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

2022

"Deep Learning Lecture"

Lecture 10：Deep Cognitive Networks

Yan Huang

Center for Research on Intelligent Perception and Computing (CRIPAC)

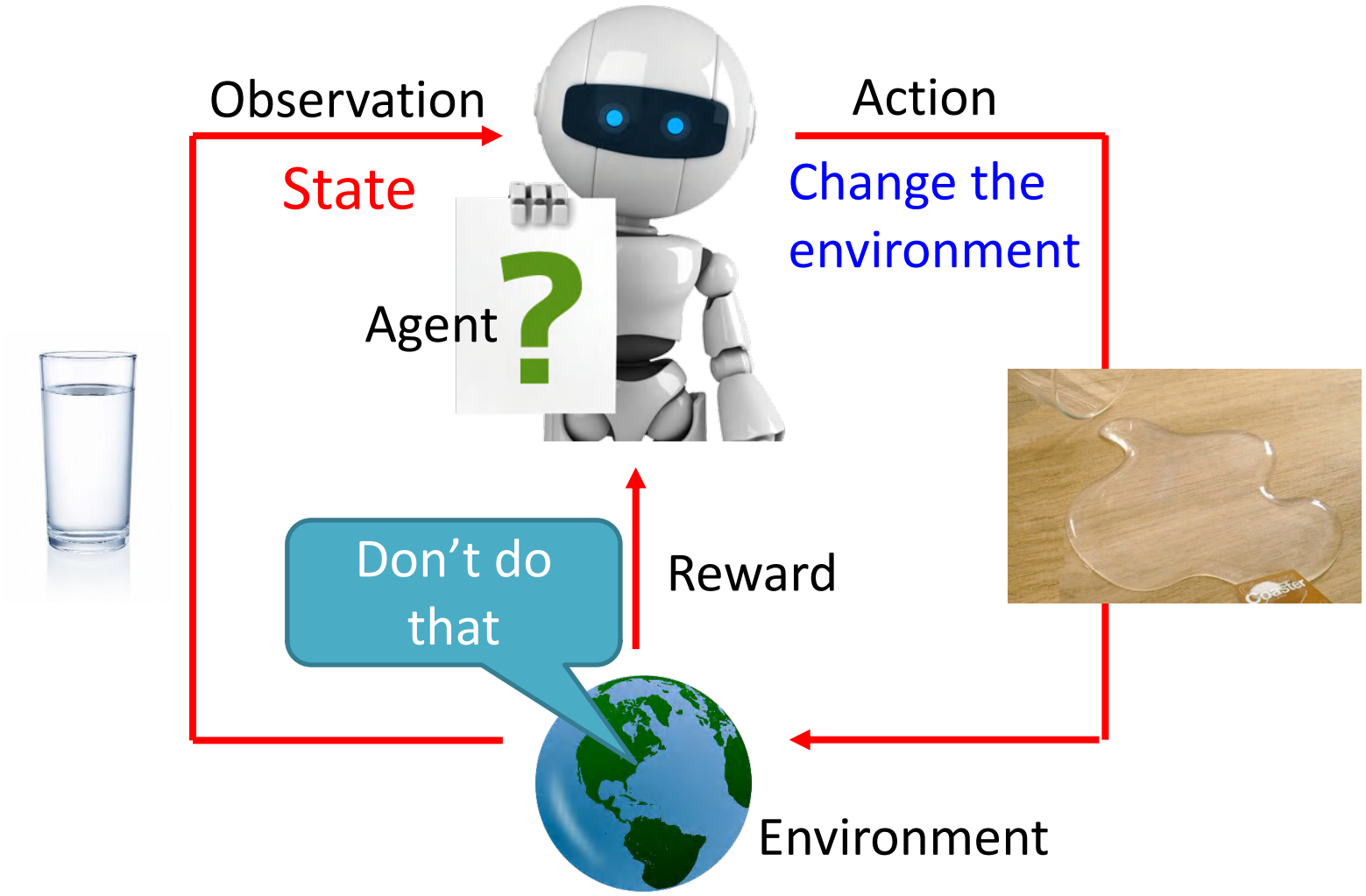
National Laboratory of Pattern Recognition (NLPR)

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

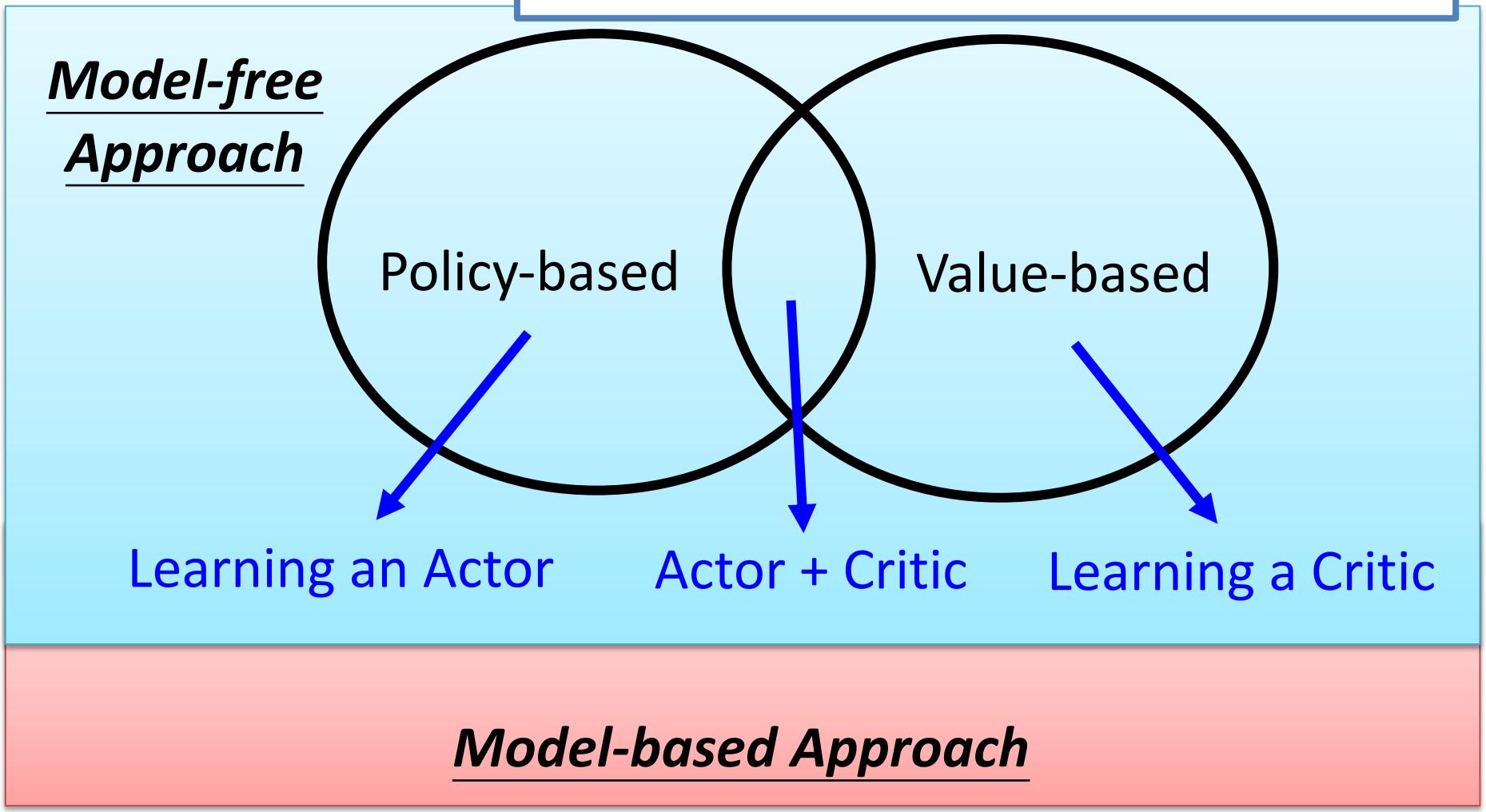


Scenario of Reinforcement Learning

https://yoast.com/how-to-clean-site-structure/

Methods of Reinforcement Learning

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



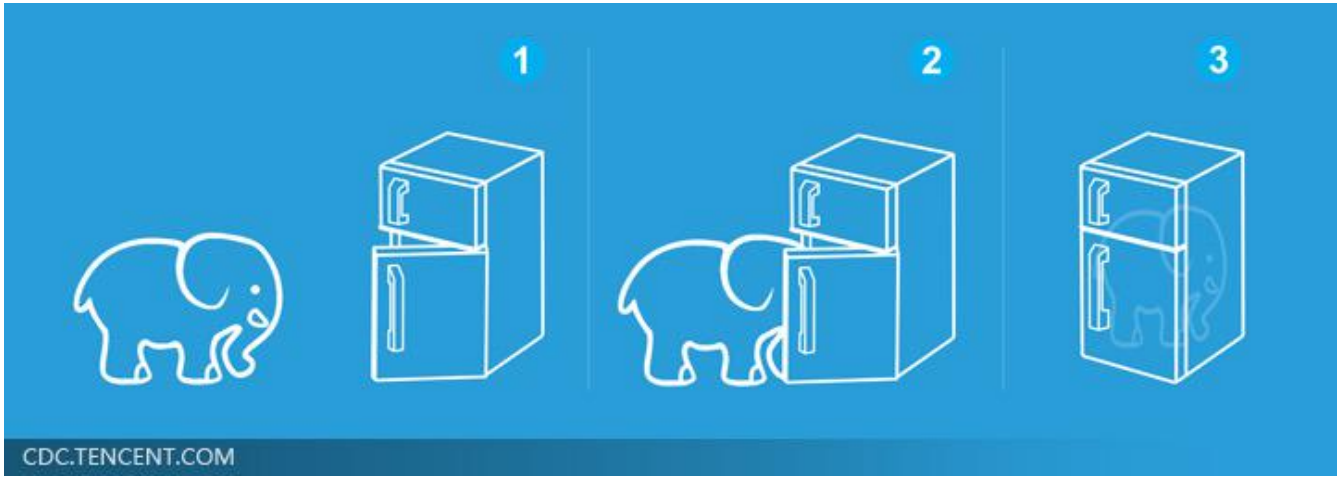
Three Steps for Deep Learning

Step 1: Neural Network as Actor

Step 2: goodness of function

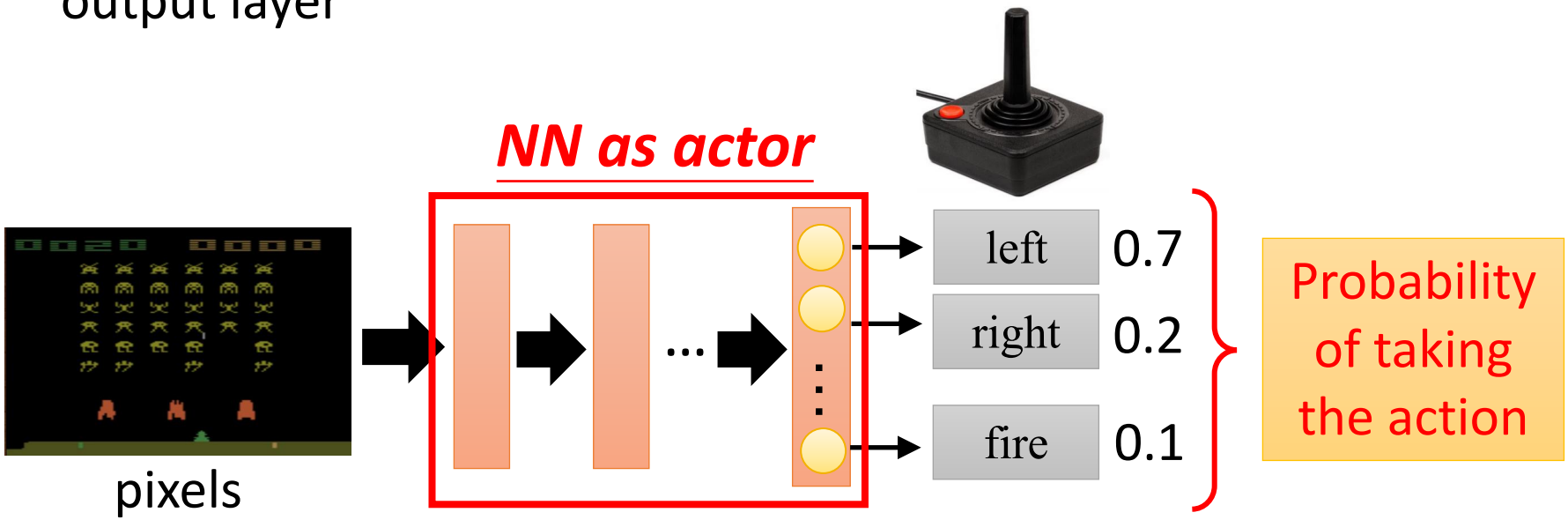
Step 3: pick the best function

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

Gradient Ascent

- Problem statement

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

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

- Gradient ascent
  - Start with θ^0
  - $\theta^1 \leftarrow \theta^0 + \eta \nabla \bar{R}_{\theta^0}$
  - $\theta^2 \leftarrow \theta^1 + \eta \nabla \bar{R}_{\theta^1}$
  - ……

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

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.

$$= \sum_\tau R(\tau)P(\tau|\theta)\nabla logP(\tau|\theta)$$

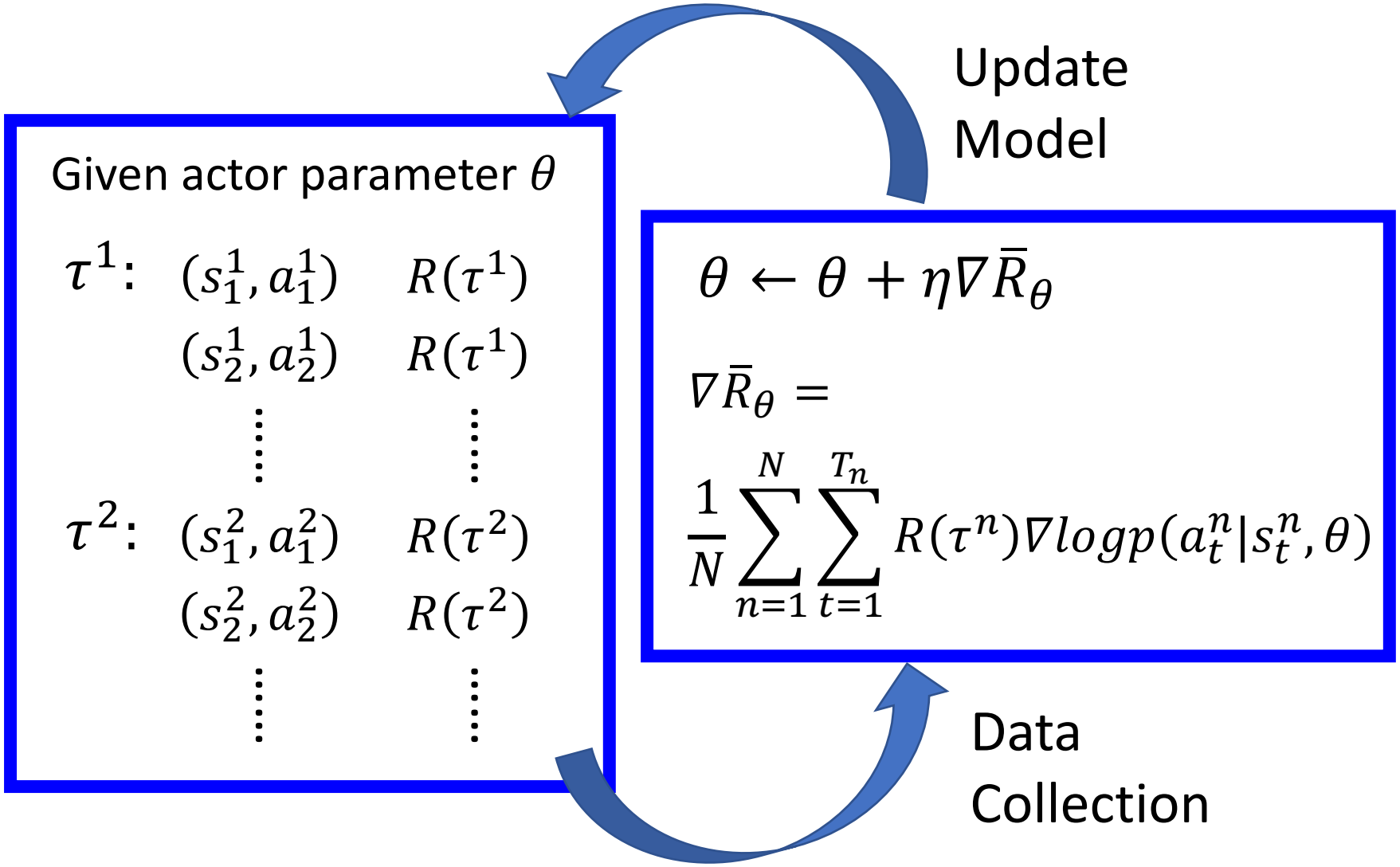
$$\frac{dlog\big(f(x)\big)}{dx} = \frac{1}{f(x)}\frac{df(x)}{dx}$$

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

Use π_θ to play the game N times,

Obtain $\{\tau^1, \tau^2, \cdots, \tau^N\}$

Policy Gradient



Critic

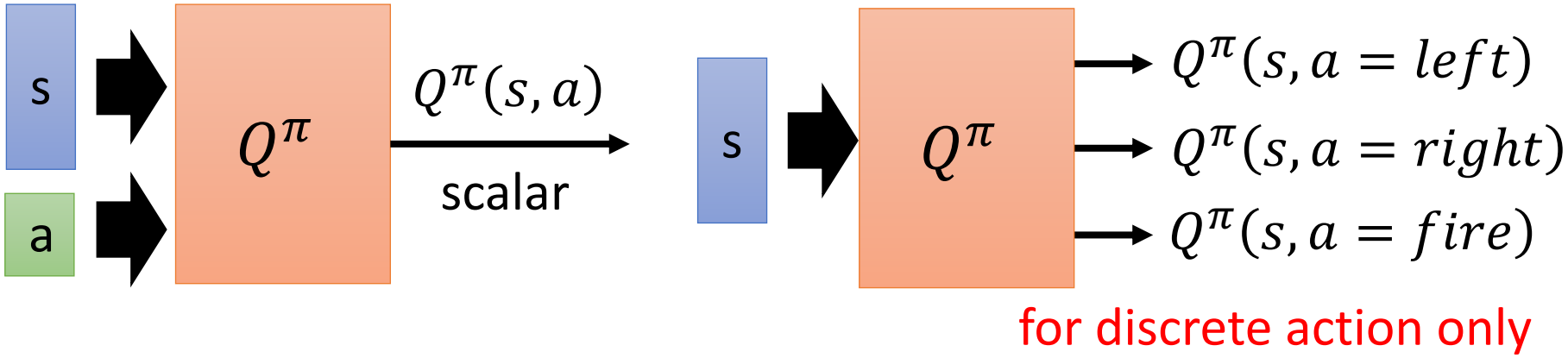
- A critic does not determine the action
- Given an actor π, it evaluates the how good the actor is

http://combiboilersleeds.com/picaso/critics/critics-4.html

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



for discrete action only

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 -> infinity

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

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):

