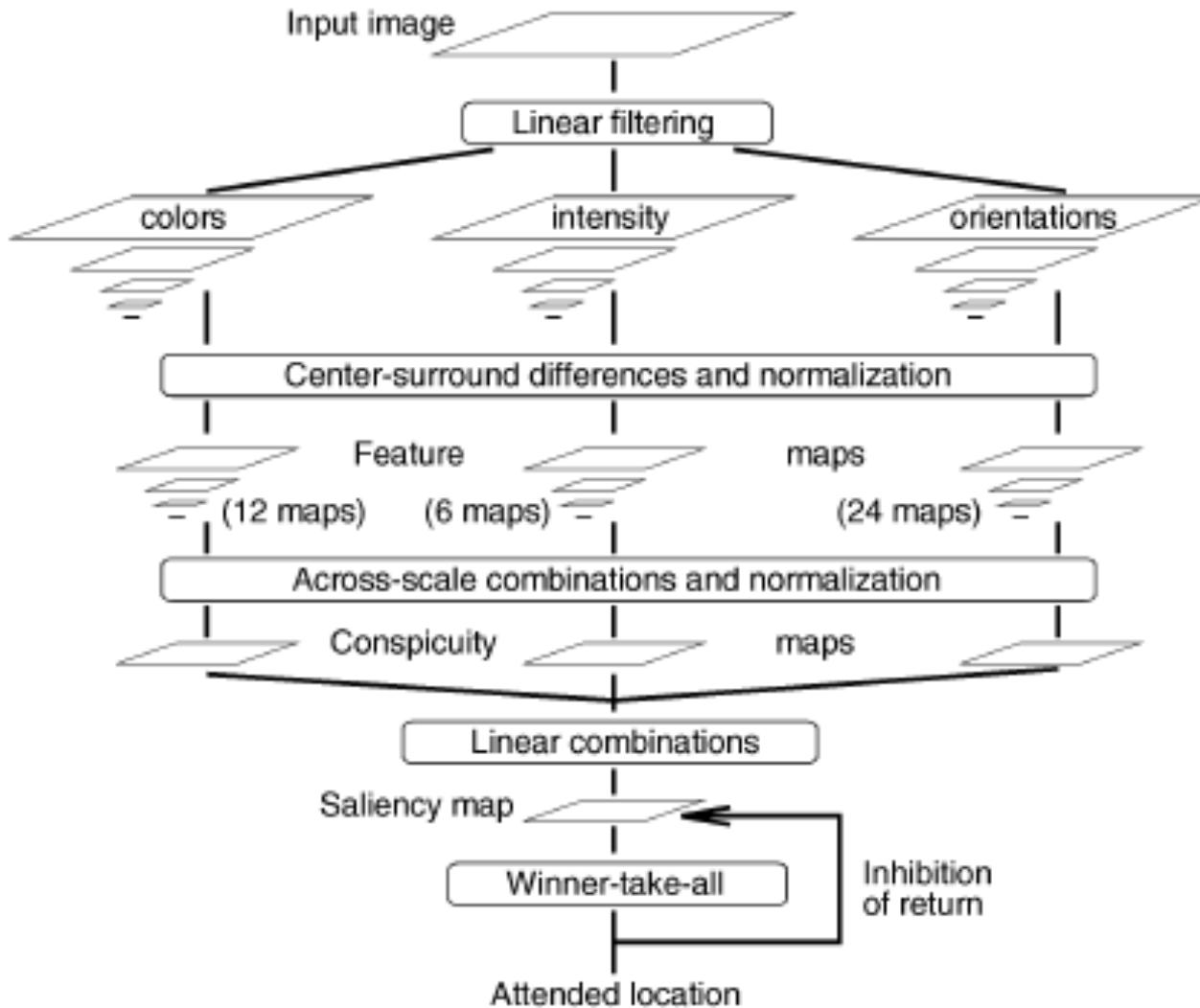
Visual Saliency

A bottom up saliency map in the primary visual cortex (初级视皮层) (V1 saliency)

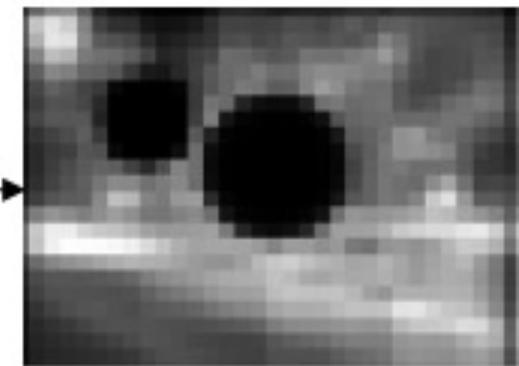
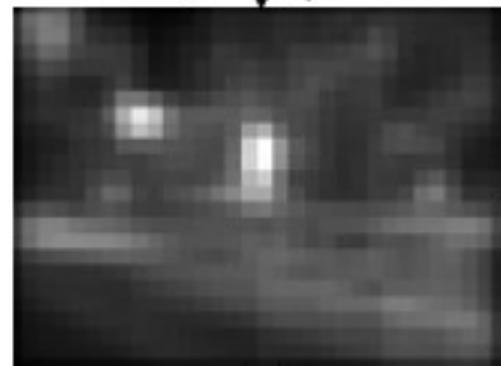
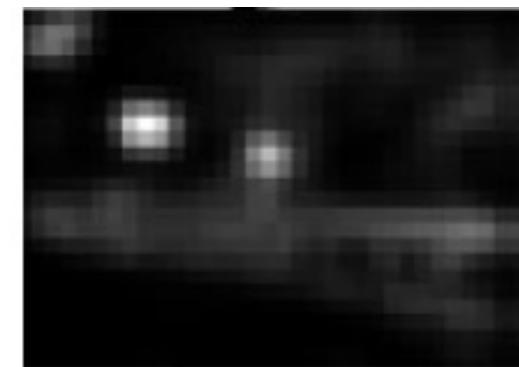
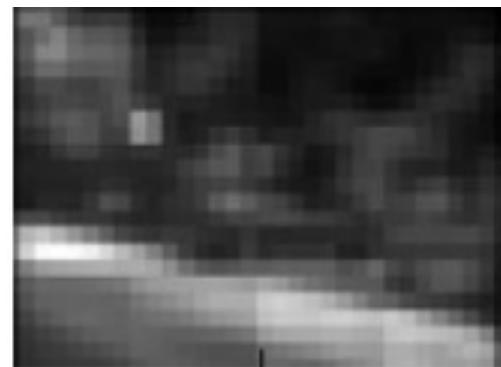
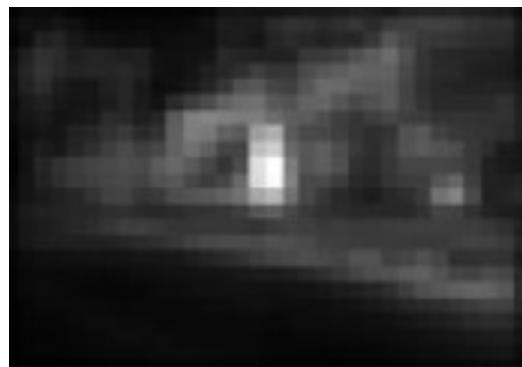
Li Zhaoping@UCL



Center-Surround Difference

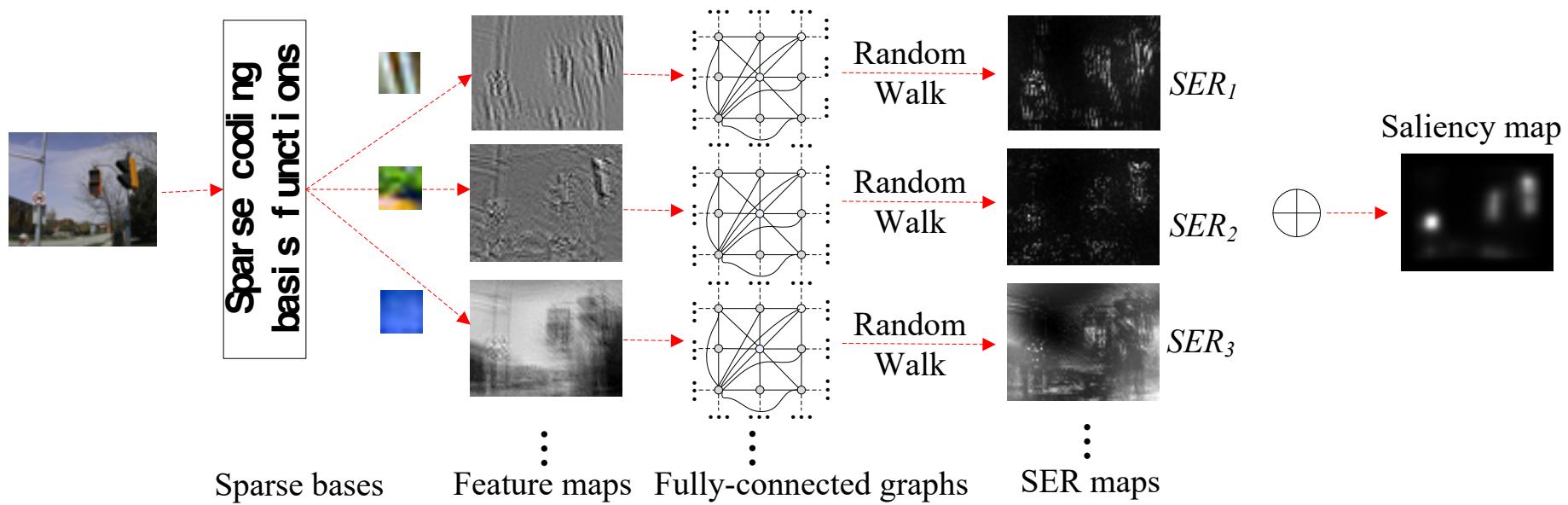


Center-Surround Difference



$$S = \frac{1}{3} (\mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O}))$$

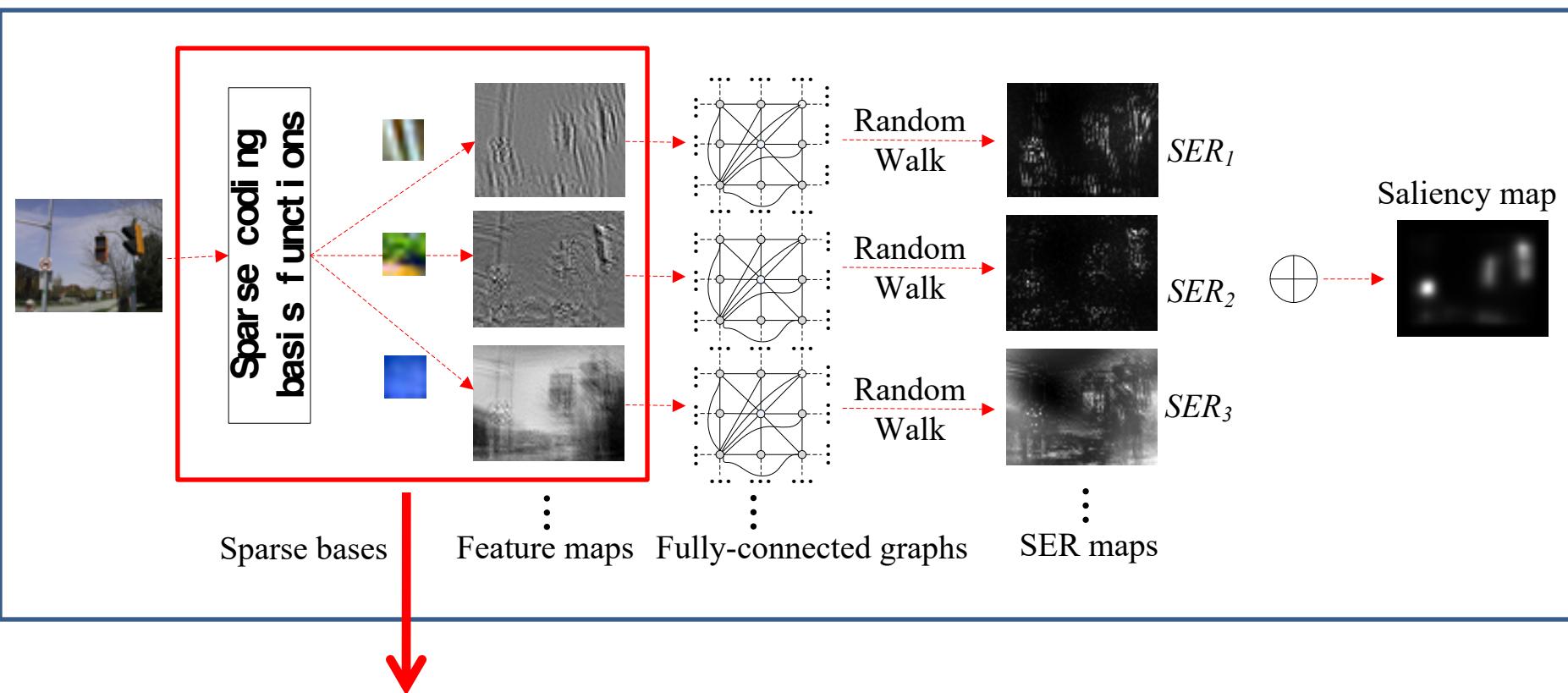
Site Entropy Rate



The procedure of computing the saliency map of an image

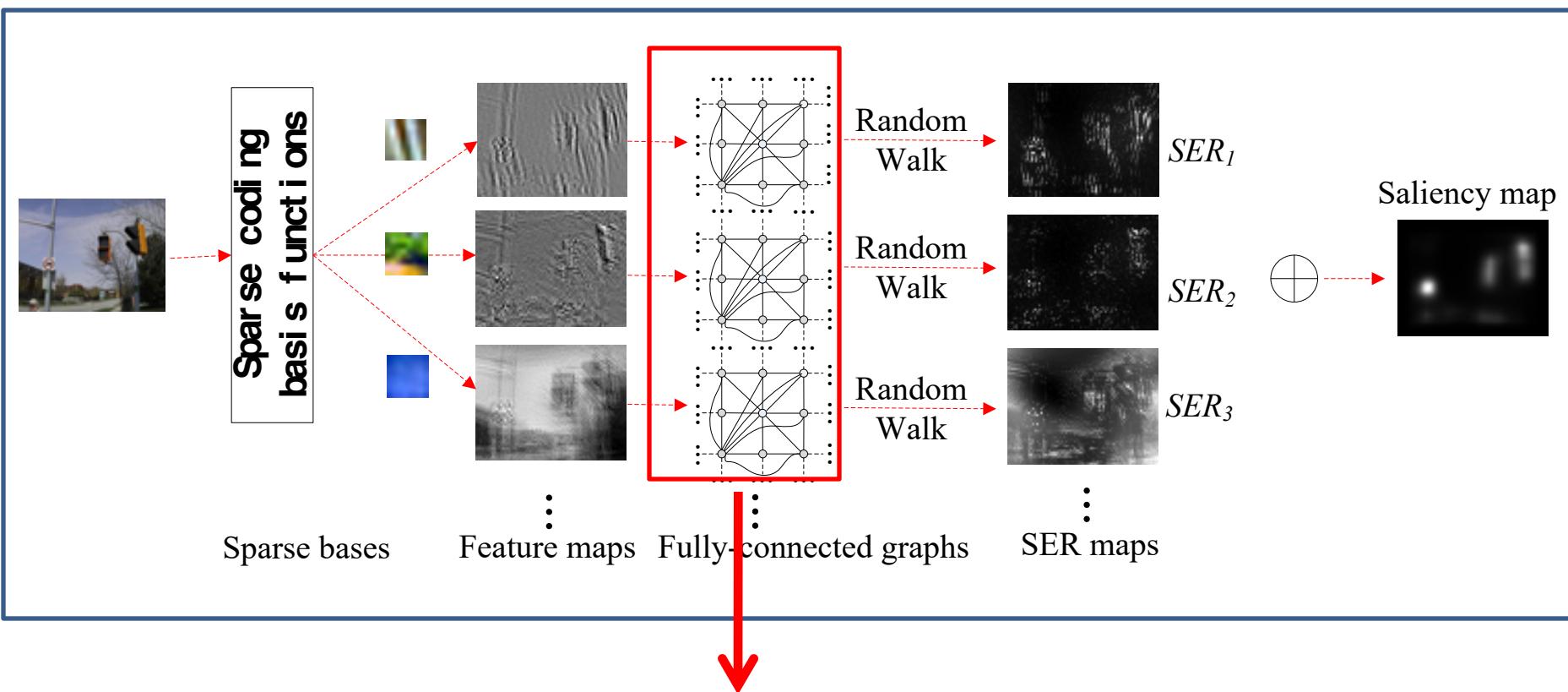
Wei Wang, Yizhou Wang, Qingming Huang, Wen Gao, "Measuring Visual Saliency by Site Entropy Rate", IEEE Computer Vision and Pattern Recognition, CVPR, San Francisco, Jun, 2010.
(oral, google citation: 180)

Site Entropy Rate

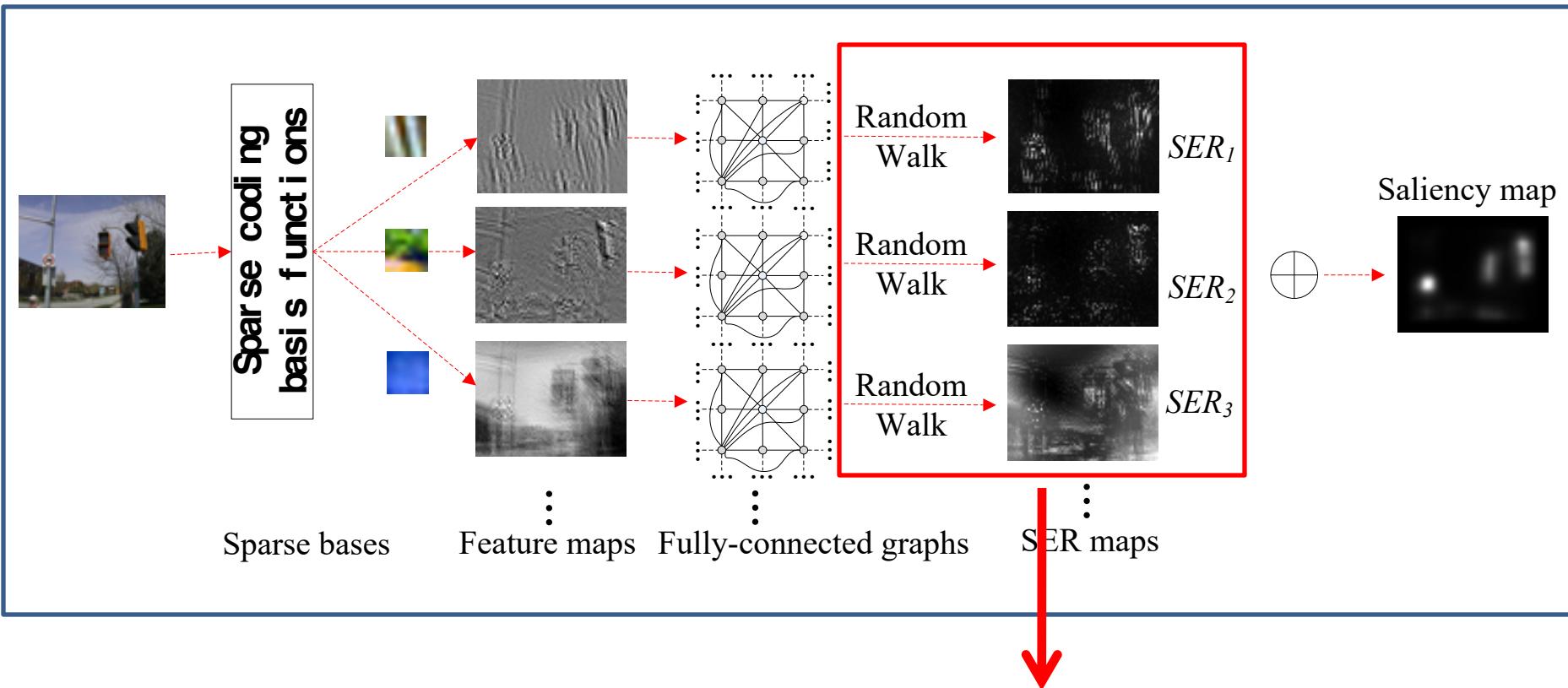


- ① Filter an input image into multi-band filter response maps using number of learned sparse coding basis functions.

Site Entropy Rate



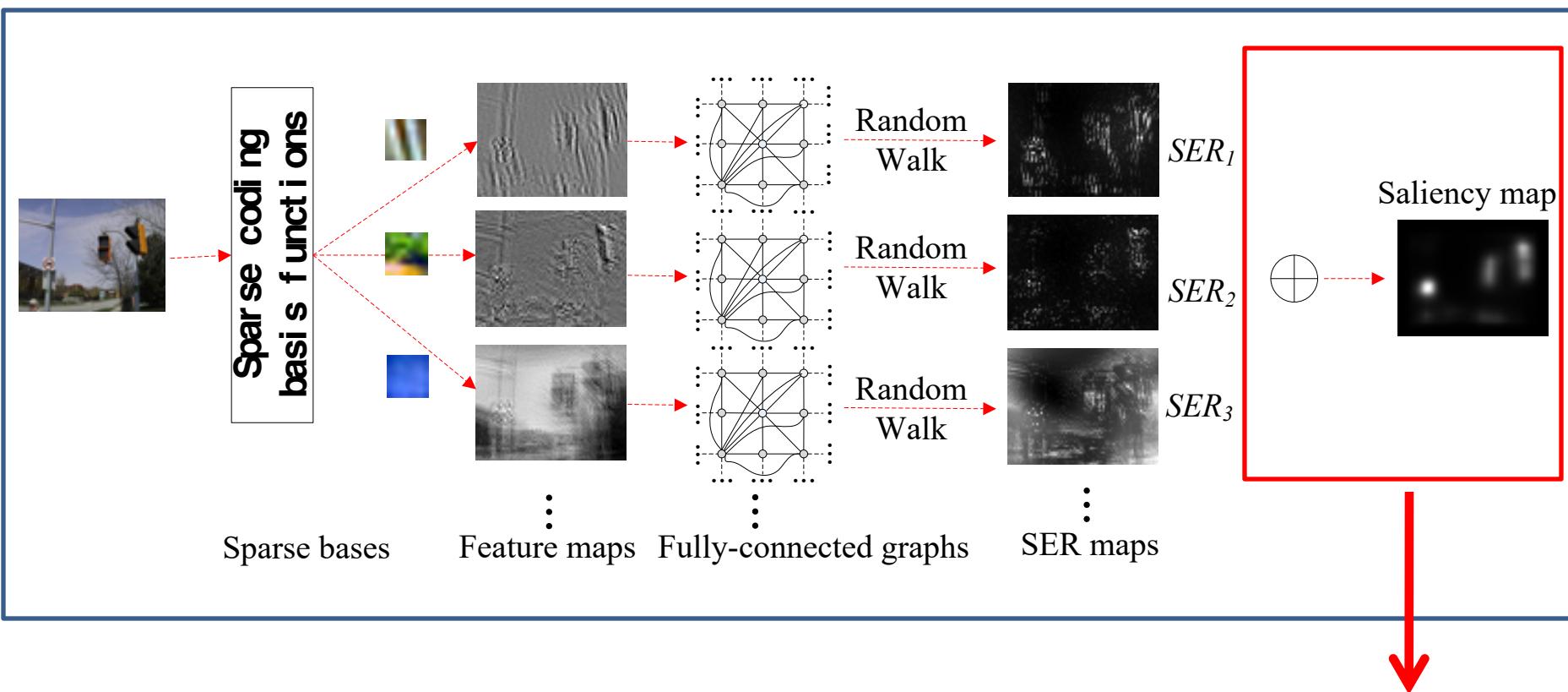
Site Entropy Rate



③ A random walk is adopted on each sub-band graph.

Site Entropy Rate (SER) is proposed to measure the average information from a node to all the others.

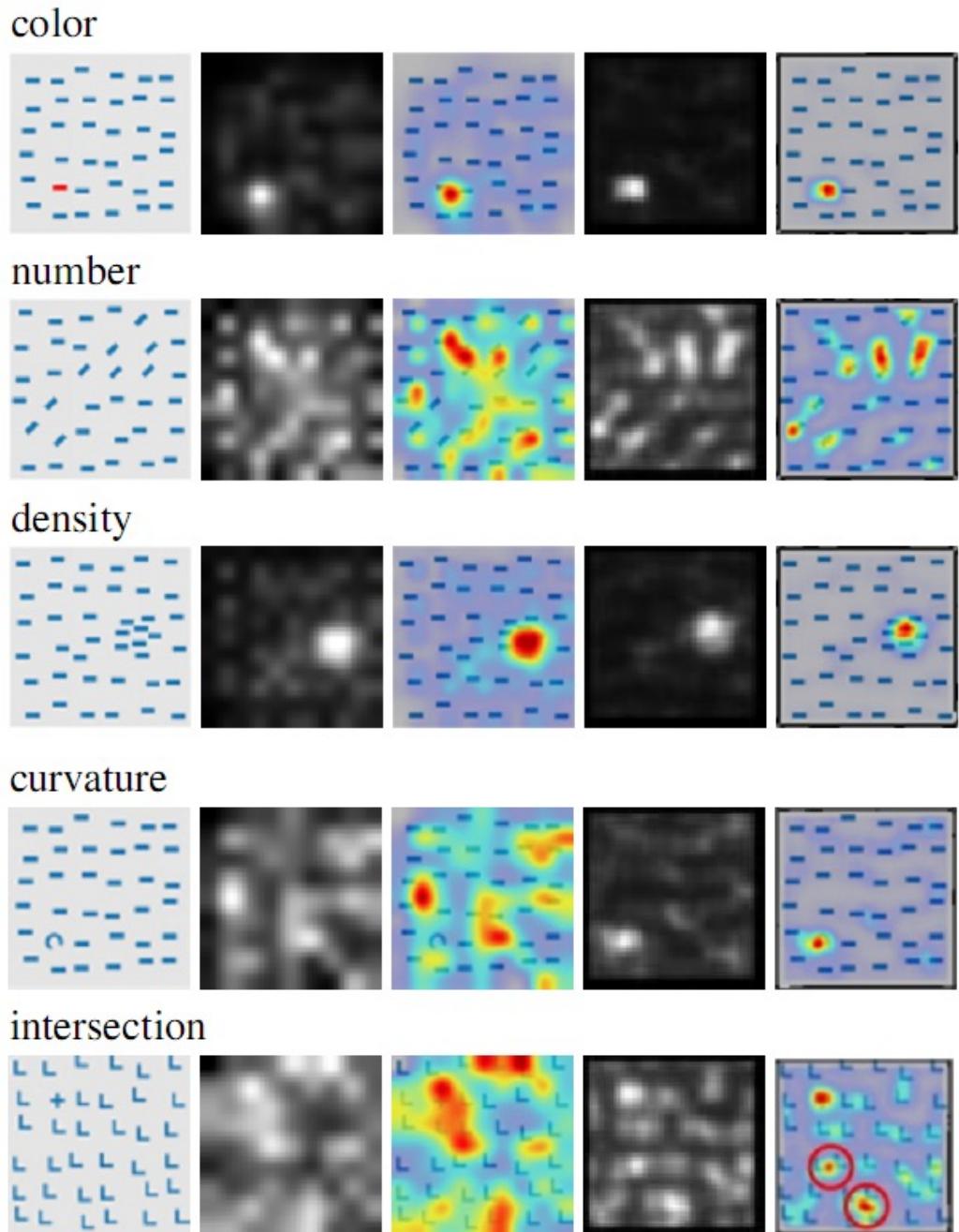
Site Entropy Rate



- ④ The saliency map is computed by summing over all the sub-band SER maps.

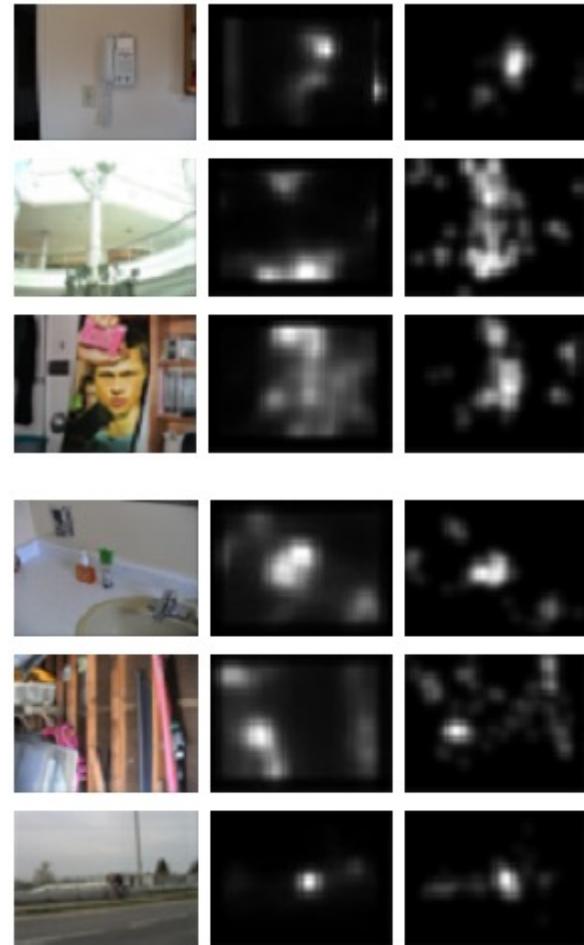
Experiments on Psychological Stimuli

■ Comparison results of five psychological stimuli between our model and Itti et al.'s [PAMI1998]



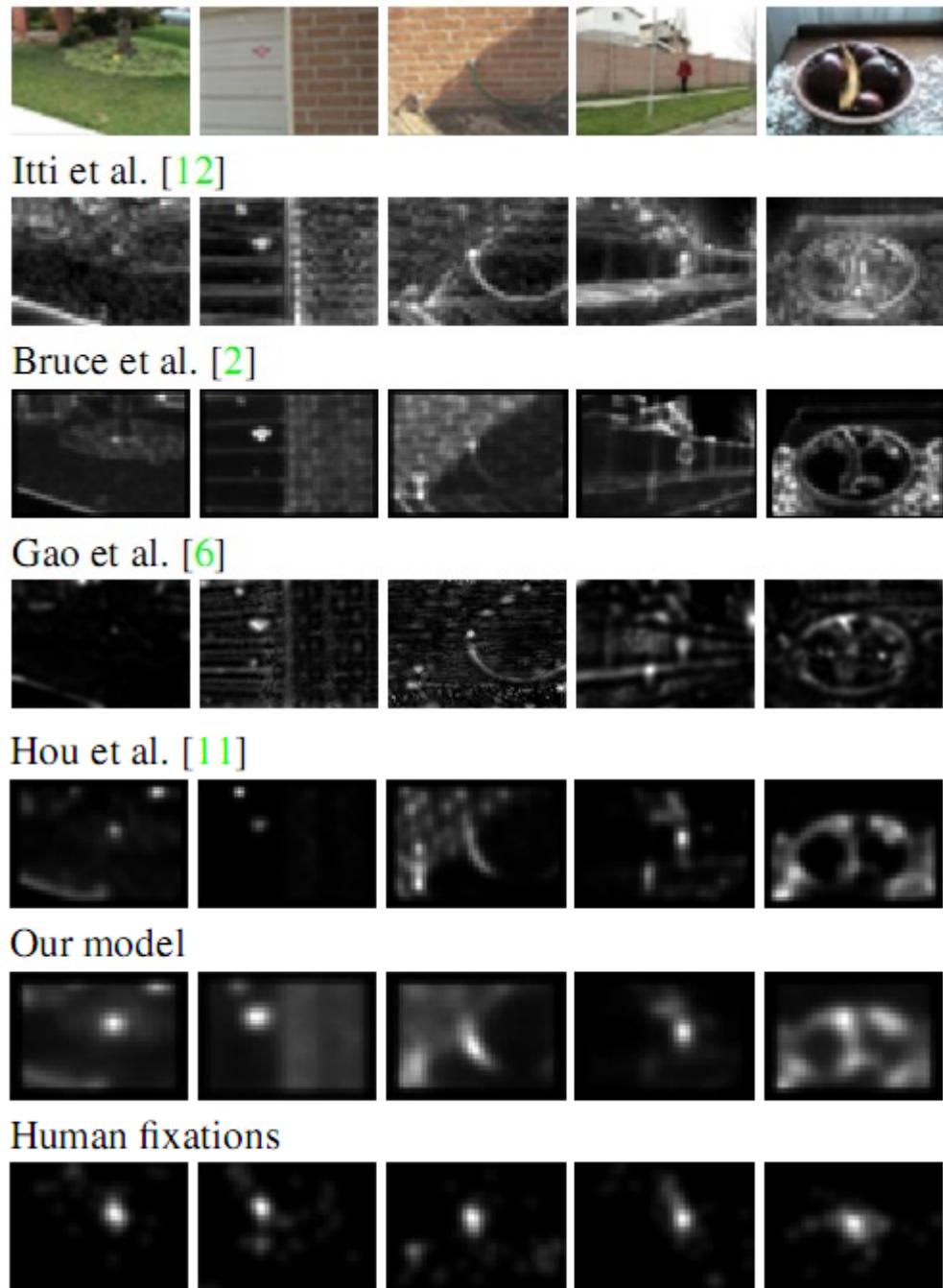
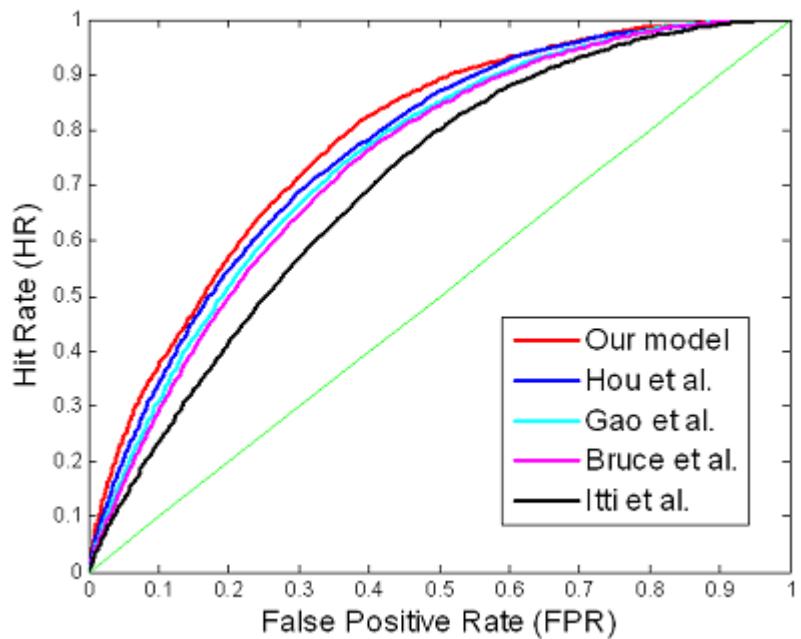
Experiments on Color Image Data Set