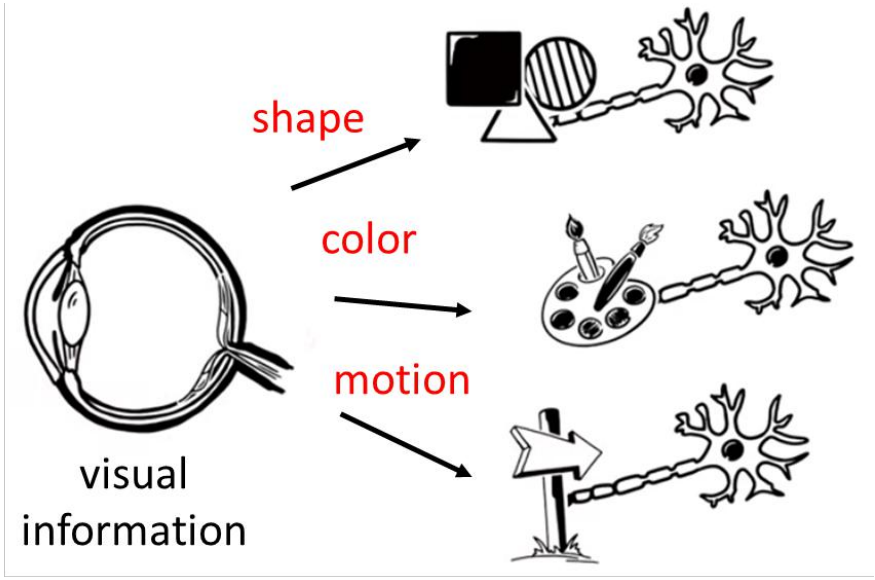
$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot); s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i) \right]$$

Outline

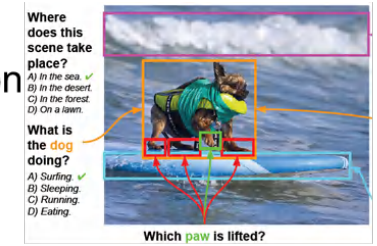
1. Course Review
2. Deep Cognitive Networks
3. Attention Models
4. Memory Models

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%)

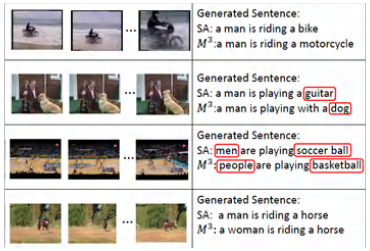


Visual question answering, visual dialog



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

Visual captioning, cross-modal retrieval



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

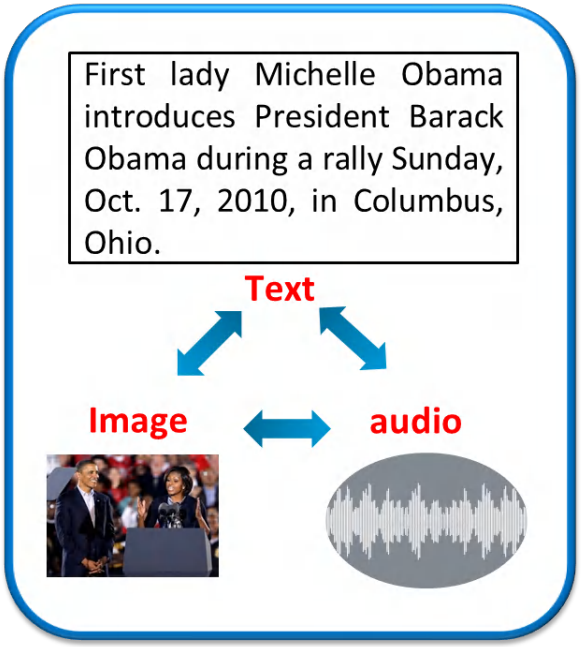
Performance on complex vision tasks

[Zhu et al., CVPR15; Li et al., ICCV17; Wang et al., CVPR18; Liu et al., ICCV17]

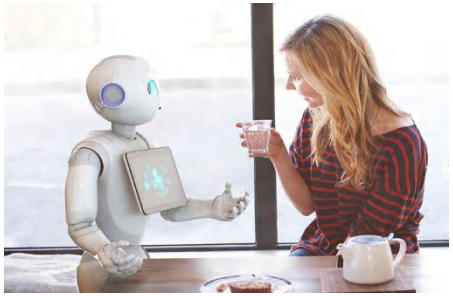
Challenges in Complex Scenarios

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

autonomous driving



Multimodal surveillance

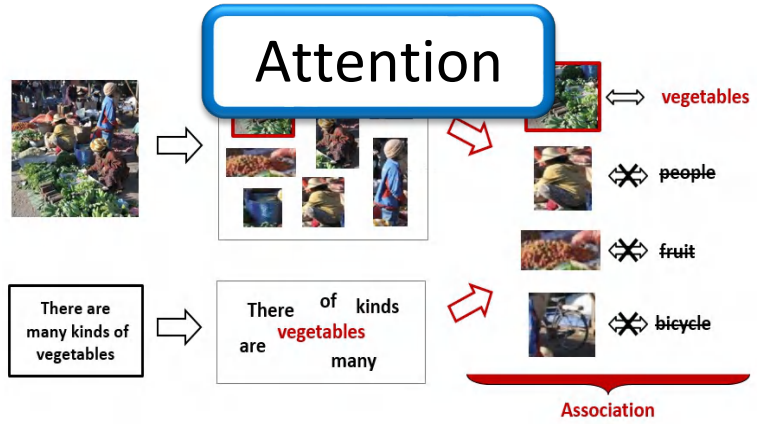


Human-robot interaction

Smart home

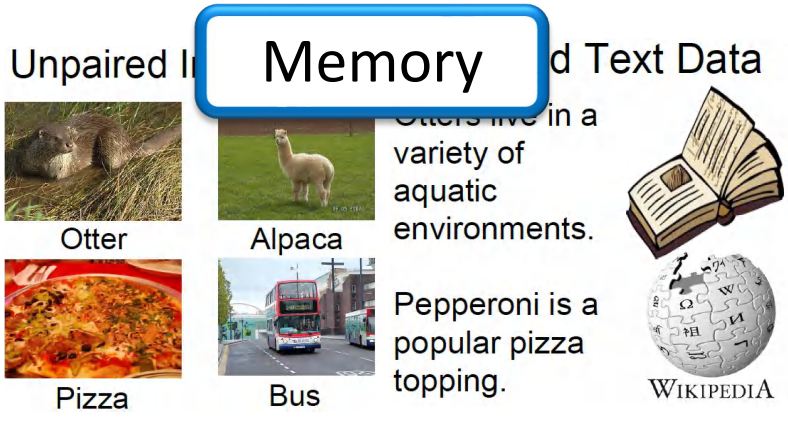
[Karpathy et al., CVPR15, Wang et al., ICCV13, Gao et al., NeurIPS15]

Challenges in Complex Scenarios



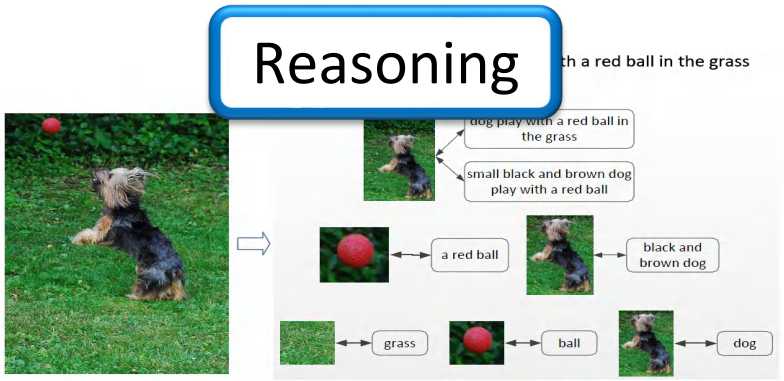
Redundant and task-irrelevant content

Content redundancy reduction



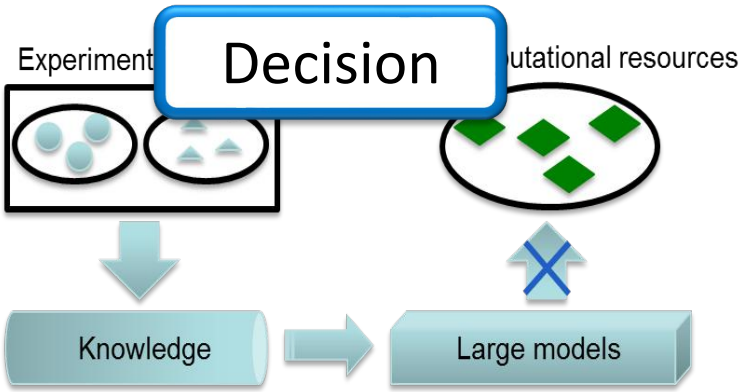
Annotating paired multimodal data

Cross-modal few-shot data



Relations exist in different granularities

Complex cross-modal relation

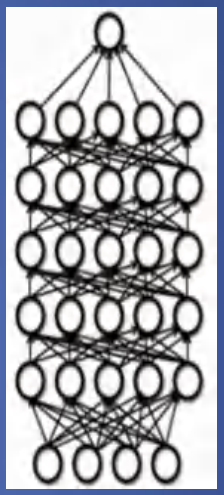


Hard to adapt to new scenarios

Poor model generalization

Cognitive Psychology

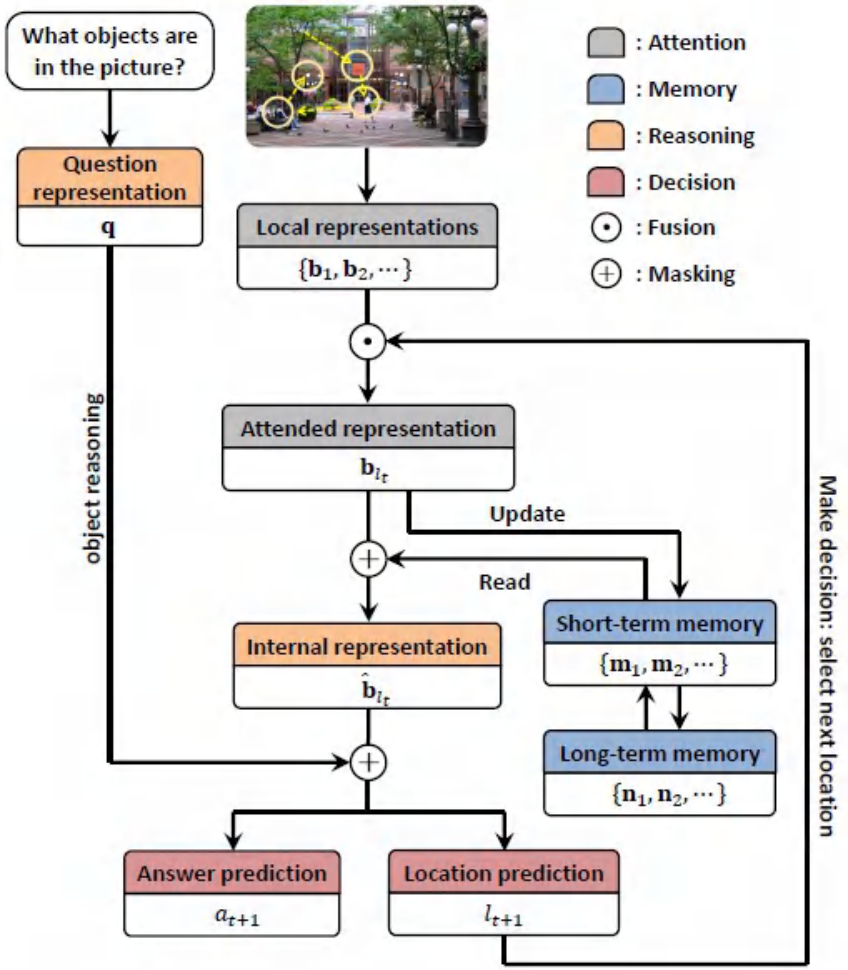
Nicole Conner



https://viquepedia.com/disciplines/cognitive-psychology

Deep Cognitive Networks

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

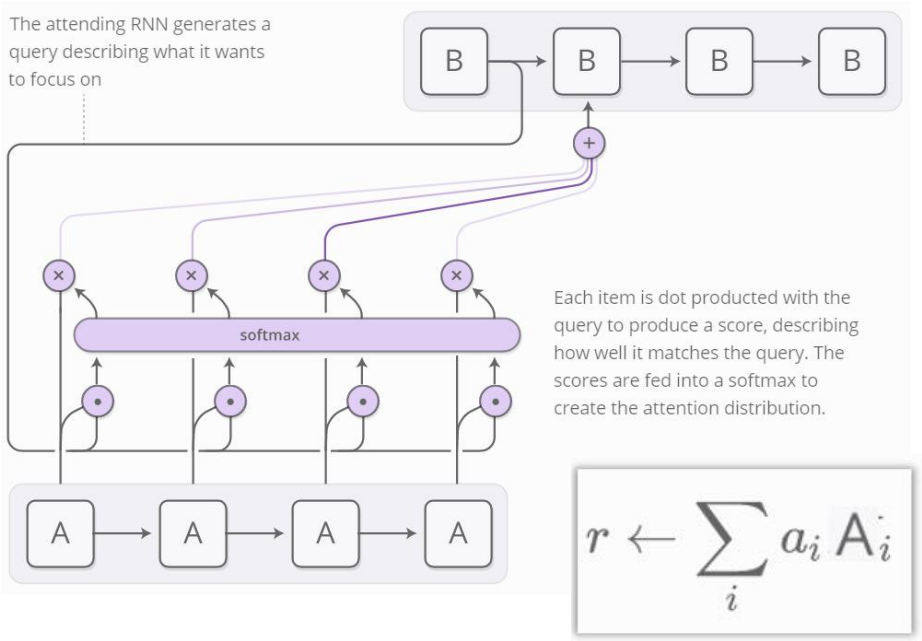


Cognitive functions

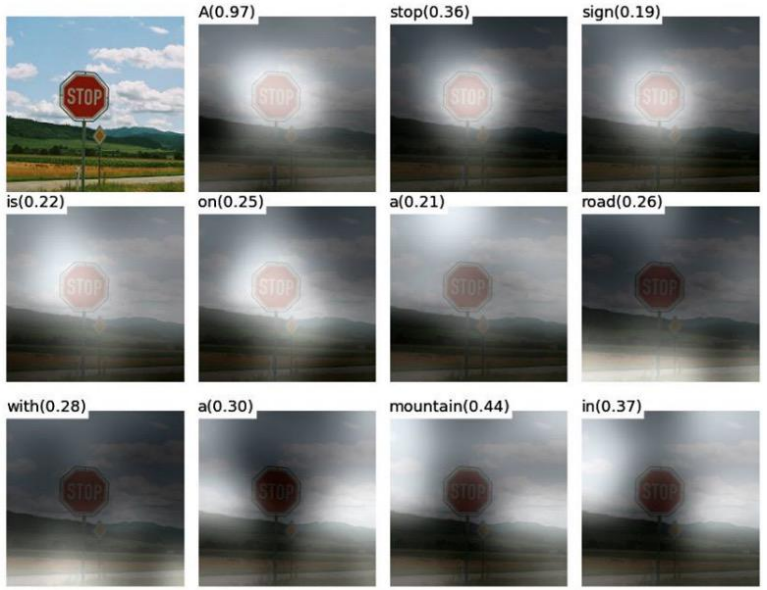
few-shot Learning, knowledge Transfer, relation reasoning, decision making

Recent Progress—Attention

- First predict weights for all components and then weighted sum them
- Hard and soft versions, useful in various complex vision tasks



Details of soft attention

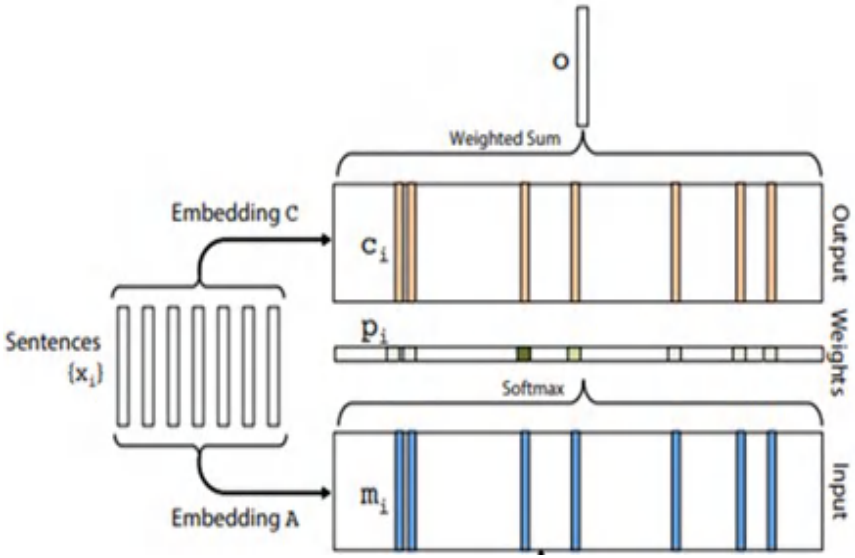


Attention maps in image captioning

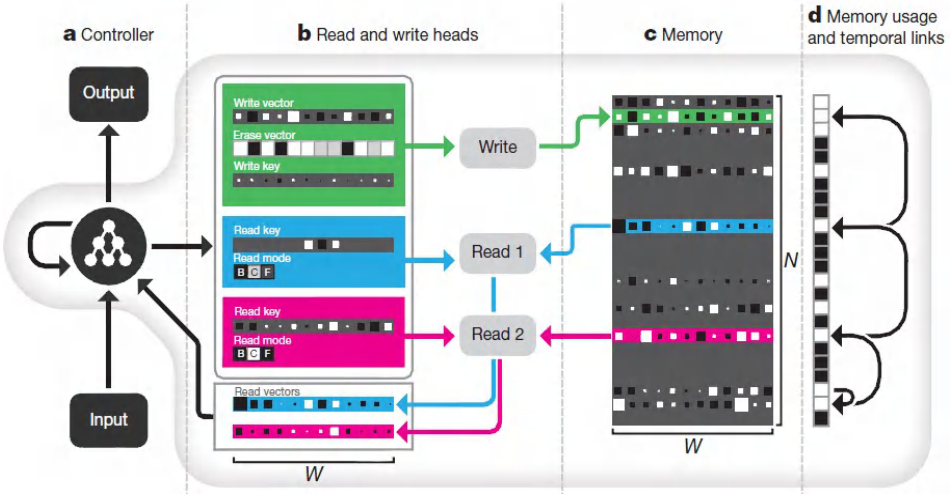
[https://distill.pub/2016/augmented-rnns/, Xu et al., ICML15]