- N. Bruce's dataset [NIPS06]
 - 120 color images (indoor and outdoor scenes)
 - Eye fixations recorded from 20 subjects
- Evaluation method
 - quantitative comparison between human fixation density map and saliency map
 - ROC curve/area



Comparison Results

	ROC Area
Itti et al. [12]	0.7031
Bruce et al. [2]	0.7522
Gao et al. [6]	0.7644
Hou et al. [11]	0.7808
Our model	0.8049



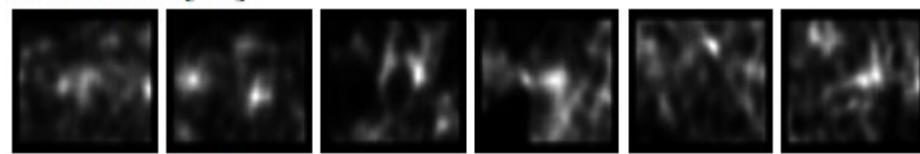
Experiments on Gray Image Data Set

- W. Einhäuser et al's dataset [Vision Research 06]
 - 108 gray images (natural scenes)
 - Fixations from 7 subjects

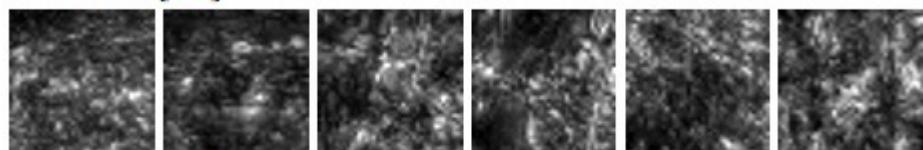
	ROC area
Harel et al. [9]	0.5028
Gao et al. [6]	0.5203
Itti et al. [12]	0.5241
Hou et al. [11]	0.6094
Bruce et al. [2]	0.6420
Our model	0.6537



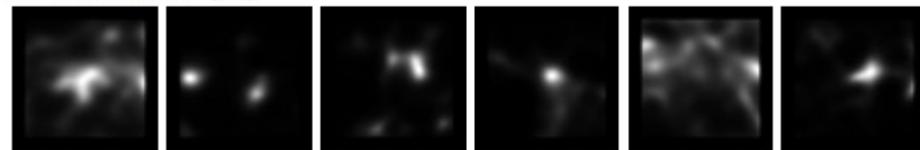
Hou et al. [11]



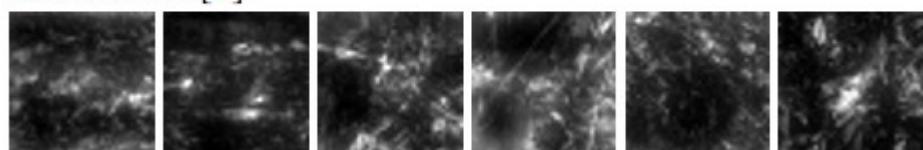
Itti et al. [12]



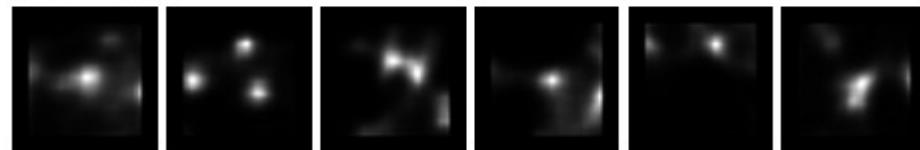
Bruce et al. [2]



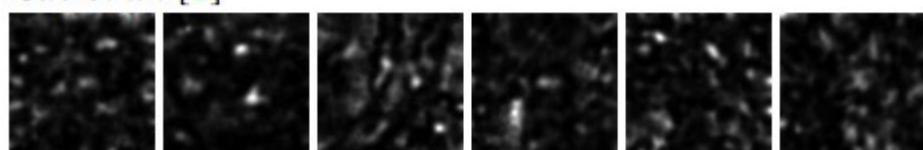
Harel et al. [9]



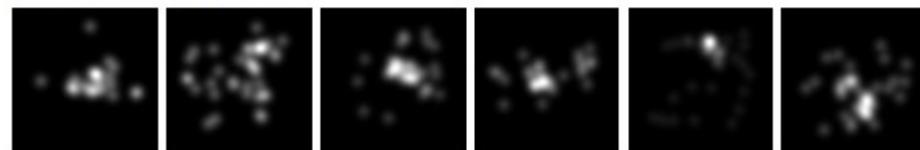
Our model



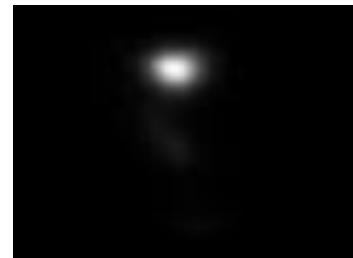
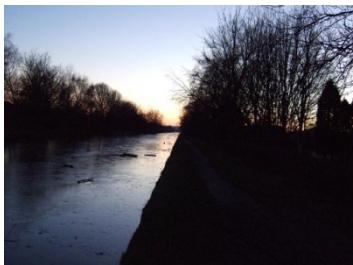
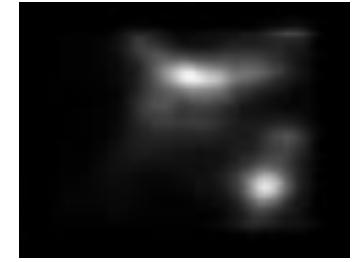
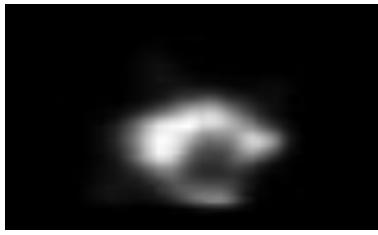
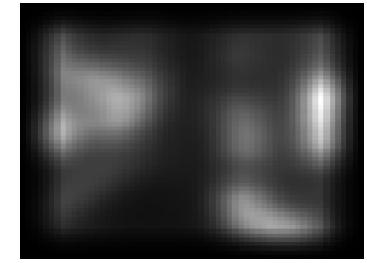
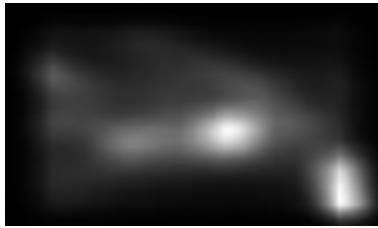
Gao et al. [6]



Human fixations



More Results

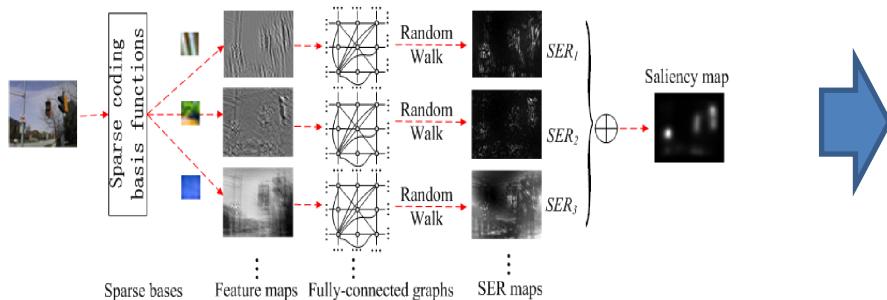


My First Paper

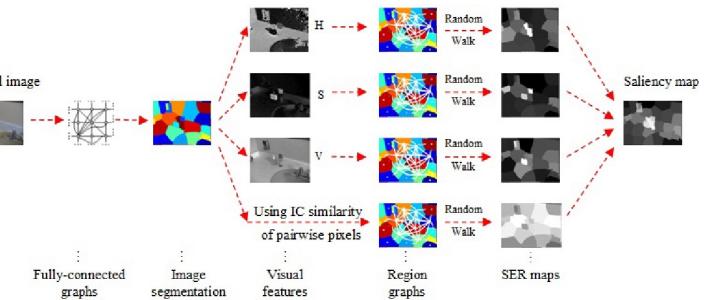
- Extend the SER from pixel-level to region-level

Future work of CVPR10 paper:

semantic aspects. The real power of our model lies in its ability to attend on these aspects and seamlessly fuse global and local information for better caption. For next steps, we plan to experiment with phrase-based visual attribute with its distributed representations, as well as exploring new models for our proposed semantic attention mechanism.

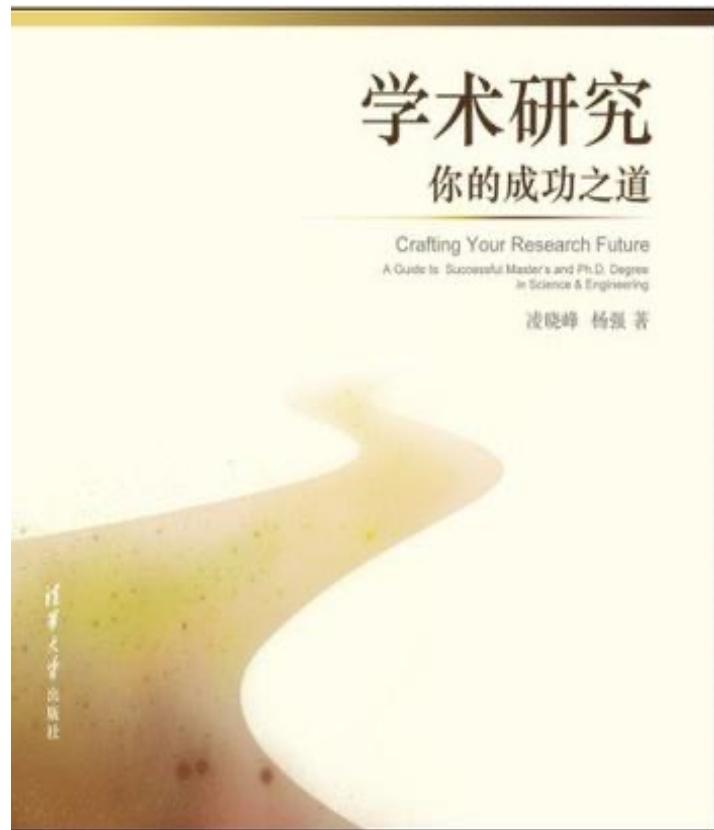


Pixel-level SER, CVPR10



Region-level SER, ICPR12

My First Paper

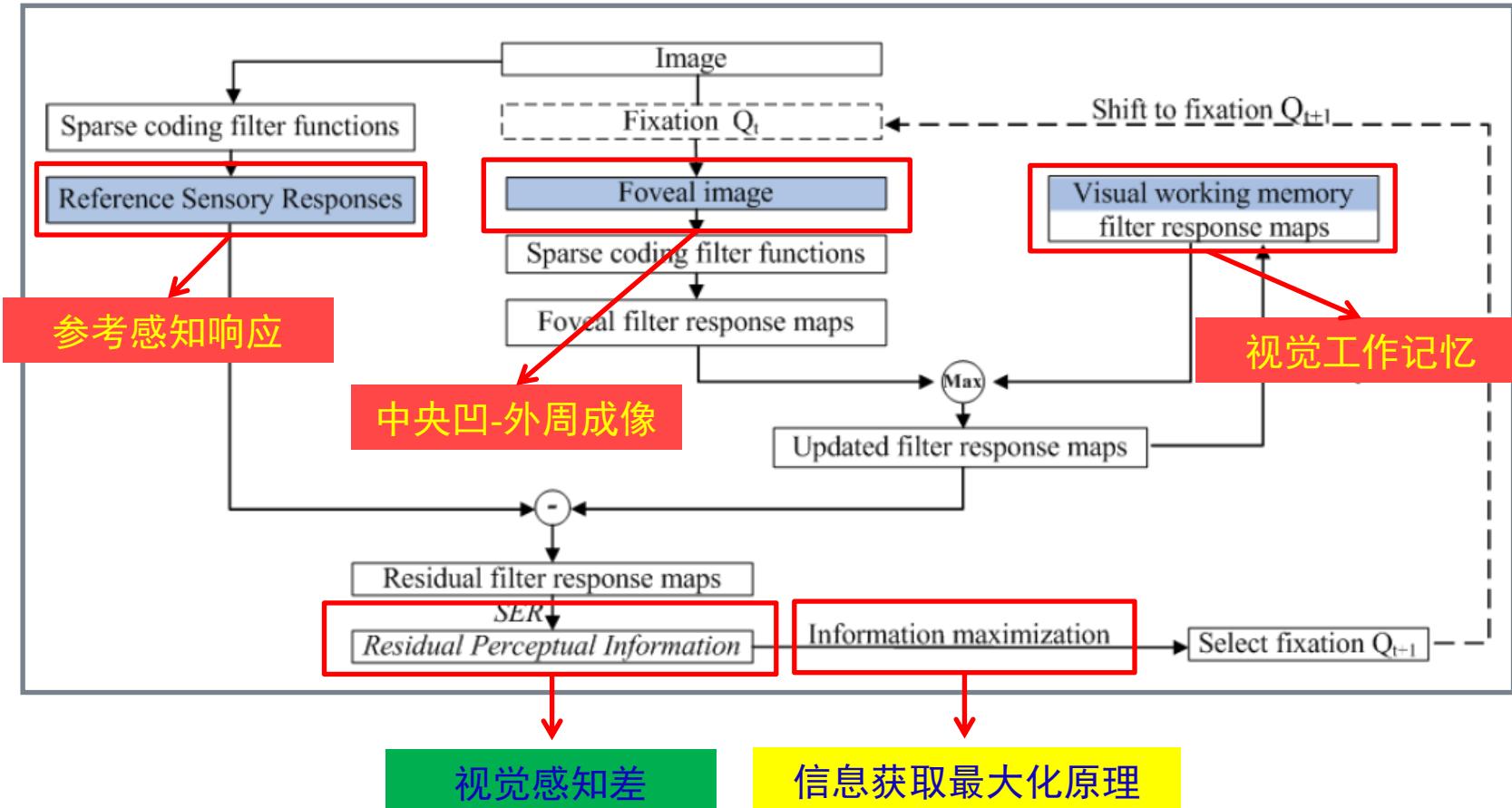


		方法			
		1	2	3	4
问题和应用	1	[3]		[17]	[4]
	2	[55]		[23]	[43]
	3	[1]			[44]
	4				

Dynamic Visual Attention

Computational models:

One principle + One mechanism + Three factors

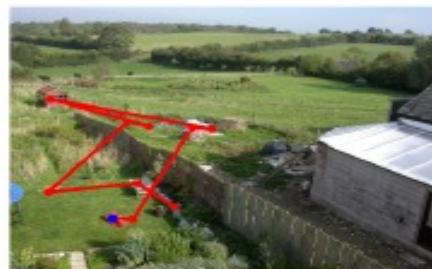


Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, Yuan Yao, Simulating Human Saccadic Scanpath on Natural Images, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Colorado Springs, USA, Jun.21-25, 2011

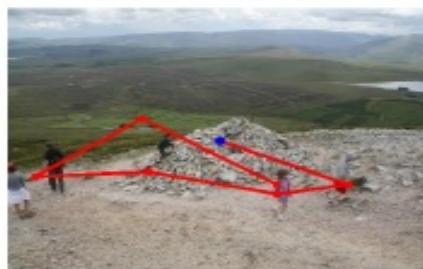
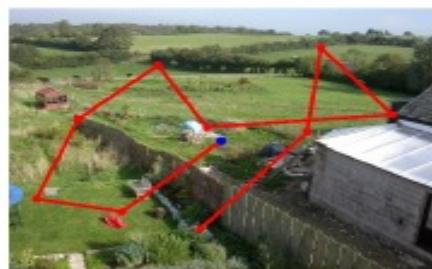
Dynamic Visual Attention



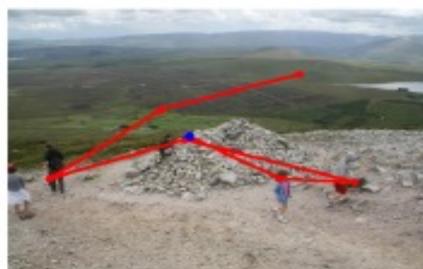
(a) Itti et al. (PAMI 1998)



(b) Wang et al. (CVPR 2010)



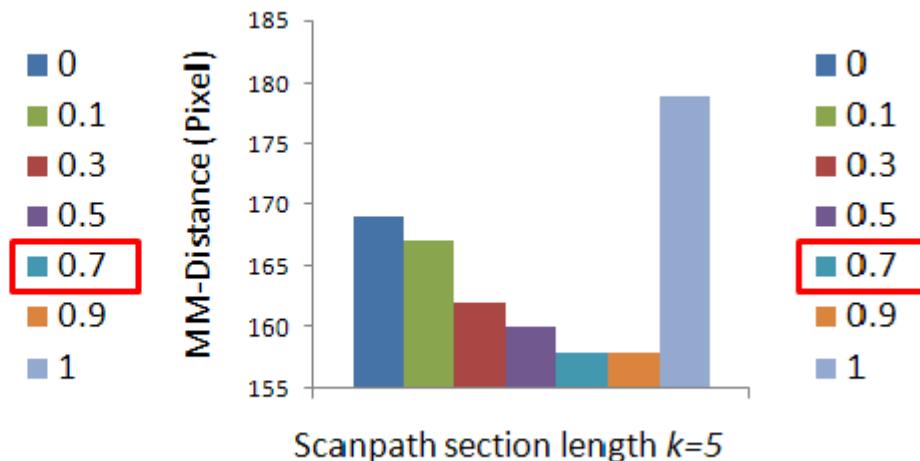
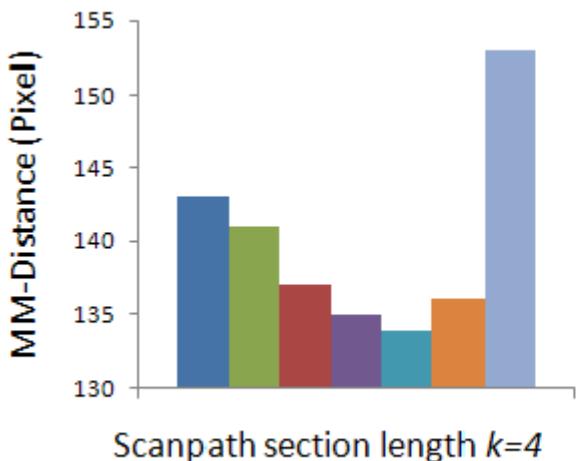
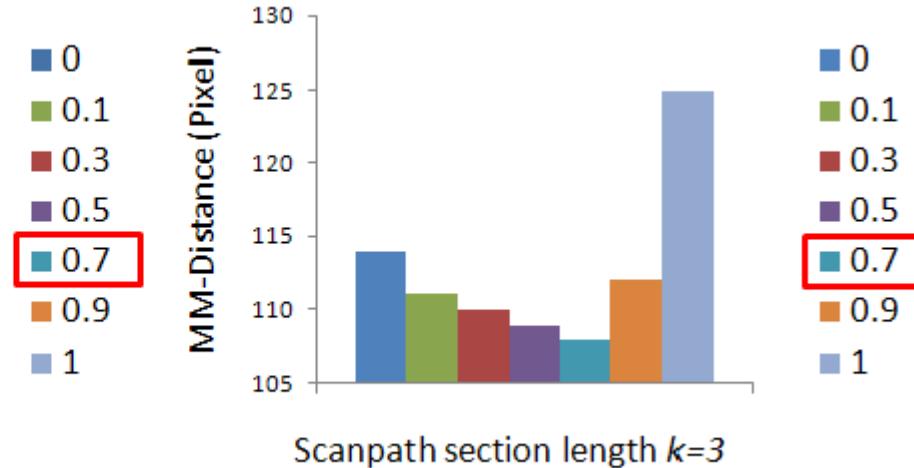
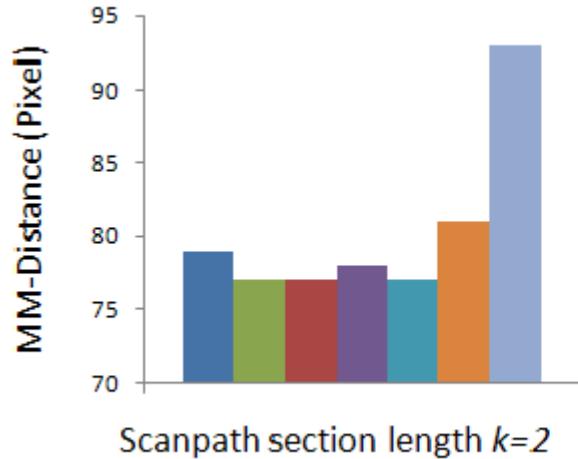
(c) Our model



(d) Eye tracking data

Dynamic Visual Attention

Forgetting factor in working memory



Deep Attention Model

Since 2013/2014

Learning to combine foveal glimpses with a third-order Boltzmann machine

Hugo Larochelle and Geoffrey Hinton

Department of Computer Science, University of Toronto
6 King's College Rd, Toronto, ON, Canada, M5S 3G4
{larocheh,hinton}@cs.toronto.edu

Abstract

We describe a model based on a Boltzmann machine with third-order connections that can learn how to accumulate information about a shape over several fixations. The model uses a retina that only has enough high resolution pixels to cover a small area of the image, so it must decide on a sequence of fixations and it must combine the “glimpse” at each fixation with the location of the fixation before integrating the information with information from other glimpses of the same object. We evaluate this model on a synthetic dataset and two image classification datasets, showing that it can perform at least as well as a model trained on whole images.

1 Introduction

Like insects with unmovable compound eyes, most current computer vision systems use images of uniform resolution. Human vision, by contrast, uses a retina in which the resolution falls off rapidly

On Learning Where To Look

Marc'Aurelio Ranzato

Google Inc.*

Mountain View CA, U.S.A.

2014

arXiv:1405.5488v1 [cs.LG] 24 Apr 2014

arXiv:1406.6247v1 [cs.LG] 24 Jun 2014

On Learning Where To Look

Recurrent Models of Visual Attention

Volodymyr Mnih Nicolas Heess Alex Graves Koray Kavukcuoglu
Google DeepMind

arXiv:1405.5488v1 [cs.LG] 24 Apr 2014

arXiv:1406.6247v1 [cs.CV] 24 Jun 2014

arXiv:1407.3068v2 [cs.CV] 28 Jul 2014

On Learning Where To Look

Recurrent Models of Visual Attention

Deep Networks with Internal Selective Attention
through Feedback Connections

A version of this paper was submitted to ICML 2014 on 31-01-2014.

Marijn Stollenga marijn@idsia.ch^{*},
Jonathan Masci jonathan@idsia.ch^{*},
Faustino Gomez tino@idsia.ch, and
Juergen Schmidhuber juergen@idsia.ch

- They all aim to solve two major challenges
 - scalability
 - variability of appearance

MULTIPLE OBJECT RECOGNITION WITH VISUAL ATTENTION

Jimmy Lei Ba*

University of Toronto

jimmy@psi.utoronto.ca

Volodymyr Mnih

Google DeepMind

vmnih@google.com

Koray Kavukcuoglu

Google DeepMind

korayk@google.com

ABSTRACT

We present an attention-based model for recognizing multiple objects in images. The proposed model is a deep recurrent neural network trained with reinforcement learning to attend to the most relevant regions of the input image. We show that the model learns to both localize and recognize multiple objects despite being given only class labels during training. We evaluate the model on the challenging task of transcribing house number sequences from Google Street View images and show that it is both more accurate than the state-of-the-art convolutional networks and uses fewer parameters and less computation.

MULTIPLE OBJECT RECOGNITION WITH VISUAL ATTENTION

Spatial Transformer Networks

Max Jaderberg Karen Simonyan Andrew Zisserman Koray Kavukcuoglu

Google DeepMind, London, UK
`{jaderberg, simonyan, zisserman, korayk}@google.com`

Abstract

Convolutional Neural Networks define an exceptionally powerful class of models, but are still limited by the lack of ability to be spatially invariant to the input data in a computationally and parameter efficient manner. In this work we introduce a new learnable module, the *Spatial Transformer*, which explicitly allows the spatial manipulation of data within the network. This differentiable module can be inserted into existing convolutional architectures, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimisation process. We show that the use of spatial transformers results in models which learn invariance to translation, scale, rotation and more generic warping, resulting in state-of-the-art performance on several benchmarks, and for a number of classes of transformations.

2015

MULTIPLE OBJECT RECOGNITION WITH VISUAL ATTENTION

Spatial Transformer Networks

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu

Jimmy Lei Ba

Ryan Kiros

Kyunghyun Cho

Aaron Courville

Ruslan Salakhutdinov

Richard S. Zemel

Yoshua Bengio

KELVIN.XU@UMONTREAL.CA

JIMMY@PSI.UTORONTO.CA

RKIROS@CS.TORONTO.EDU

KYUNGHYUN.CHO@UMONTREAL.CA

AARON.COURVILLE@UMONTREAL.CA

RSALAKHU@CS.TORONTO.EDU

ZEMEL@CS.TORONTO.EDU

FIND-ME@THE.WEB

Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention classes of transformations.

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

23 Apr 2015

1 [cs.CV] 5 Jun 2015

11 Feb 2015

Recall: RNN for Captioning

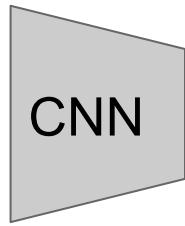


Image:
 $H \times W \times 3$

Recall: RNN for Captioning



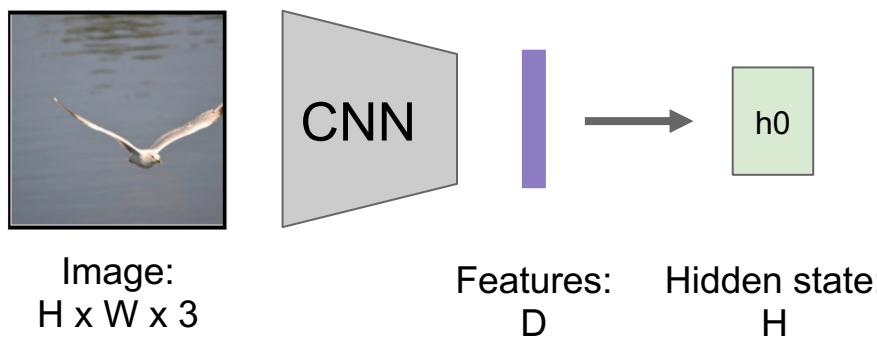
Image:
 $H \times W \times 3$



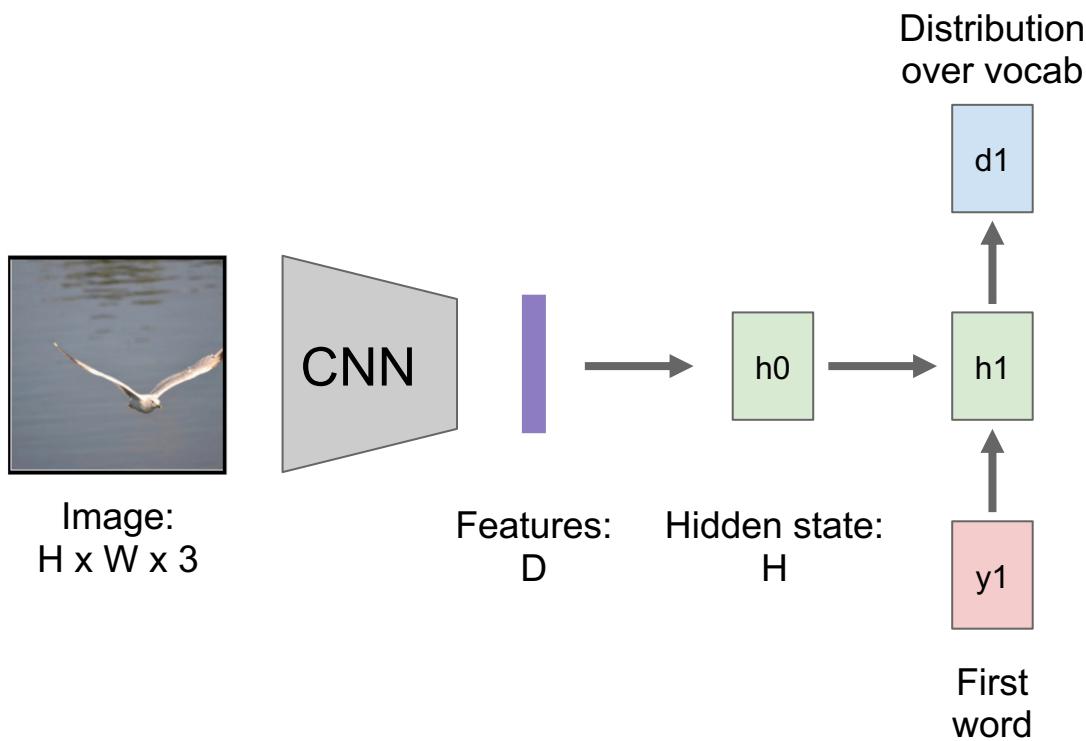
Features:
 D



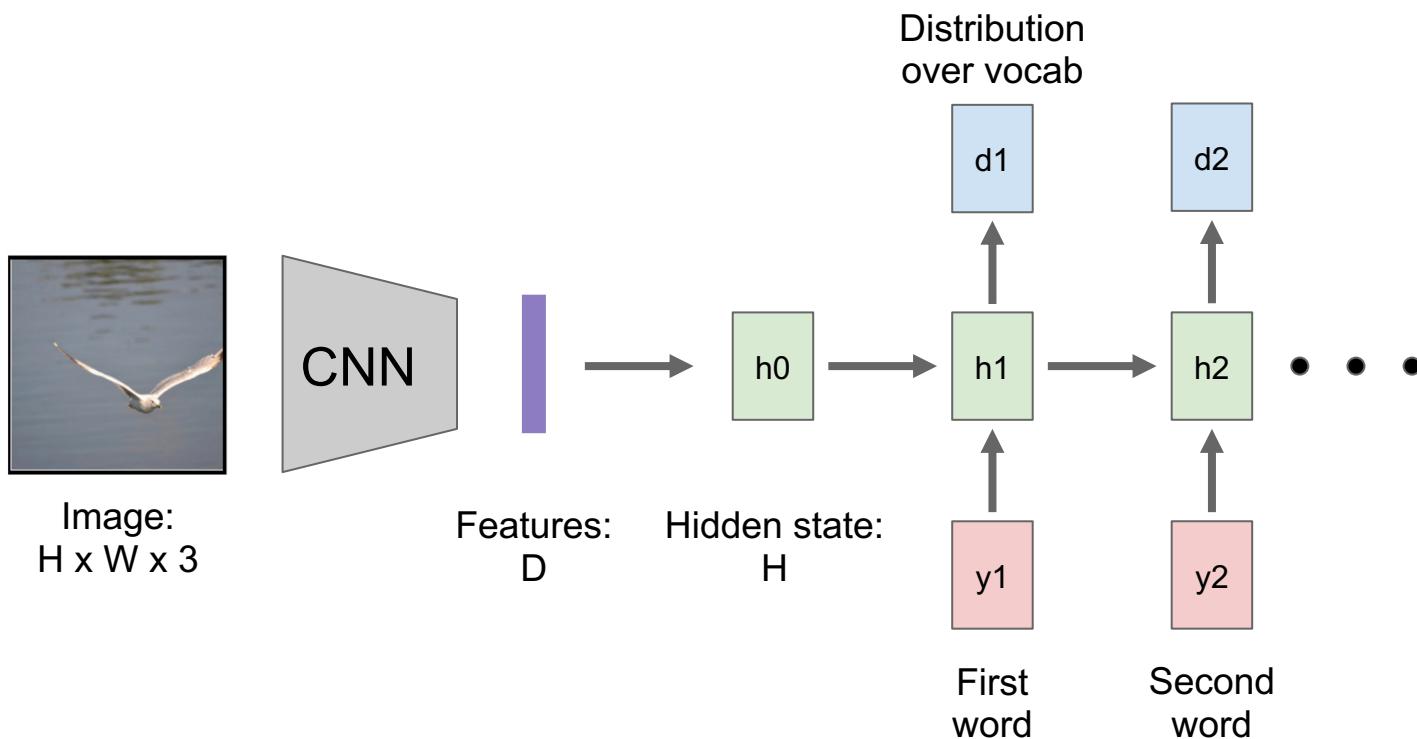
Recall: RNN for Captioning



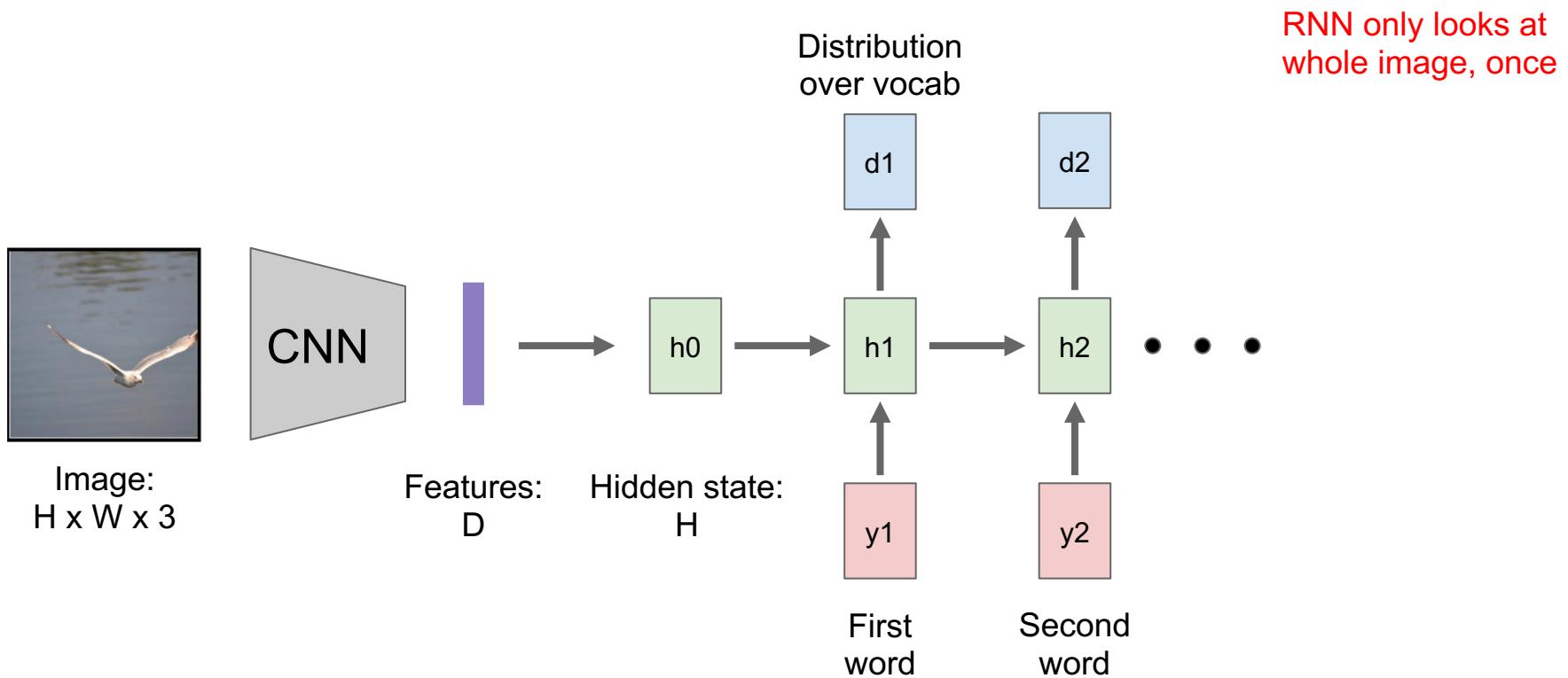
Recall: RNN for Captioning



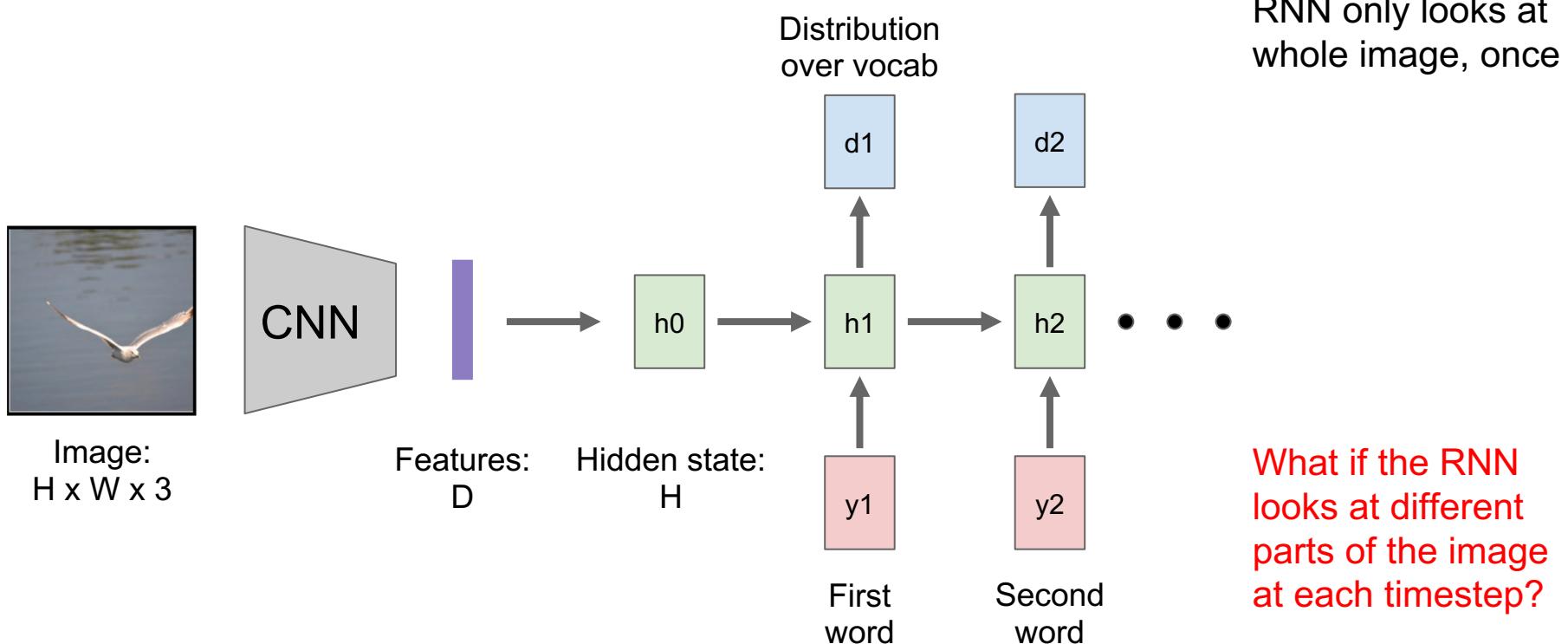
Recall: RNN for Captioning



Recall: RNN for Captioning



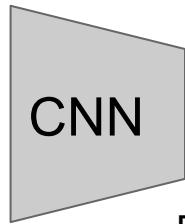
Recall: RNN for Captioning



Soft Attention for Captioning



Image:
 $H \times W \times 3$

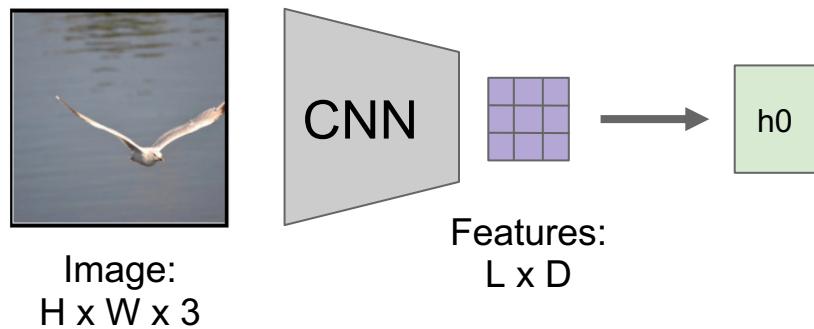


Features:
 $L \times D$



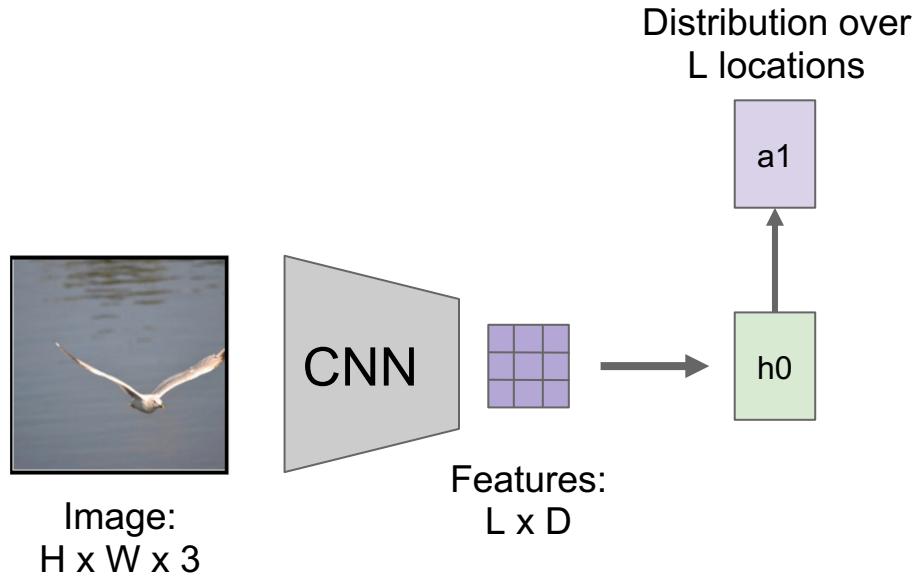
Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning



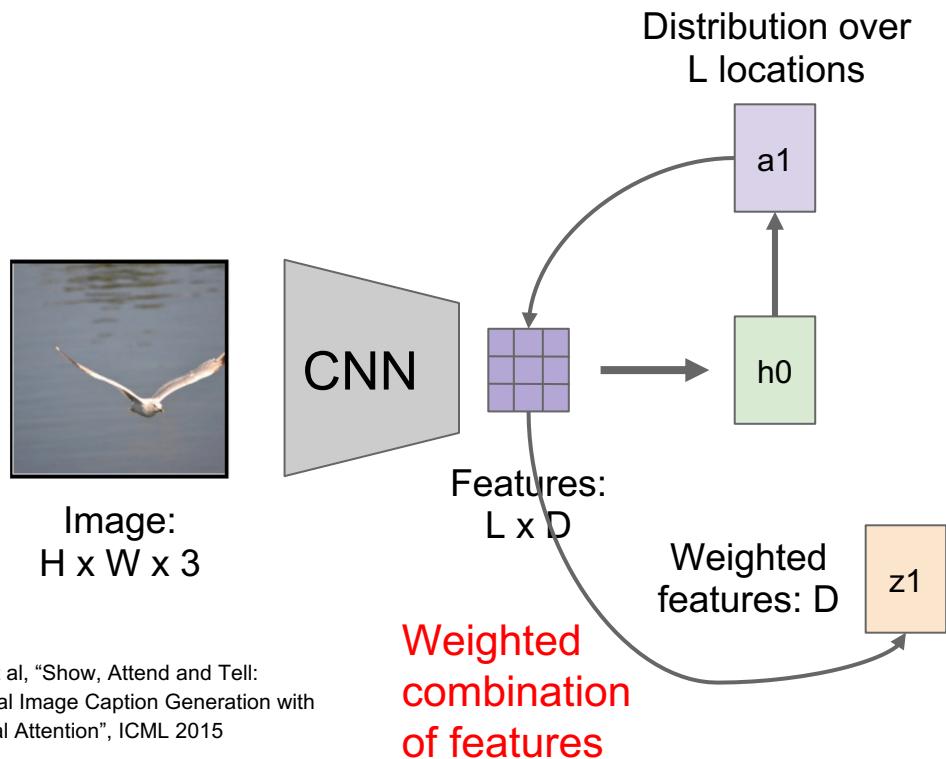
Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning



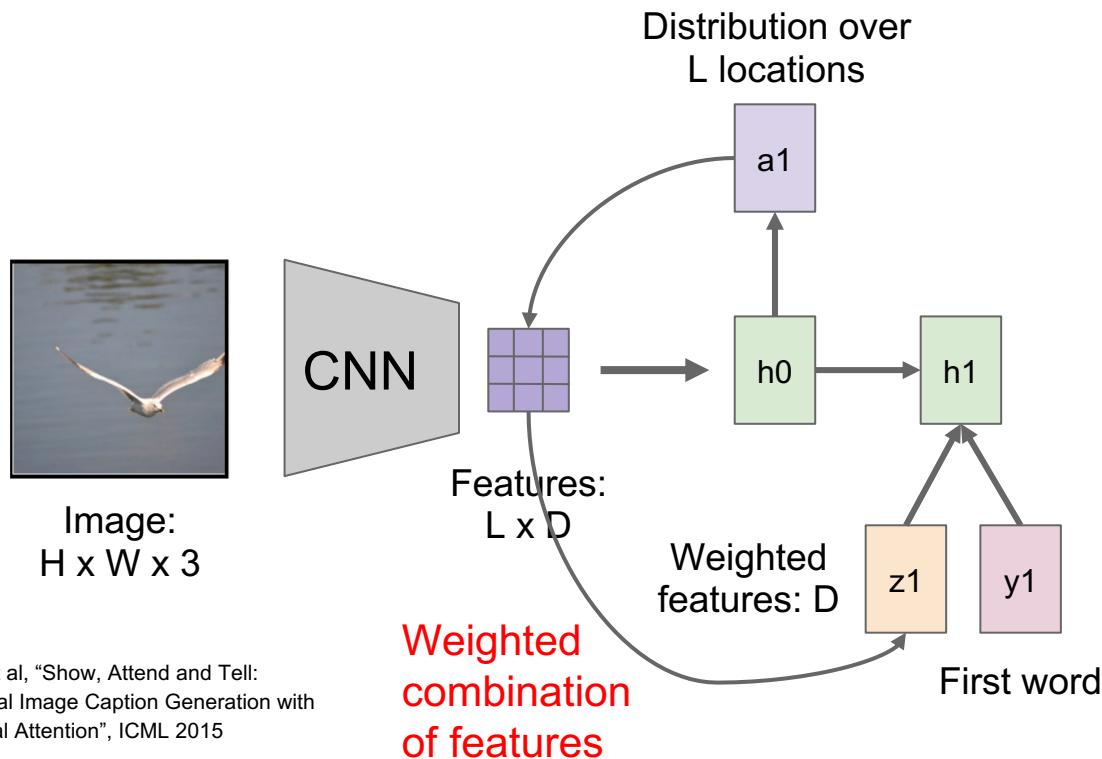
Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning



Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning

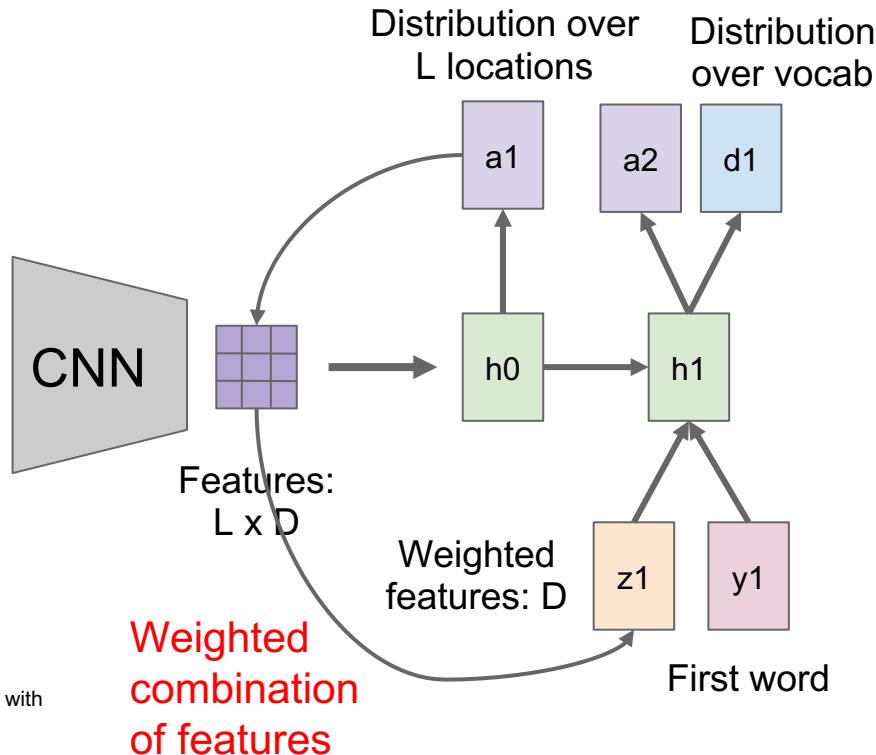


Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning



Image:
 $H \times W \times 3$

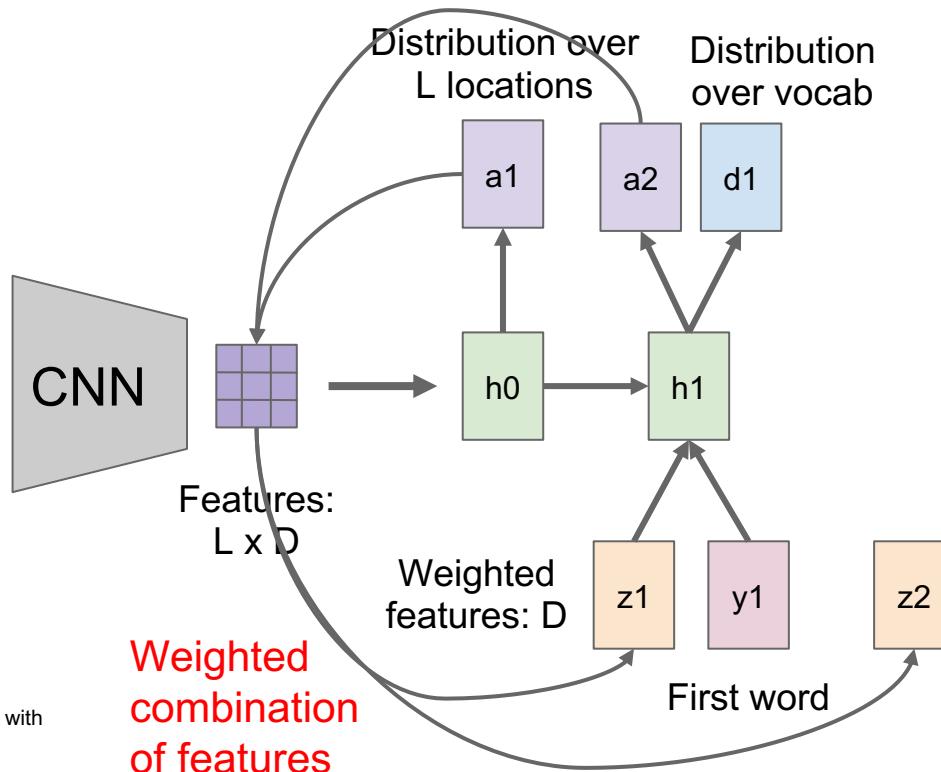


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Soft Attention for Captioning



Image:
 $H \times W \times 3$

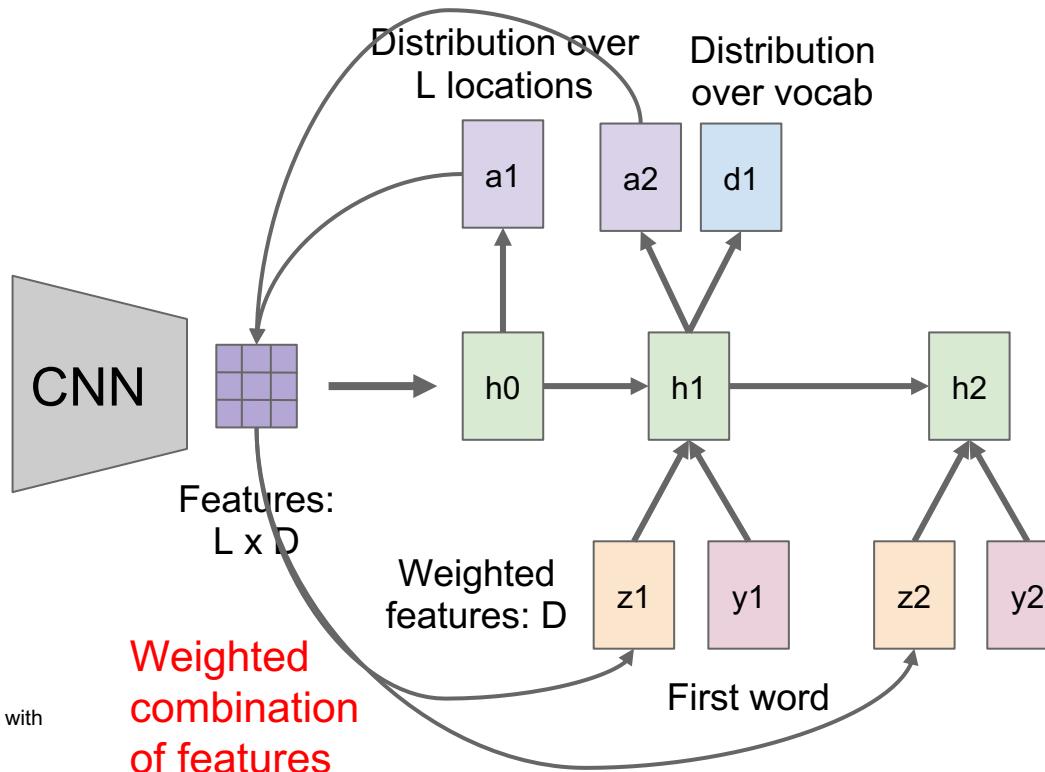


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Soft Attention for Captioning

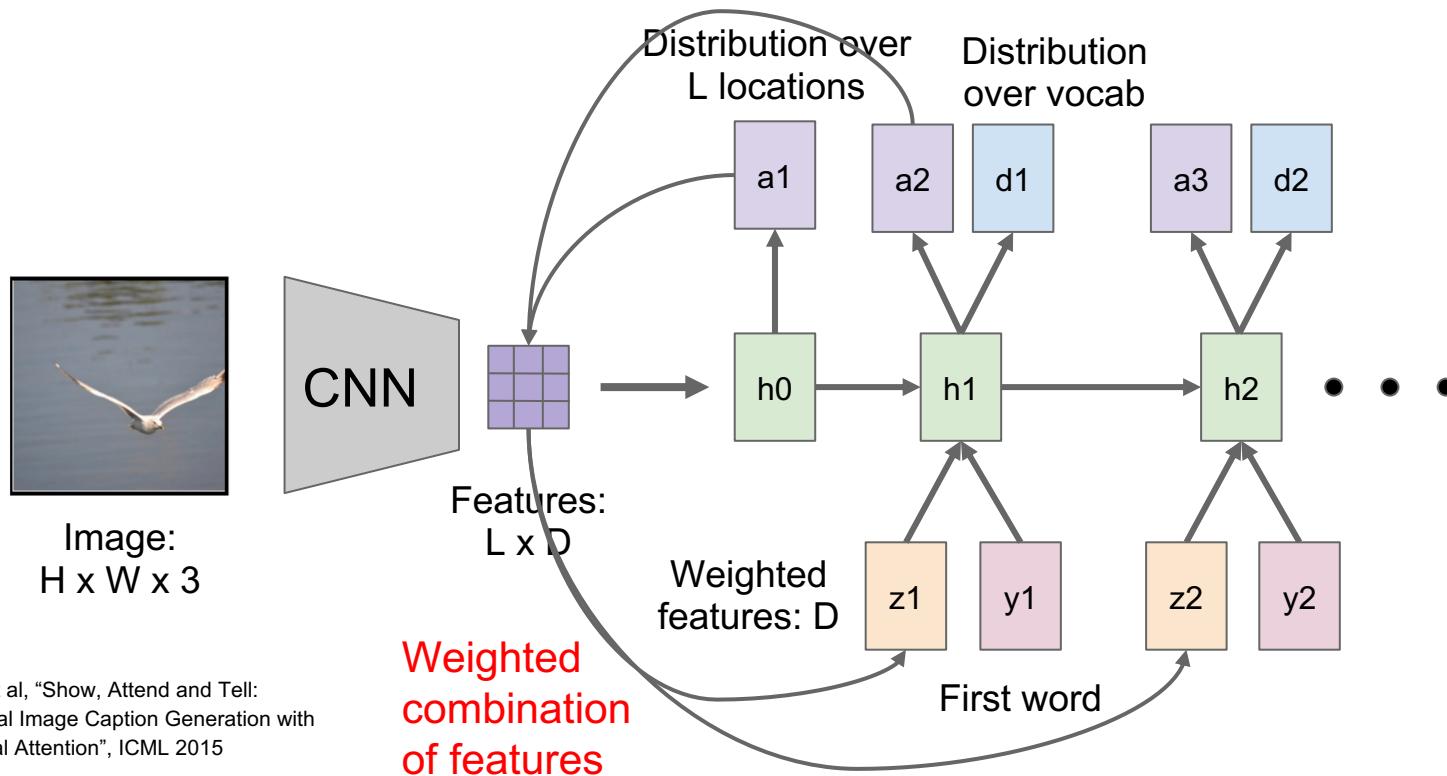


Image:
 $H \times W \times 3$



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Soft Attention for Captioning



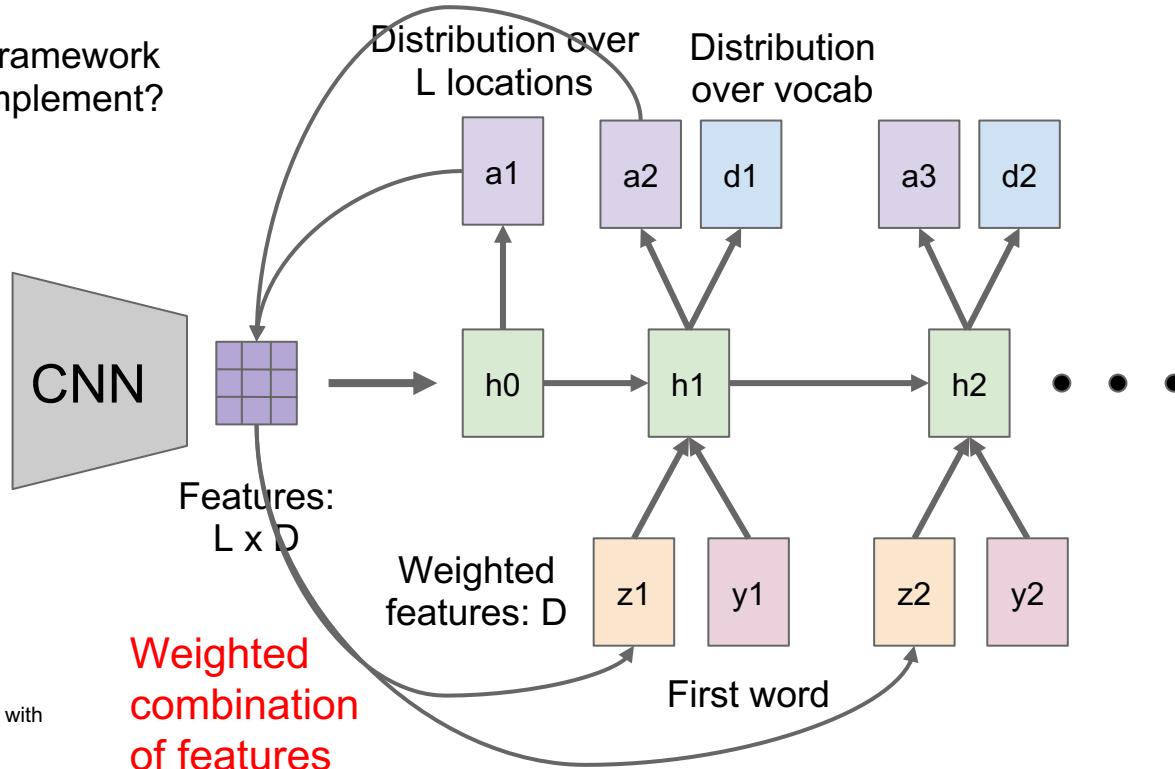
Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning

Guess which framework was used to implement?



Image:
 $H \times W \times 3$



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

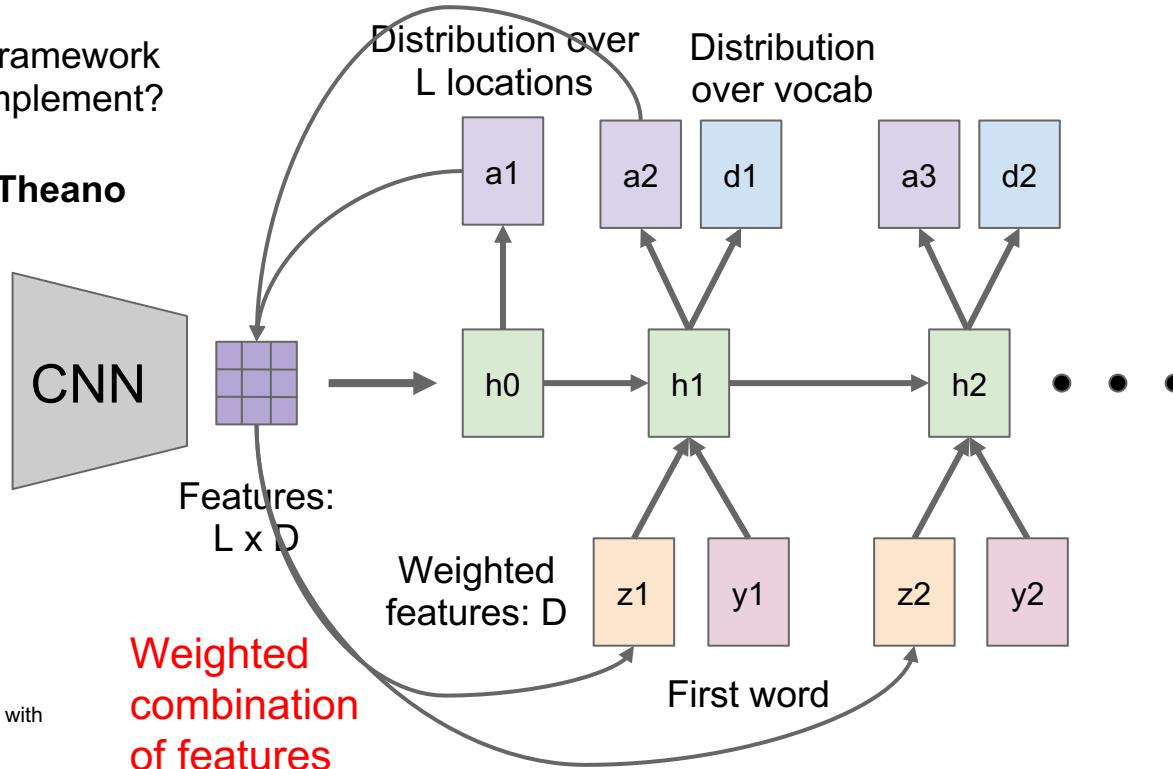
Soft Attention for Captioning

Guess which framework was used to implement?

Crazy RNN = **Theano**



Image:
 $H \times W \times 3$

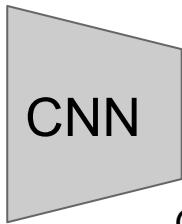


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Soft vs Hard Attention



Image:
 $H \times W \times 3$



Grid of features
(Each D-dimensional)

a	b
c	d

From
RNN:

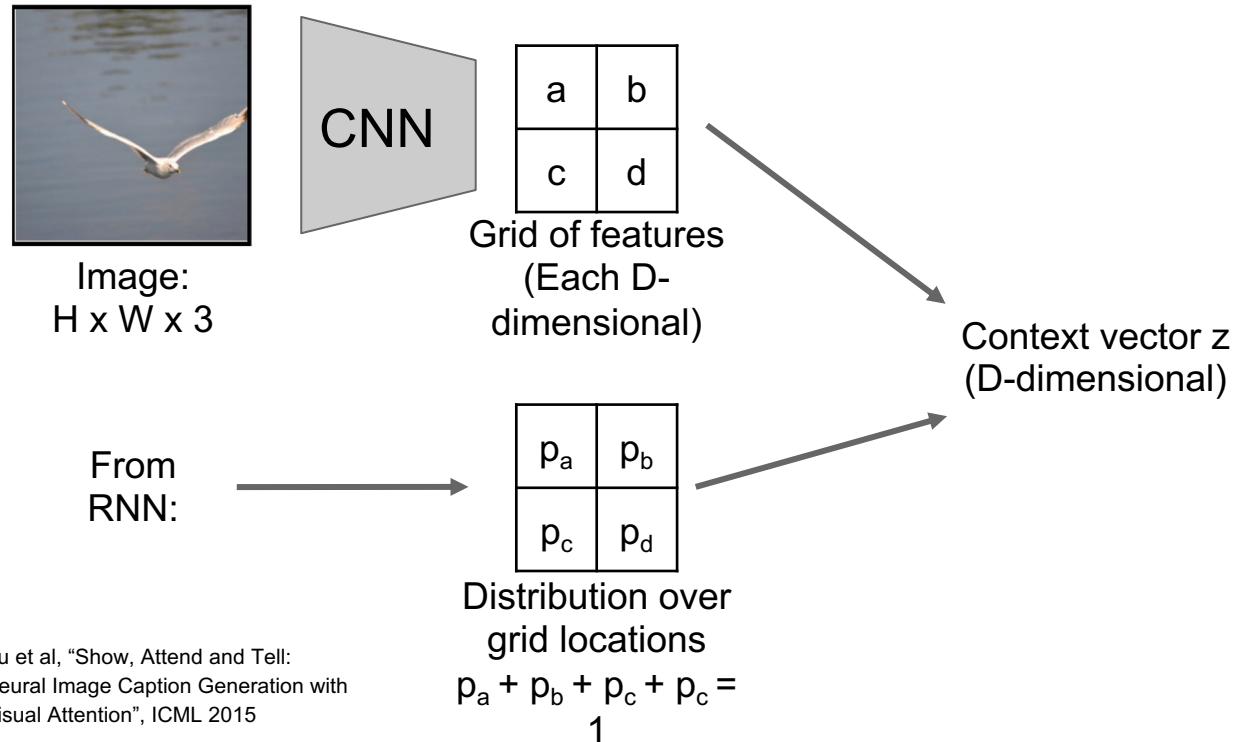


p_a	p_b
p_c	p_d

Distribution over
grid locations

$$p_a + p_b + p_c + p_d = 1$$

Soft vs Hard Attention

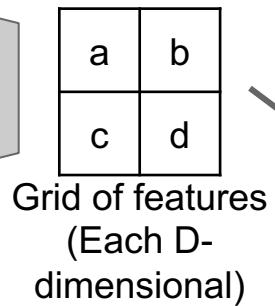
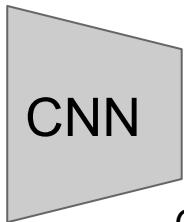


Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

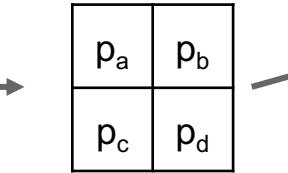
Soft vs Hard Attention



Image:
 $H \times W \times 3$



From
RNN:



Distribution over
grid locations

$$p_a + p_b + p_c + p_d = 1$$

Context vector z
(D-dimensional)

Soft attention:

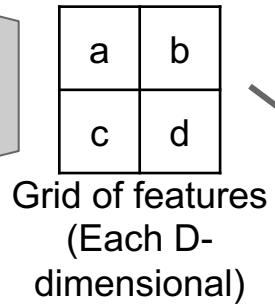
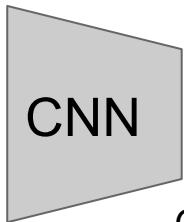
Summarize ALL locations
$$z = p_a a + p_b b + p_c c + p_d d$$

Derivative dz/dp is nice!
Train with gradient descent

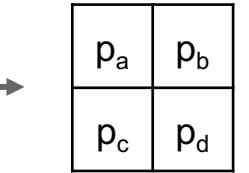
Soft vs Hard Attention



Image:
 $H \times W \times 3$



From
RNN:



Distribution over
grid locations

$$p_a + p_b + p_c + p_d = 1$$

Context vector z
(D-dimensional)

Soft attention:

Summarize ALL locations
$$z = p_a a + p_b b + p_c c + p_d d$$

Derivative dz/dp is nice!
Train with gradient descent

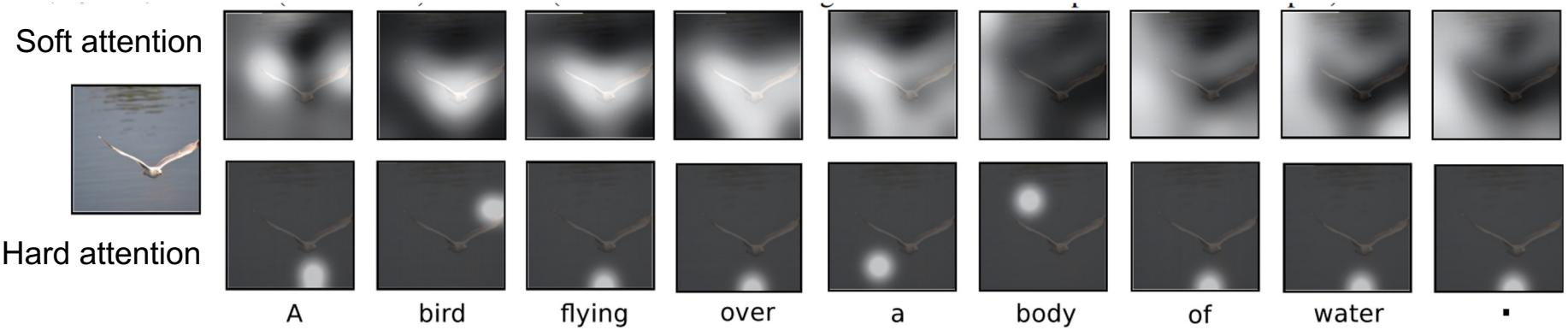
Hard attention:

Sample ONE location
according to p , $z = \text{that vector}$

With argmax, dz/dp is zero
almost everywhere ...