Recent Progress—Memory

- 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)

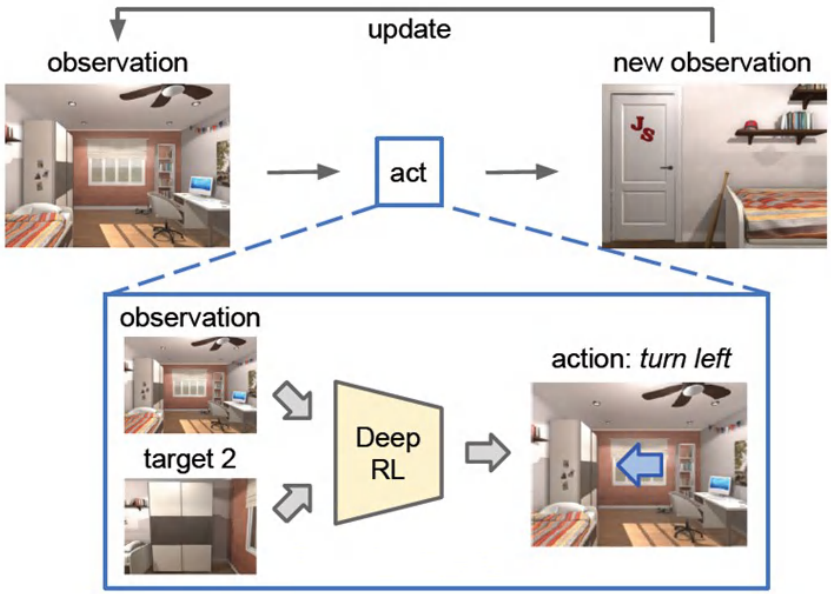


Neural Turing memory
(memory items can be updated)

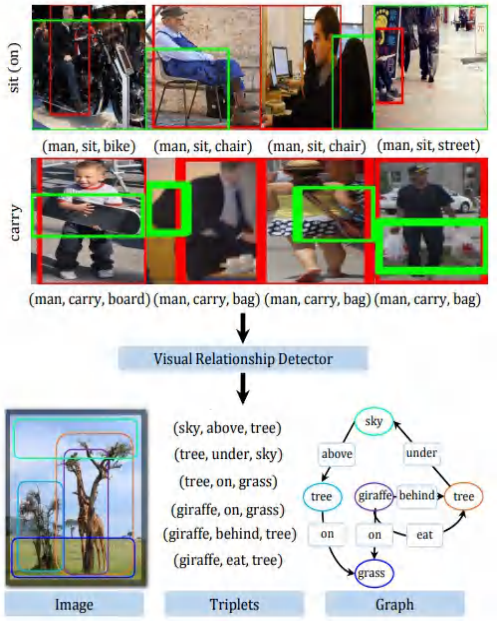
[Sukhbaatar et al., NeurIPS15; Graves et al., Nature16]

Recent Progress—Reasoning

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



Target reasoning and decision making



Relation reasoning

[Zhu et al., arXiv17, Dai et al., CVPR17]

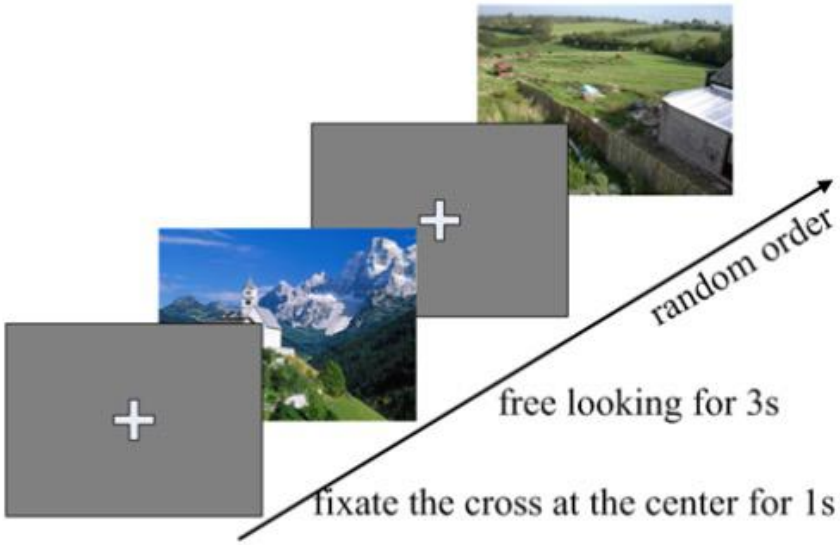
Outline

1. Course Review
2. Deep Cognitive Networks
3. Attention Models
4. Memory Models


WELLS FARGO

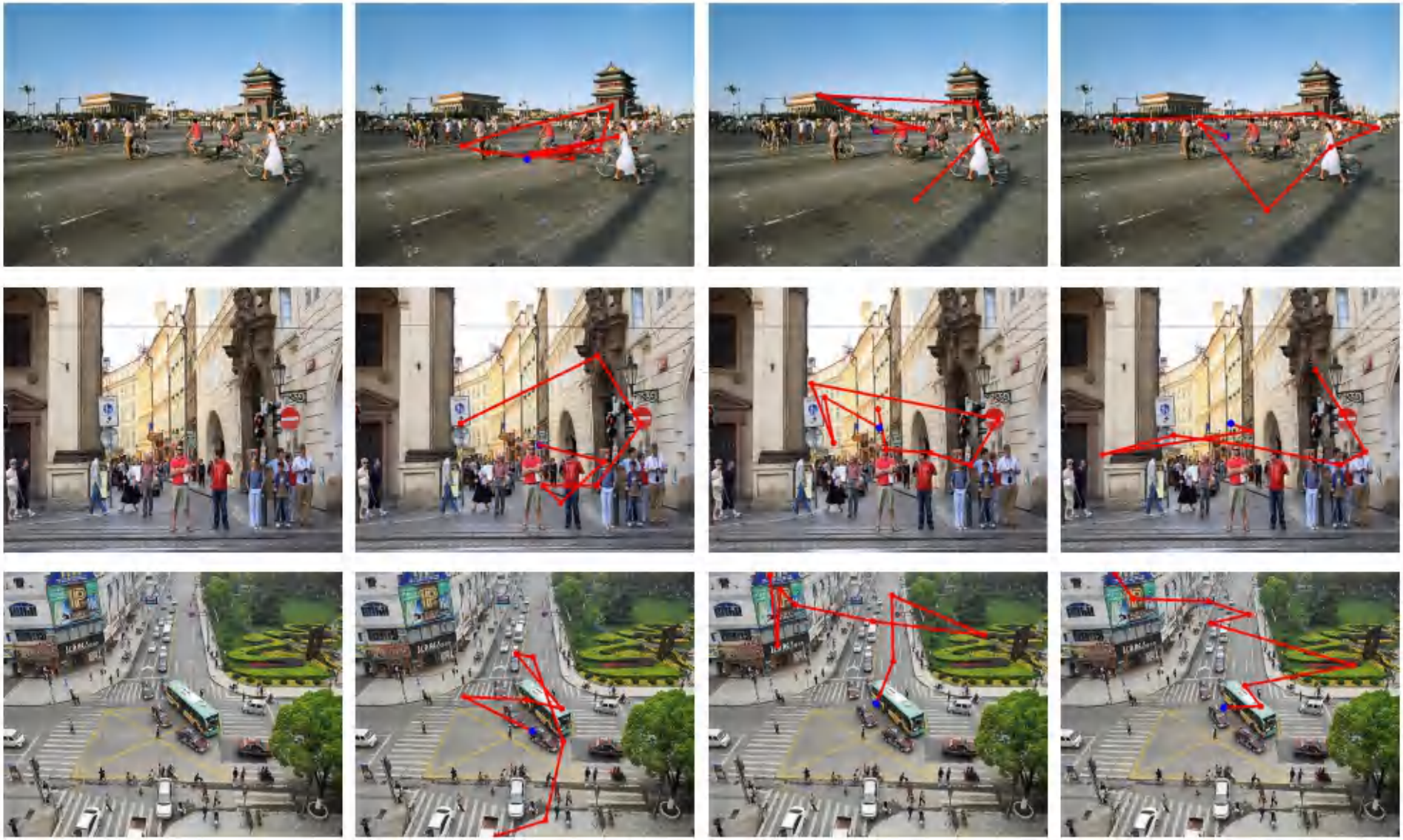

WELLS
FARGO

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

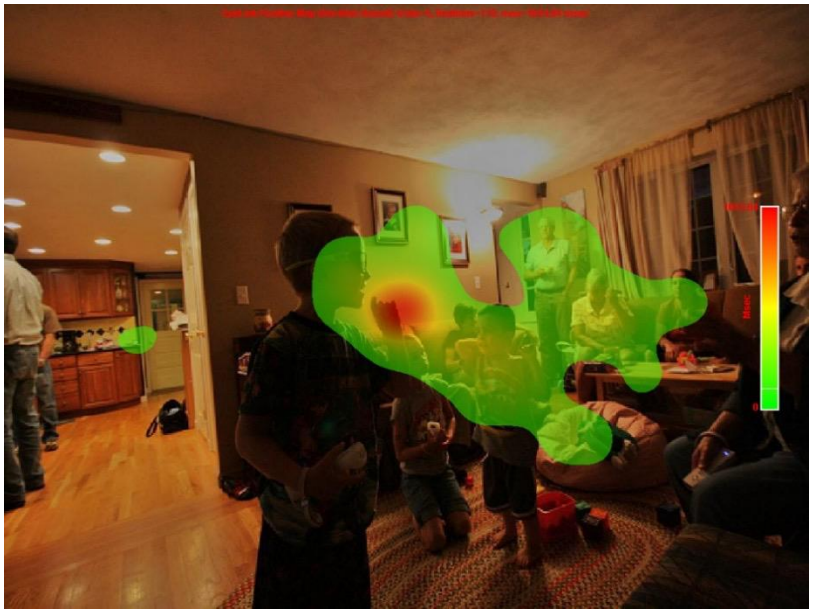


Given Image | No. 1 | No. 2 | No. 3

Task-Driven Saccadic Scanpath

Tasks:

1. 房间里有一群人在嬉闹
2. 晚饭过后一家人在客厅里享受悠闲的时光
3. 在一个灯光昏暗的房间里有许多小孩在玩耍
4. 屋子里一群孩子围在一起玩耍
5. 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

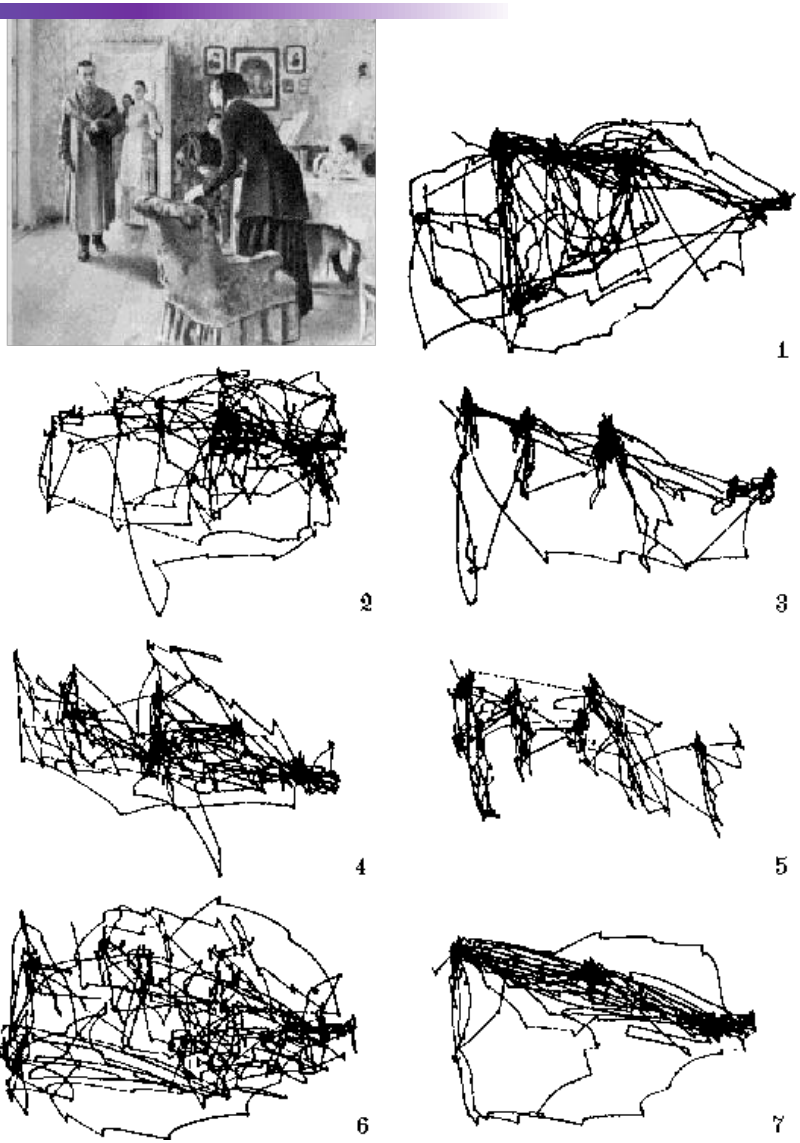
注意在人眼视觉信息处理过程中体现为信息的选择与过滤，从而减少向高级皮层传递的信息，避免冗余或者噪声信息对高级视觉任务的干扰，这种选择过程发生在视觉通道的多个阶段，而且这种选择过程既有各阶段视觉特征的刺激，也有高级皮层向下的反馈调节，对选择性注意的研究不仅有理论上的意义，而且有现实的迫切需求

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



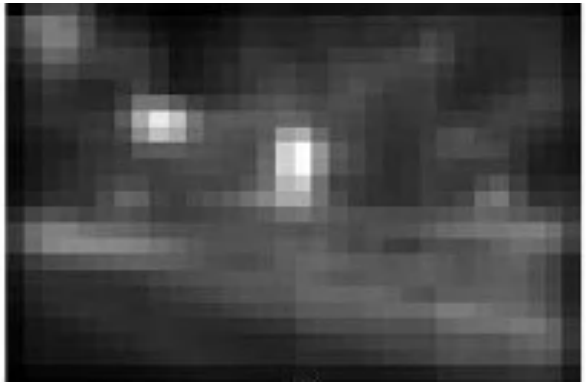
Yarbus, A. L. (1967). **Eye Movements and Vision**. New York: Plenum.