Can't use gradient descent;
need reinforcement learning

Soft Attention for Captioning



Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning



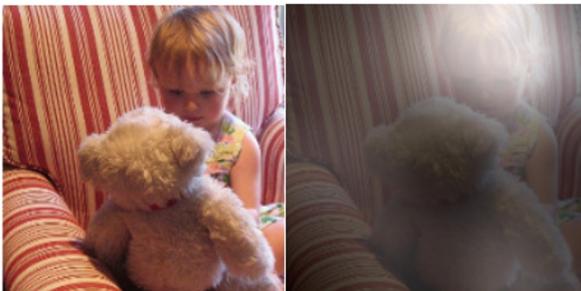
A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.

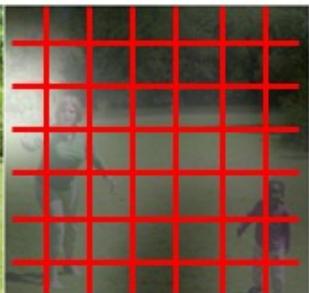
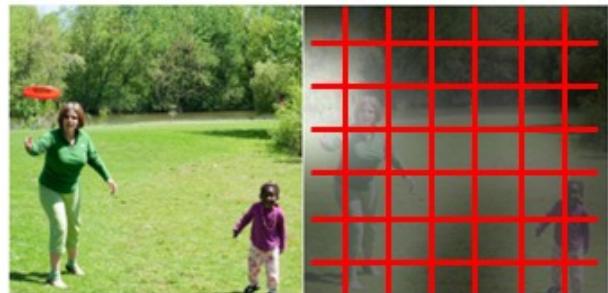


A giraffe standing in a forest with trees in the background.

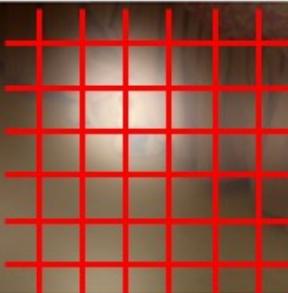
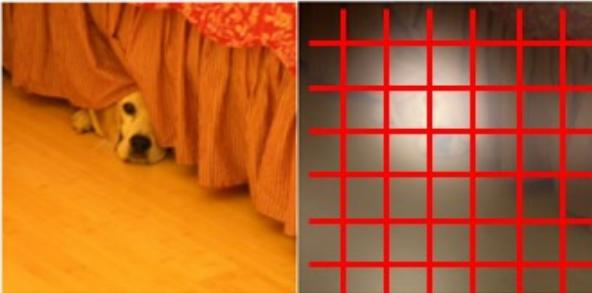
Xu et al, "Show, Attend and Tell:
Neural Image Caption Generation with
Visual Attention", ICML 2015

Soft Attention for Captioning

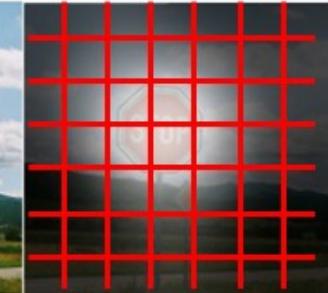
Attention constrained to fixed grid! We'll come back to this



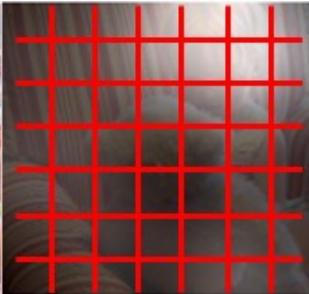
A woman is throwing a frisbee in a park.



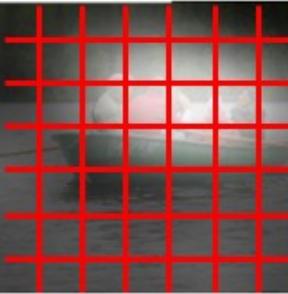
A dog is standing on a hardwood floor.



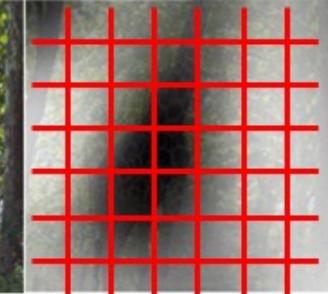
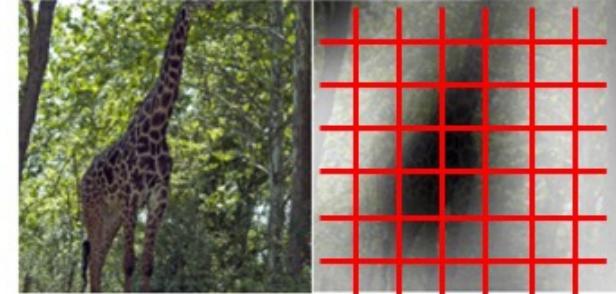
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.

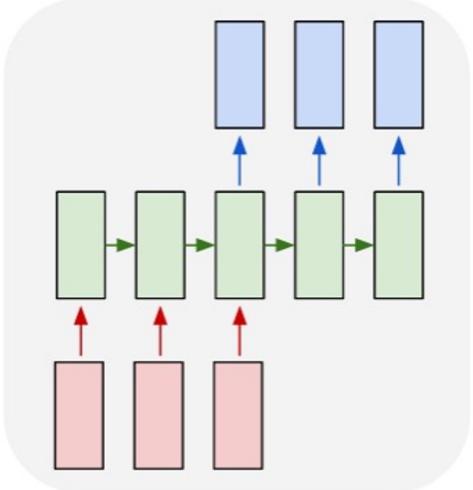


A giraffe standing in a forest with trees in the background.

Soft Attention for Translation

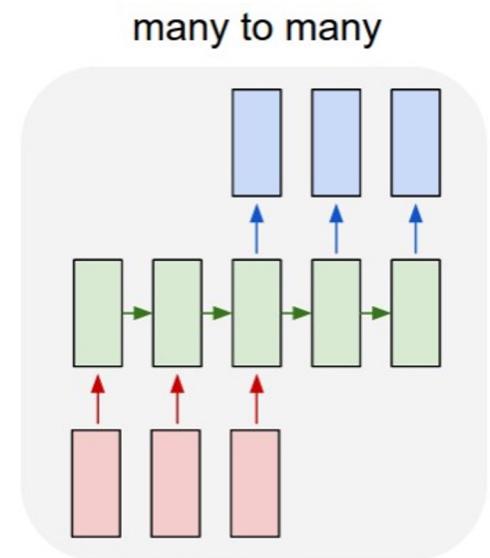
“Mi gato es el mejor” -> “My cat is the best”

many to many



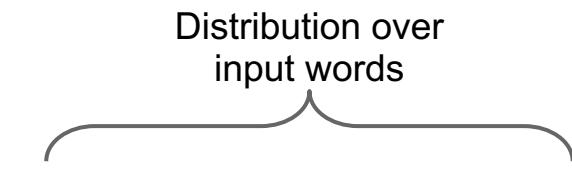
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Soft Attention for Translation



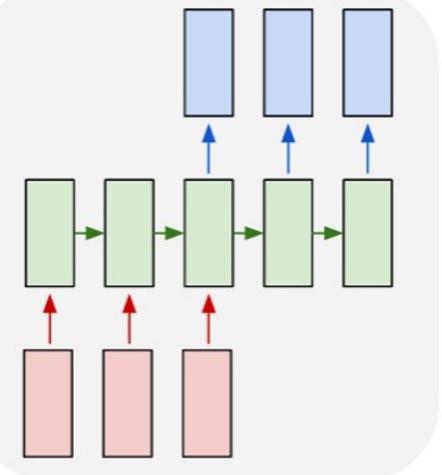
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Soft Attention for Translation



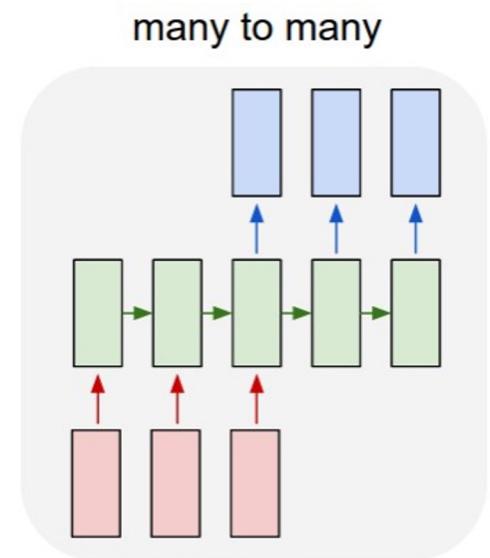
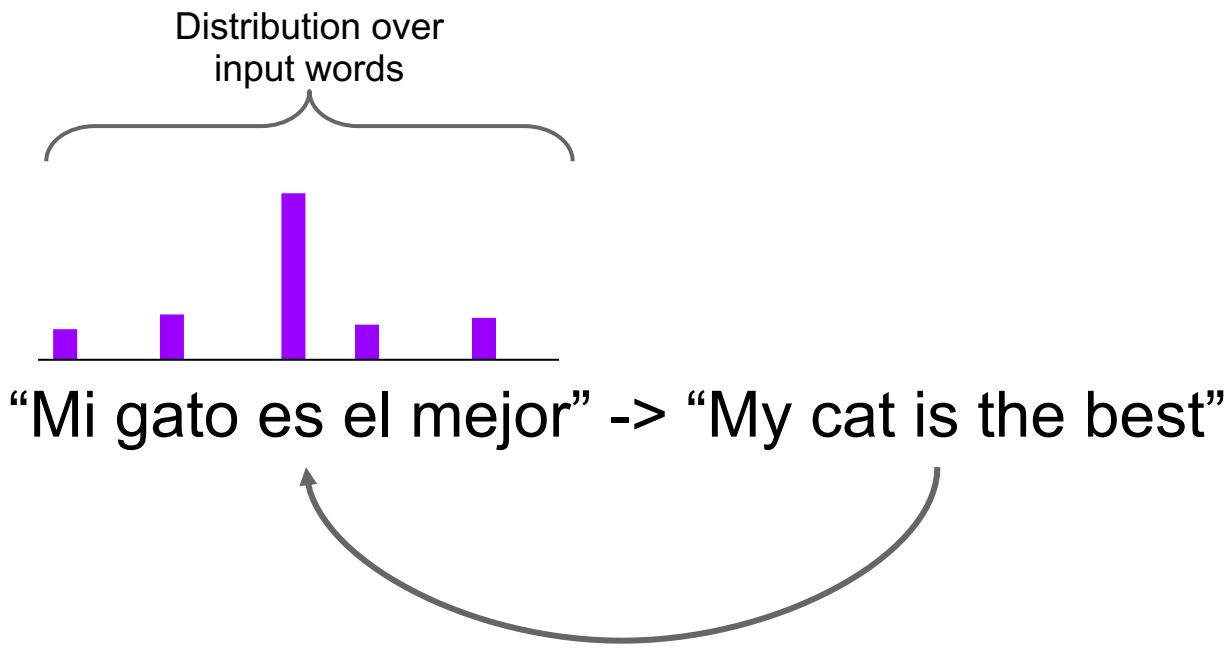
“Mi gato es el mejor” -> “My cat is the best”

many to many



Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Soft Attention for Translation



Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Soft Attention for Everything!

Machine Translation, attention over input:

- Luong et al, "Effective Approaches to Attention-based Neural Machine Translation," EMNLP 2015



Video captioning, attention over input frames:

- Yao et al, "Describing Videos by Exploiting Temporal Structure", ICCV 2015

Speech recognition, attention over input sounds:

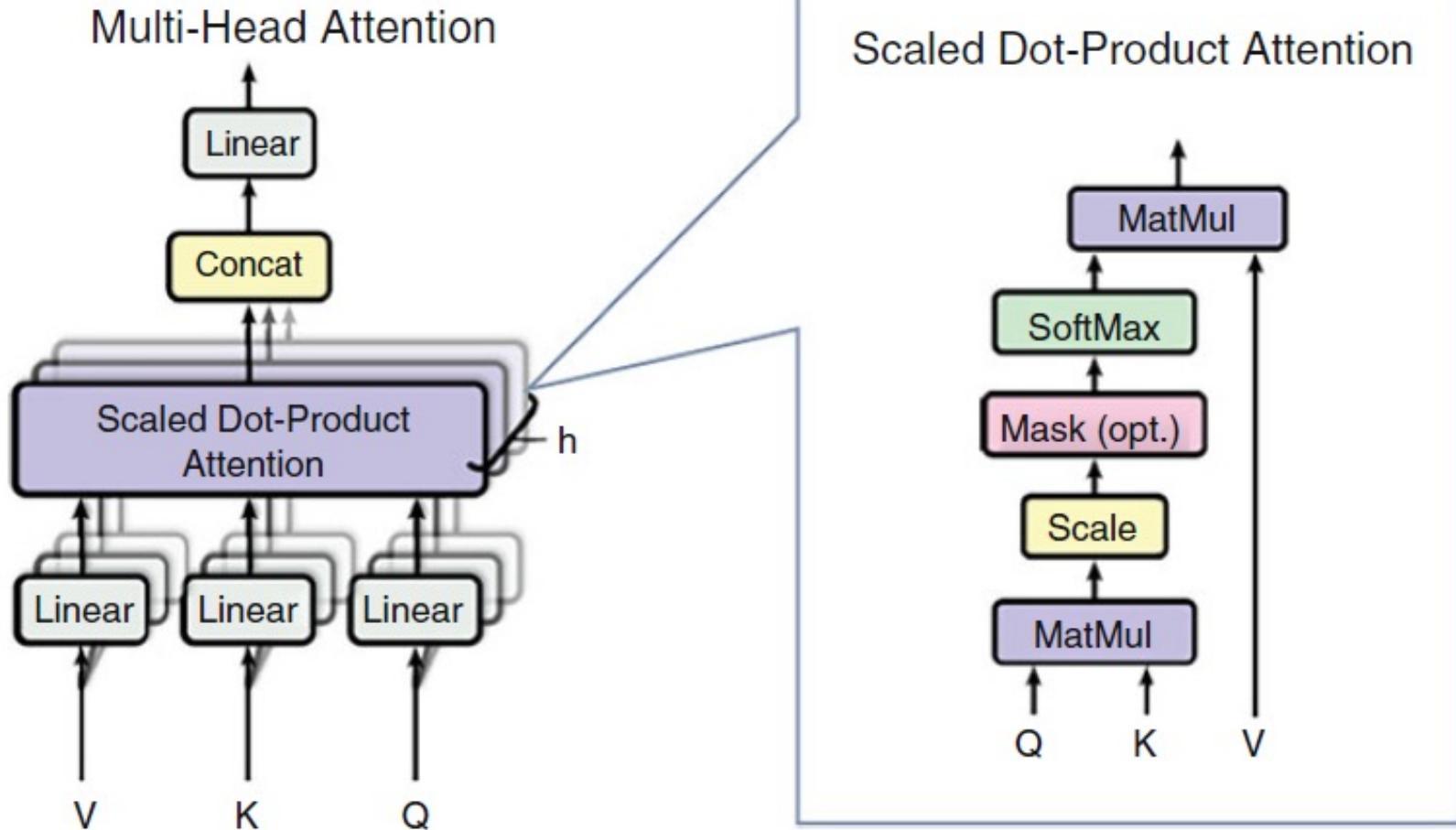
- Chan et al, "Listen, Attend, and Spell", arXiv 2015
- Chorowski et al, "Attention-based models for Speech Recognition", NIPS 2015



Image, question to answer, attention over image:

- Xu and Saenko, "Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering", arXiv 2015
- Zhu et al, "Visual7W: Grounded Question Answering in Images", arXiv 2015

Self Attention

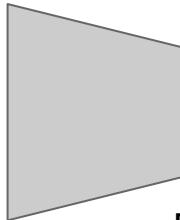


Attending to Arbitrary Regions?

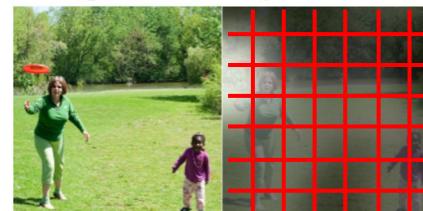
Attention mechanism from Show, Attend, and Tell only lets us softly attend to **fixed grid positions** ... can we do better?



Image:
 $H \times W \times 3$



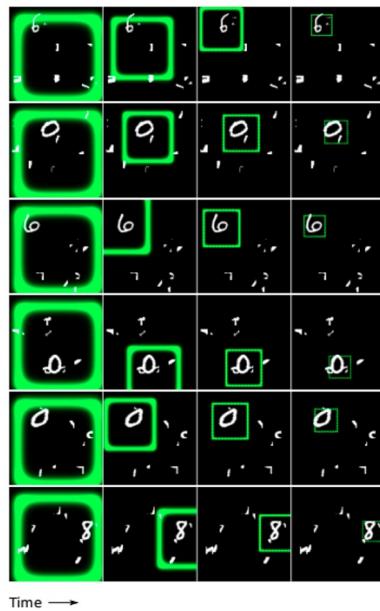
Features:
 $L \times D$



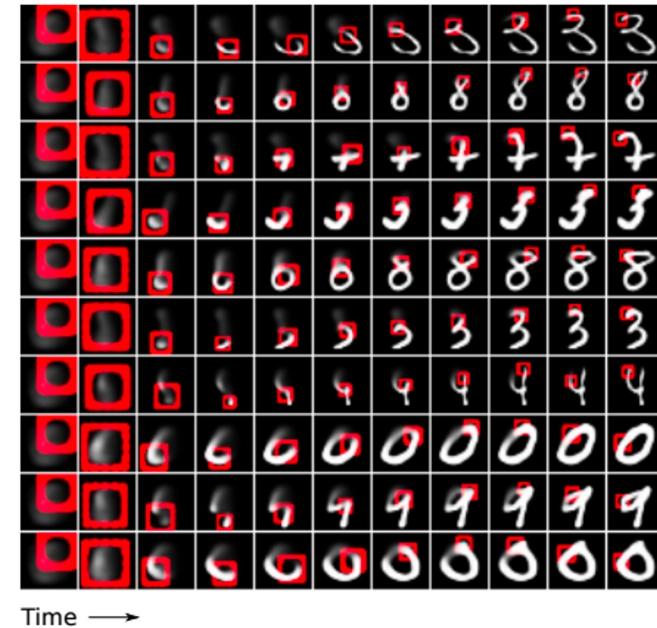
A woman is throwing a frisbee in a park.

Attending to Arbitrary Regions: DRAW

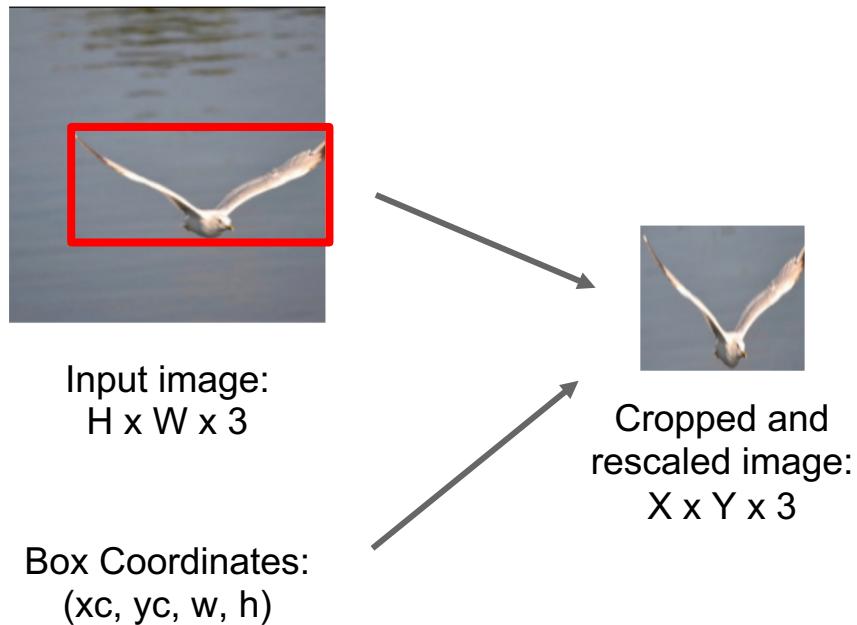
Classify images by attending to arbitrary regions of the *input*



Generate images by attending to arbitrary regions of the *output*



Spatial Transformer Networks



Jaderberg et al, "Spatial Transformer Networks", NIPS 2015

Spatial Transformer Networks



Can we make this
function differentiable?

Input image:
 $H \times W \times 3$

Box Coordinates:
 (x_c, y_c, w, h)



Cropped and
rescaled image:
 $X \times Y \times 3$

Spatial Transformer Networks



Input image:
 $H \times W \times 3$

Box Coordinates:
 (x_c, y_c, w, h)

Can we make this
function differentiable?

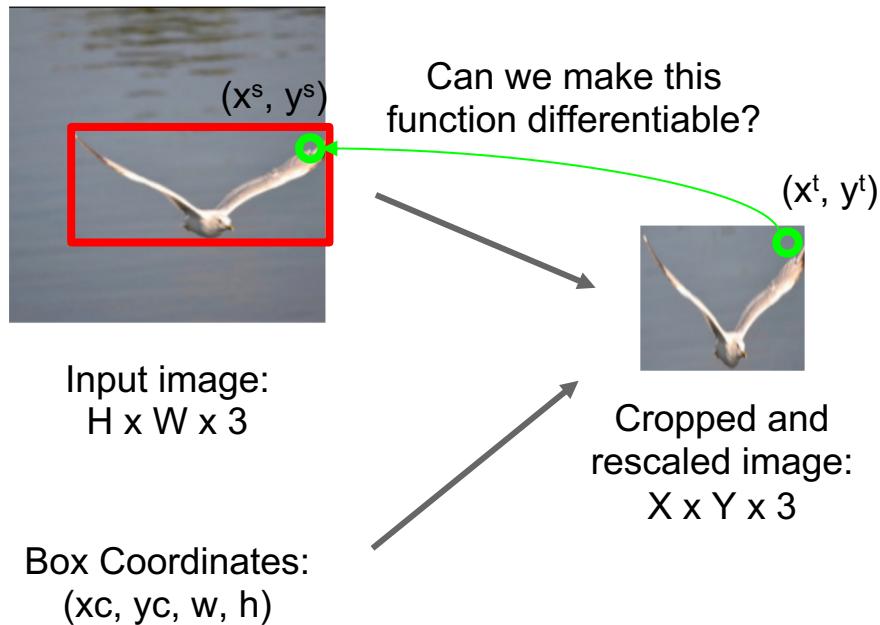


Cropped and
rescaled image:
 $X \times Y \times 3$

Idea: Function mapping
pixel coordinates (x_t, y_t) of
output to *pixel coordinates*
(x_s, y_s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

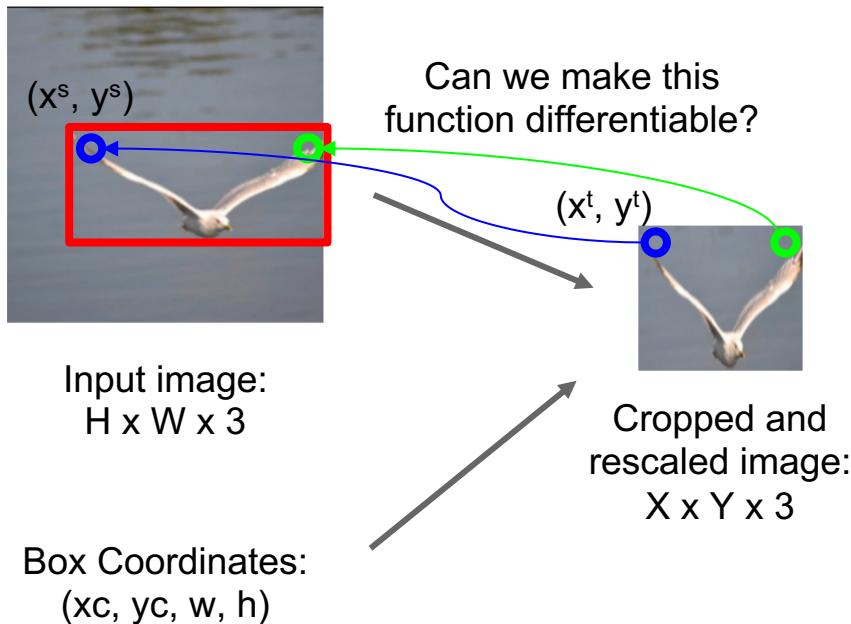
Spatial Transformer Networks



Idea: Function mapping *pixel coordinates* (x^t, y^t) of output to *pixel coordinates* (x^s, y^s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

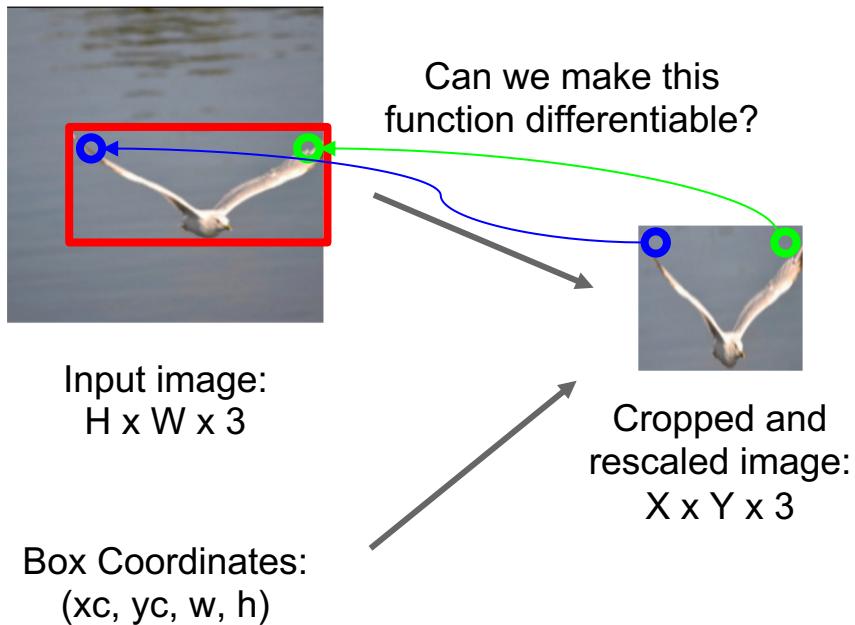
Spatial Transformer Networks



Idea: Function mapping *pixel coordinates* (xt, yt) of output to *pixel coordinates* (xs, ys) of input

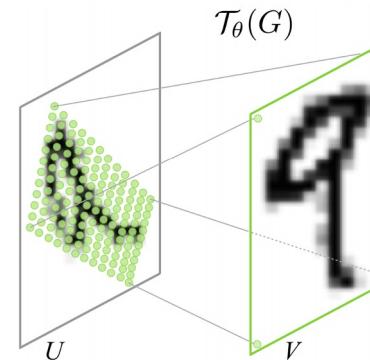
$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

Spatial Transformer Networks



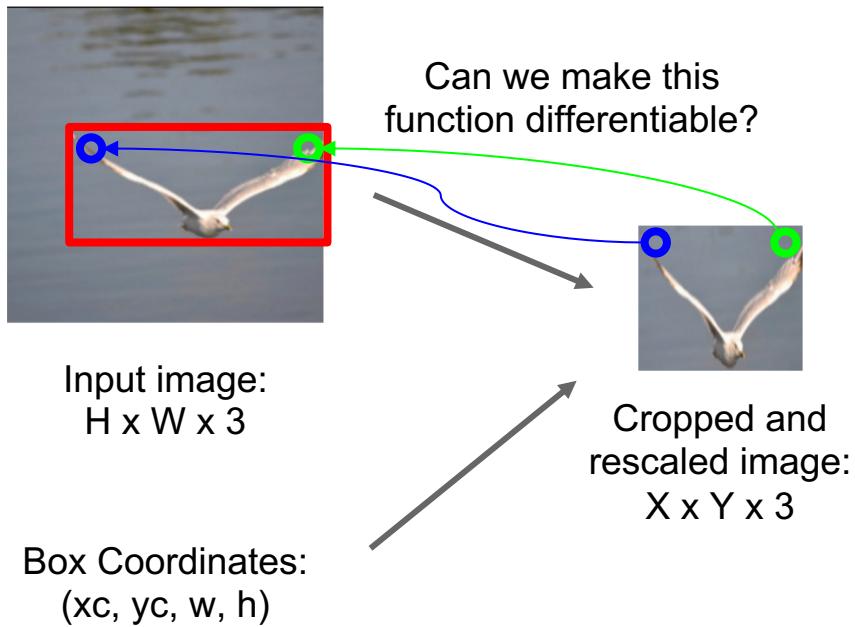
Idea: Function mapping *pixel coordinates* (x_t, y_t) of output to *pixel coordinates* (x_s, y_s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



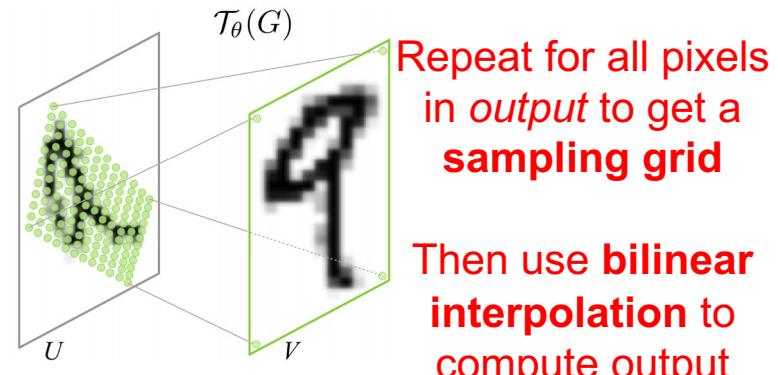
Repeat for all pixels in *output* to get a **sampling grid**

Spatial Transformer Networks

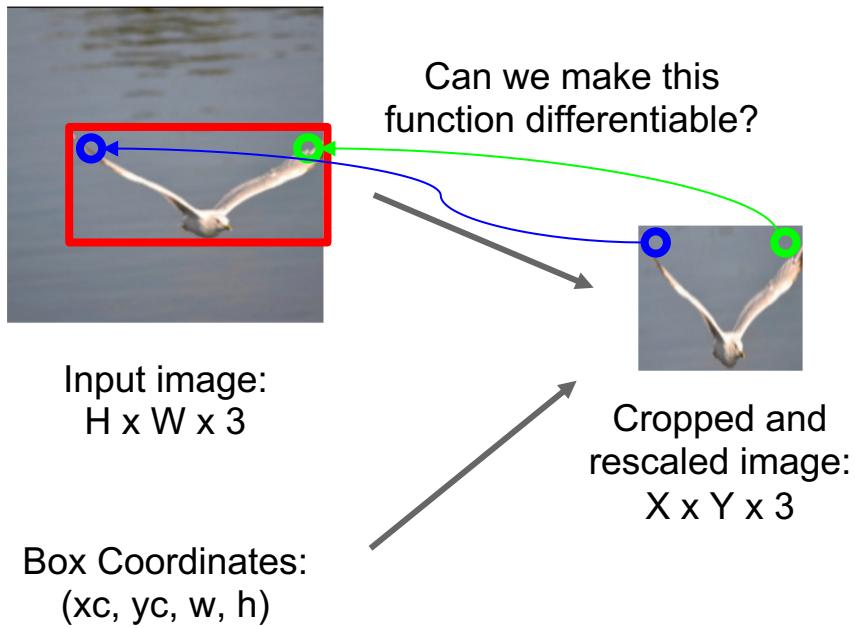


Idea: Function mapping *pixel coordinates* (x_t, y_t) of output to *pixel coordinates* (x_s, y_s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

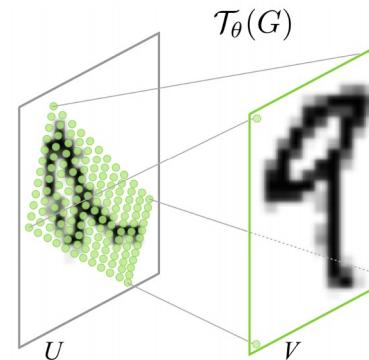


Spatial Transformer Networks



Idea: Function mapping *pixel coordinates* (x^t, y^t) of output to *pixel coordinates* (x^s, y^s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

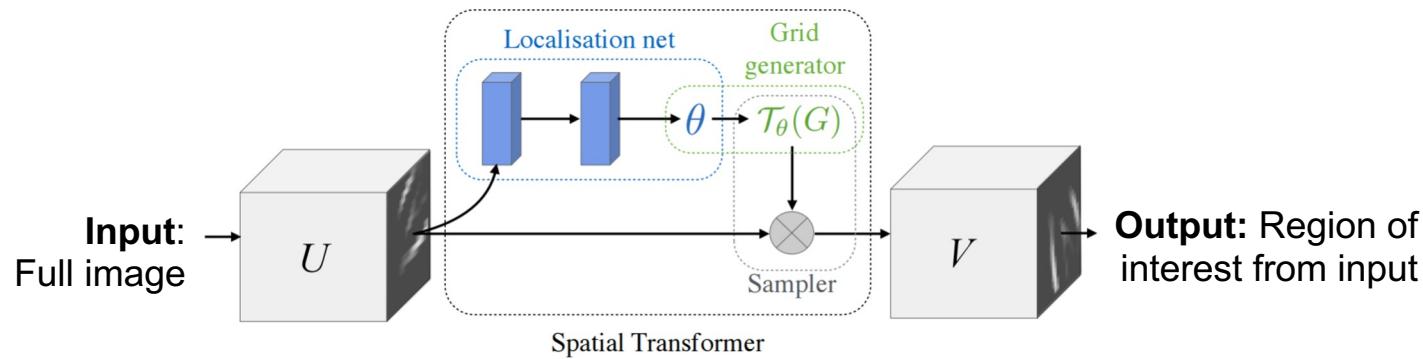


Repeat for all pixels in *output* to get a **sampling grid**

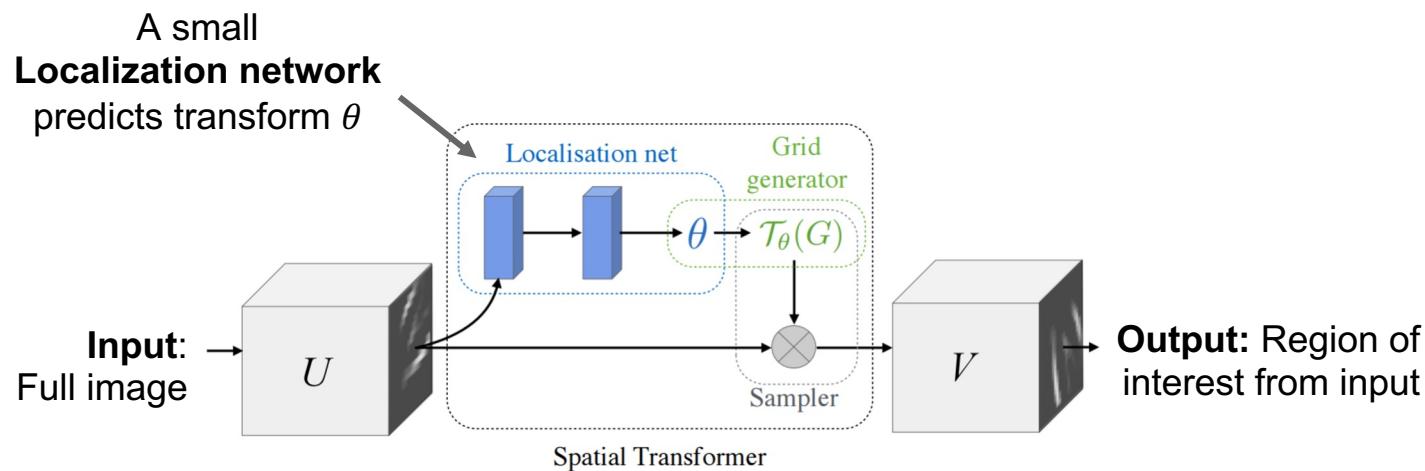
Then use **bilinear interpolation** to compute output