Early Computational Models of Visual Attention

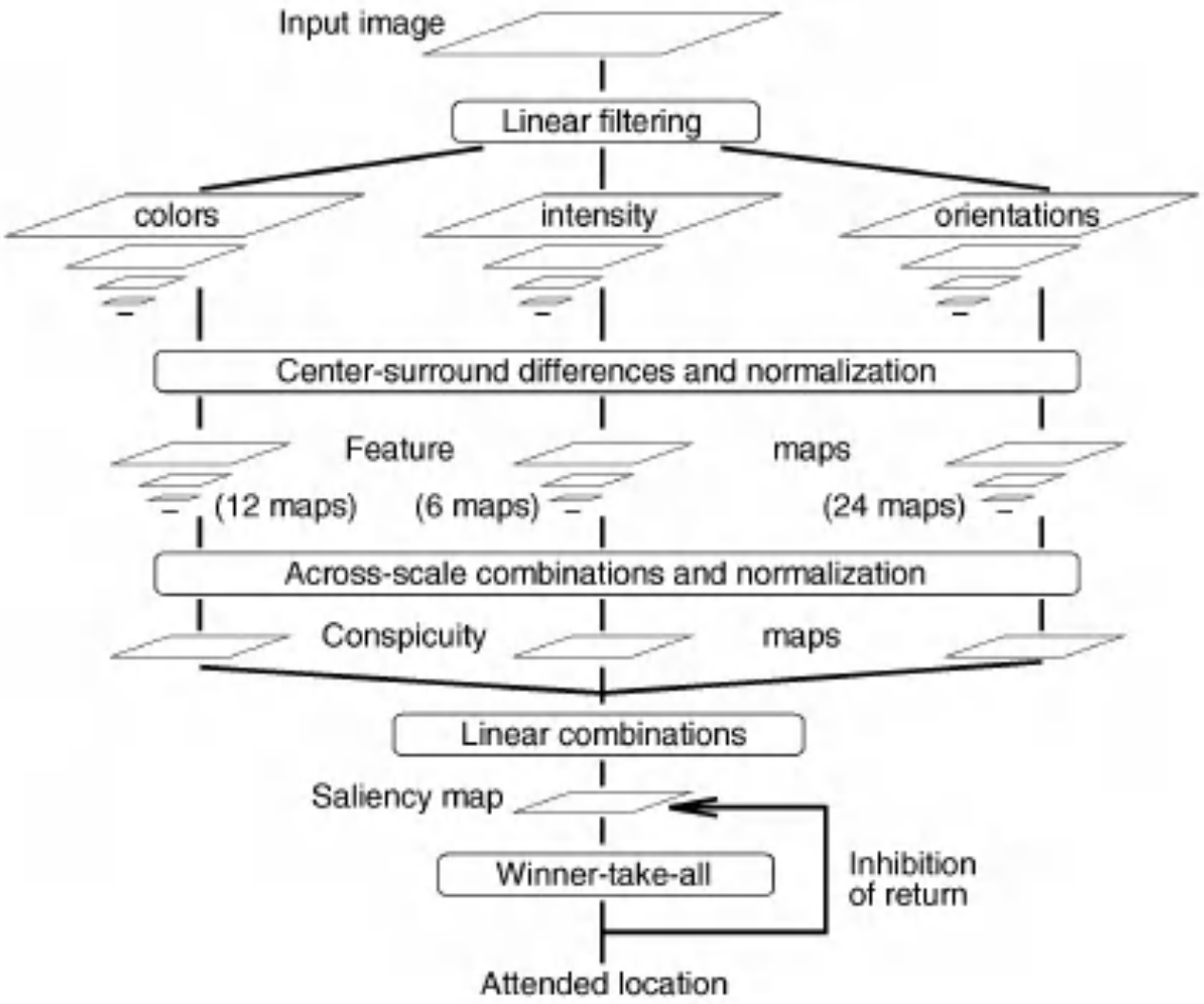
Visual Saliency

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

Li Zhaoping@UCL

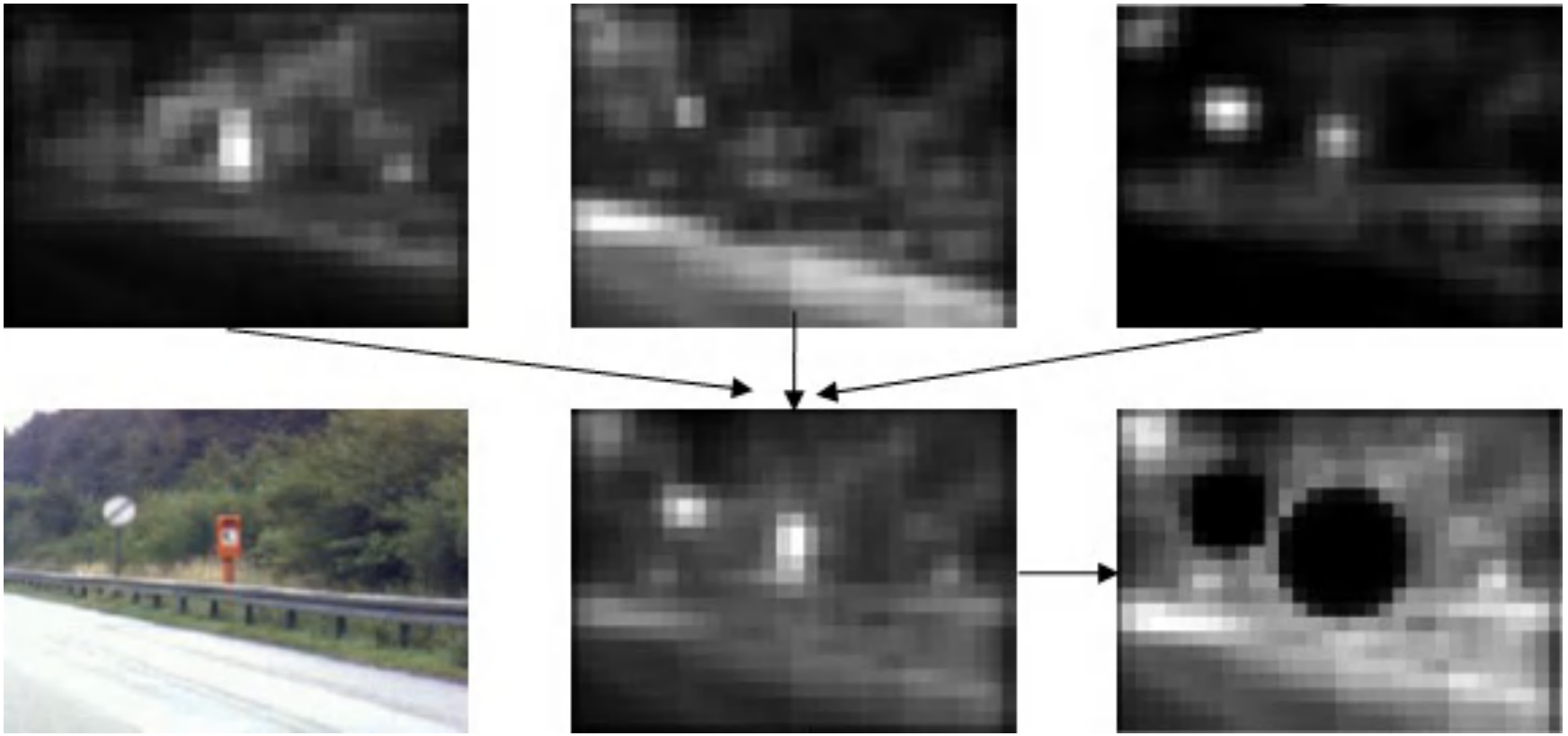


Center-Surround Difference



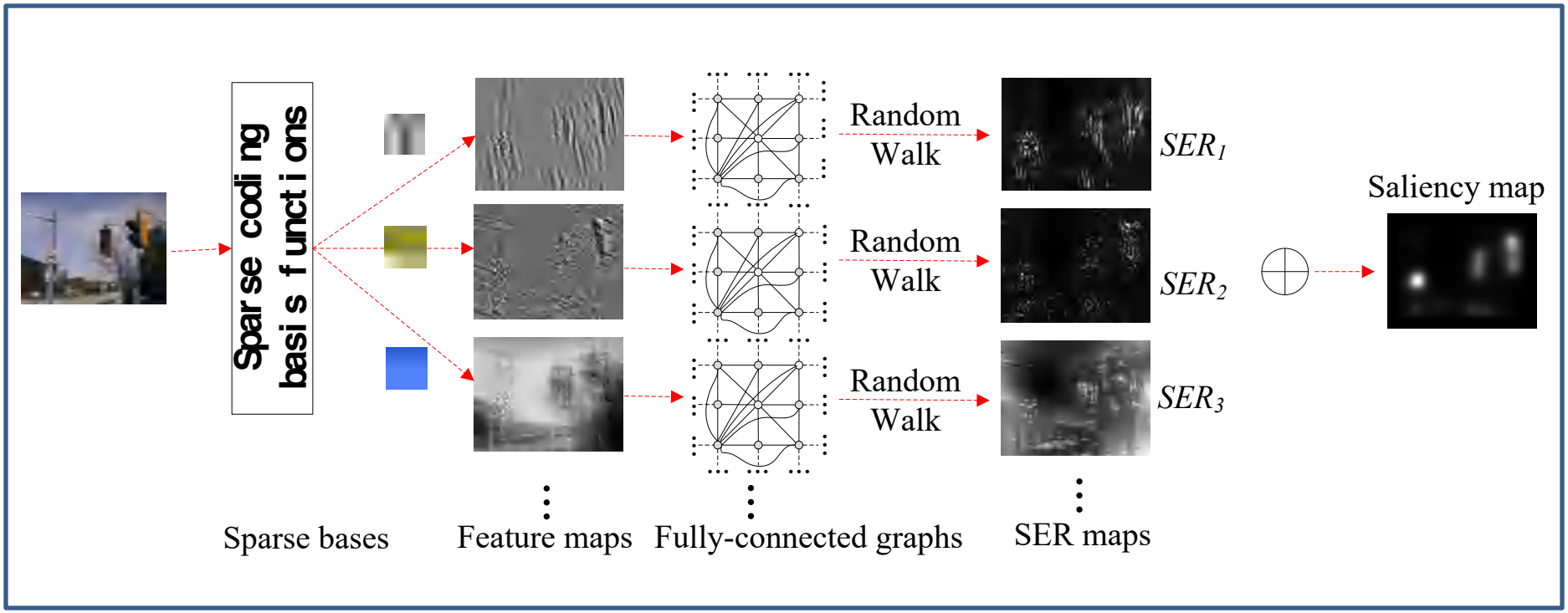
Itti et al. saliency-based visual attention for rapid scene analysis. TPAMI 1998

Center-Surround Difference



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

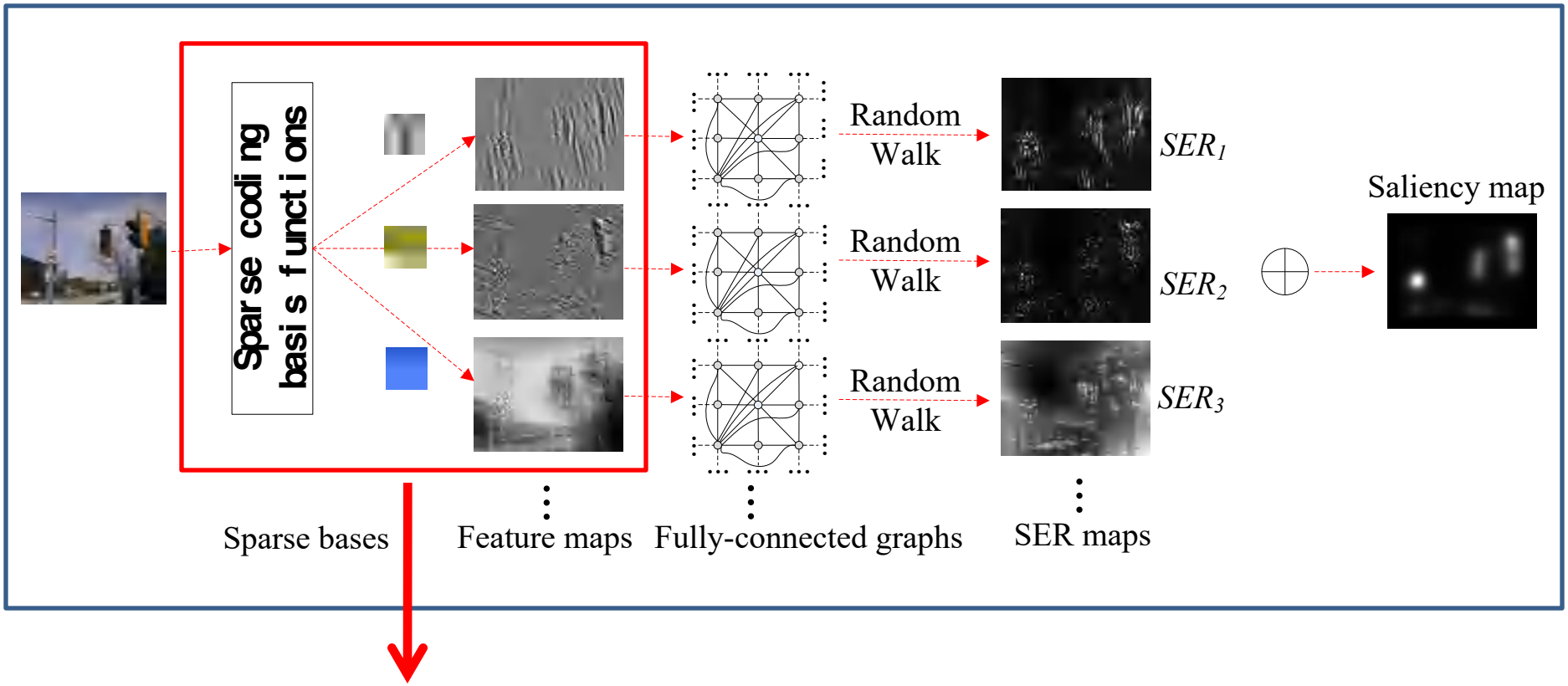
Site Entropy Rate



The procedure of computing the saliency map of an image

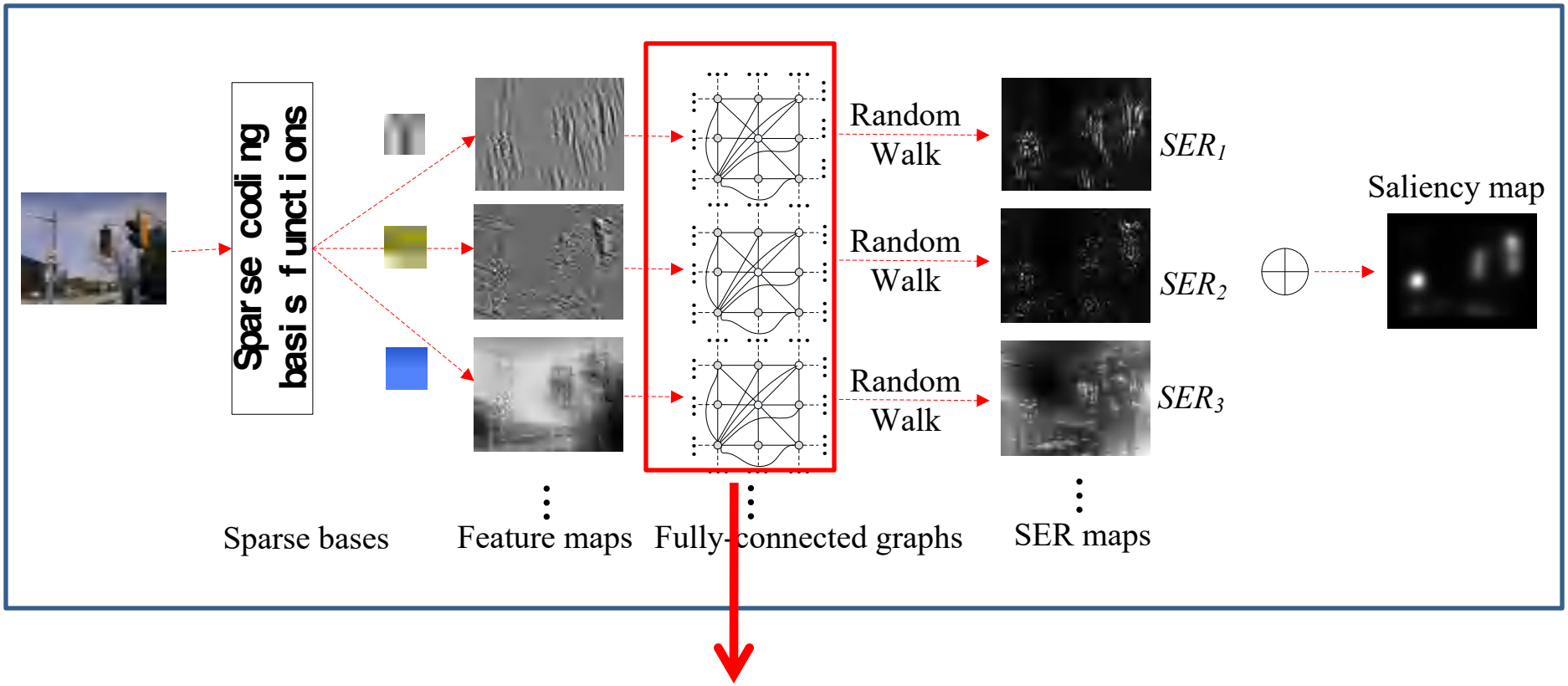
Wei Wang, Yizhou Wang, Qingming Huang, Wen Gao, "Measuring Visual Saliency by Site Entropy Rate", IEEE Computer Vision and Pattern Recognition, CVPR, San Francisco, Jun, 2010. (oral, google citation: 180)

Site Entropy Rate



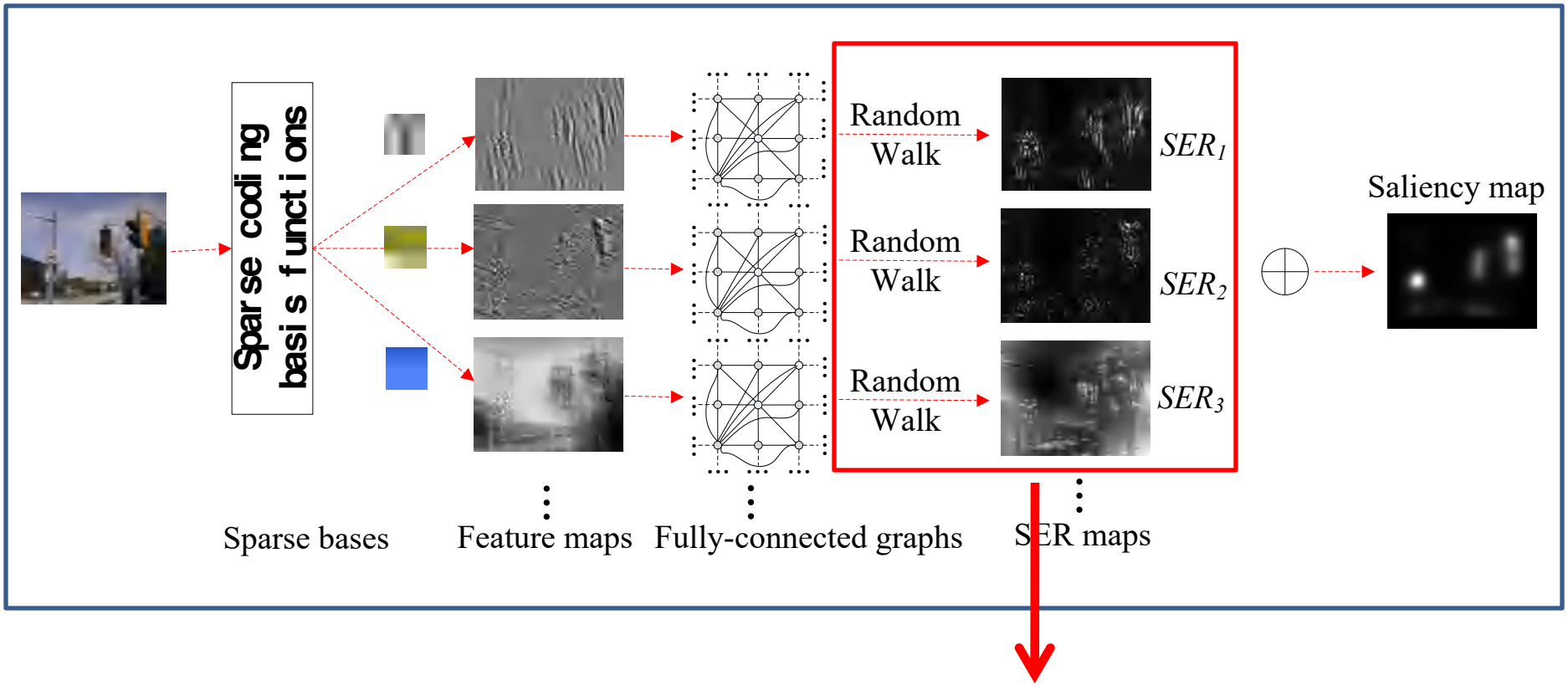
① Filter an input image into multi-band filter response maps using number of learned sparse coding basis functions.

Site Entropy Rate



② To simulate the cortical connectivity, a fully-connected graph representation is adopted for each feature map.

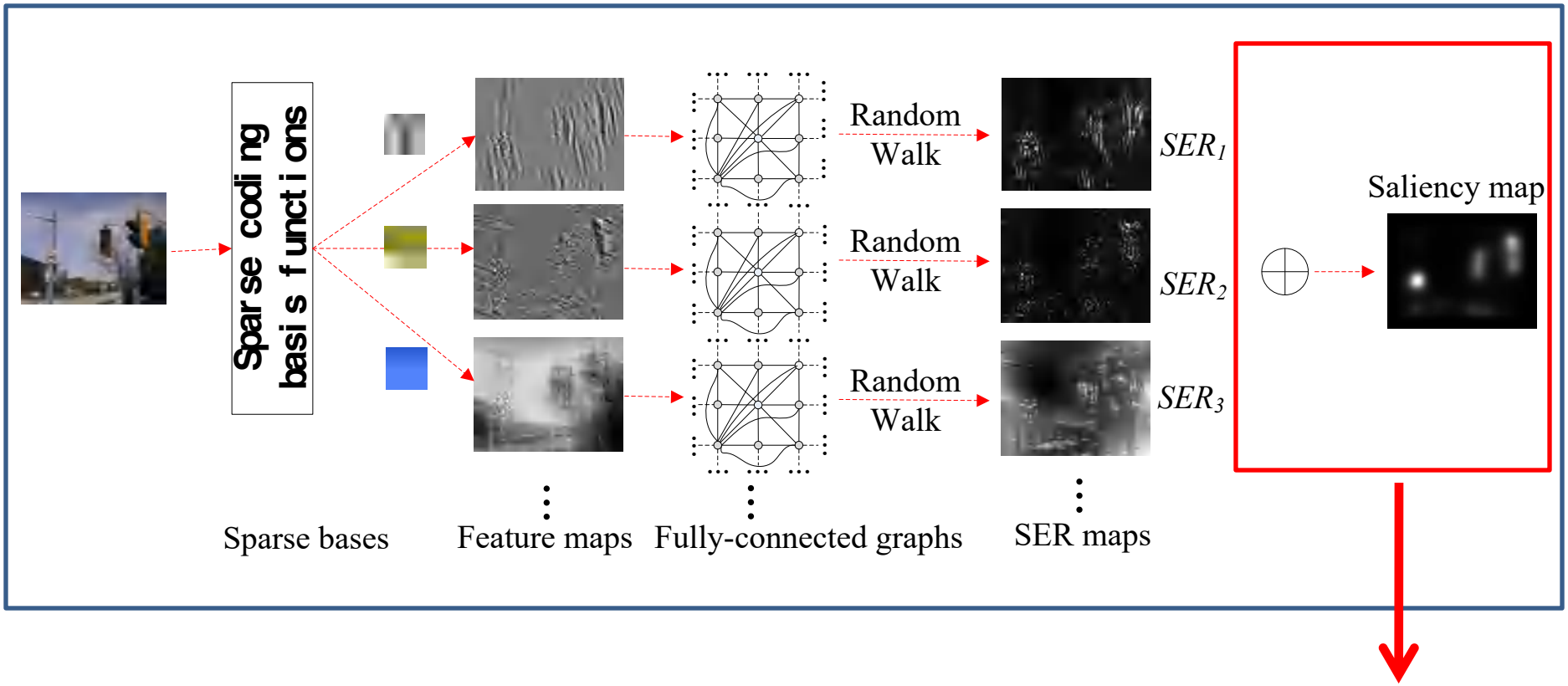
Site Entropy Rate



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.

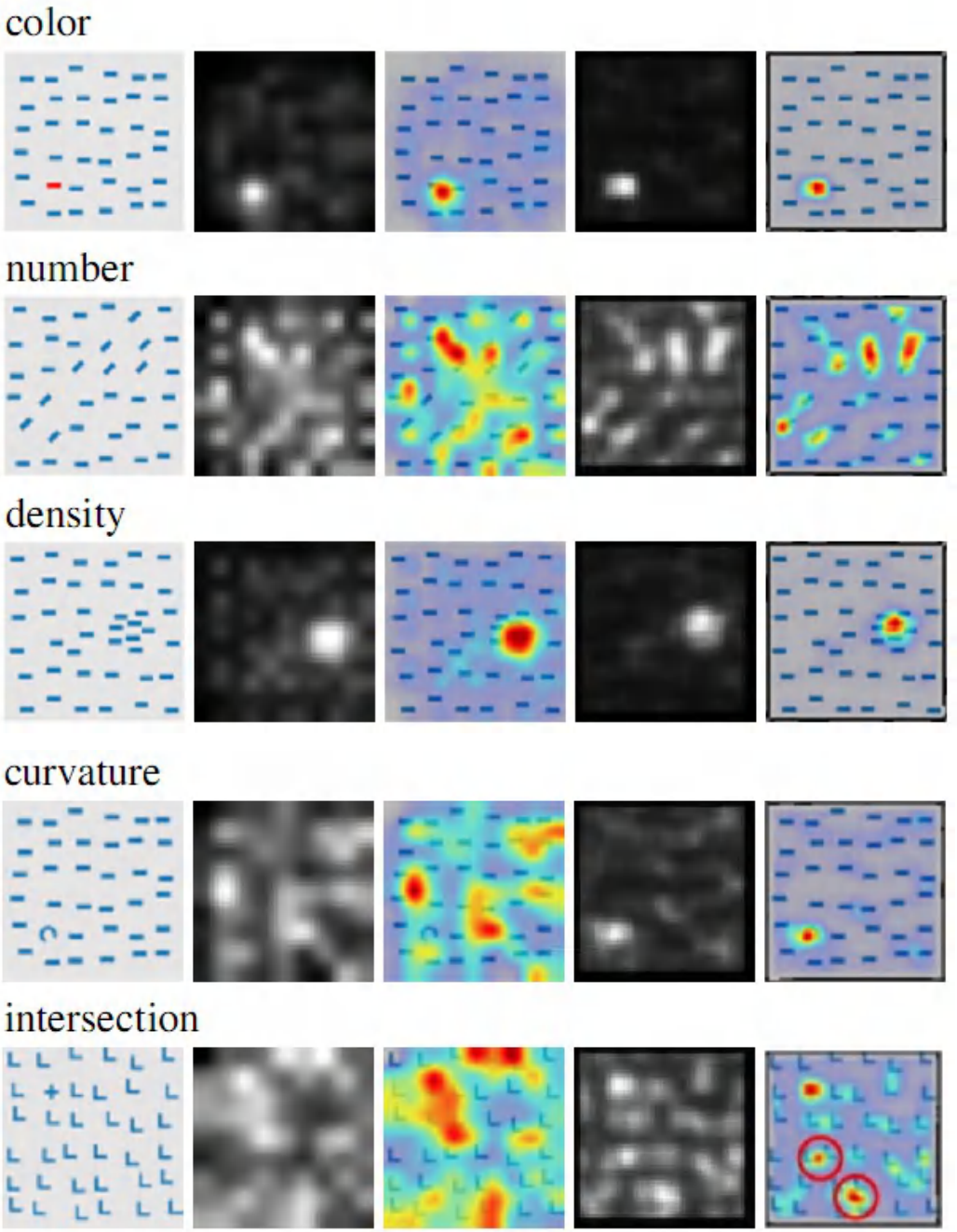
Site Entropy Rate



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

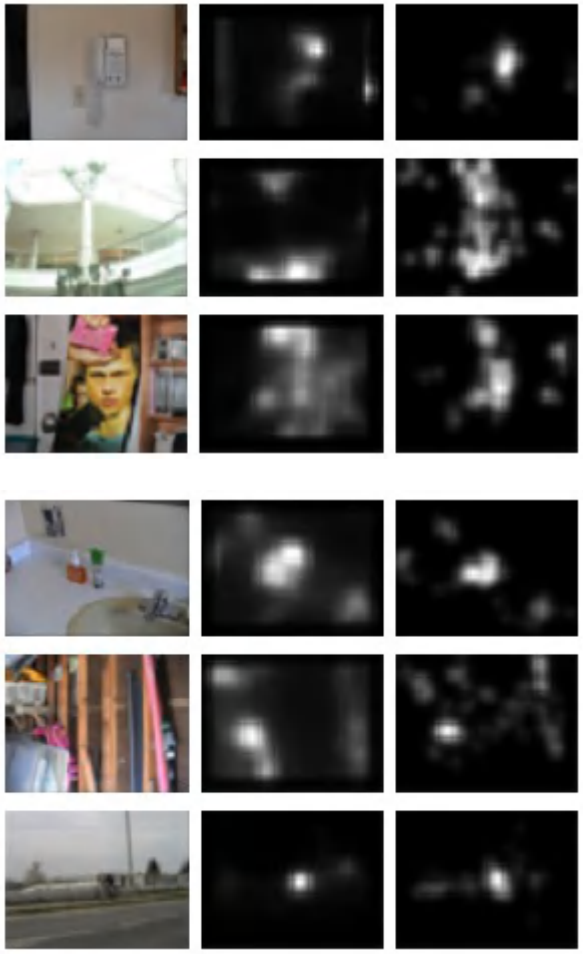
Experiments on Psychological Stimuli

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



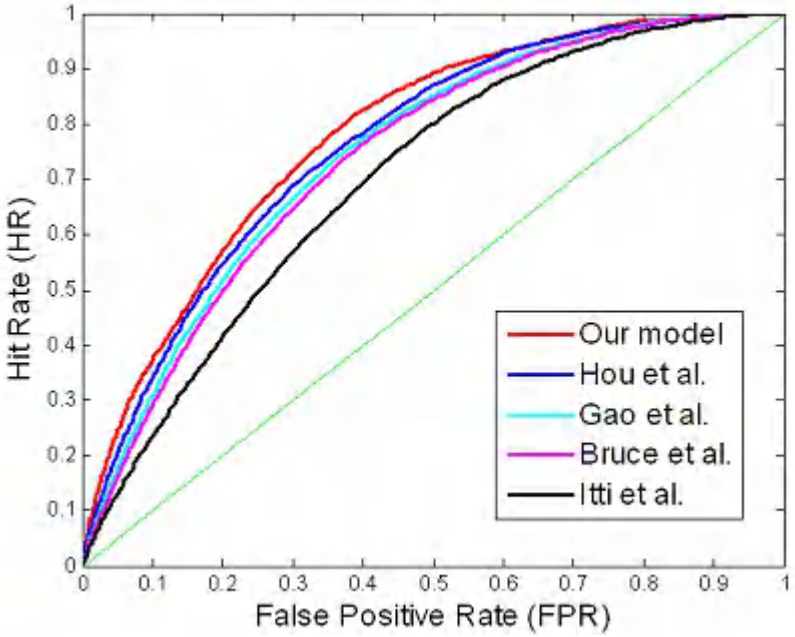
Experiments on Color Image Data Set

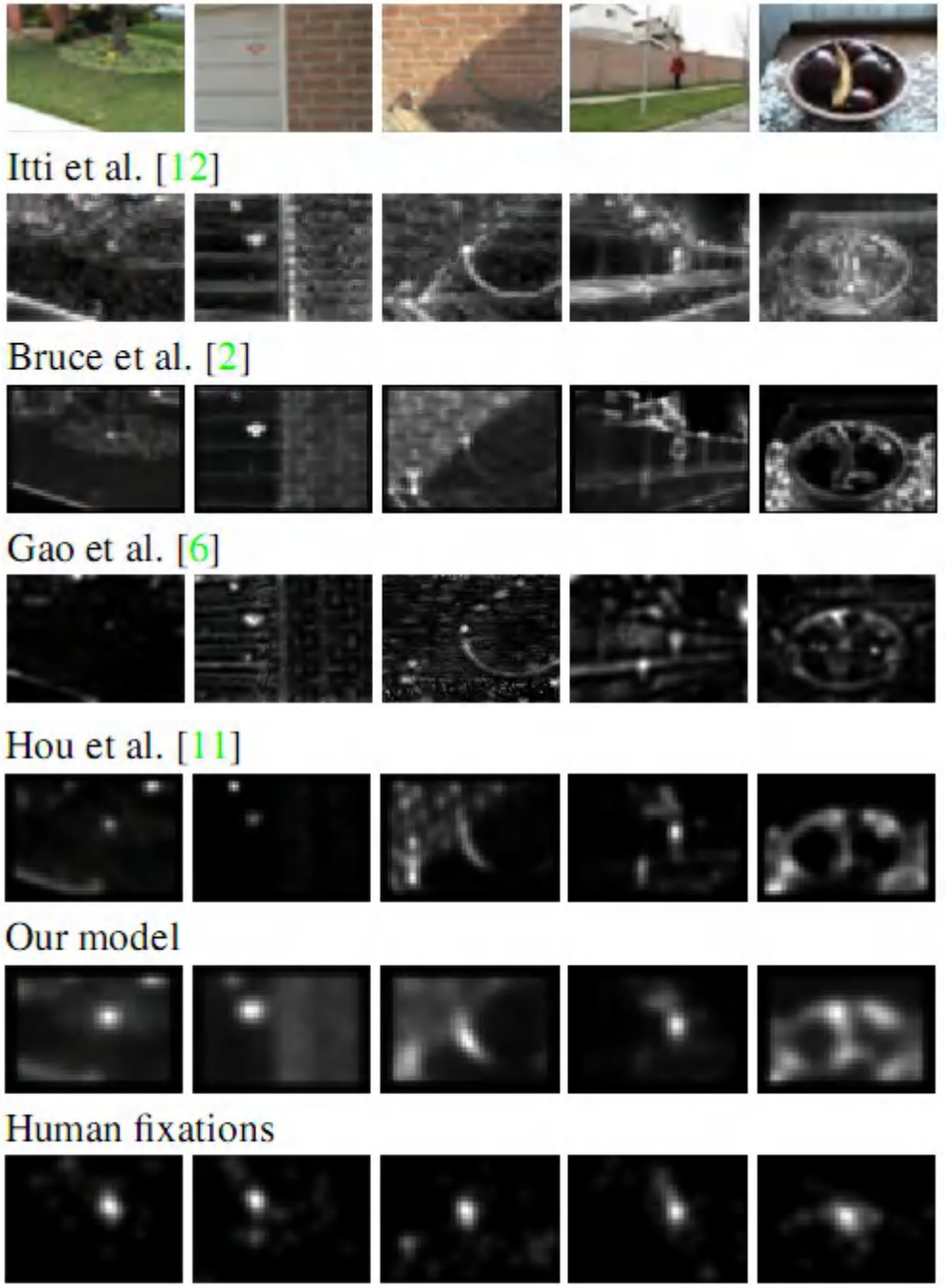
- N. Bruce's dataset [NIPS06]
  - 120 color images (indoor and outdoor scenes)
  - Eye fixations recorded from 20 subjects
- Evaluation method
  - quantitative comparison between human fixation density map and saliency map
  - ROC curve/area



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

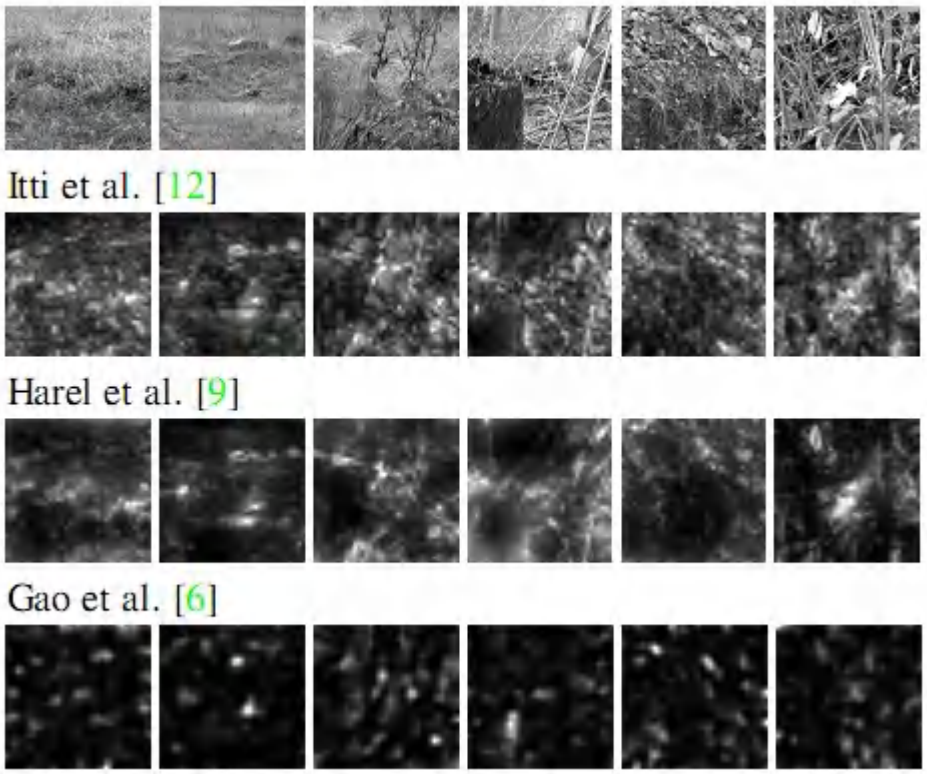


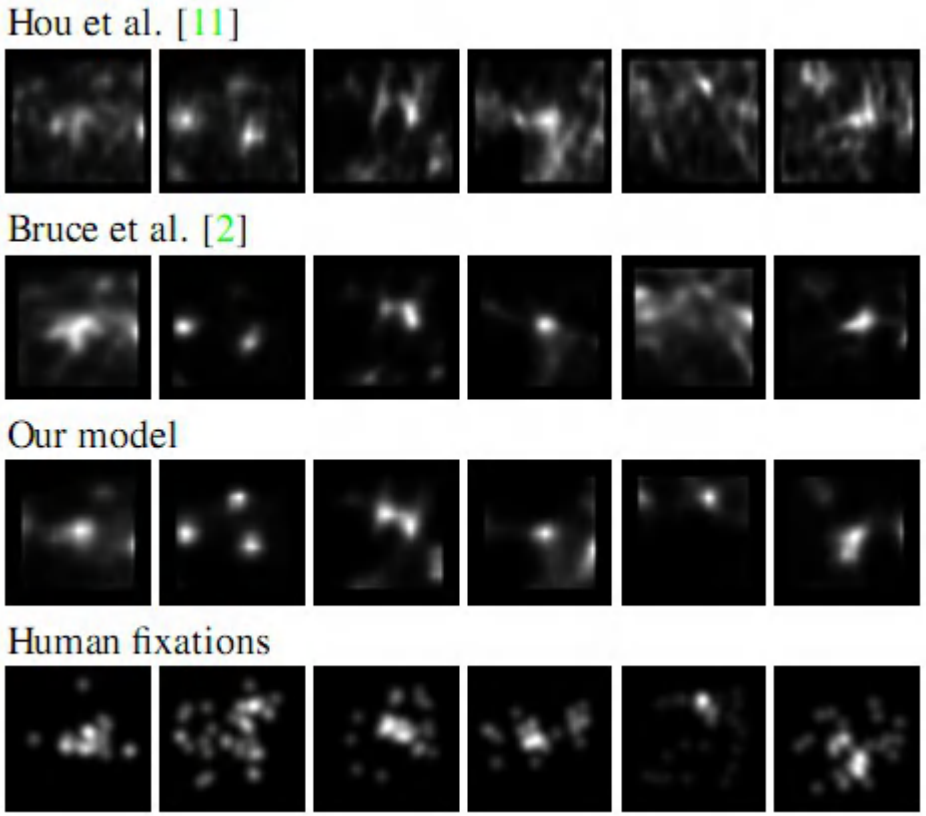


Experiments on Gray Image Data Set

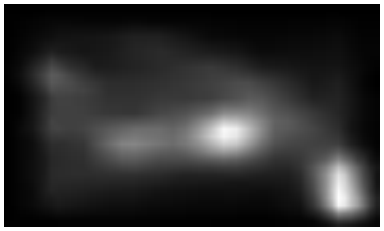
- W. Einhäuser et al's dataset [Vision Research 06]
  - 108 gray images (natural scenes)
  - Fixations from 7 subjects

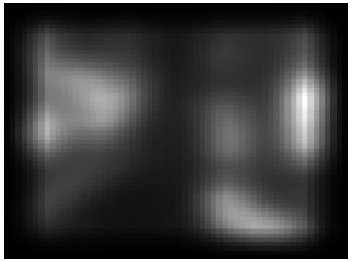
	ROC area
Harel et al. [9]	0.5028
Gao et al. [6]	0.5203
Itti et al. [12]	0.5241
Hou et al. [11]	0.6094
Bruce et al. [2]	0.6420
Our model	0.6537



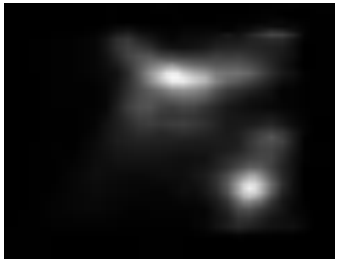
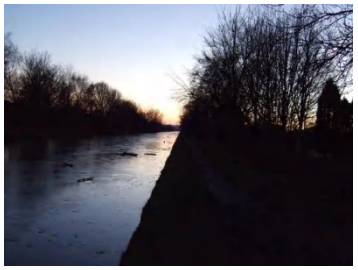


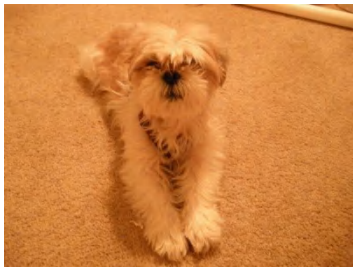
More Results







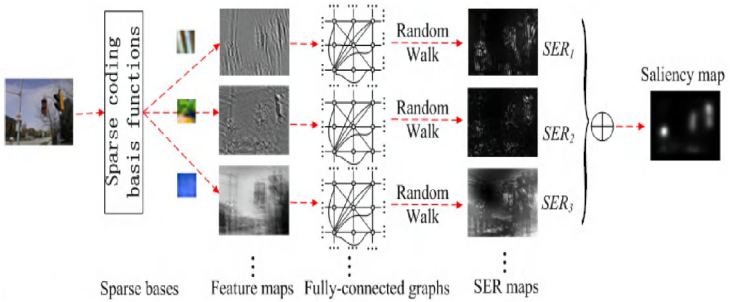



My First Paper

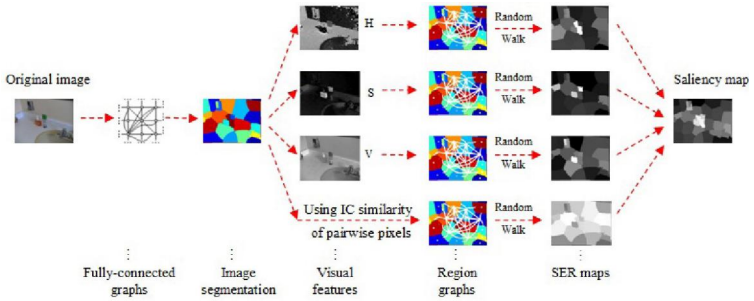
- Extend the SER from pixel-level to region-level

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.