Network attends to input by predicting θ

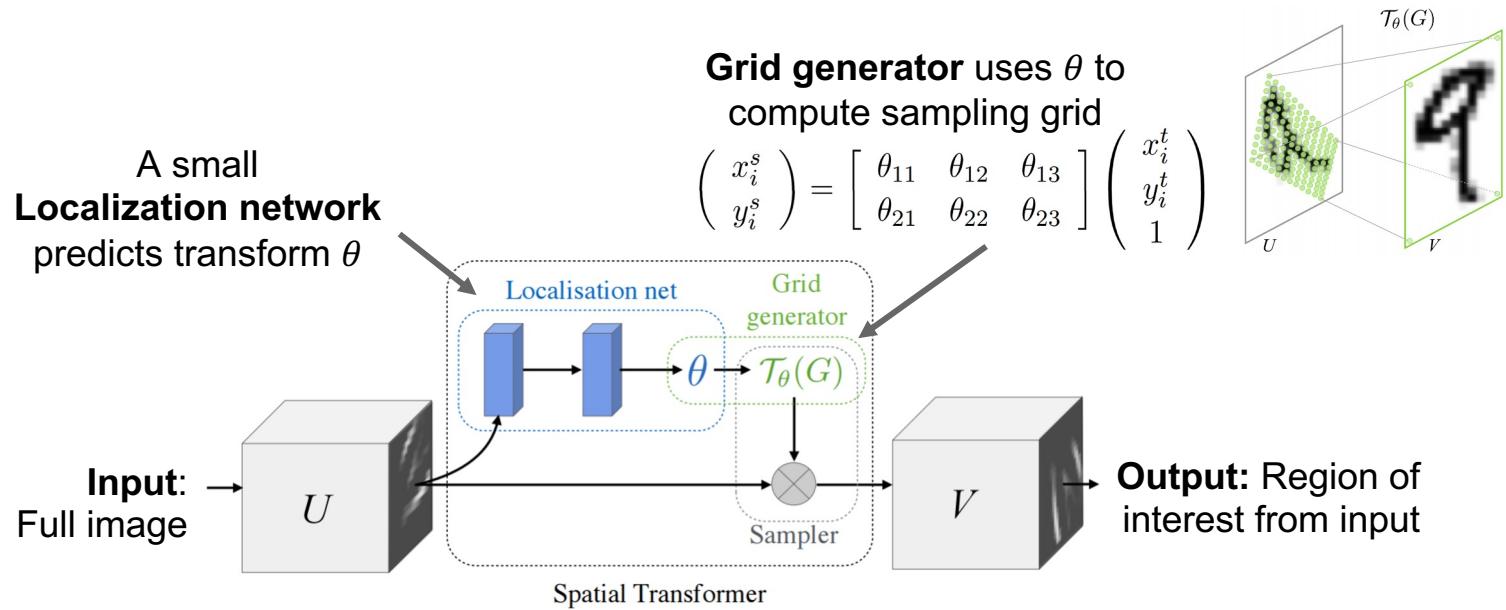
Spatial Transformer Networks



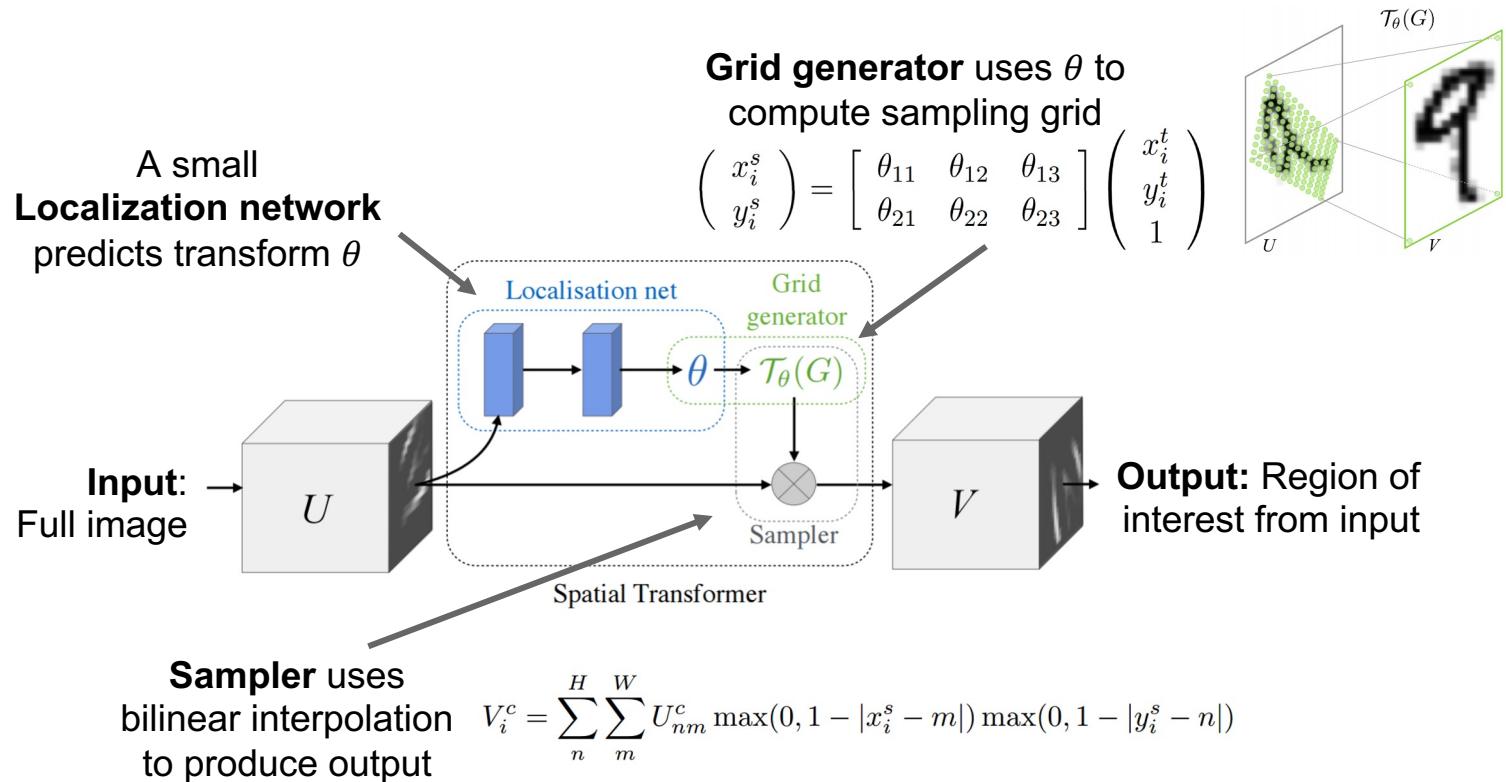
Spatial Transformer Networks



Spatial Transformer Networks

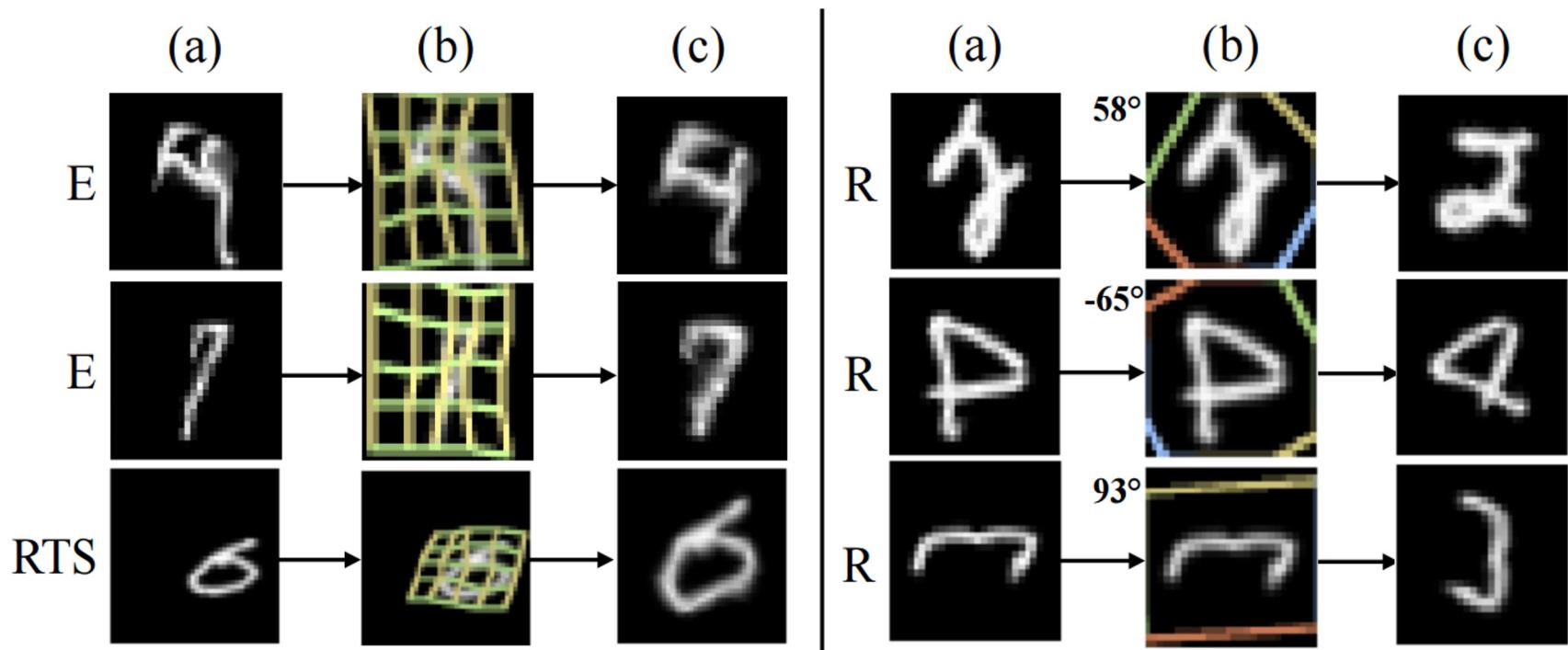


Spatial Transformer Networks



Spatial Transformer Networks

Insert spatial transformers into a classification network
and it **learns to attend and transform the input**



Attention Recap

- Soft attention:
 - Easy to implement: produce distribution over input locations, reweight features and feed as input
 - Attend to arbitrary input locations using spatial transformer networks
- Hard attention:
 - Attend to a single input location
 - Can't use gradient descent!
 - Need **reinforcement learning!**

Outline

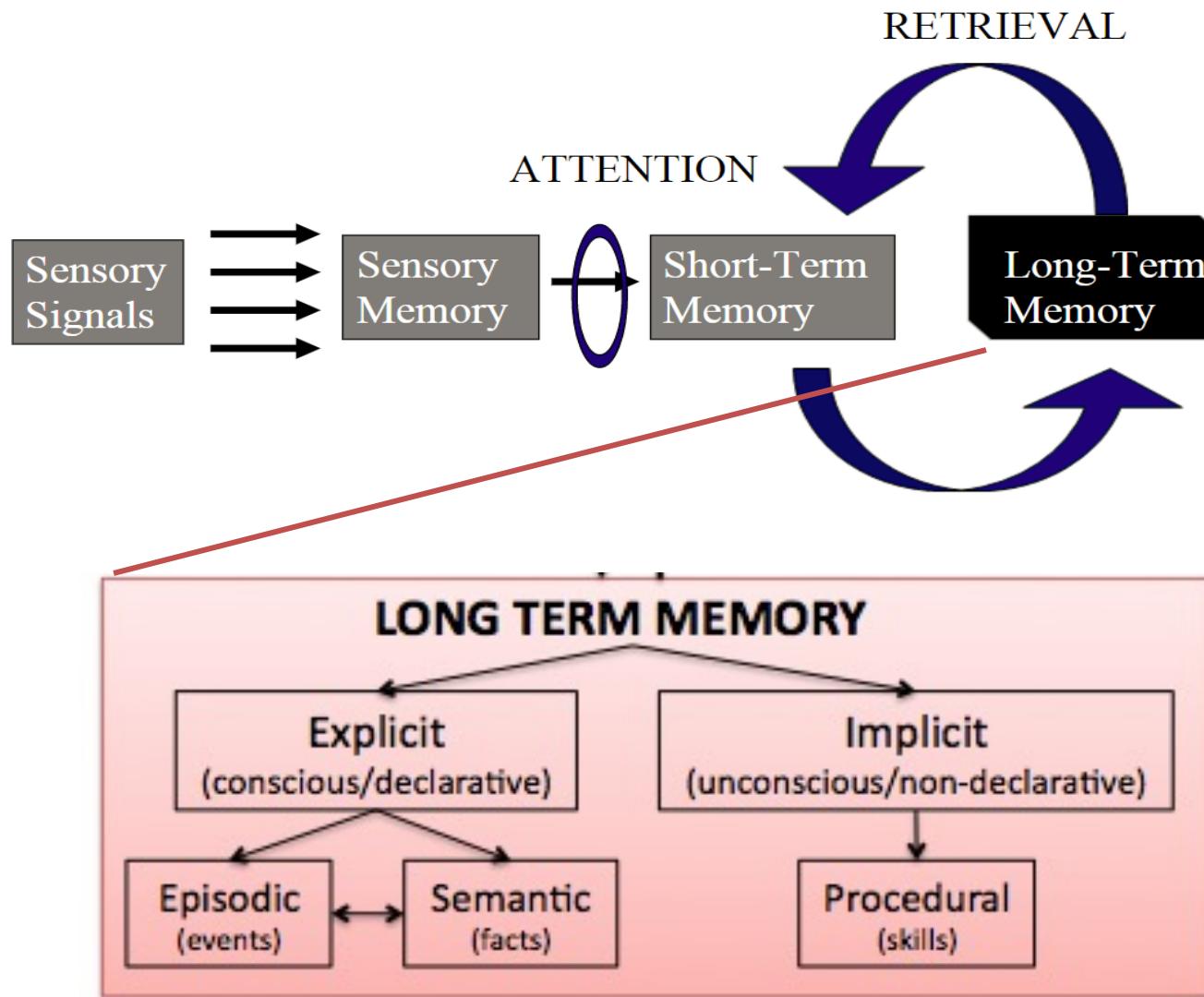
1 Course Review

2 Deep Cognitive Networks

3 Attention Models

4 Memory Models

Short-term Memory & Long-term Memory



Story Comprehension

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to his office. Joe left the milk. Joe went to the bathroom.

Questions from
Joe's angry
mother:

Q1 : Where is Joe?

Q2 : Where is the milk now?

Q3 : Where was Joe before the office?

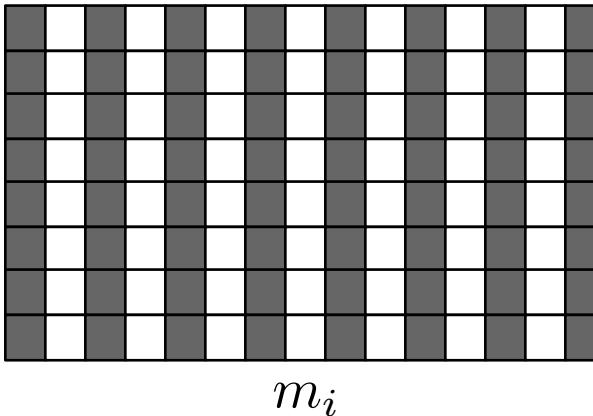
Why Need Memory

AI tasks require explicit memory and
multi-hop reasoning over it

- RNNs only have short memory
- Cannot increase memory without increasing number of parameters

Memory Networks

- Class of models with memory m - array of objects m_i



Four Components :

I - Input Feature Map : Input manipulation

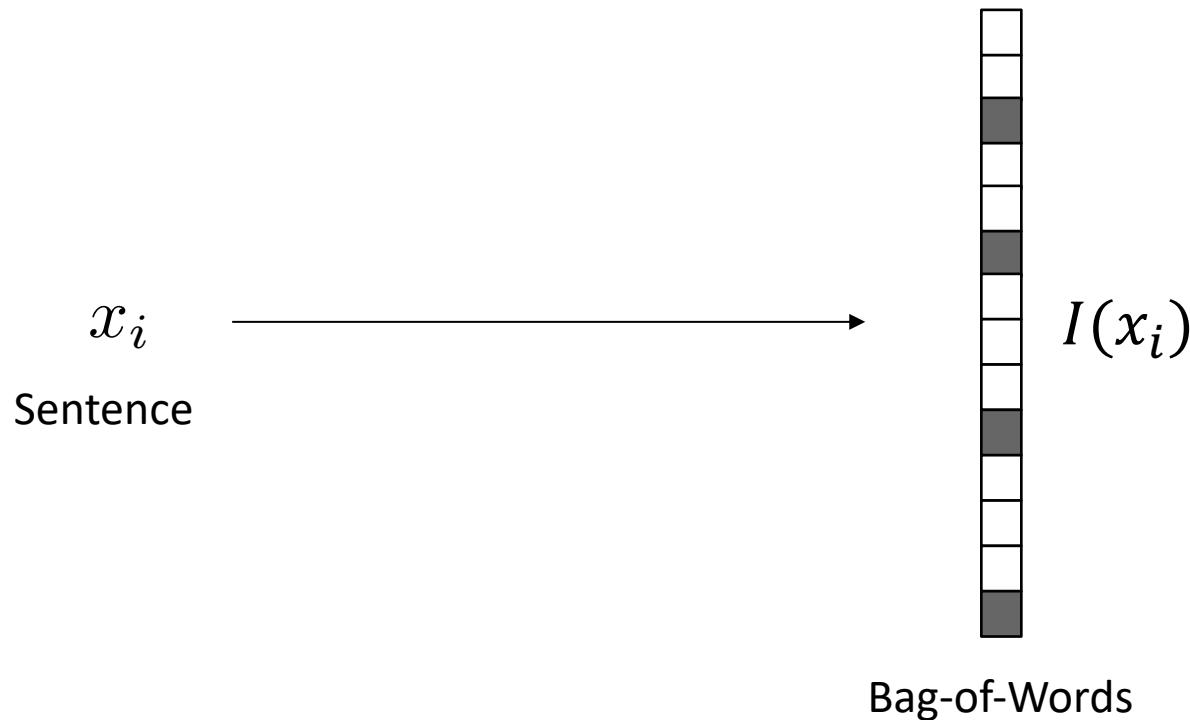
G - Generalization : Memory manipulation

O - Output Feature Map : Output representation generator

R - Response : Response generator

Simple MemNN for Text

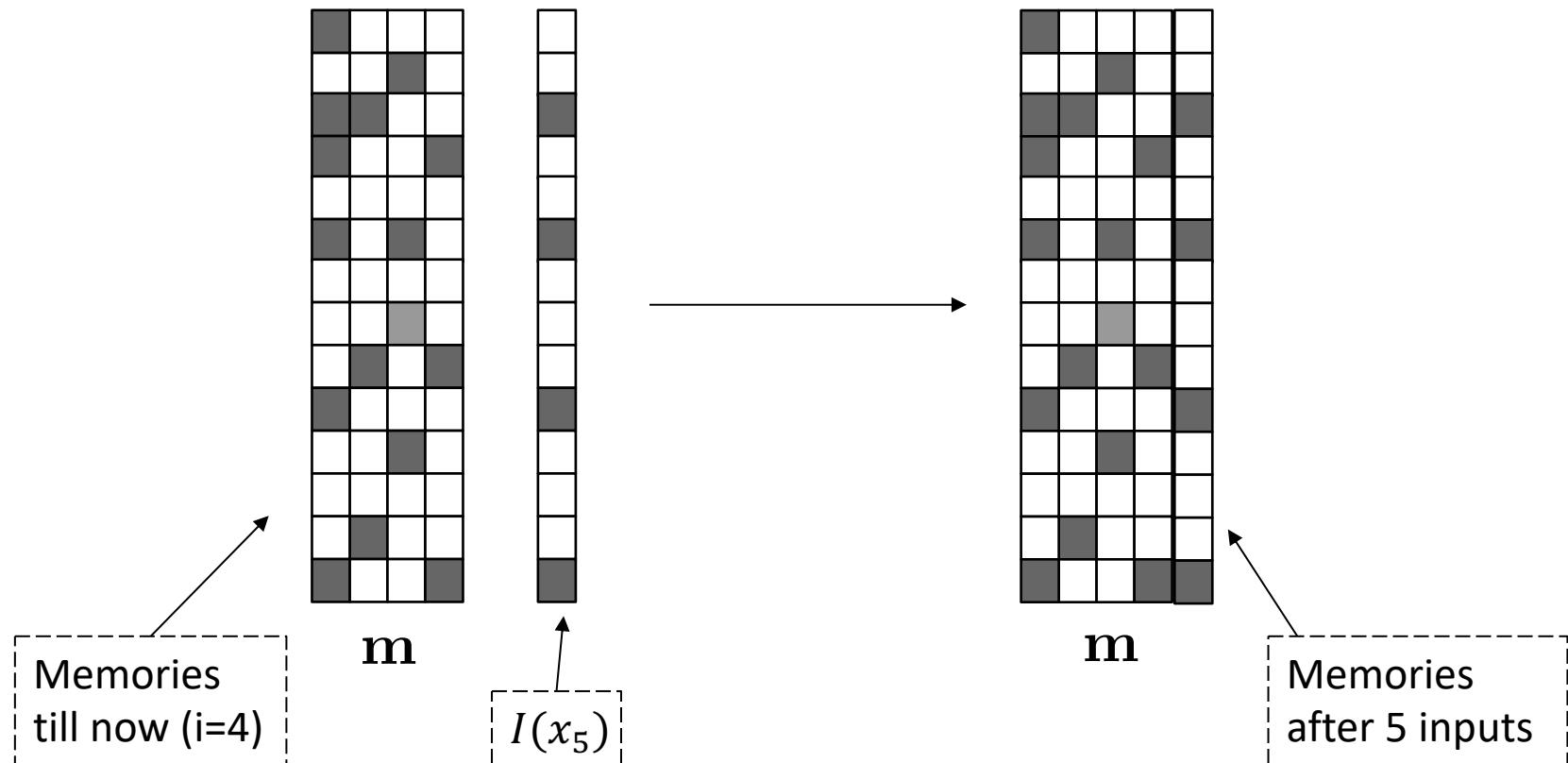
1. Input Feature Map - Bag-of-Words representation



Simple MemNN for Text

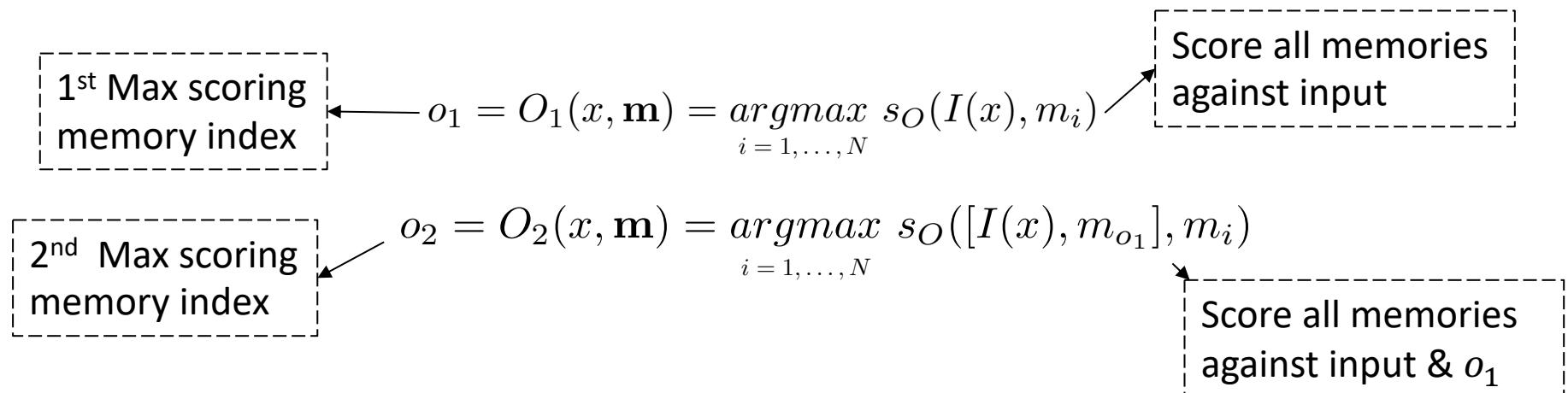
2. Generalization : Store input in new memory

$$m_i = I(x_i)$$

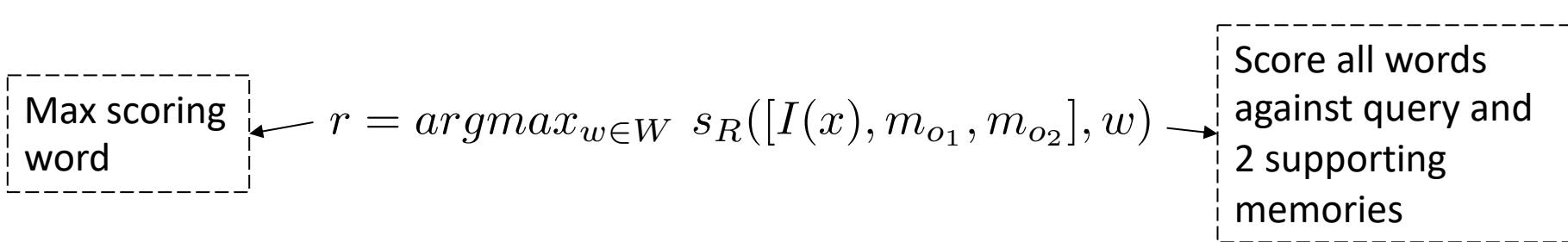


Simple MemNN for Text

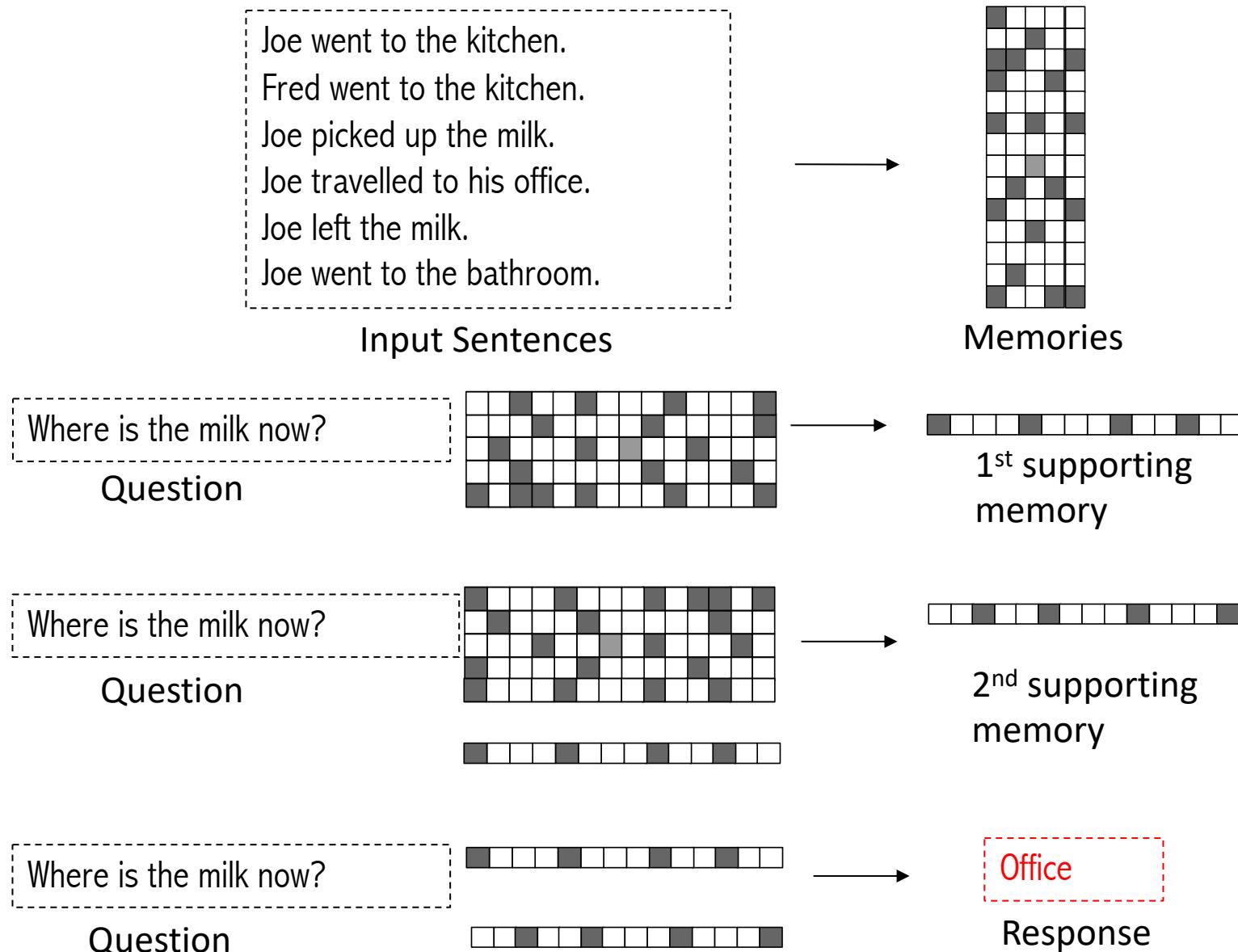
3. Output: Using $k = 2$ memory hops with query x



4. Response - Single Word Answer



Whole Pipeline



Limitations

- **Strong supervision**, i.e., for intermediate memories is needed

End-to-End Memory Networks (MemN2N)

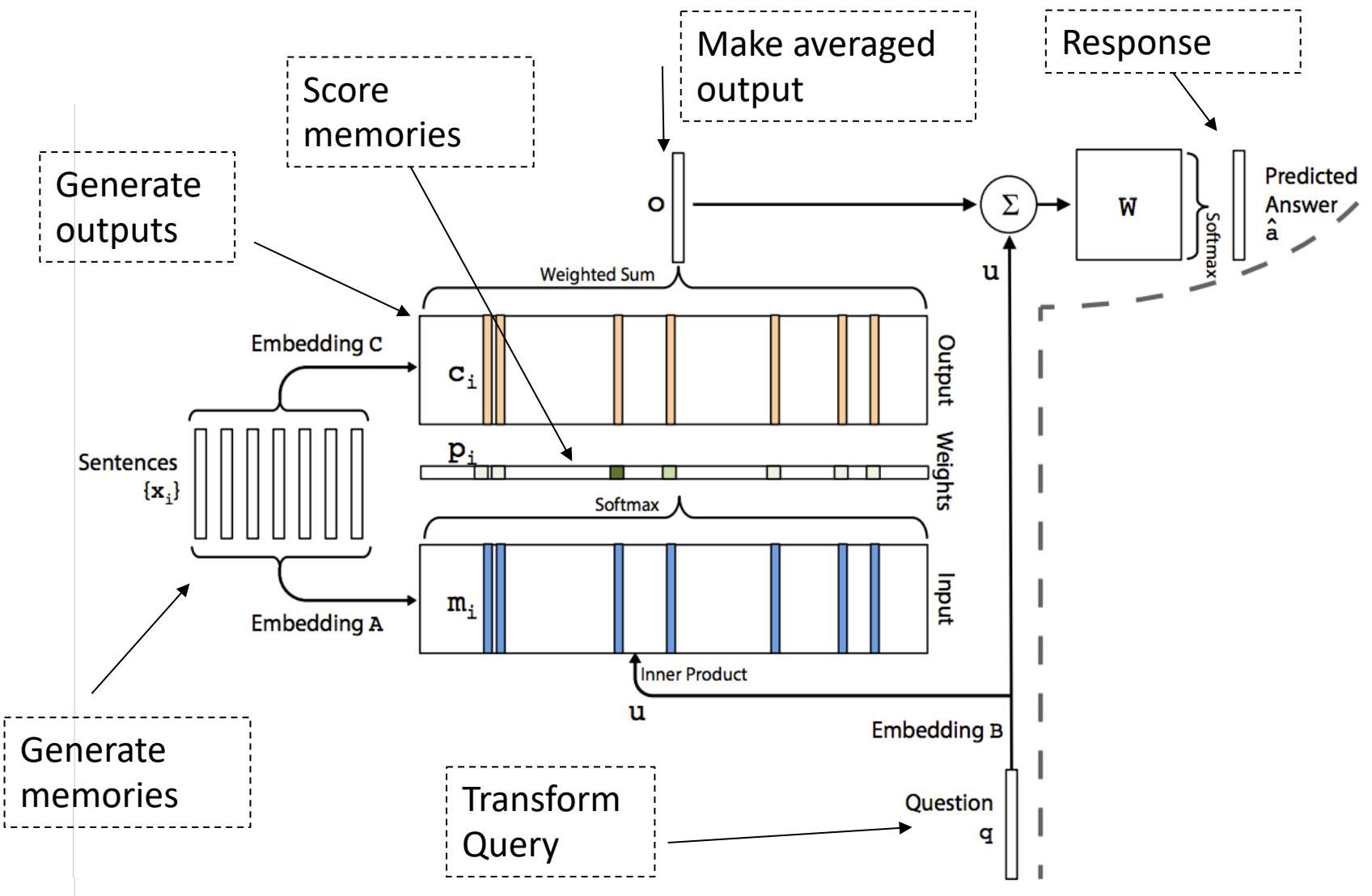
- What if the annotation is:
 - Input sentences x_1, x_2, \dots, x_n
 - Query q
 - Answer a
- Model performs by:
 - Generating memories from inputs
 - Transforming query into suitable representation
 - Process query and memories jointly using multiple hops
only to produce the answer
 - Backpropagate through the whole procedure

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to his office. Joe left the milk. Joe went to the bathroom.

Where is the milk now?

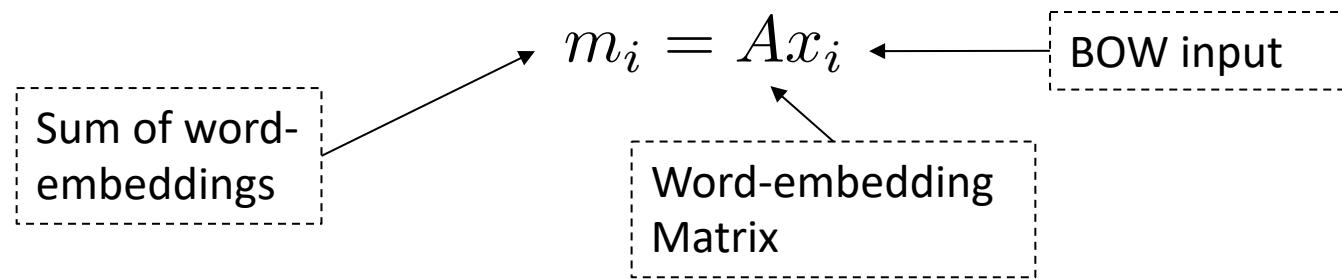
Office

End-to-End Memory Networks (MemN2N)



End-to-End Memory Networks (MemN2N)

1. Convert input to memories $x_i \rightarrow m_i$



2. Transform query q into same representation space

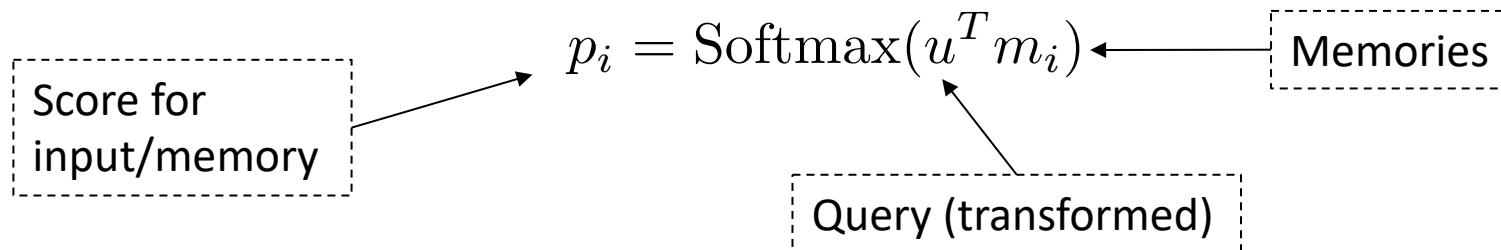
$$u = Bq$$

3. Output vectors $x_i \rightarrow c_i$

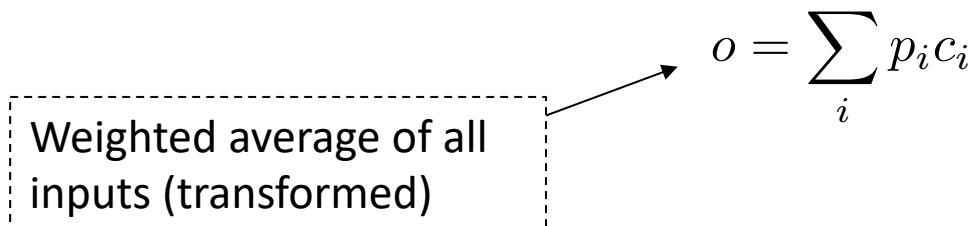
$$c_i = Cx_i$$

End-to-End Memory Networks (MemN2N)

3. Scoring memories against query



4. Generate output



End-to-End Memory Networks (MemN2N)

5. Generating Response

$$\hat{a} = \text{Softmax}(W(u + o))$$

Distribution over response words

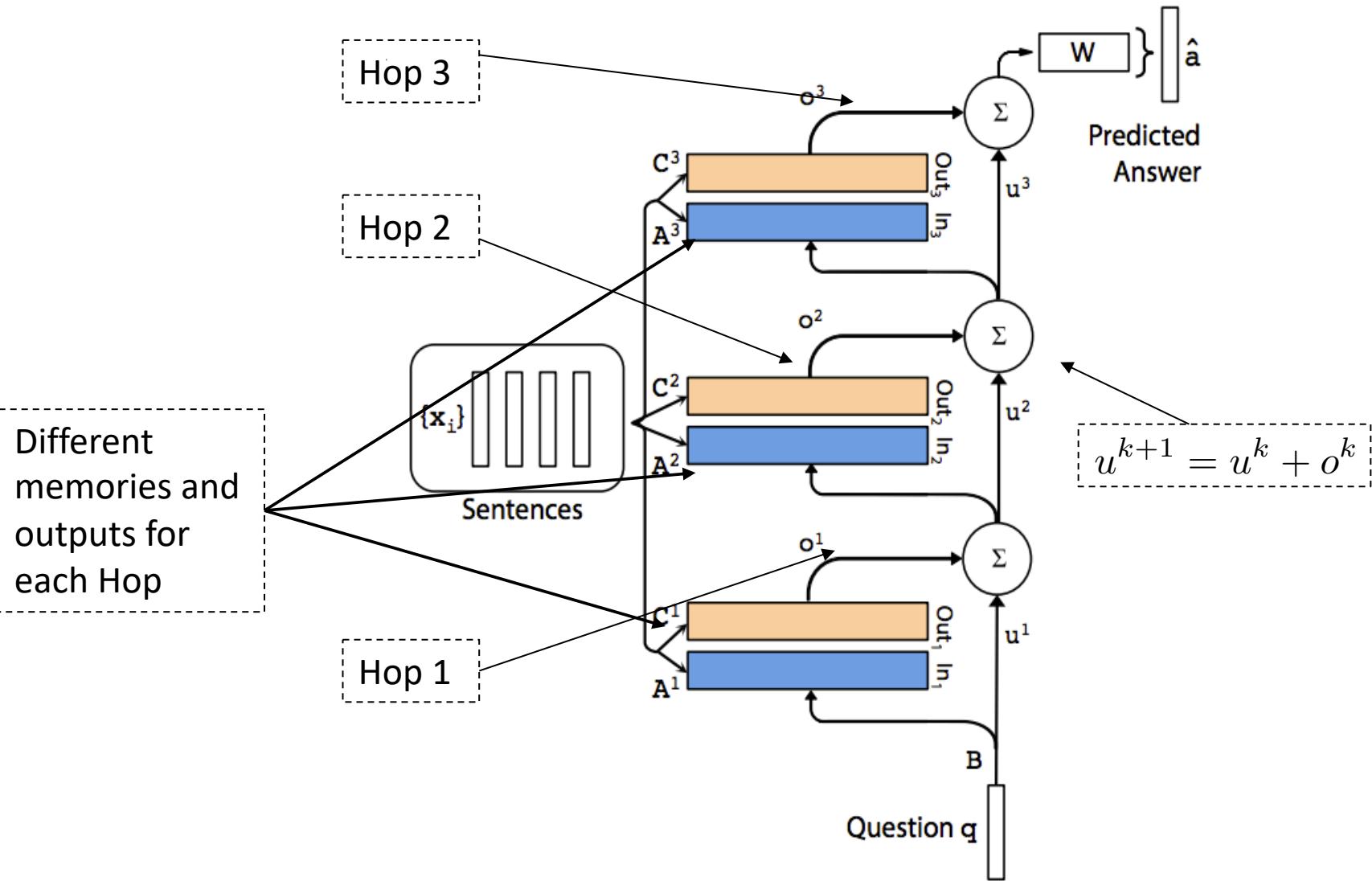
Query Averaged-output

The diagram illustrates the formula $\hat{a} = \text{Softmax}(W(u + o))$. It features three dashed boxes: one on the left labeled "Distribution over response words" with an arrow pointing to the formula; one below the formula labeled "Query" with an arrow pointing to the first term $W(u)$; and one to the right labeled "Averaged-output" with an arrow pointing to the second term $W(o)$.

Training Objective – Maximum Likelihood / Cross Entropy

$$\hat{\Theta} = \operatorname{argmax} \sum_{s=1}^N \log P(\hat{a}_s)$$

Multi-hop MemN2N



Experiments

- Simulated World QA
 - 20 tasks from bAbI dataset - 1K and 10K instances per task
 - Vocabulary = 177 words only
 - 60 epochs
 - Linear start with different learning rate
 - “*Model diverged very often, hence trained multiple models*”

Experiments

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

Movie Trivia Time

- Which was Stanley Kubrick's first movie?

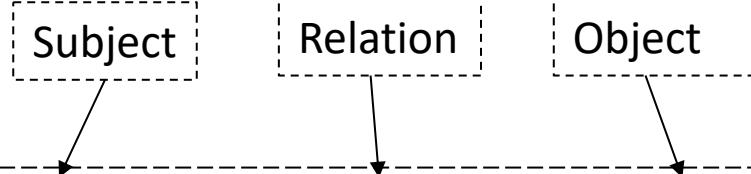
Fear and Dark

- When did 2001:A Space Odyssey release?

1968

- After The Shining, which movie did its director direct?

Full Metal Jacket



(2001:a_space_odyssey, directed_by, stanley_kubrick)

(fear_and_dark, directed_by, stanley_kubrick)

...

(fear_and_dark, released_in, 1953)

(full_metal_jacket, released_in, 1987)

...

(2001:a_space_odyssey, released_in, 1968)

...

(the_shining, directed_by, stanley_kubrick)

...

(AI:artificial_intelligence, written_by, stanley_kubrick)

Knowledge Base

Key-Value MemNNs

- Structured memories as **key-value pairs**
 - Regular MemNNs have single vector for each memory
 - Key more related to question and values to answer

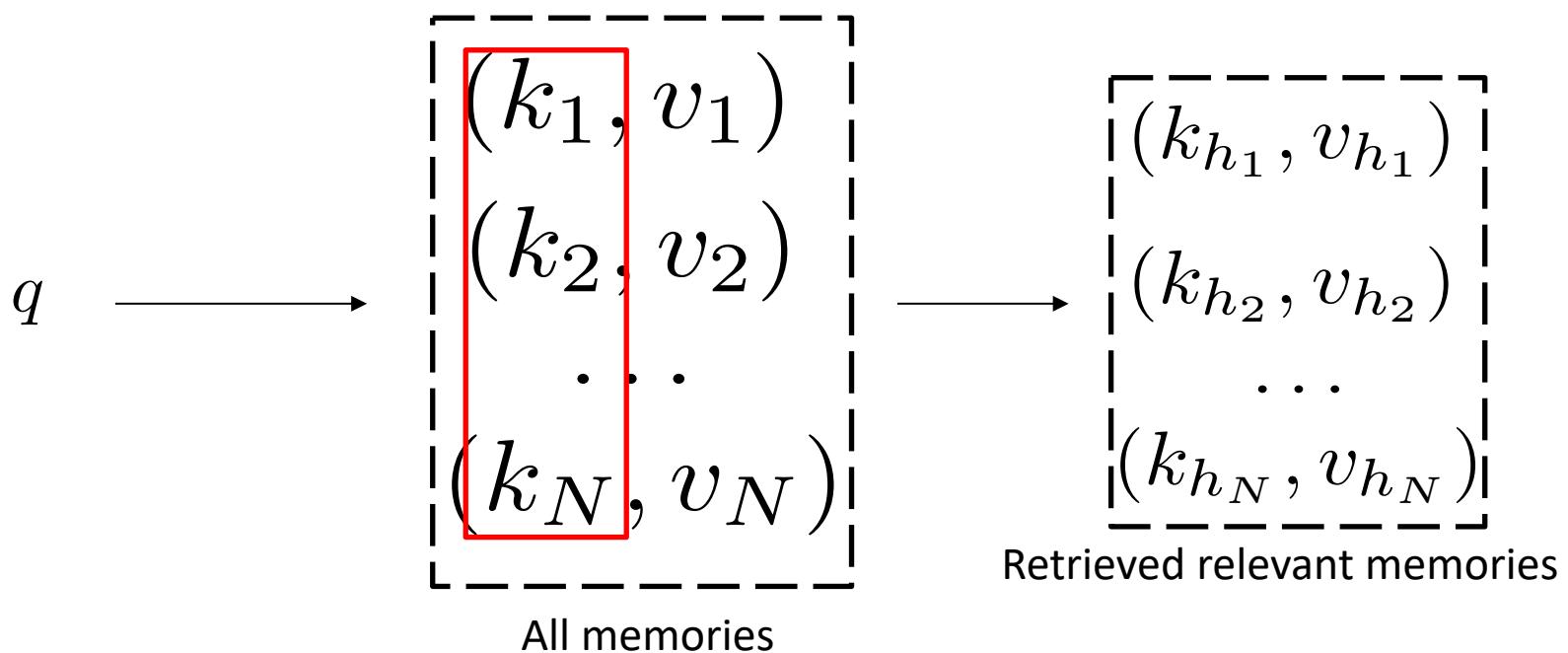
Memories = $(k_1, v_1), (k_2, v_2), \dots, (k_N, v_N)$

(k : Kubrick's first movie was, v : Fear and Dark)

Keys and values can be words, sentences, vectors, etc.

Key-Value MemNNs

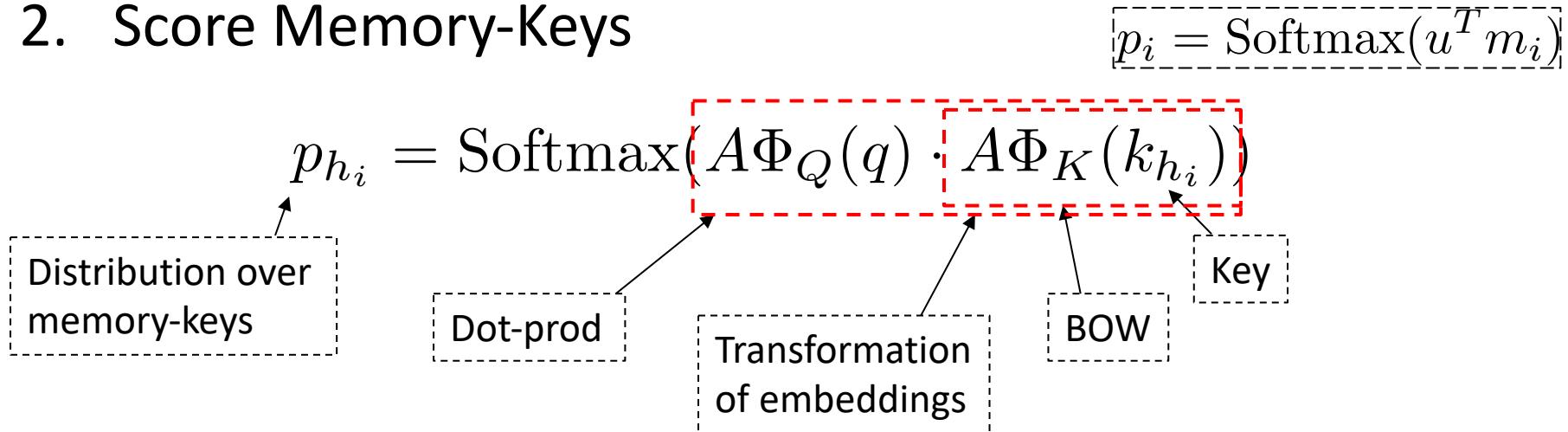
1. Retrieve relevant memories using keys



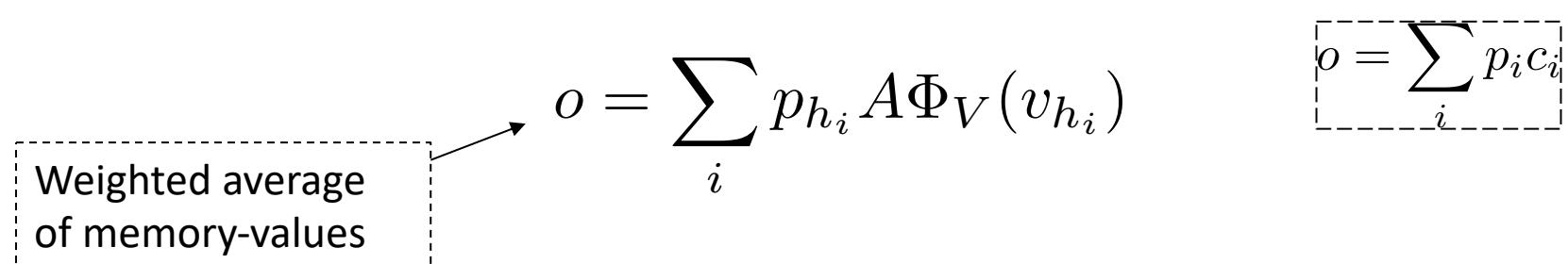
Use inverted index, locality sensitive hashing, something sensible

Key-Value MemNNs

2. Score Memory-Keys



3. Generate Output



KV-MemNN – What to Store in Memories?

1. KB Based

Key: (subject, relation); Value: Object

K: (2001:a_space_odyssey, directed_by); V: stanley_kubrick

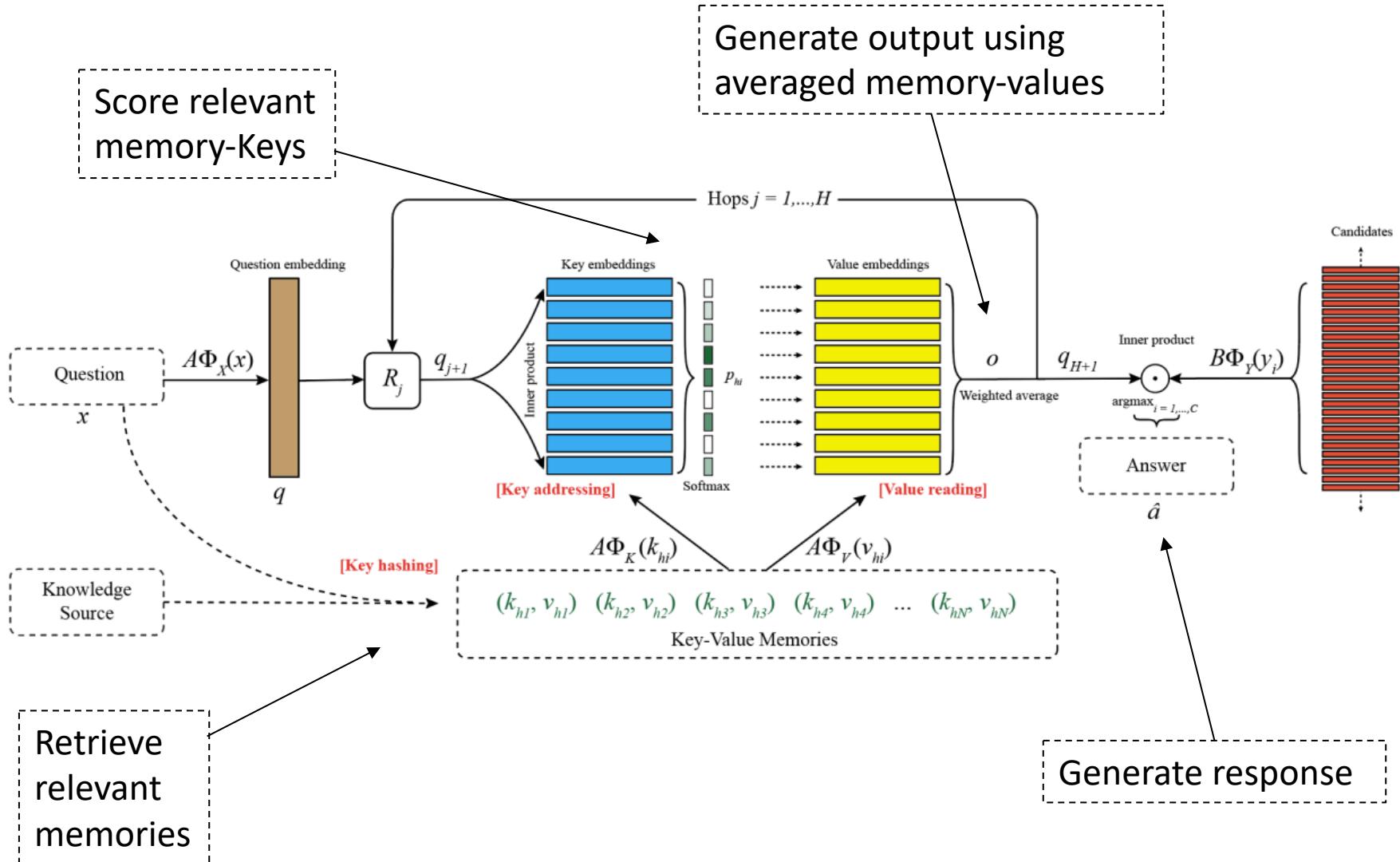
2. Document Based

For each entity in document, extract 5-word window around it

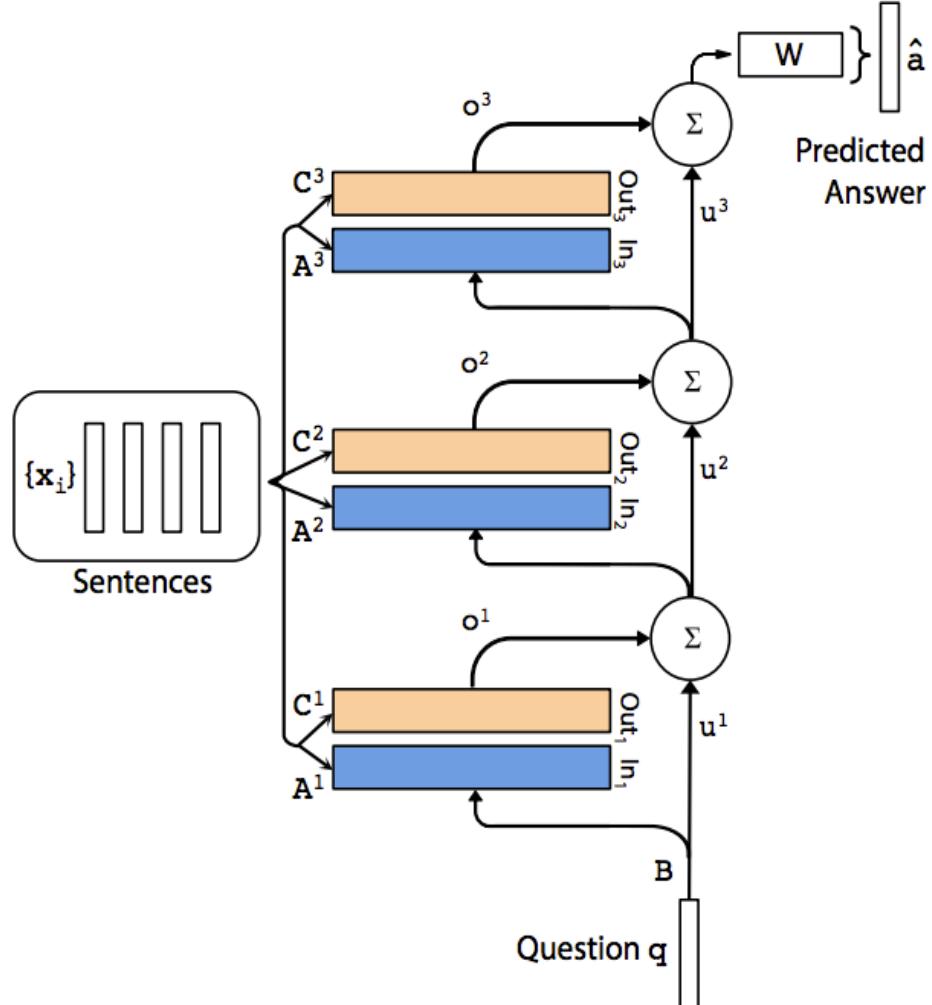
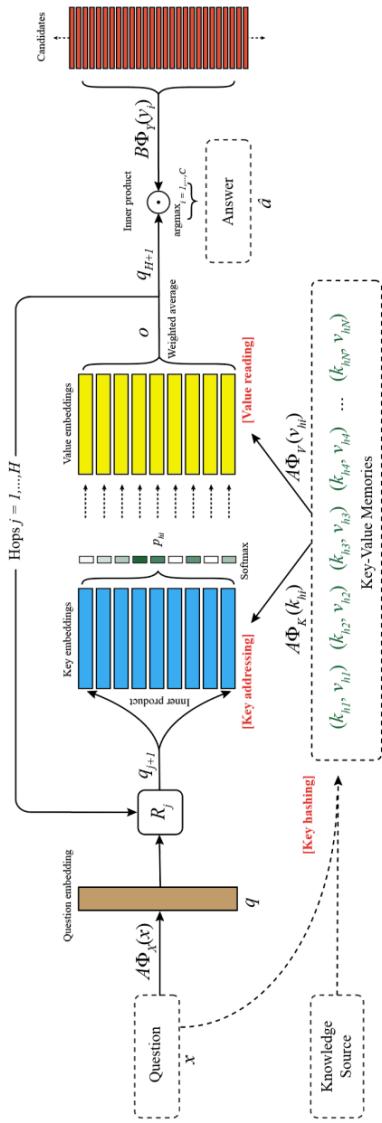
Key: window; Value: Entity

K: screenplay written by and; V: Hampton

KV-MemNN



KV-MemNN



KV-MemNN – Experiments

- WikiMovies Benchmark
 - Total 100K QA-pairs
 - 10% for testing

Method	KB	Doc
E2E Memory Network	78.5	69.9
Key-Value Memory Network	93.9	76.2

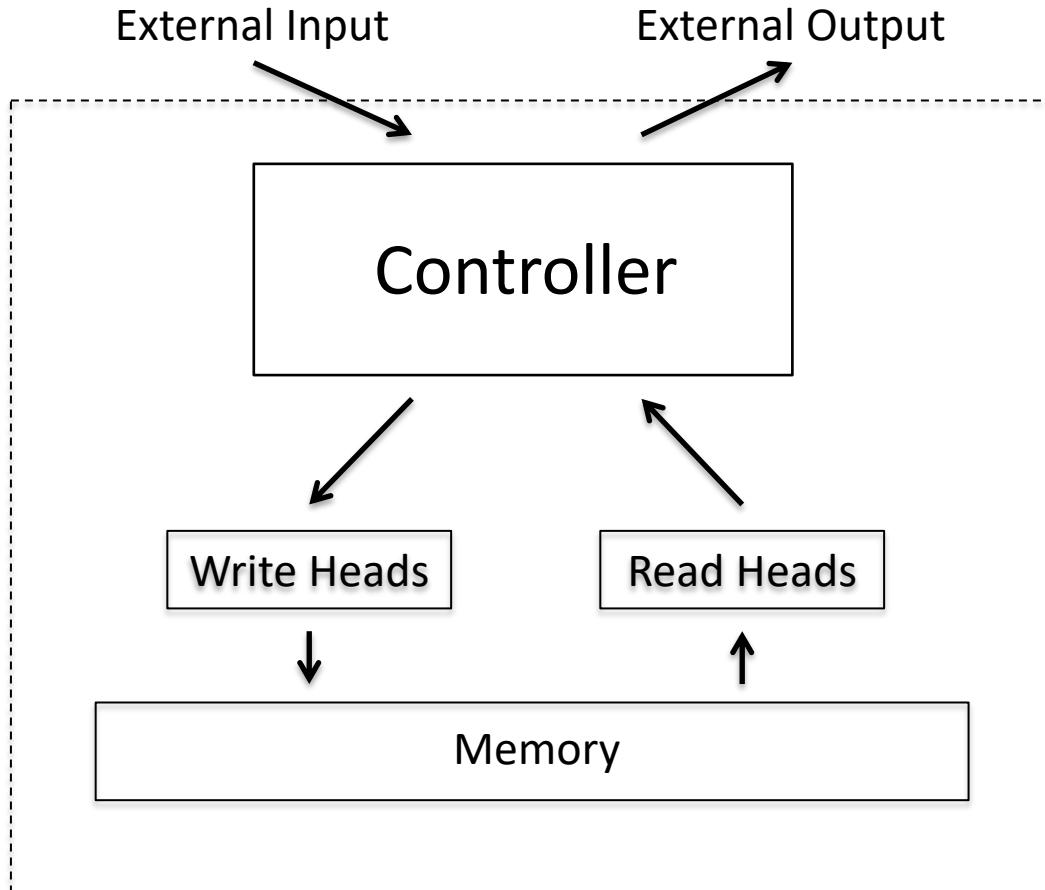
Neural Turing Machine

A more general memory

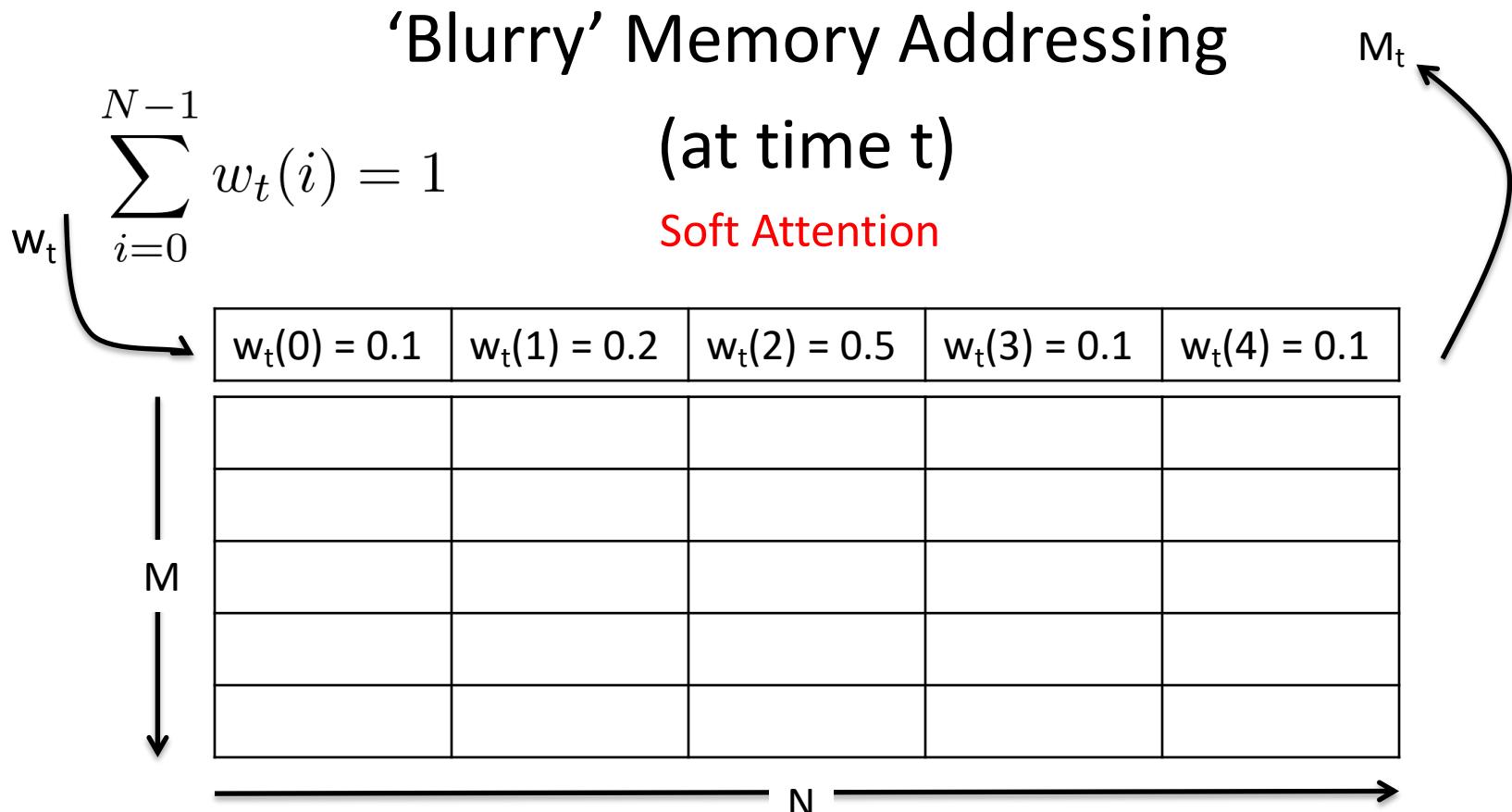
Neural Turing Machine learns:

1. What to write to memory
2. When to write to memory
3. When to stop writing
4. Which memory cell to read from
5. How to convert result of read into final output

Neural Turing Machine



Neural Turing Machine



Neural Turing Machine

More formally,

Blurry Read Operation

Given: M_t (memory matrix) of size $N \times M$

w_t (weight vector) of length N

t (time index)

$$r_t = \sum_{i=0}^{N-1} w_t(i) M_t(i)$$

Neural Turing Machines: Blurry Writes

Blurry Write Operation

Decomposed into blurry erase + blurry add

Given: M_t (memory matrix) of size $N \times M$

w_t (weight vector) of length N

t (time index)

e_t (erase vector) of length M

a_t (add vector) of length M

$$M_t(i) = \underbrace{M_{t-1}(i)(1 - w_t(i)e_t)}_{\text{Erase Component}} + \underbrace{w_t(i)a_t}_{\text{Add Component}}$$

Neural Turing Machines: Erase

$$\mathbf{M}_t(i) = \mathbf{M}_{t-1}(i)(1 - w_t(i)\mathbf{e}_t)$$

1xN

w ₁ (0) = 0.1	w ₁ (1) = 0.2	w ₁ (2) = 0.5	w ₁ (3) = 0.1	w ₁ (4) = 0.1
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

M₀ \Rightarrow

5	7	9	2	12
11	6	3	1	2
3	7	3	10	6
4	2	5	9	9
3	5	12	8	4

e₁

Mx1

1.0
0.7
0.2
0.5
0.0

Neural Turing Machines: Erase

$$\mathbf{M}_t(i) = \mathbf{M}_{t-1}(i)(1 - w_t(i)\mathbf{e}_t)$$

$w_1(0) = 0.1$	$w_1(1) = 0.2$	$w_1(2) = 0.5$	$w_1(3) = 0.1$	$w_1(4) = 0.1$
4.5	5.6	4.5	1.8	10.8
10.23	5.16	1.95	0.93	1.86
2.94	6.72	2.7	9.8	5.88
3.8	1.8	3.75	8.55	8.55
3	5	12	8	4

Neural Turing Machines: Addition

$$\mathbf{M}_t(i) = \mathbf{M}_{t-1}(i)(1 - w_t(i)\mathbf{e}_t) + w_t(i)\mathbf{a}_t$$

$w_1(0) = 0.1$	$w_1(1) = 0.2$	$w_1(2) = 0.5$	$w_1(3) = 0.1$	$w_1(4) = 0.1$	a_1
4.5	5.6	4.5	1.8	10.8	3
10.23	5.16	1.95	0.93	1.86	4
2.94	6.72	2.7	9.8	5.88	-2
3.8	1.8	3.75	8.55	8.55	0
3	5	12	8	4	2