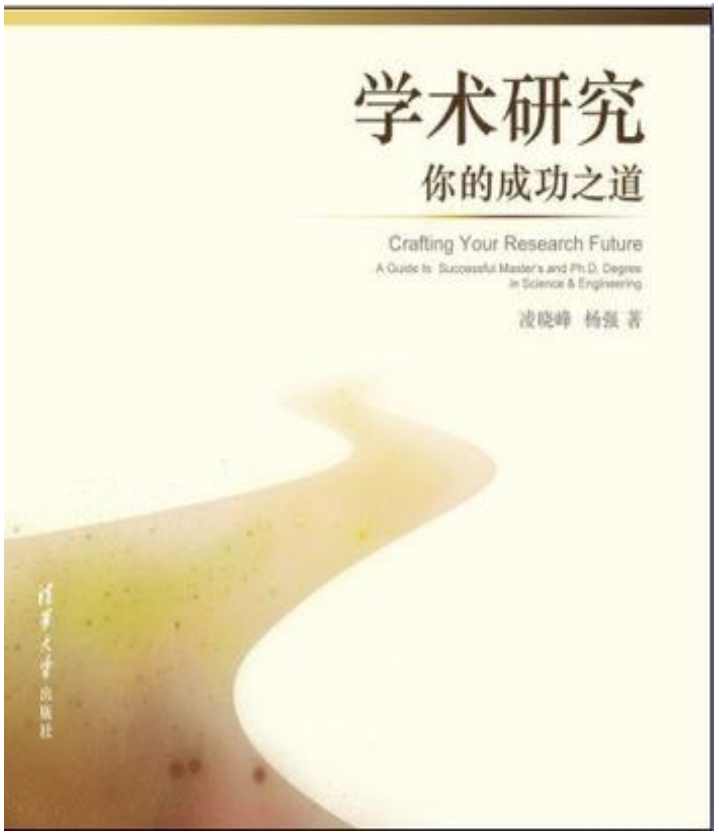


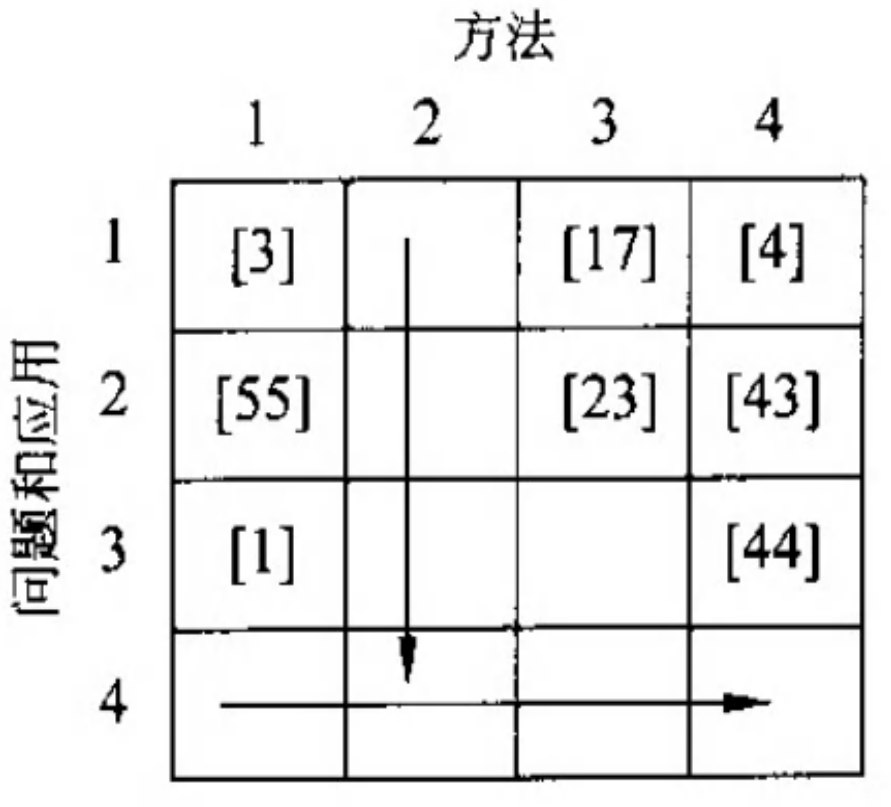
Pixel-level SER, CVPR10



Region-level SER, ICPR12

My First Paper

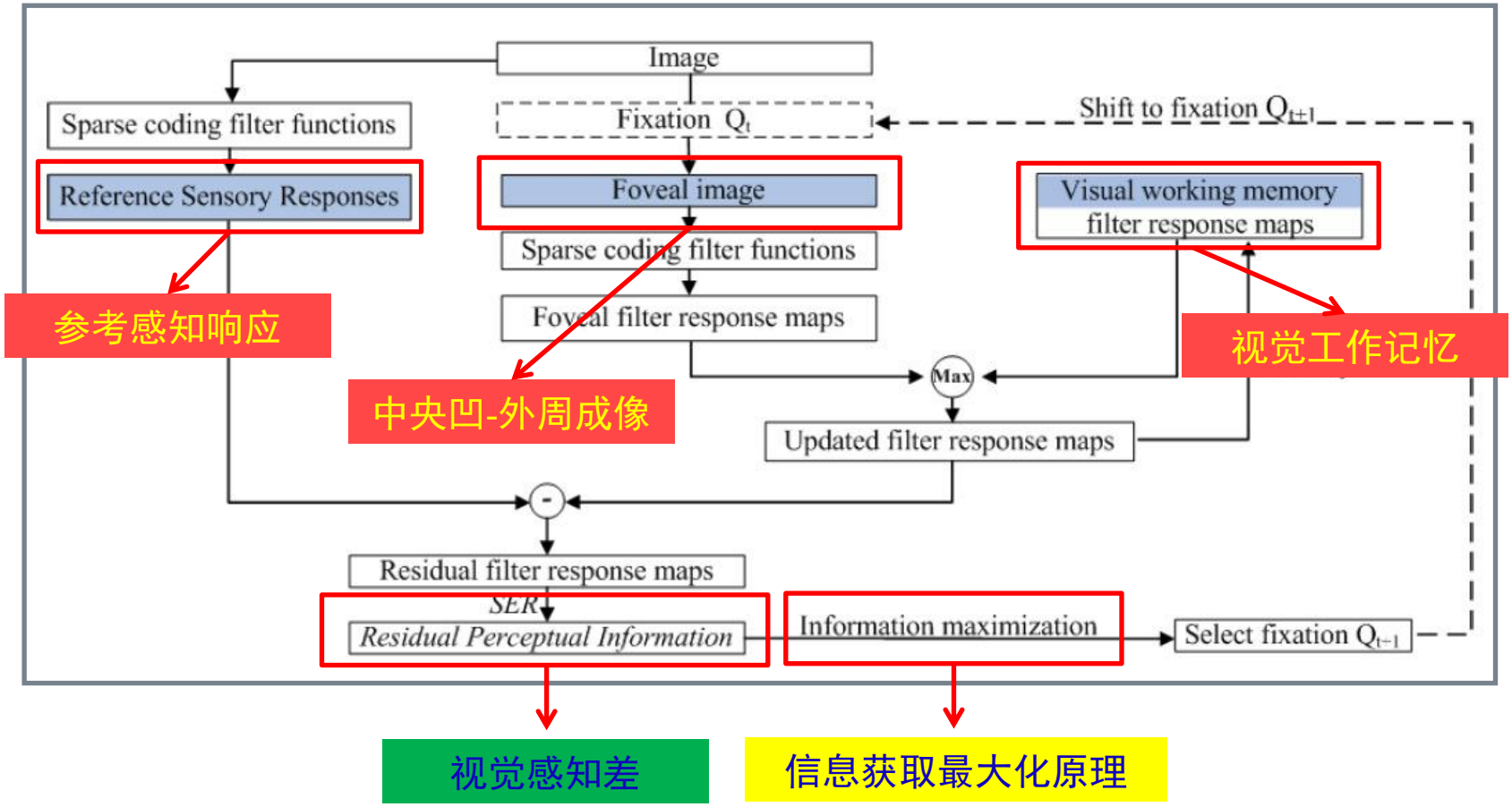




Dynamic Visual Attention

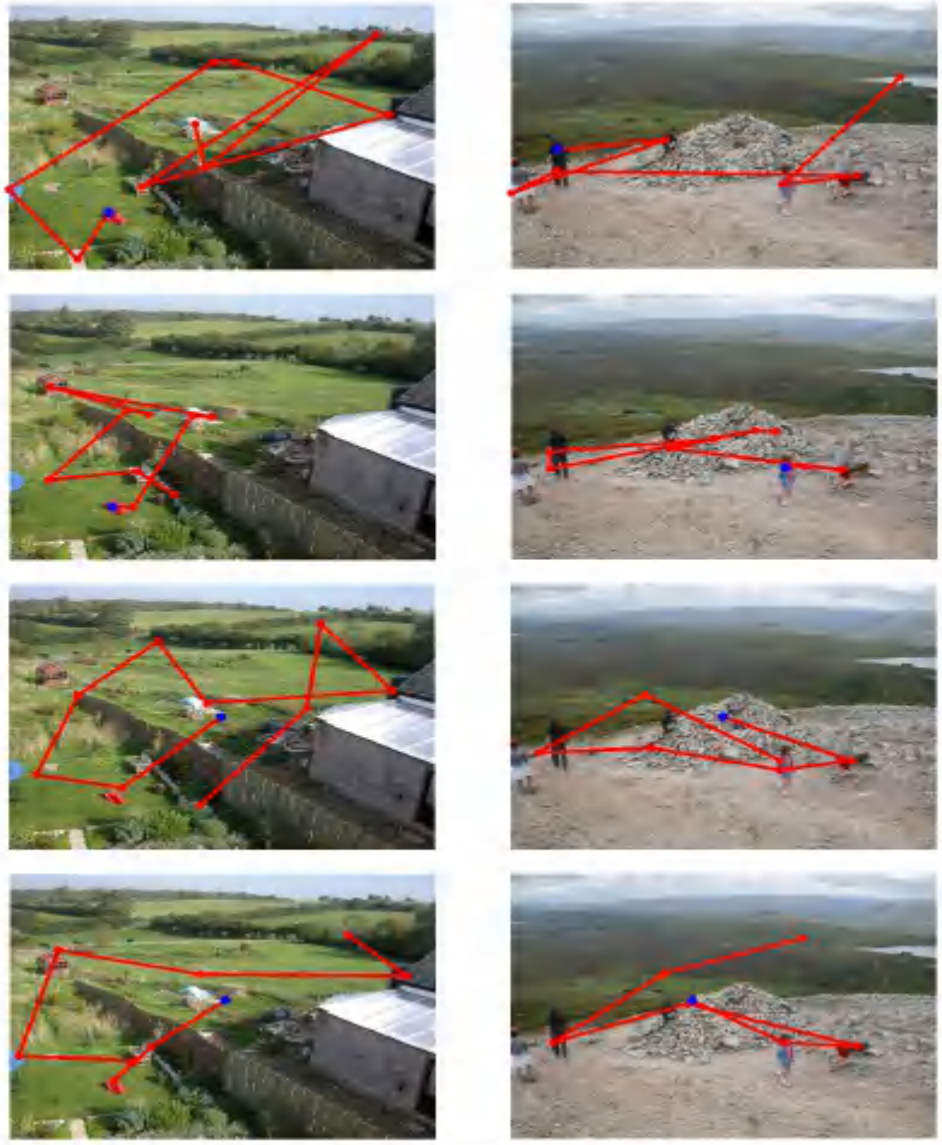
Computational models：

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)

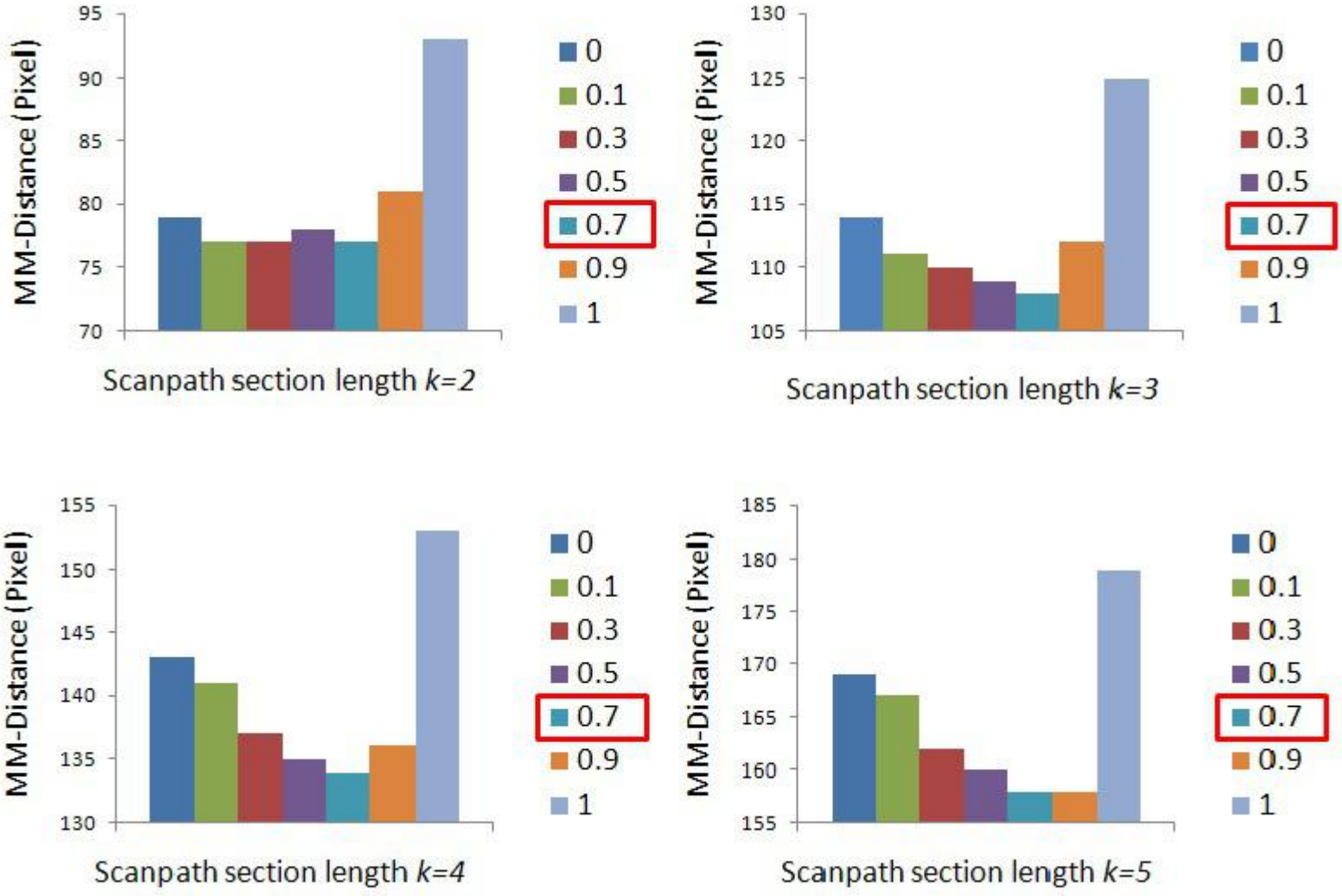
(b) Wang et al. (CVPR 2010)

(c) Our model

(d) Eye tracking data

Dynamic Visual Attention

Forgetting factor in working memory



Deep Attention Model

Since 2013/2014

2010

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

2014

On Learning Where To Look

Marc'Aurelio Ranzato
Google Inc.*
Mountain View CA, U.S.A.

2014

On Learning Where To Look

-

Recurrent Models of Visual Attention

Volodymyr Mnih Nicolas Heess Alex Graves Koray Kavukcuoglu
Google DeepMind

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

2015

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.