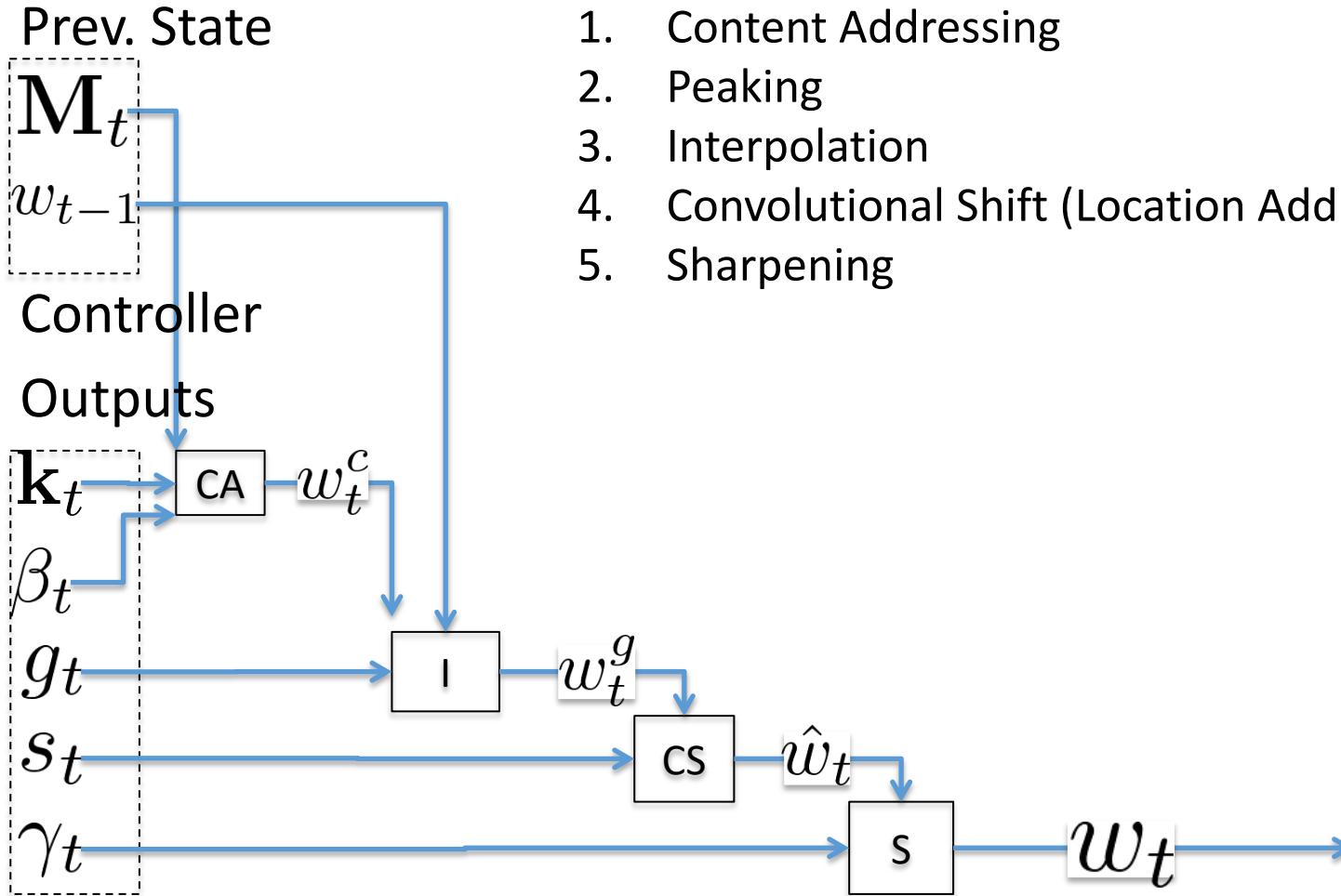
Neural Turing Machines: Blurry Writes

$$\mathbf{M}_t(i) = \mathbf{M}_{t-1}(i)(1 - w_t(i)\mathbf{e}_t) + w_t(i)\mathbf{a}_t$$

$\mathbf{M}_1 \Rightarrow$

4.8	6.2	6	2.1	11.1
10.63	5.96	3.95	1.33	2.26
2.74	6.32	1.7	9.6	5.68
3.8	1.8	3.75	8.55	8.55
3.2	5.4	13	8.2	4.2

Neural Turing Machines: Generate w_t



Steps for generating w_t

1. Content Addressing
2. Peaking
3. Interpolation
4. Convolutional Shift (Location Addressing)
5. Sharpening

Neural Turing Machines: Generate w_t

Prev. State

M_t

Controller

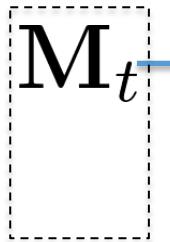
Outputs

k_t

k_t : Vector (length M) produced by Controller

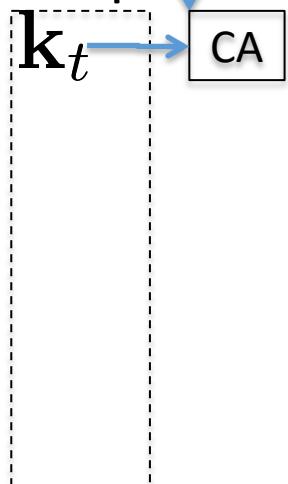
Neural Turing Machines: Generate w_t

Prev. State



Controller

Outputs

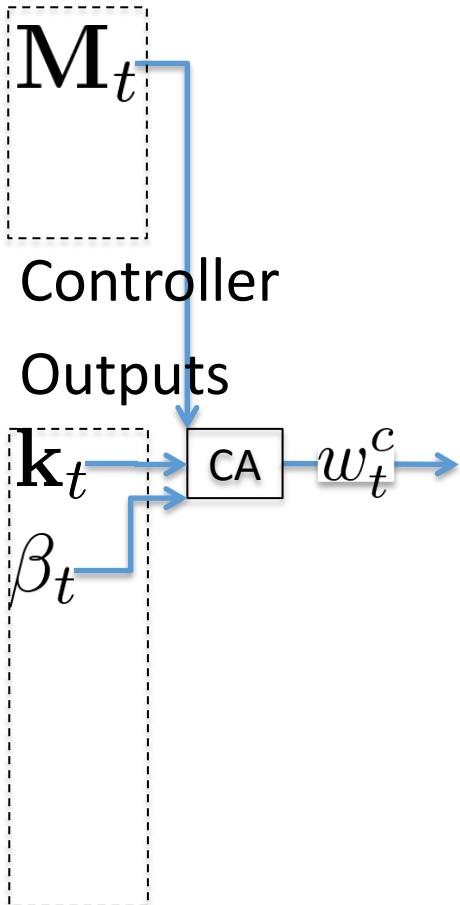


Step 1: Content Addressing (CA)

$$w_t^c(i) = \frac{\exp \langle \mathbf{M}_t(i), \mathbf{k}_t \rangle}{\sum_i \exp \langle \mathbf{M}_t(i), \mathbf{k}_t \rangle}$$

Neural Turing Machines: Generate w_t

Prev. State

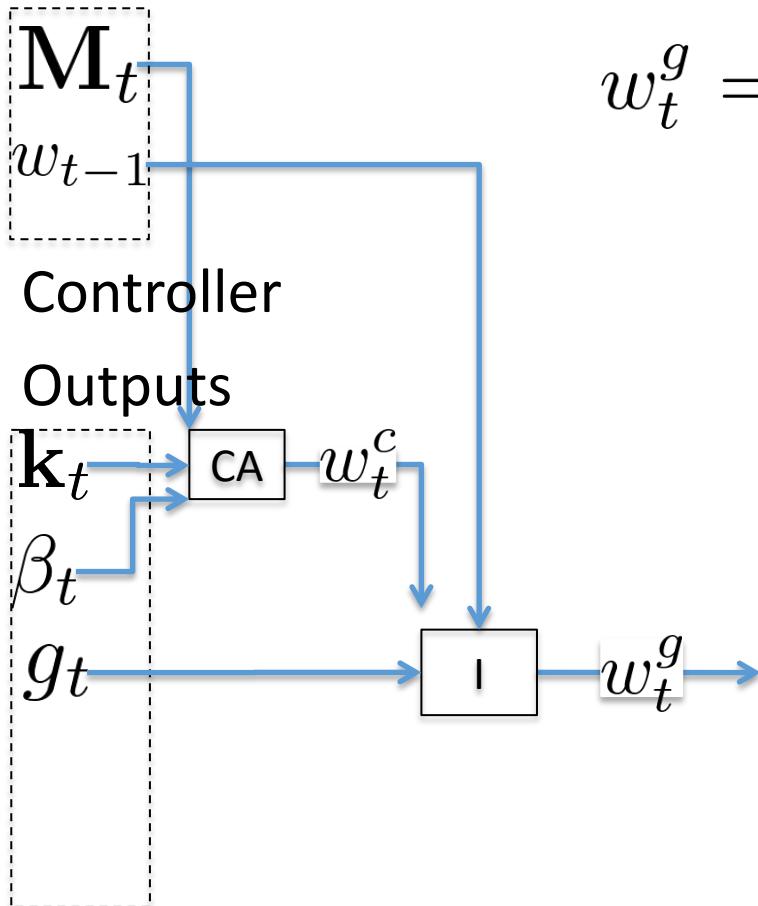


Step 2: Peaking

$$w_t^c(i) = \frac{\exp(\beta_t(\langle \mathbf{M}_t(i), \mathbf{k}_t \rangle))}{\sum_i \exp(\beta_t(\langle \mathbf{M}_t(i), \mathbf{k}_t \rangle))}$$

Neural Turing Machines: Generate w_t

Prev. State

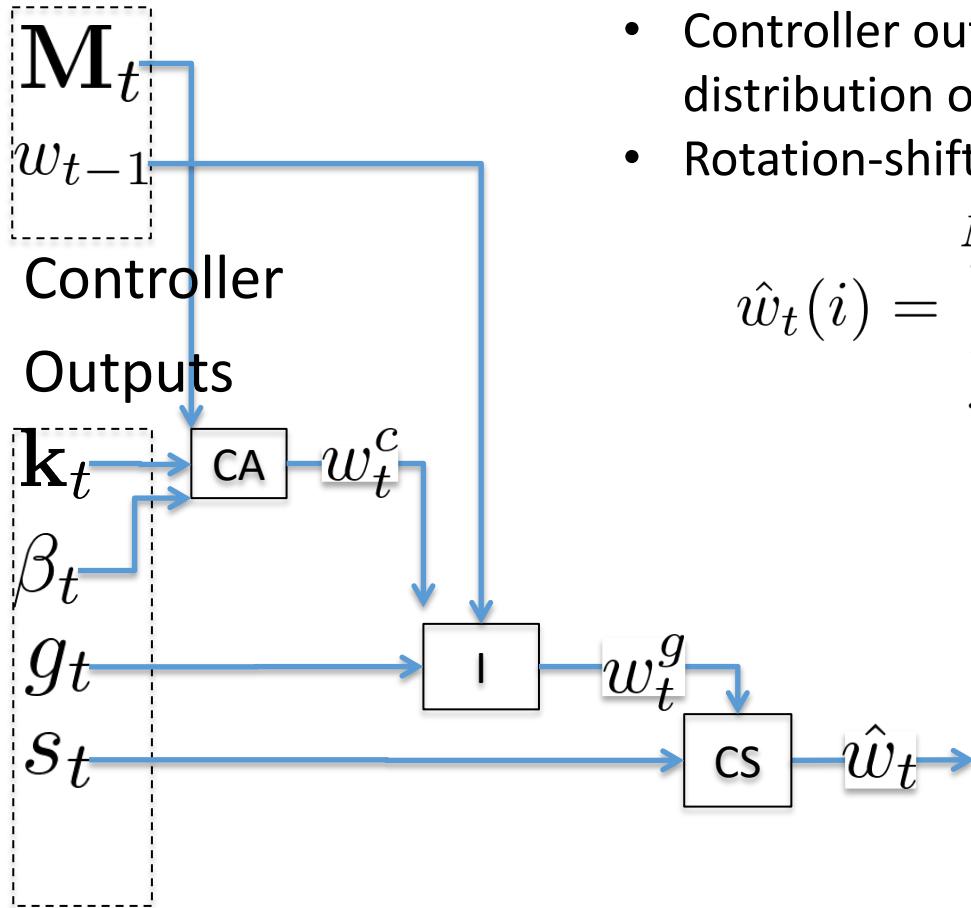


Step 3: Interpolation (I)

$$w_t^g = g_t w_t^c + (1 - g_t) w_{t-1}$$

Neural Turing Machines: Generate w_t

Prev. State



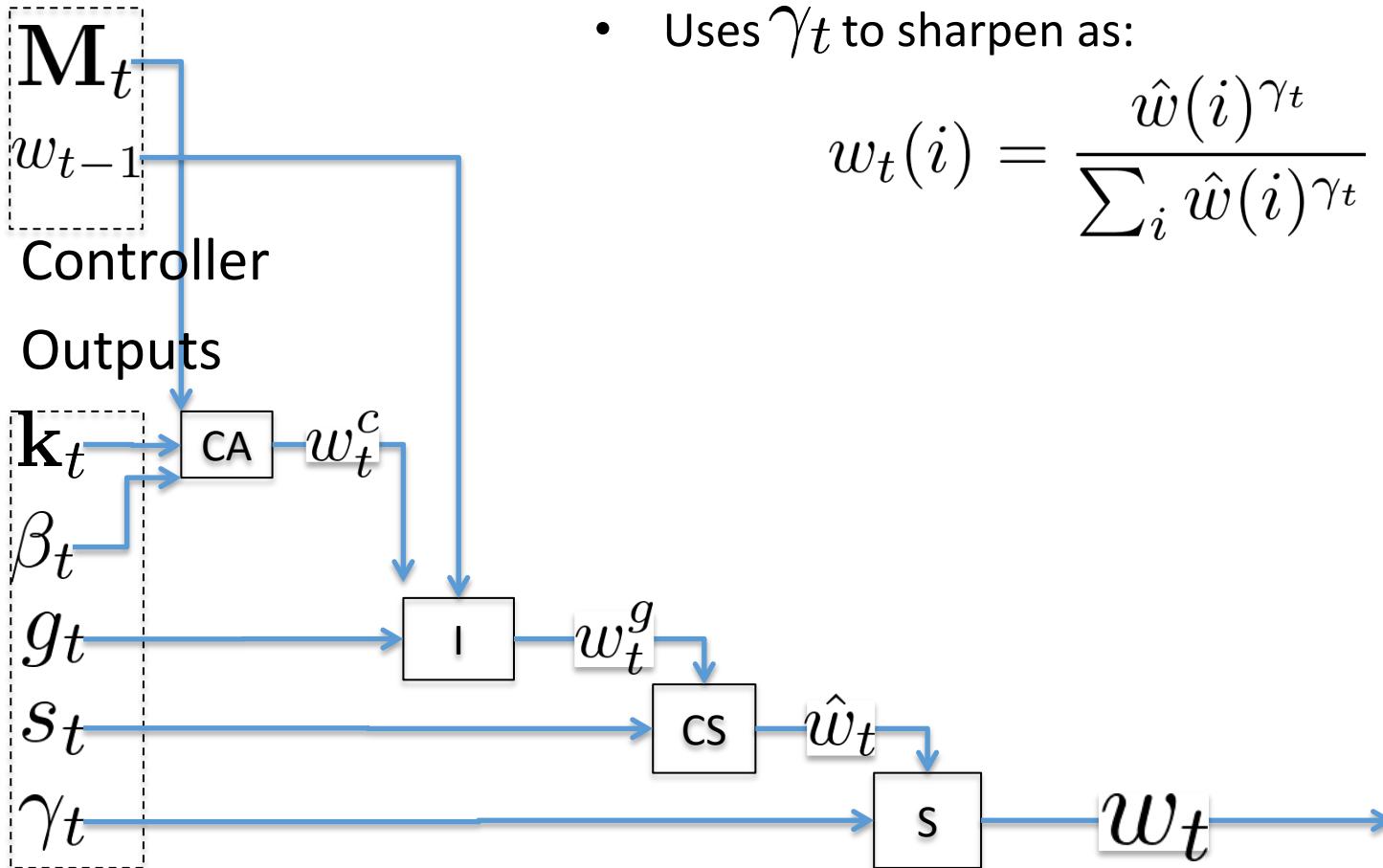
Step 4: Convolutional Shift (CS)

- Controller outputs S_t , a normalized distribution over all N possible shifts
- Rotation-shifted weights computed as:

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

Neural Turing Machines: Generate w_t

Prev. State



Step 5: Sharpening (S)

- Uses γ_t to sharpen as:

$$w_t(i) = \frac{\hat{w}(i)^{\gamma_t}}{\sum_i \hat{w}(i)^{\gamma_t}}$$

Neural Turing Machine: Controller Design

- **Feed-forward**: faster, more transparency & interpretability about function learnt
- **LSTM**: more expressive power, doesn't limit the number of computations per time step

Both are end-to-end differentiable!

1. Reading/Writing -> Convex Sums
2. w_t generation -> Smooth

Neural Turing Machine: Network Overview

Unrolled Feed-forward Controller

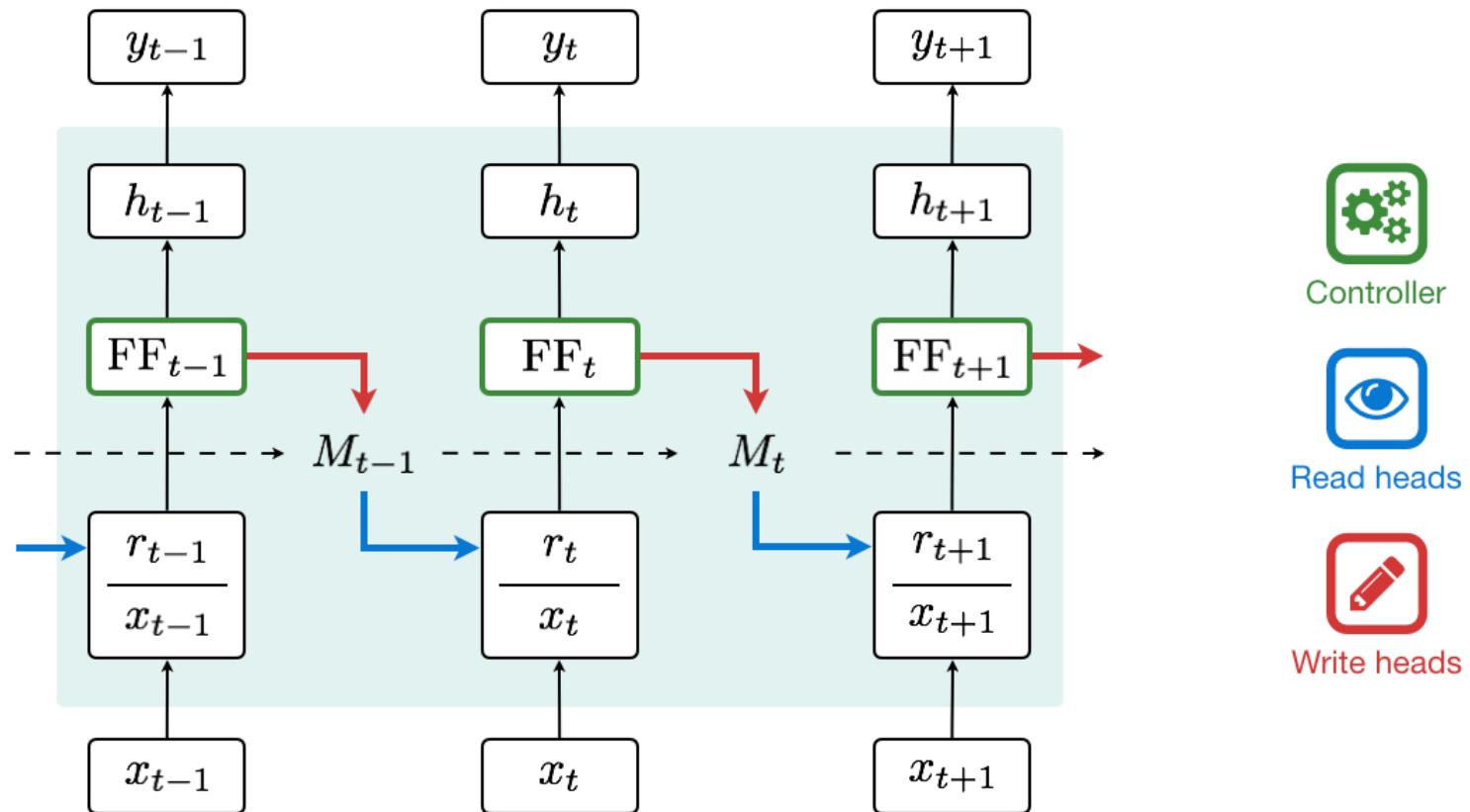


Figure from [Snips AI's Medium Post](#)

Neural Turing Machines vs. MemNNs

MemNNs

- Memory is **static**, with focus on retrieving (**reading**) information from memory

NTMs

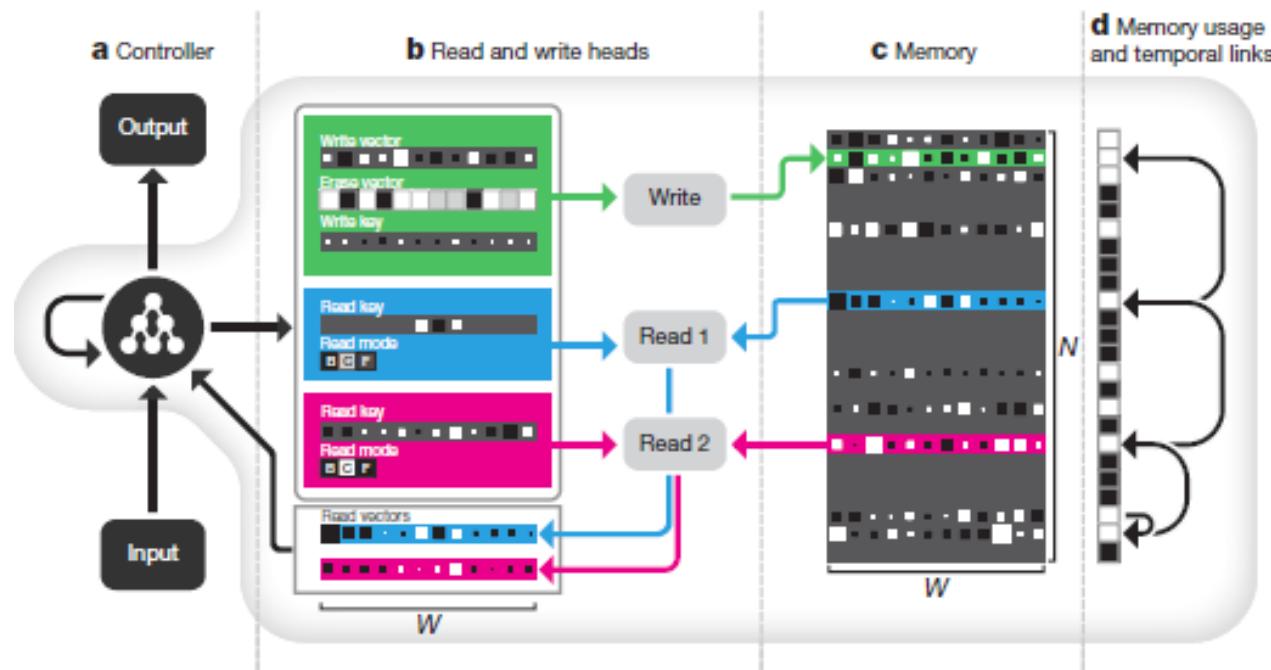
- Memory is continuously written to and read from, with network learning when to perform memory **read and write**

Differentiable Neural Computers

doi:10.1038/nature20101

Hybrid computing using a neural network with dynamic external memory

Alex Graves^{1*}, Greg Wayne^{1*}, Malcolm Reynolds¹, Tim Harley¹, Ivo Danihelka¹, Agnieszka Grabska-Barwińska¹, Sergio Gómez Colmenarejo¹, Edward Grefenstette¹, Tiago Ramalho¹, John Agapiou¹, Adrià Puigdomènech Badia¹, Karl Moritz Hermann¹, Yori Zwols¹, Georg Ostrovski¹, Adam Cain¹, Helen King¹, Christopher Summerfield¹, Phil Blunsom¹, Koray Kavukcuoglu¹ & Demis Hassabis¹



Differentiable Neural Computers

Advanced addressing mechanisms:

- Content Based Addressing
- Temporal Addressing
 - Maintains notion of sequence in addressing
 - Temporal Link Matrix L (size $N \times N$), $L[i,j]$ = degree to which location i was written to after location j
- Usage Based Addressing

DNC: Usage Based Addressing

- Writing increases usage of cell, reading decreases usage of cell
- Least used location has highest **usage-based weighting**
- Interpolate b/w usage & content based weights for final write weights

DNC: Improvements over NTMs

NTM

- Large contiguous blocks of memory needed
- No way to free up memory cells after writing

DNC

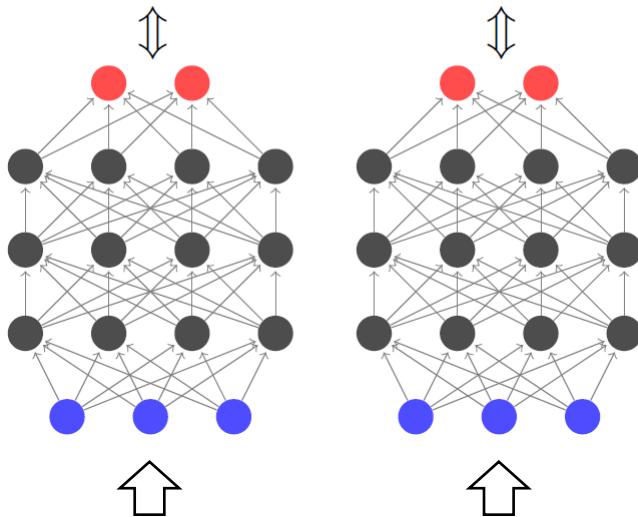
- Memory locations **non-contiguous, usage-based**
- **Regular re-allocation** based on usage-tracking

Multimodal Memory

- Wang et al., M3: Multimodal Memory Modelling for Video Captioning, **CVPR**, 2018.
- Kim et al., Multimodal Dual Attention Memory for Video Story Question Answering, **ECCV**, 2018.
- Huang et al., ACMM: Aligned Cross-Modal Memory for Few-Shot Image and Sentence Matching, **ICCV**, 2019.
- Zhu et al., Vision-Dialog Navigation by Exploring Cross-Modal Memory, **CVPR**, 2020.
- Khademi et al., Multimodal Neural Graph Memory Networks for Visual Question Answering, **ACL**, 2020.
- Zhang et al., Few-Shot Activity Recognition with Cross-Modal Memory Network, **PR**, 2020.
- Huang et al., Few-Shot Image and Sentence Matching via Aligned Cross-Modal Memory, **IEEE TPAMI**, 2021.

Case Study: Image and Sentence Matching

Canonical Correlation Analysis



There are many kinds of vegetables



Image-sentence retrieval



Until April, the Polish forces had been slowly but steadily advancing eastward

In southeastern Washington, a stretch of the river passes through the Hanford Site, established in 1943 as part of the Manhattan Project

Tropical Storm Edouard was the first of eight named storms to form in September 2002, the most such storms for any month in the Atlantic



Image caption

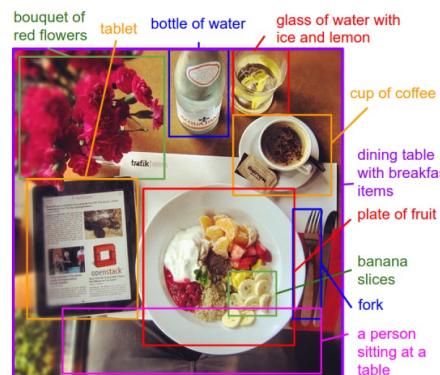


Image question answering



Image

GT Question

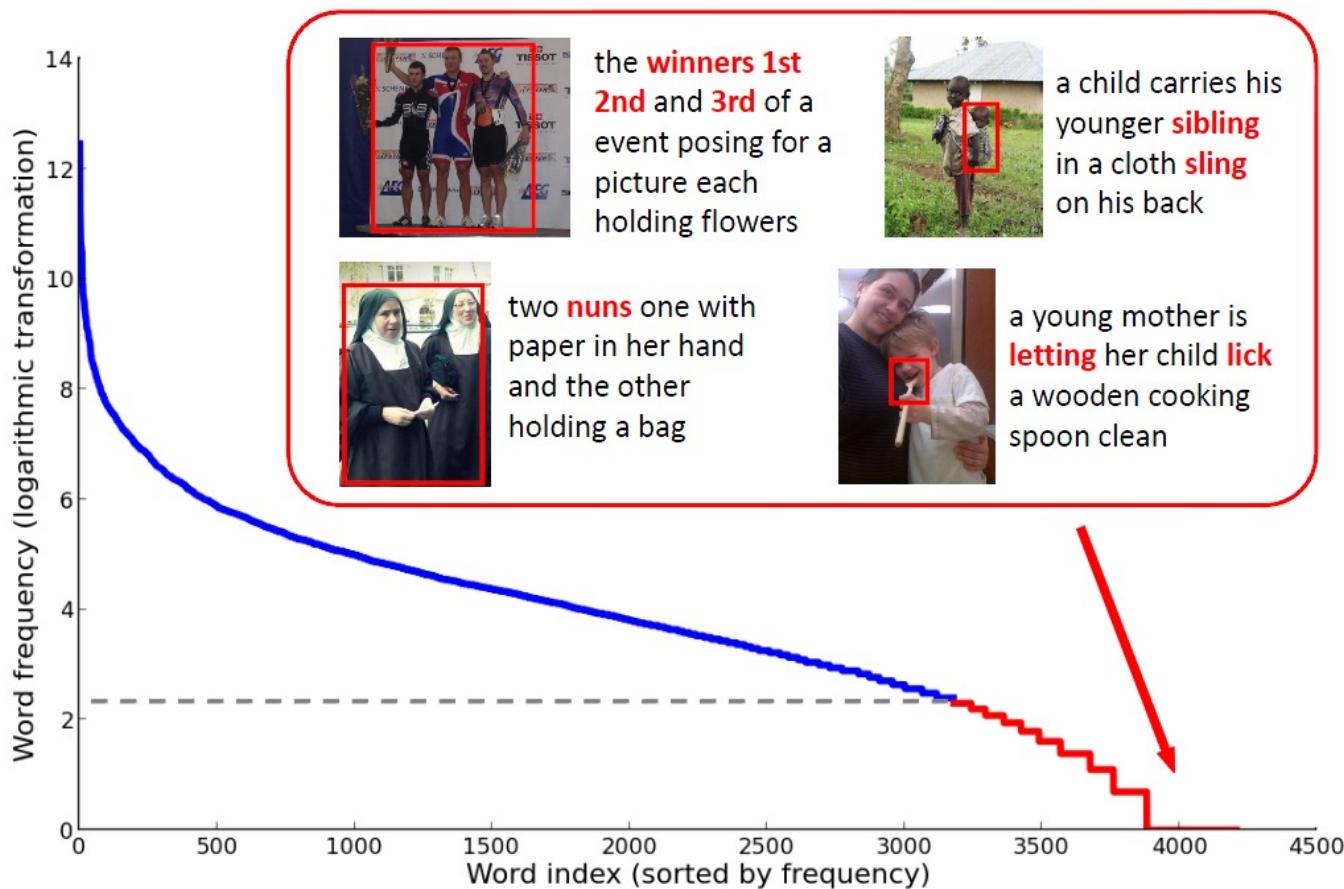
戴帽子的男孩在干什么?
What is the boy in green cap doing?

GT Answer

他在玩滑板。
He is playing skateboard.