2015

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 KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Abstract

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

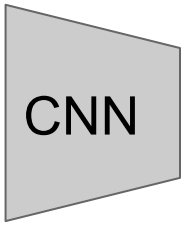
Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

Recall: RNN for Captioning

Image:
H x W x 3

Recall: RNN for Captioning

Image:
H x W x 3



Features:
D

Recall: RNN for Captioning

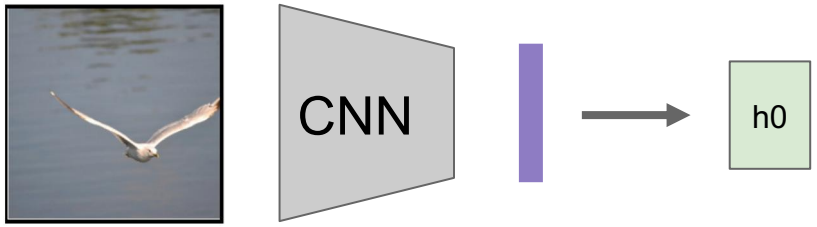
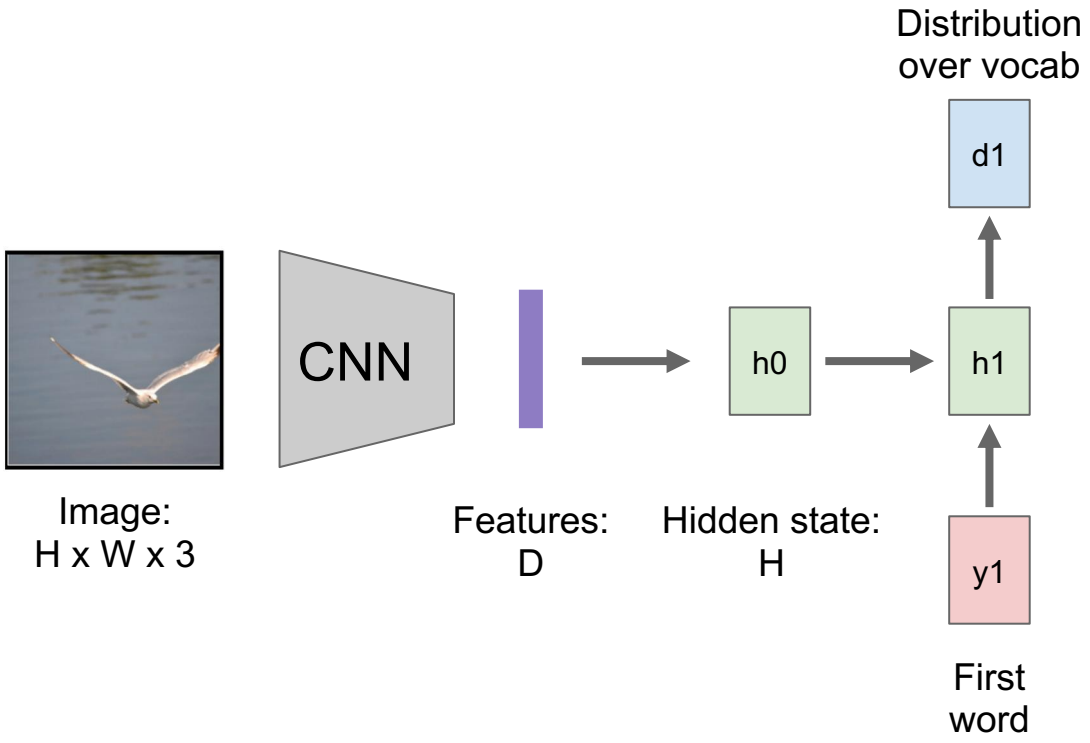


Image:
H x W x 3

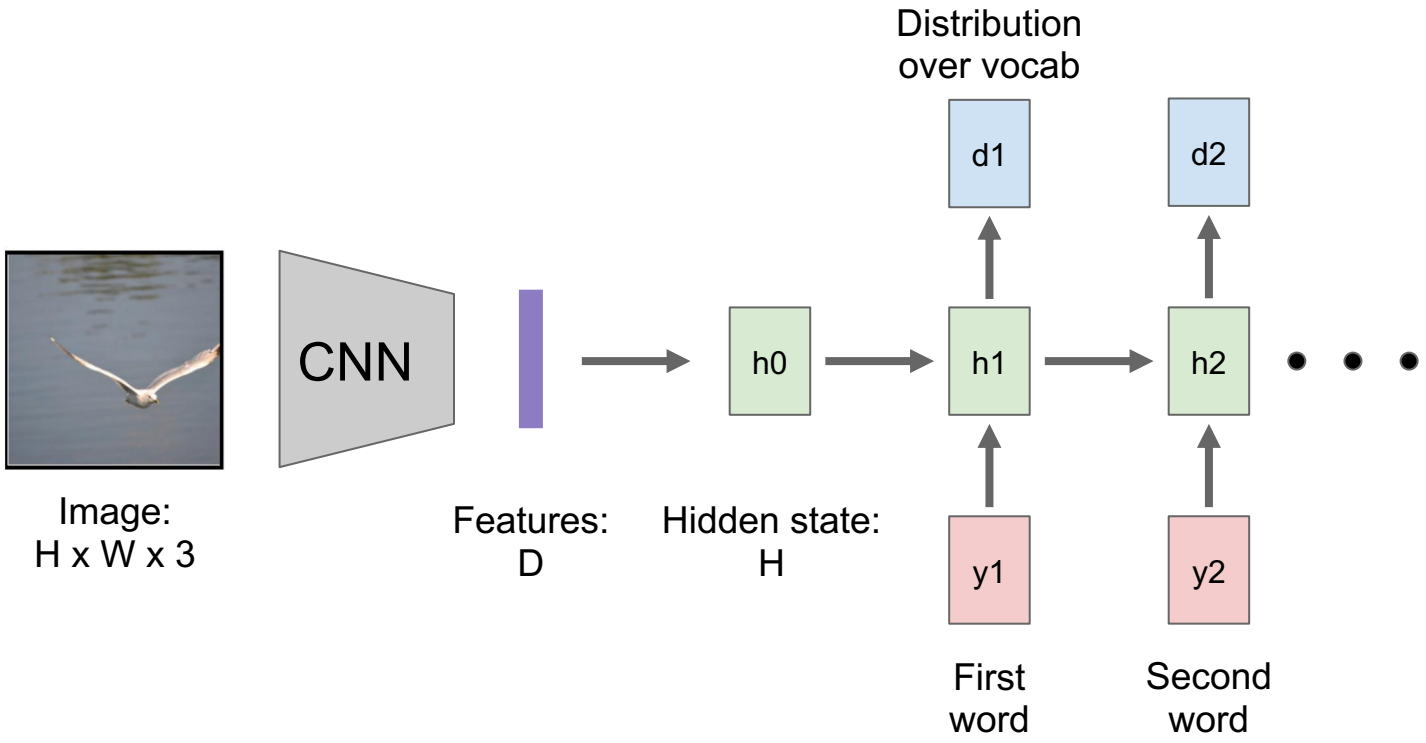
Features:
D

Hidden state:
H

Recall: RNN for Captioning



Recall: RNN for Captioning



Recall: RNN for Captioning

Image:
H x W x 3

CNN

Features:
D

h0

Hidden state:
H

Distribution
over vocab

d1

h1

y1

First
word

d2

h2

y2

Second
word

RNN only looks at
whole image, once

Recall: RNN for Captioning

Image:
H x W x 3

CNN

Features:
D

h0

Hidden state:
H

Distribution
over vocab

d1

h1

y1

First
word

d2

h2

y2

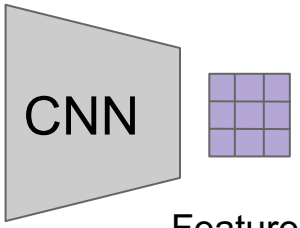
Second
word

RNN only looks at whole image, once

What if the RNN looks at different parts of the image at each timestep?

Soft Attention for Captioning

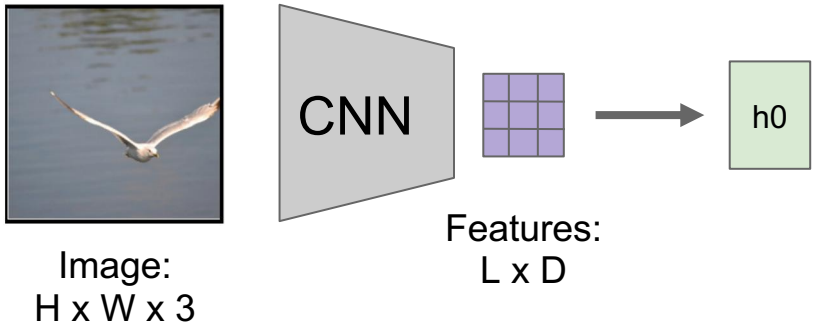
Image:
H x W x 3



Features:
L x D

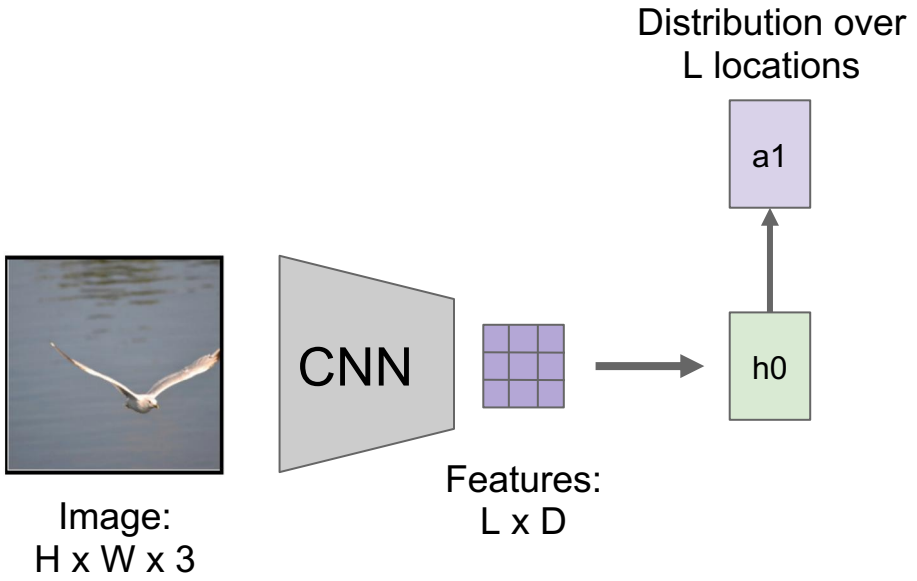
Xu et al, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML 2015

Soft Attention for Captioning



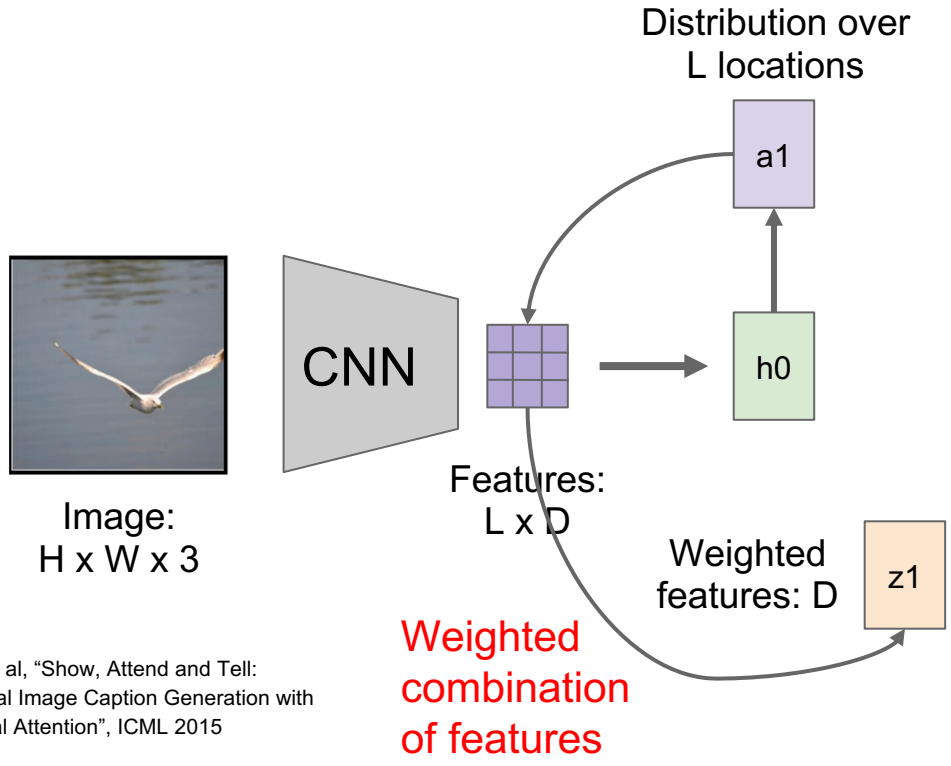
Xu et al, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML 2015

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

Image:
H x W x 3

CNN

Features:
L x D

Distribution over
L locations

a1

h0

h1

z1

y1

Weighted
features: D

First word

Weighted
combination
of features

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

Soft Attention for Captioning

Image:
H x W x 3

CNN

Features:
L x D

Distribution over
L locations

Distribution
over vocab

a1

a2

d1

h0

h1

Weighted
features: D

z1

y1

First word

Weighted
combination
of features

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

Soft Attention for Captioning

Image:
H x W x 3

CNN

Features:
L x D

Distribution over
L locations

Distribution
over vocab

a1

a2

d1

h0

h1

Weighted
features: D

z1

y1

z2

First word

Weighted
combination
of features

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

Soft Attention for Captioning

Image:
H x W x 3

CNN

Features:
L x D

Distribution over
L locations

Distribution
over vocab

a1

a2

d1

h0

h1

h2

Weighted
features: D

z1

y1

z2

y2

First word

Weighted
combination
of features

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

Soft Attention for Captioning

Image:
H x W x 3

CNN

Features:
L x D

Distribution over
L locations

Distribution
over vocab

a1 a2 d1 a3 d2

h0 h1 h2

z1 y1 z2 y2

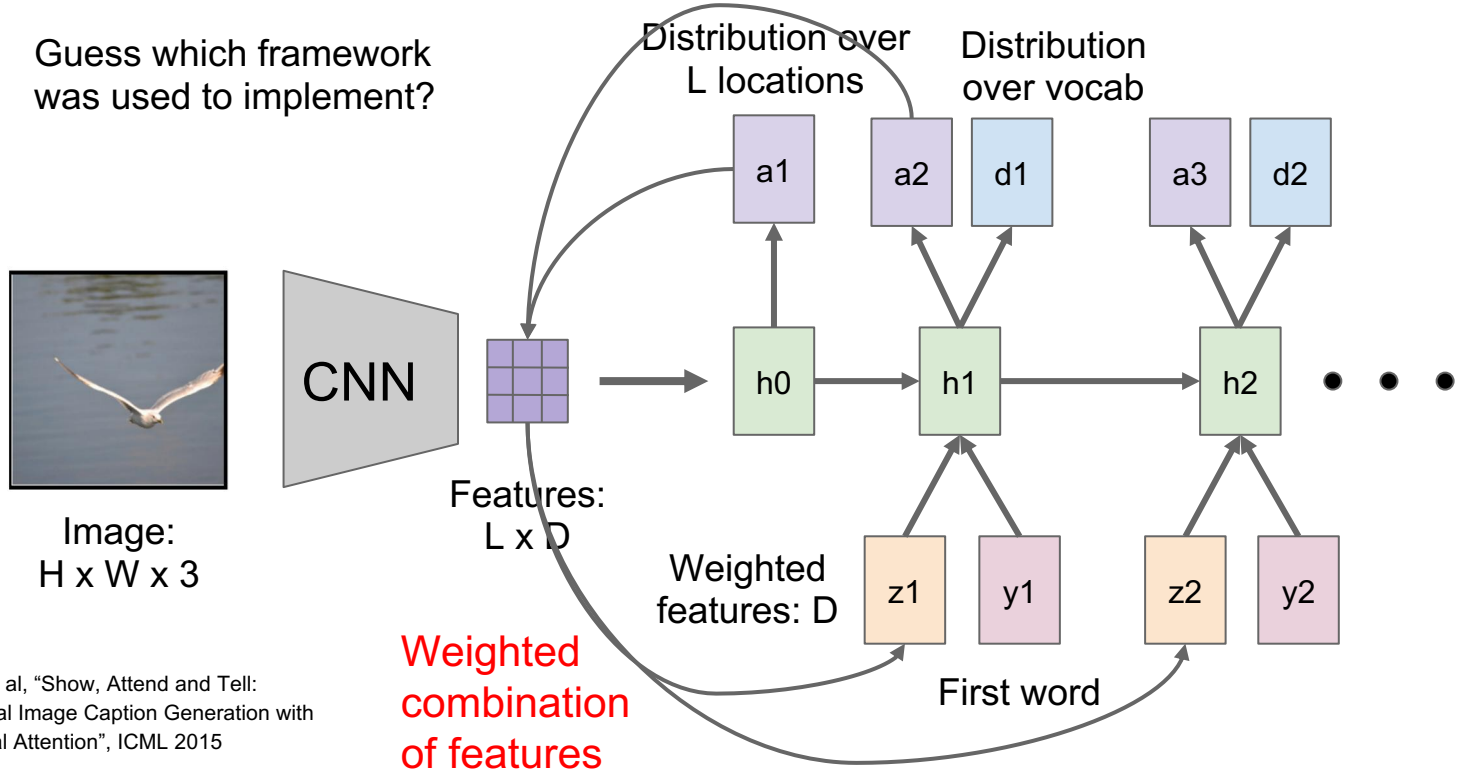
Weighted
features: D

Weighted
combination
of features

First word

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?