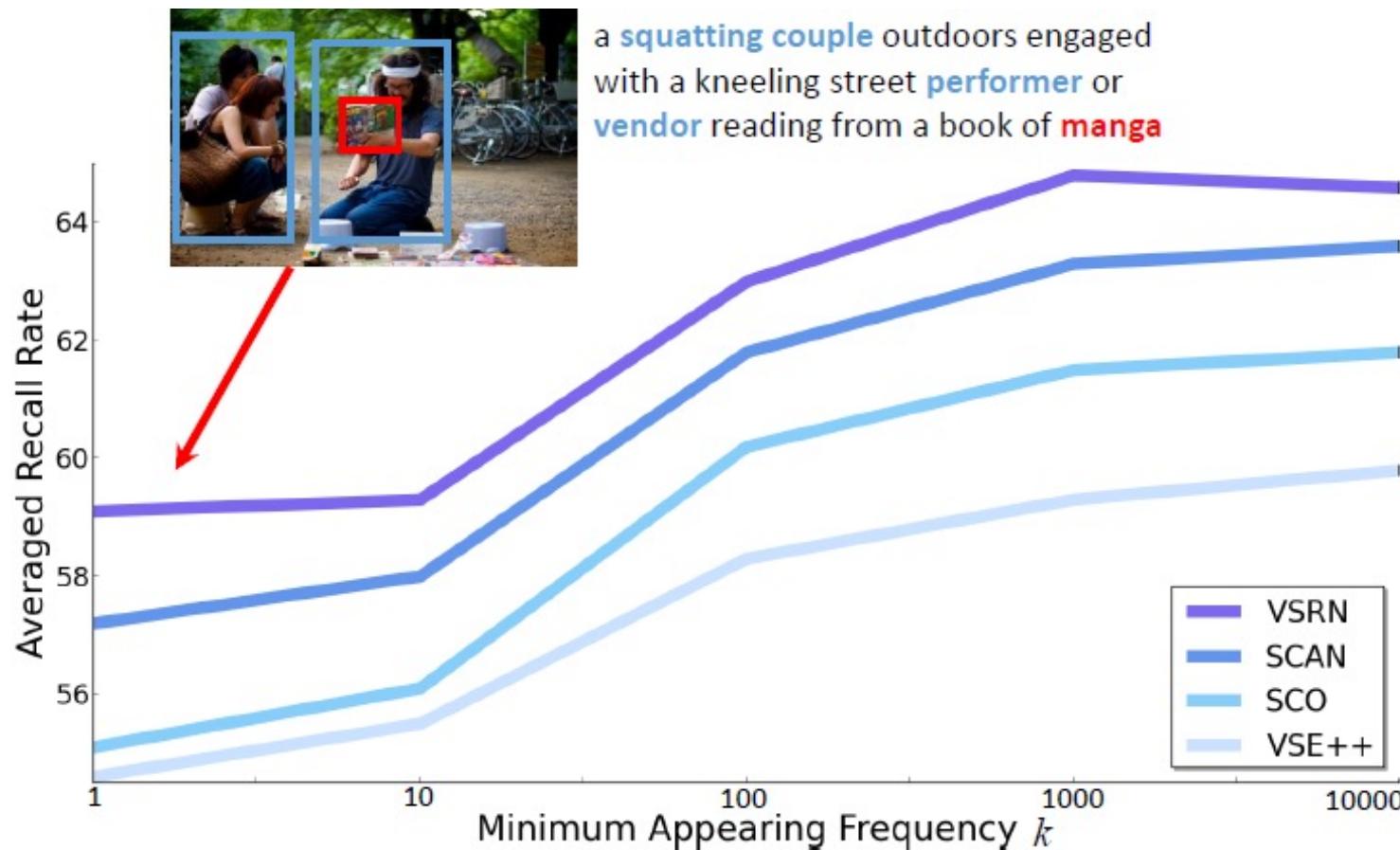
Few-shot Matching (1/2)

- Either an image or a sentence is a mixture of few-shot and common content



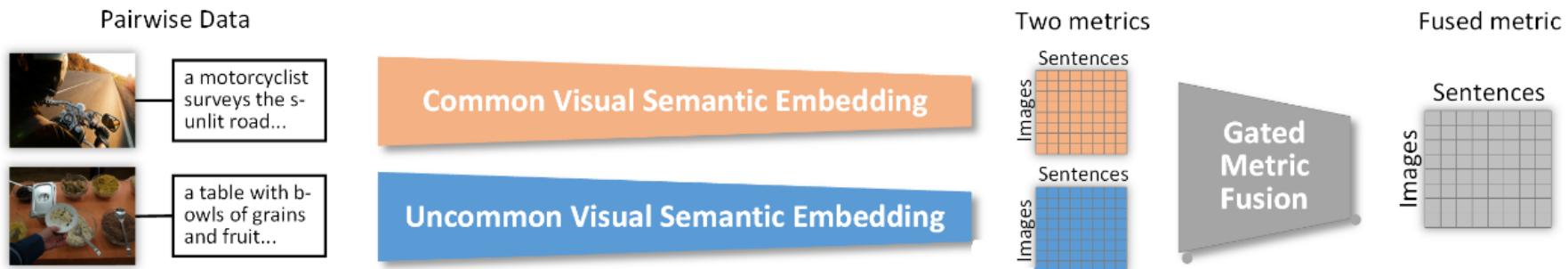
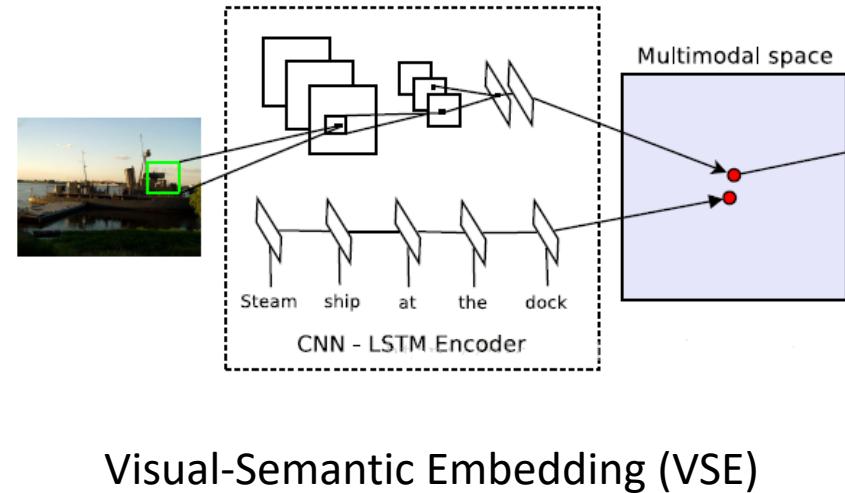
Few-shot Matching (2/2)

- A bottleneck for performance improvement, but commonly ignored



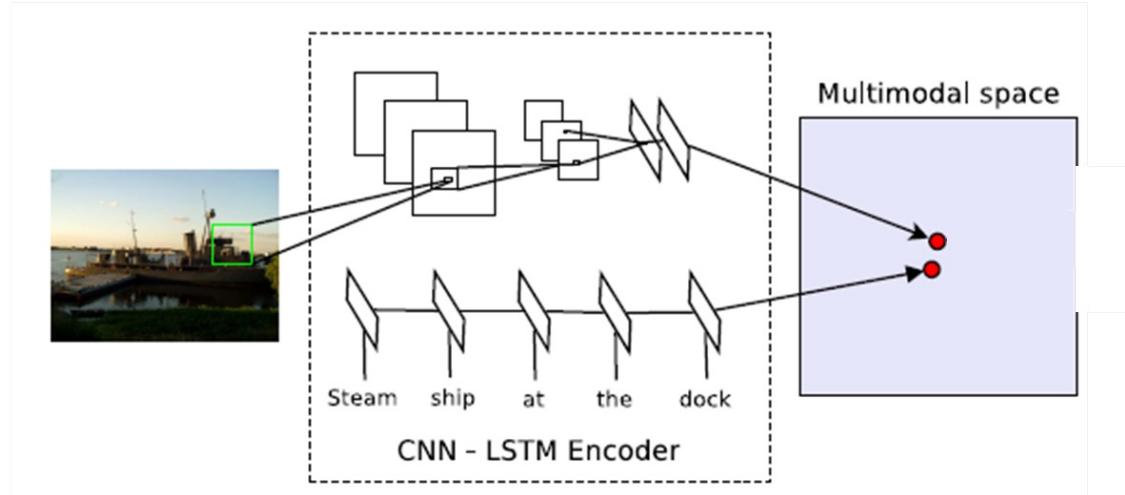
A Straightforward Idea: Two-Stream Framework

- Two separated visual-semantic embedding (VSE) modules
- A gated metric fusion module



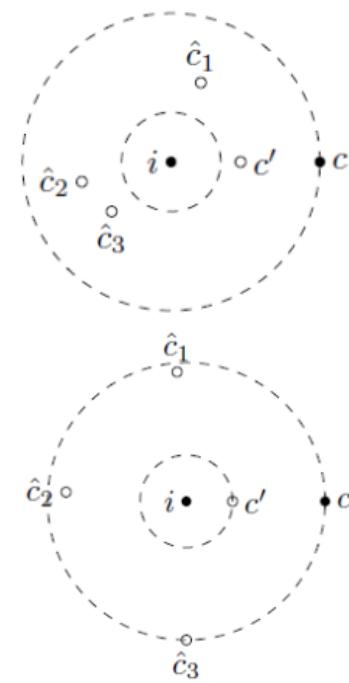
Common VSE

- Image and sentence features are **directly learnt** from scratch



Visual-Semantic Embedding (VSE)

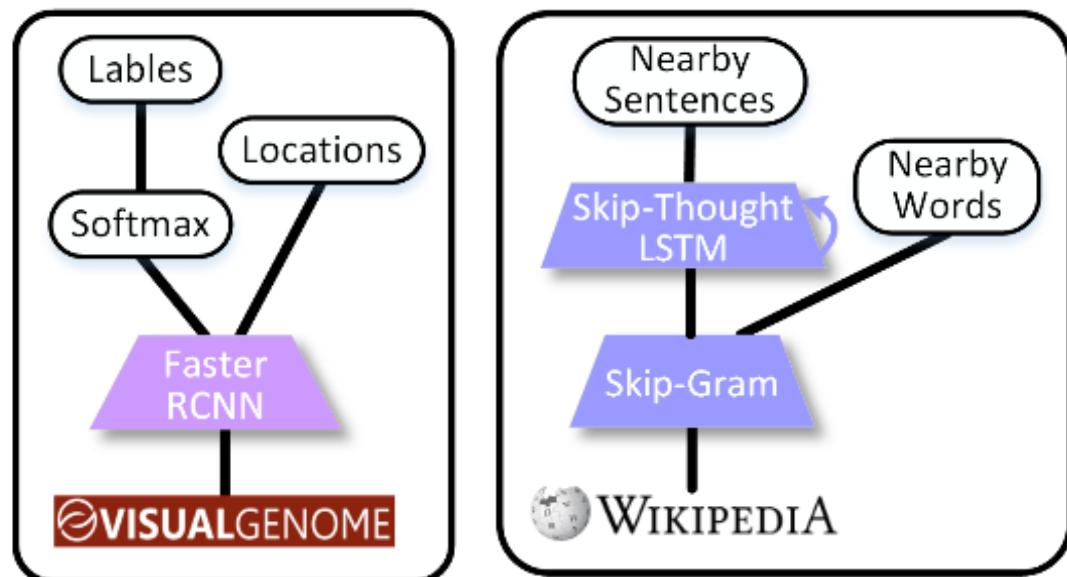
sum loss



max loss

Uncommon VSE

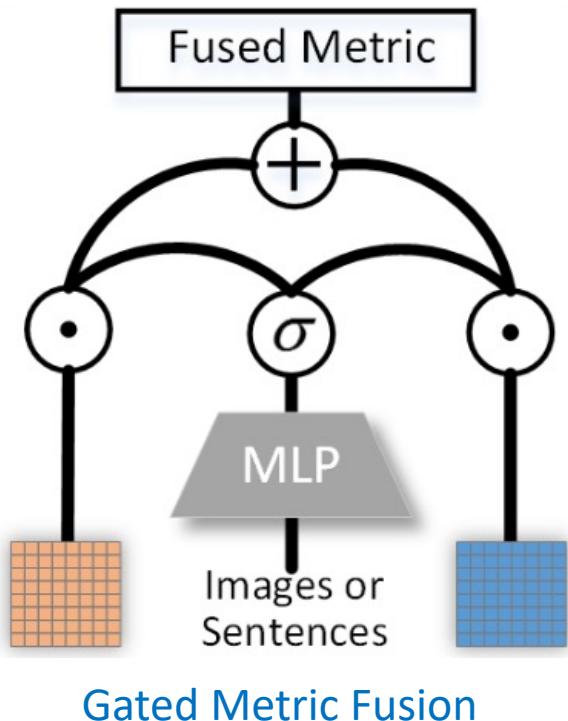
- Use a **pretrained** Faster RCNN to output region-level features
- Use a **pretrained** Skip-Gram to obtain word-level features
- Feed them into another VSE to correlate images and sentences



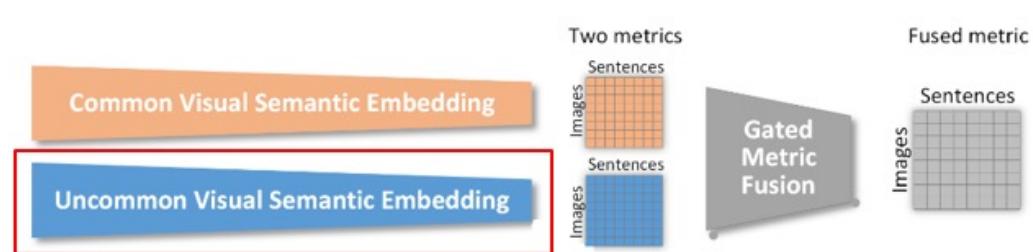
Pretrained Faster-RCNN and Skip-Gram

Similarity Measurement

- The two VSE modules obtain two different similarity metrics for uncommon and common content, respectively
- Perform gated feature fusion with the gated metric fusion module

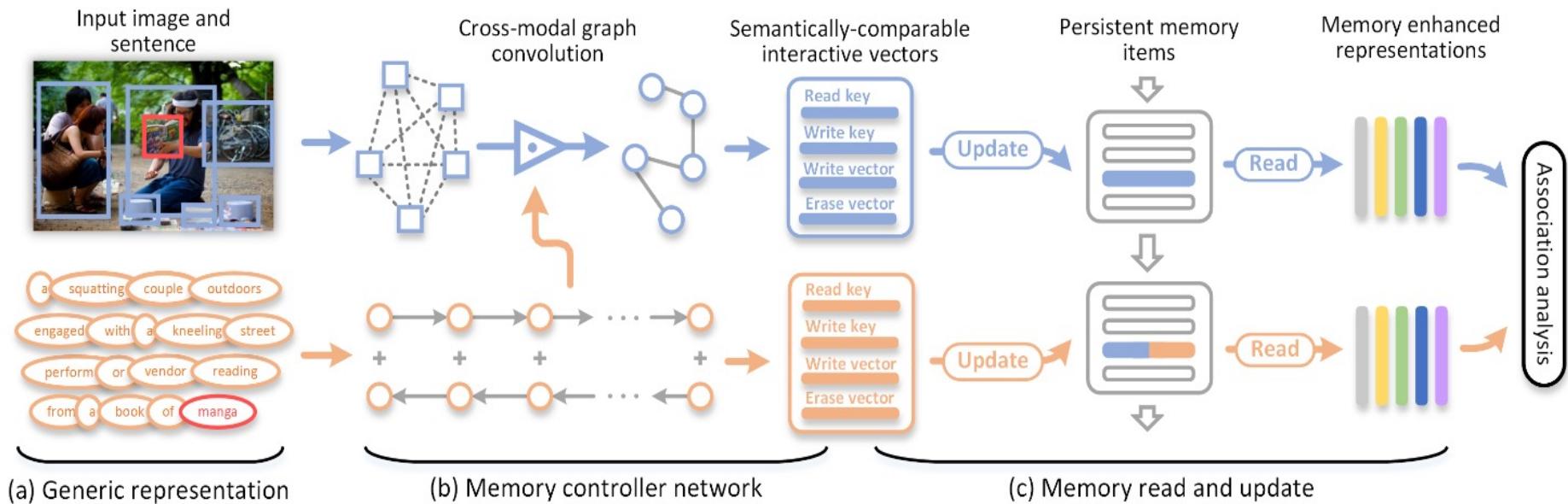


$$t = \sigma(f(\mathbf{x})), \quad \hat{\mathbf{m}} = t \odot \mathbf{m}_0 + (1 - t) \odot \mathbf{m}_1$$



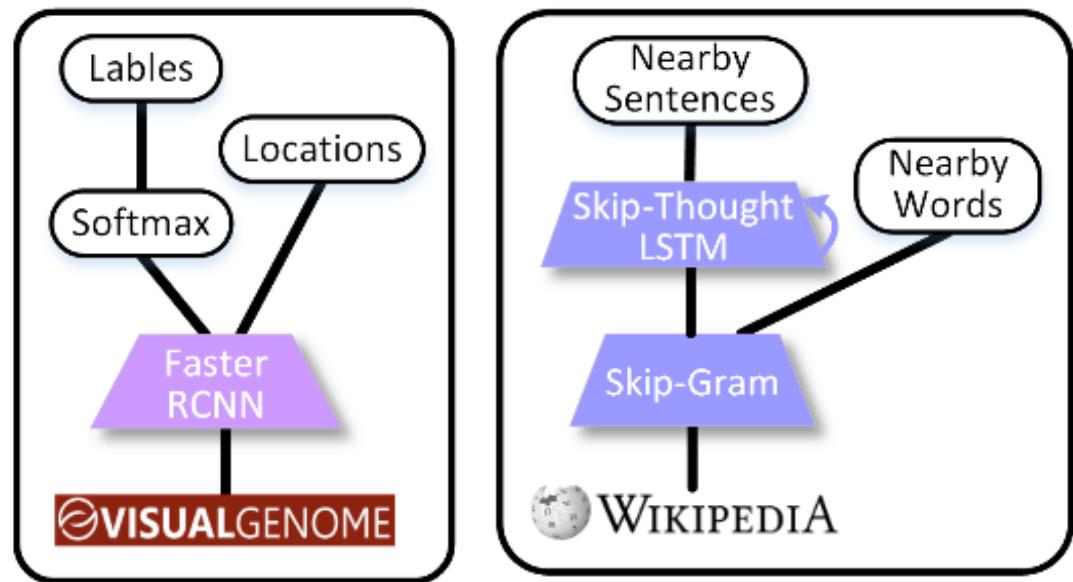
Uncommon VSE >>> Cross-Modal Memory

- Learn to memorize cross-modal shared few-shot object-word features



Generic Representation

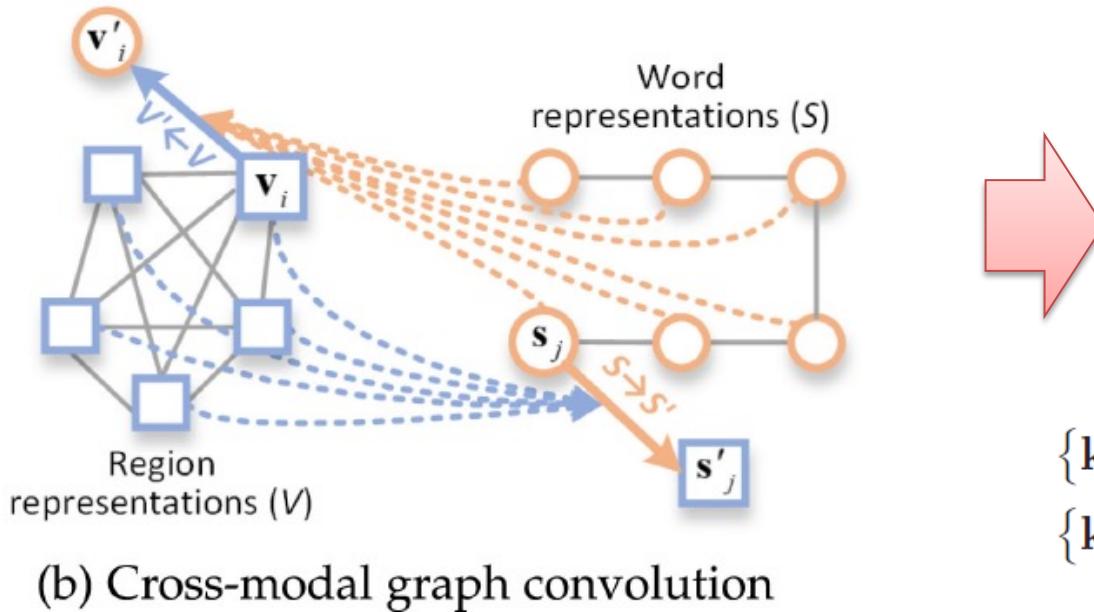
- Use a Faster RCNN pretrained on Visual Genome to output detected instances
- Use a Skip-Gram pretrained on Wikipedia to encode arbitrary words



Pretrained Faster-RCNN and Skip-Gram

Memory Controller Network

- Perform cross-modal **alignment** to make them semantically comparable
- Generate modality-specific interface vectors to connect with shared memory



$$g(\mathbf{s}_j, \mathbf{v}_i) = \alpha(\mathbf{s}_j)^T \varphi(\mathbf{v}_i),$$

$$\mathbf{G}_{ji}^{S \rightarrow S'} = \frac{e^{g(\mathbf{s}_j, \mathbf{v}_i)}}{\sum_i e^{g(\mathbf{s}_j, \mathbf{v}_i)}},$$

$$\mathbf{S}' = \mathbf{G}^{S \rightarrow S'} \mathbf{V} \mathbf{W}^{S \rightarrow S'},$$

Cross-modal graph convolutional network

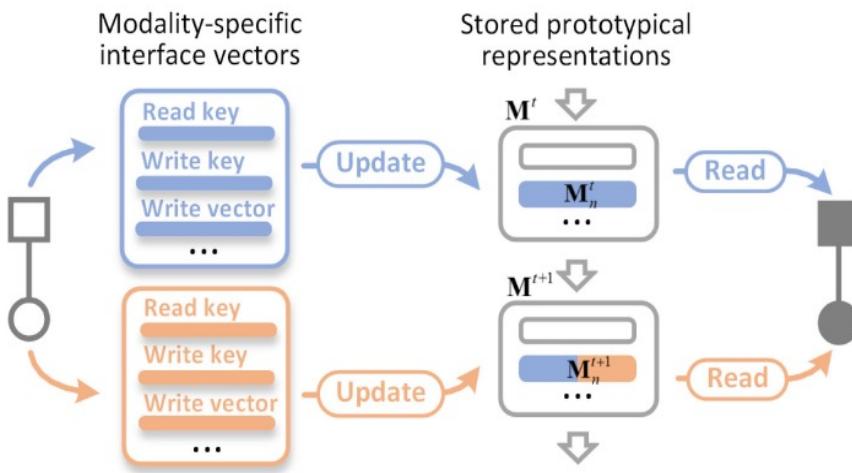
$$\{\mathbf{k}^{Vr}, \beta^{Vr}, \mathbf{k}^{Vw}, \beta^{Vw}, \mathbf{e}^V, \mathbf{u}^V\} = t^V(\mathbf{v}_i),$$

$$\{\mathbf{k}^{Sr}, \beta^{Sr}, \mathbf{k}^{Sw}, \beta^{Sw}, \mathbf{e}^S, \mathbf{u}^S\} = t^S(\mathbf{v}'_i),$$

Interface Vectors

Memory Update and Read

- Generate modality-specific interface vectors to connect with cross-modal shared memory
- Memory items are updated in a persistent manner



$$\theta(\mathbf{k}, \mathbf{M}_n^t, \beta) = \frac{e^{\cos(\mathbf{k}, \mathbf{M}_n^t) \cdot \beta}}{\sum_n e^{\cos(\mathbf{k}, \mathbf{M}_n^t) \cdot \beta}},$$

Memory update:

$$w_n^{Vw} = \theta(\mathbf{k}^{Vw}, \mathbf{M}_n^t, \beta^{Vw}), \quad w_n^{Sw} = \theta(\mathbf{k}^{Sw}, \mathbf{M}_n^t, \beta^{Sw}),$$

$$\mathbf{M}_n^t = \mathbf{M}_n^t \circ (1 - w_n^{Vw} \mathbf{e}^V) + w_n^{Vw} \mathbf{u}^V,$$

$$\mathbf{M}_n^{t+1} = \mathbf{M}_n^t \circ (1 - w_n^{Sw} \mathbf{e}^S) + w_n^{Sw} \mathbf{u}^S,$$

Memory read:

$$\mathbf{r}_i^V = \sum_n w_n^{Vr} \mathbf{M}_n^t, \quad w_n^{Vr} = \theta(\mathbf{k}^{Vr}, \mathbf{M}_n^t, \beta^{Vr}),$$

$$\mathbf{r}_i^S = \sum_n w_n^{Sr} \mathbf{M}_n^t, \quad w_n^{Sr} = \theta(\mathbf{k}^{Sr}, \mathbf{M}_n^t, \beta^{Sr}),$$

Experimental Datasets

- Flickr 30k dataset
 - from the Flickr.com website
 - 31784 images
 - each image has 5 captions
 - use the public training, validation and testing splits, which contain 28000, 1000 and 1000 images, respectively
- Microsoft COCO dataset
 - 82783 images
 - each image has 5 captions
 - use the public training, validation and testing splits, with 82783, 4000 and 1000 images, respectively



1. A man in street racer armor is examining the tire of another racers motor bike.
2. The two racers drove the white bike down the road.
3.

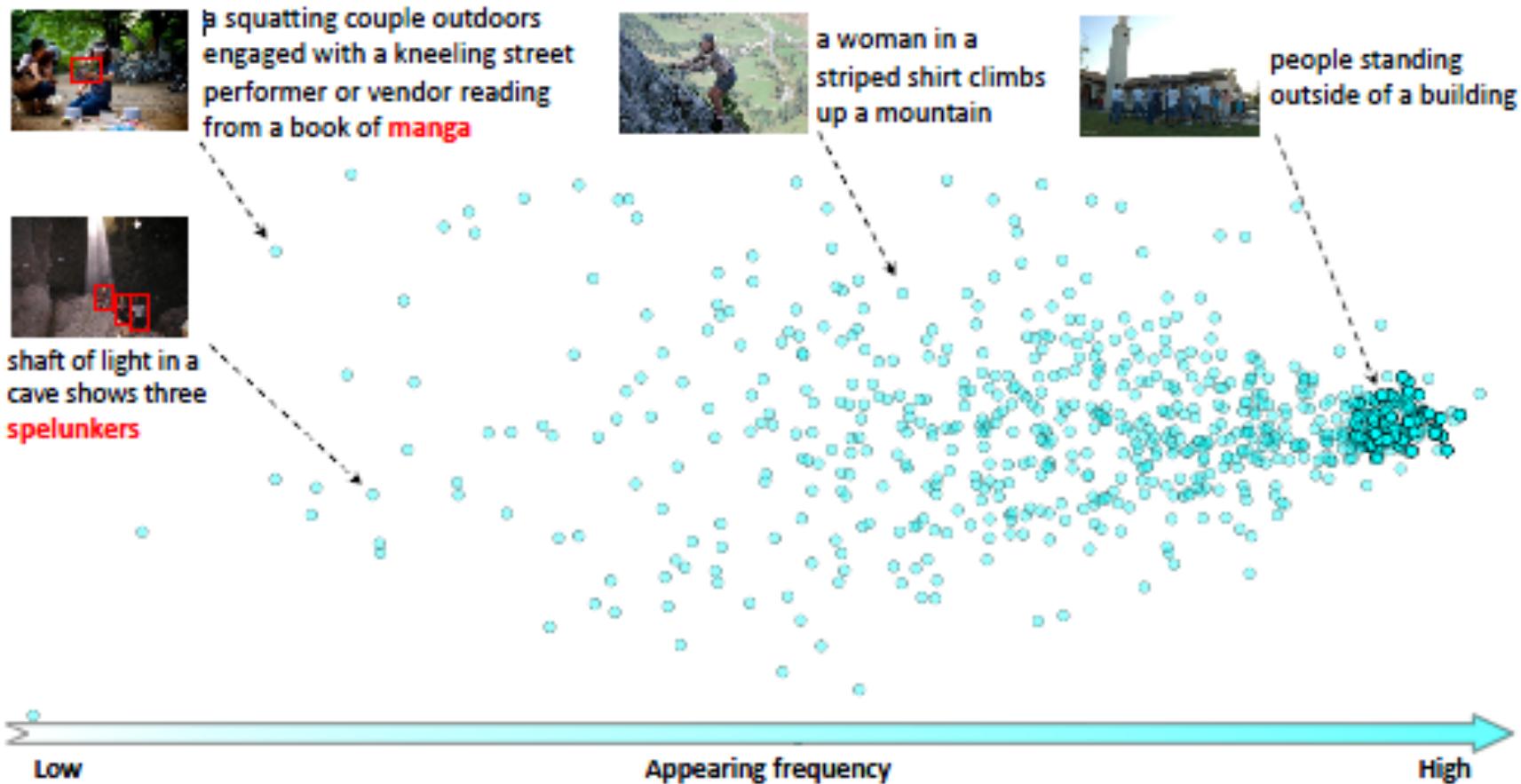
Ablation Study

Few-shot image and sentence matching by ablation models on the Flickr30k datasets.

Method	Flickr30k dataset							
	Image Annotation			Image Retrieval			mR	
	R@1	R@5	R@10	R@1	R@5	R@10		
Cross-modal region-word alignment	v2v (nr)	61.7	86.2	92.6	40.8	70.0	80.1	71.9
	v2v	66.5	88.2	92.9	41.9	72.2	82.0	74.0
	s2s (nr)	53.2	80.8	90.3	40.6	69.1	78.3	68.7
	s2s	65.3	90.6	94.7	47.7	76.7	84.1	76.5
	bd (tk)	67.9	89.7	95.8	48.4	77.5	85.2	77.4
	bd (p)	68.1	90.9	96.3	48.5	78.1	85.9	78.0
	bd	67.6	90.0	95.5	47.7	77.2	85.1	77.2
Cross-modal shared memory	mem	1.3	6.5	12.6	1.0	4.7	8.3	5.7
	v2v + mem	81.6	96.0	97.8	49.3	77.2	84.2	81.0
	s2s + mem	80.0	95.5	98.2	50.2	76.8	84.7	80.9
	bd + mem (ns)	68.3	89.5	94.1	44.3	73.9	83.0	75.5
	bd + mem (np)	65.1	88.7	93.7	45.8	76.1	83.2	75.4
	bd + mem (d)	79.6	94.8	97.1	49.7	75.9	84.0	80.2
	bd + mem (m)	84.7	96.0	97.9	52.8	79.0	82.7	82.2
	bd + mem	85.2	96.7	98.4	53.8	79.8	86.8	83.5
Two-stream similarity fusion	lb (nf)	43.3	71.7	82.0	32.3	61.1	72.1	60.4
	lb	52.9	80.4	87.8	39.6	69.9	79.6	68.4
	bd + mem + lb (s)	75.6	93.3	96.9	53.1	80.5	87.5	81.1
	bd + mem + lb (f)	80.2	94.7	97.5	53.9	80.9	88.2	82.6
	bd + mem + lb (ft)	84.6	95.1	98.4	56.0	82.0	89.1	84.3
	bd + mem + lb	85.4	96.9	98.4	55.1	81.7	88.4	84.2

Ablation Study

Two-dimensional visualization of learned memory items, few-shot content are marked as red



Few-shot Image and Sentence Matching

Few-shot image and sentence matching on the Flickr30k datasets

k	N	Method	Flickr30k dataset						
			Image Annotation			Image Retrieval			mR
			R@1	R@5	R@10	R@1	R@5	R@10	
1	321 / 754	SCAN [29]	56.7	86.1	90.9	37.4	59.2	72.3	67.1
		VSRN [31]	61.9	84.1	90.5	41.1	69.2	77.6	70.7
		GVSE* [16]	62.3	88.9	92.9	46.4	73.5	83.2	74.5
		ACMM* [17]	73.0	91.3	96.4	40.5	66.7	77.6	74.2
		ACMM	74.6	93.5	97.6	49.0	73.6	83.8	78.6
5	678 / 973	SCAN [29]	62.2	87.8	93.4	37.0	64.2	74.3	69.8
		VSRN [31]	64.4	86.5	91.9	43.1	72.3	79.2	72.9
		GVSE* [16]	64.6	90.8	94.8	47.6	75.7	84.8	76.4
		ACMM* [17]	76.6	93.2	97.6	42.3	68.0	76.8	75.8
		ACMM	79.0	94.5	97.6	50.0	74.2	84.9	80.0
10	969 / 1144	SCAN [29]	62.6	88.3	93.2	39.3	66.3	75.9	70.9
		VSRN [31]	65.0	86.6	92.3	45.4	72.7	80.3	73.7
		GVSE* [16]	64.5	89.9	94.4	48.4	76.1	84.5	76.3
		ACMM* [17]	77.6	93.7	97.6	45.5	69.6	77.8	77.0
		ACMM	80.2	94.6	97.6	51.3	76.1	85.7	80.9

k: the minimum appearing frequency of all words in a sentence, which is used to obtain different k-shot test sets.

N: the total number of k-shot words in the k-shot test set of Flickr30k dataset.

Conventional Image and Sentence Matching

Conventional image and sentence matching on the Flickr30k and MSCOCO datasets

Method	Flickr30k dataset						MSCOCO dataset						mR		
	Image Annotation			Image Retrieval			Image Annotation			Image Retrieval					
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10			
VGG	RNN+FV [30]	34.7	62.7	72.6	26.2	55.1	69.2	53.4	40.8	71.9	83.2	29.6	64.8	80.5	61.8
	m-CNN [37]	33.6	64.1	74.9	26.2	56.3	69.6	54.1	42.8	73.1	84.1	32.6	68.6	82.8	64.0
	OEM [53]	-	-	-	-	-	-	-	46.7	78.6	88.9	37.9	73.7	85.9	68.6
	VQA [33]	33.9	62.5	74.5	24.9	52.6	64.8	52.2	50.5	80.1	89.7	37.0	70.9	82.9	68.5
	RTP [43]	37.4	63.1	74.3	26.0	56.0	69.3	54.3	-	-	-	-	-	-	-
	sm-LSTM [18]	42.5	71.9	81.5	30.2	60.4	72.3	59.8	53.2	83.1	91.5	40.7	75.8	87.4	72.0
	DSPE [57], [58]	43.2	71.6	79.8	31.7	61.3	72.4	60.0	54.9	84.0	92.2	43.3	76.4	87.5	73.1
	2WayNet [6]	49.8	67.5	-	36.0	55.6	-	-	55.8	75.2	-	39.7	63.3	-	-
ResNet	CSE [69]	44.6	74.3	83.8	36.9	69.1	79.6	64.7	56.3	84.4	92.2	45.7	81.2	90.6	75.1
	RRF [34]	47.6	77.4	87.1	35.4	68.3	79.9	66.0	56.4	85.3	91.5	43.9	78.1	88.6	73.9
	DAN [39]	55.0	81.8	89.0	39.4	69.2	79.1	68.9	-	-	-	-	-	-	-
	CHAIN [61]	-	-	-	-	-	-	-	59.4	88.0	94.2	43.5	79.8	90.2	75.9
	UVSE [64]	-	-	-	-	-	-	-	64.3	89.2	94.8	48.3	81.7	91.2	78.3
	VSE++ [7]	52.9	79.1	87.2	39.6	69.6	79.5	68.0	64.6	89.1	95.7	52.0	83.1	92.0	79.4
	DPCNN [71]	55.6	81.9	89.5	39.1	69.2	80.9	69.4	65.6	89.8	95.5	47.1	79.9	90.0	78.0
	PVSE [49]	-	-	-	-	-	-	-	69.2	91.6	96.6	55.2	86.5	93.7	82.1
Faster RCNN	SCO [19], [20]	58.0	84.5	90.5	43.9	72.9	81.6	71.9	71.3	93.8	98.0	58.2	88.8	95.3	84.2
	ACMM	65.0	87.1	93.6	45.6	74.2	83.7	74.9	77.7	95.2	99.0	60.0	90.1	96.9	86.5
	LIWE [62]	69.6	90.3	95.6	51.2	80.4	87.2	79.1	73.2	95.5	98.2	57.9	88.3	94.5	84.6
	SCAN [29]	67.4	90.3	95.8	48.6	77.7	85.2	77.5	72.7	94.8	98.4	58.8	88.4	94.8	84.7
	GVSE* [16]	68.5	90.9	95.5	50.6	79.8	87.6	78.8	72.2	94.1	98.1	60.5	89.4	95.8	85.0
	VSRN [31]	71.3	90.6	96.0	54.7	81.8	88.2	80.4	76.2	94.8	98.2	62.8	89.7	95.1	86.1
ACMM*	[17]	85.2	96.7	98.4	53.8	79.8	86.8	83.5	84.1	97.8	99.4	60.7	88.7	94.9	87.6
	ACMM	85.4	96.9	98.4	55.1	81.7	88.4	84.2	84.4	97.9	99.4	63.4	90.3	95.7	88.5

Error Analysis

the people are **quietly** listening while the story of the ice **cabin** was **explained** to them



275

a woman **acts** out a dramatic scene in public behind yellow **caution** tape



64

a man wearing **revolutionary period** clothes is **ringing** a **bell**



88

two women **ascend** a telephone pole with **special** boots and **straps** on their **waists**



51

an **oriental** traveler awaits his turn at the currency exchange



511

battling between the sexes who **will win**



301

fall shoppers and **bistro** food **lovers** caught in the **ebb** and **flow** of the city



58

Acknowledgement

Some of the materials in these slides are drawn inspiration from:

- Shubhendu Trivedi and Risi Kondor, University of Chicago, Deep Learning Course
- Hung-yi Lee, National Taiwan University, Machine Learning and having it Deep and Structured course
- Xiaogang Wang, The Chinese University of Hong Kong, Deep Learning Course
- Fei-Fei Li, Standord University, CS231n Convolutional Neural Networks for Visual Recognition course

Next time

- Graph Neural Networks

Questions?

Thank You !