Crazy RNN = **Theano**

Image:
H x W x 3

CNN

Features:
L x D

Distribution over L locations

Distribution over vocab

a1 a2 d1 a3 d2

h0 h1 h2

z1 y1 z2 y2

Weighted features: D

Weighted combination of features

First word

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

Soft vs Hard Attention

Image:
H x W x 3

CNN

a	b
c	d

Grid of features
(Each D-dimensional)

From RNN:

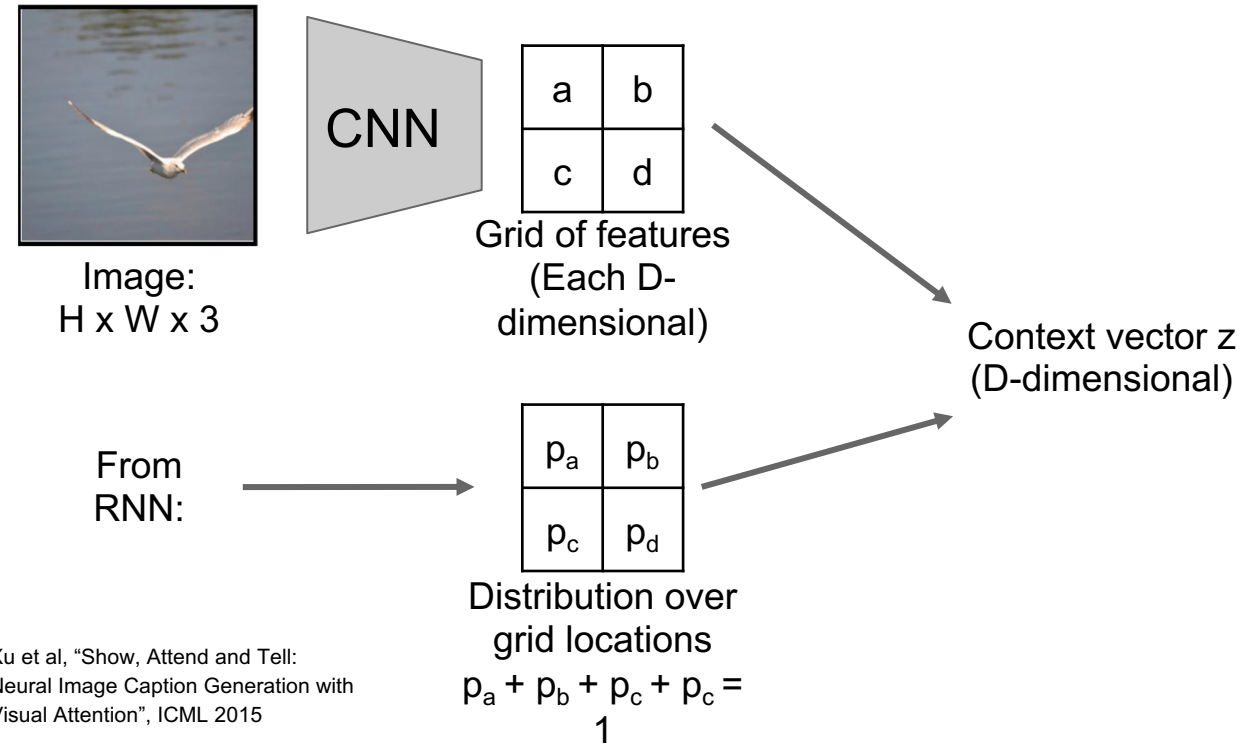
p_a	p_b
p_c	p_d

Distribution over grid locations

$p_a + p_b + p_c + p_c = 1$

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

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 x W x 3

CNN

a	b
c	d

Grid of features
(Each D-dimensional)

From RNN:

p_a	p_b
p_c	p_d

Distribution over grid locations
$p_a + p_b + p_c + p_c = 1$

Context vector z
(D-dimensional)

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

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 x W x 3

CNN

a	b
c	d

Grid of features
(Each D-dimensional)

Context vector z
(D-dimensional)

From RNN:

p_a	p_b
p_c	p_d

Distribution over grid locations
$p_a + p_b + p_c + p_c = 1$

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

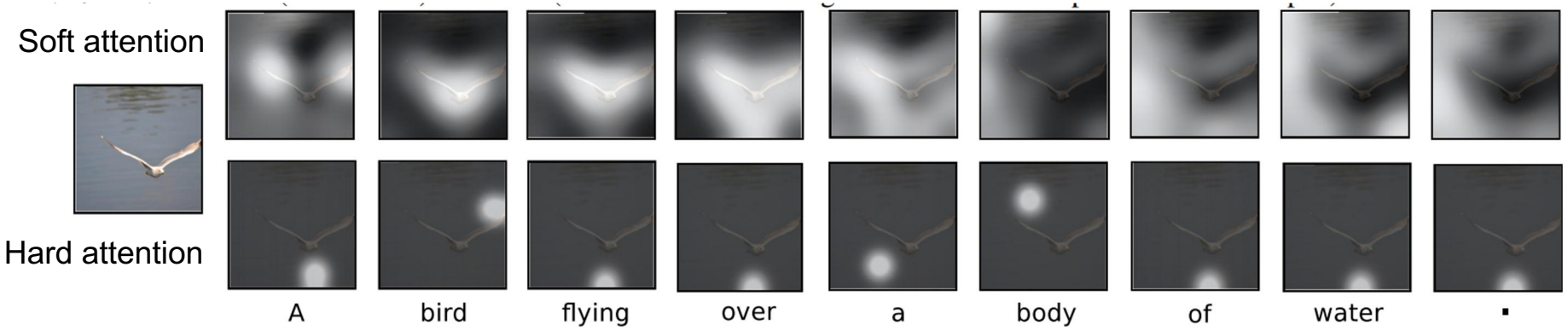
Soft attention:
Summarize ALL locations
$z = p_a a + p_b b + p_c c + p_d d$

Derivative dz/dp is nice!
Train with gradient descent

Hard attention:
Sample ONE location according to p, z = 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.

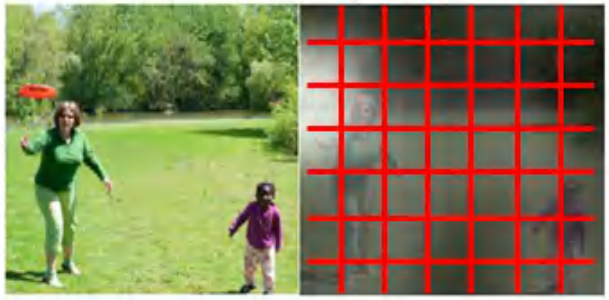
A group of people sitting on a boat in the water.

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

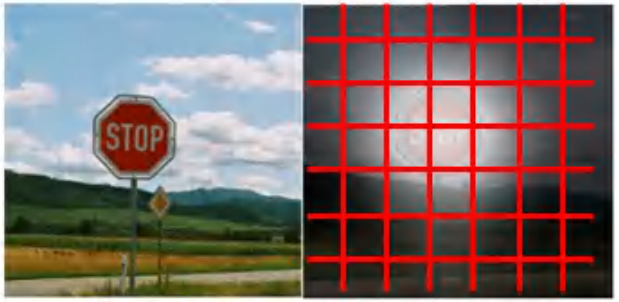
Soft Attention for Captioning

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

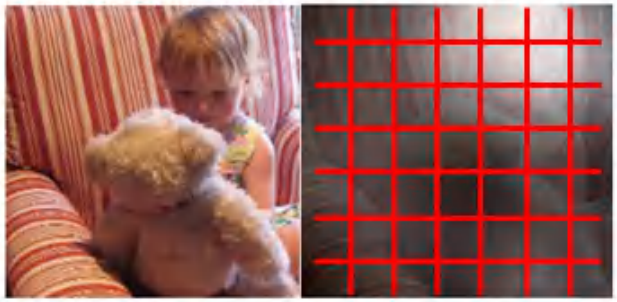


A woman is throwing a frisbee in a park.

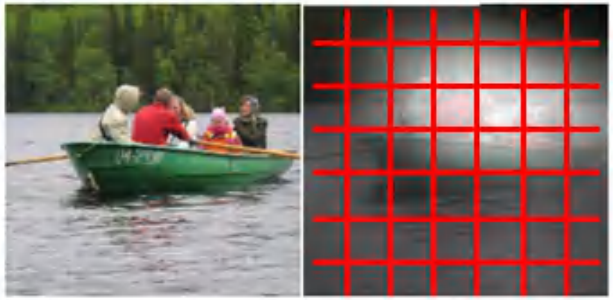
A dog is standing on a hardwood floor.



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



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



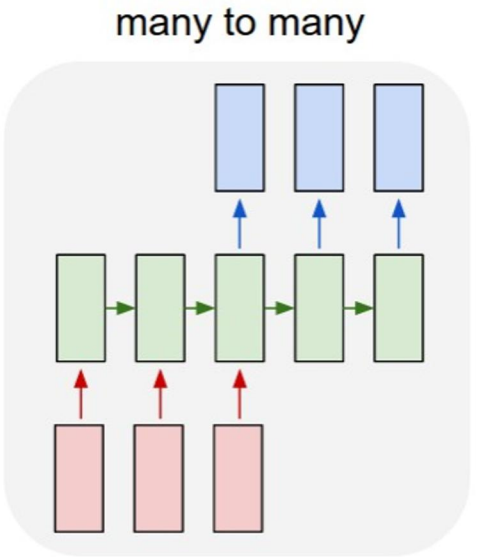
A group of people sitting on a boat in the water.

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

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

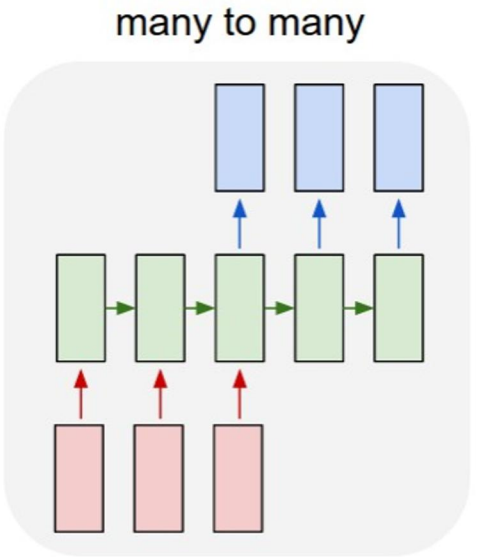
Soft Attention for Translation

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



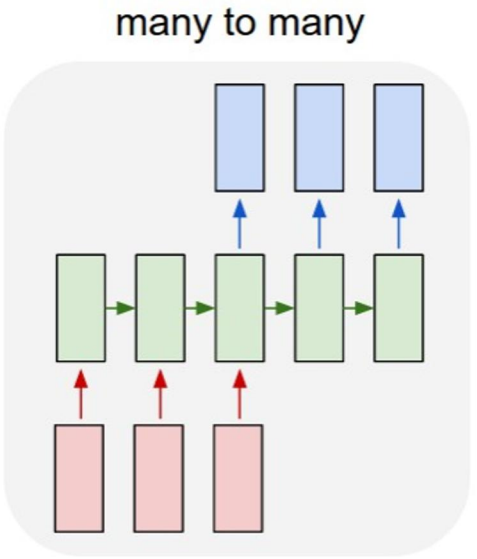
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Soft Attention for Translation



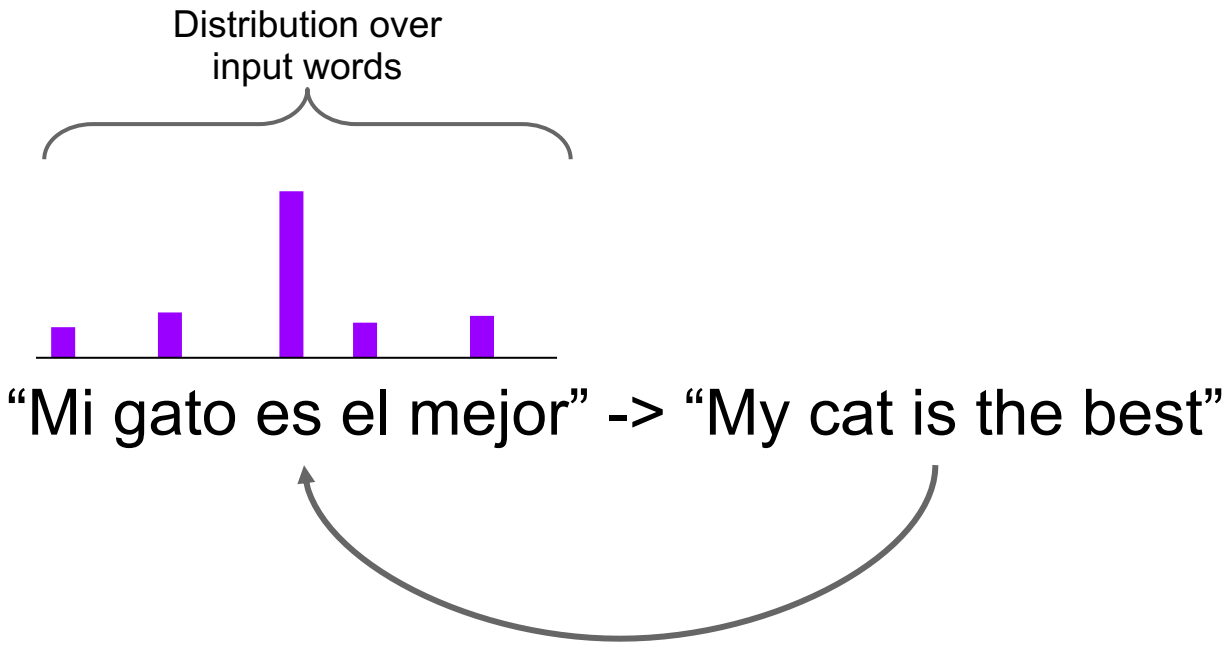
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

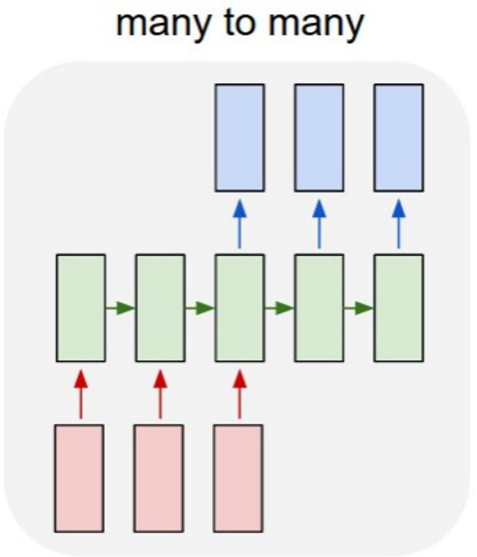
Soft Attention for Translation



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

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

Video captioning, attention over input frames:

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

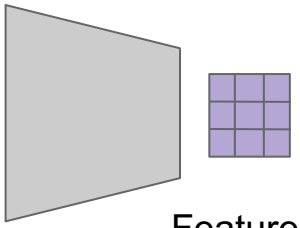
Image, question to answer, attention over image:

- 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

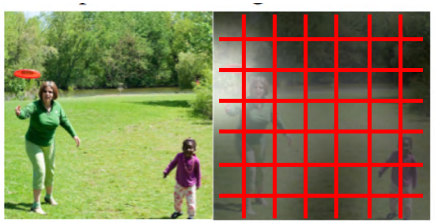
Attending to Arbitrary Regions?

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

Image:
H x W x 3



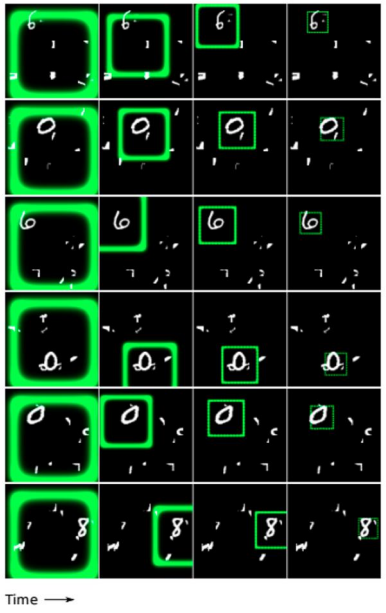
Features:
L x D



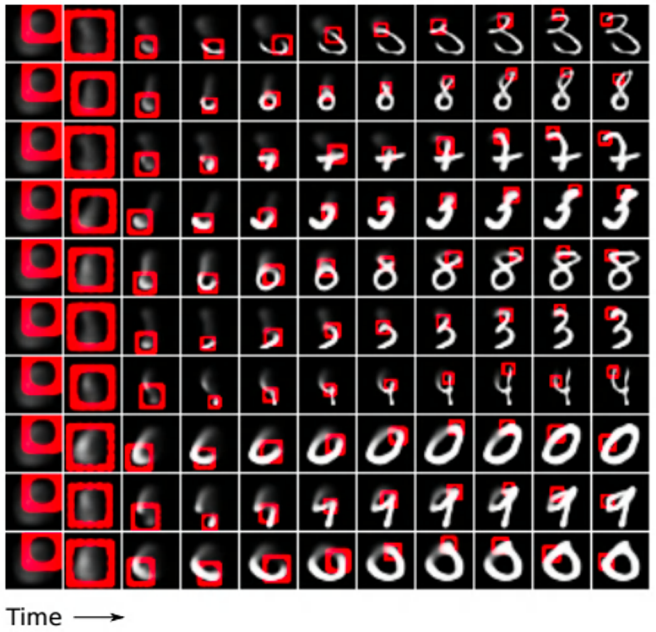
A woman is throwing a frisbee in a park.

Attending to Arbitrary Regions: DRAW

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

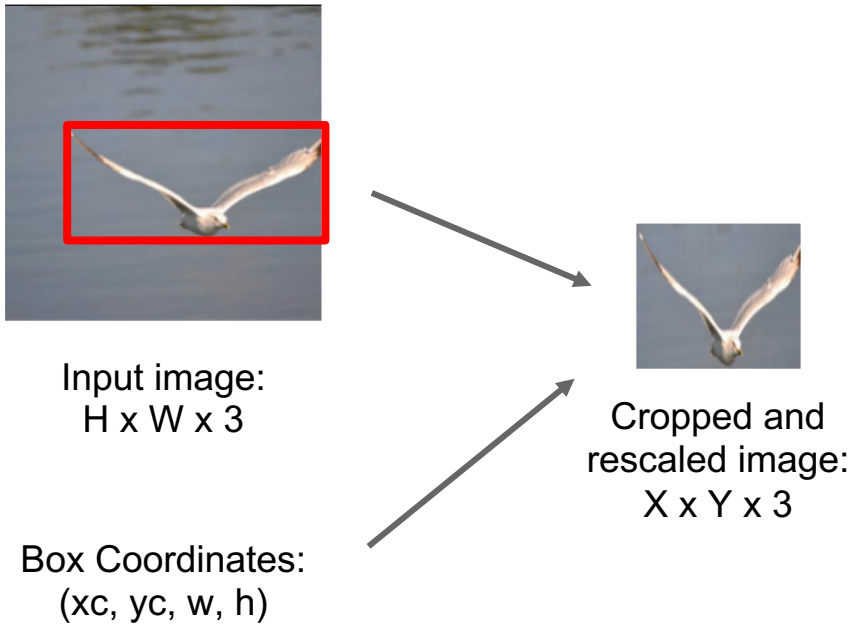


Generate images by attending to arbitrary regions of the *output*



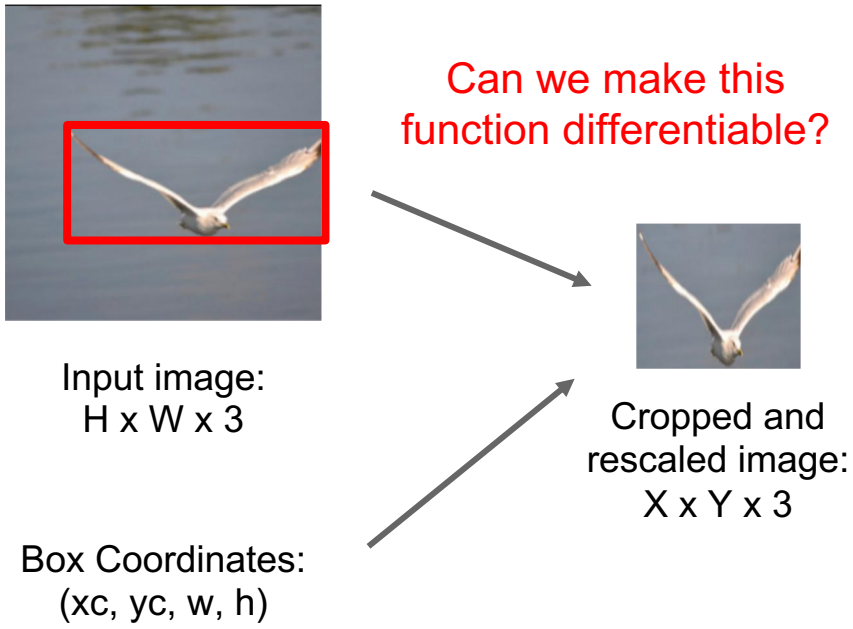
Gregor et al, “DRAW: A Recurrent Neural Network For Image Generation”, ICML 2015

Spatial Transformer Networks



Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks



Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks

Input image:
H x W x 3

Can we make this function differentiable?

Cropped and rescaled image:
X x Y x 3

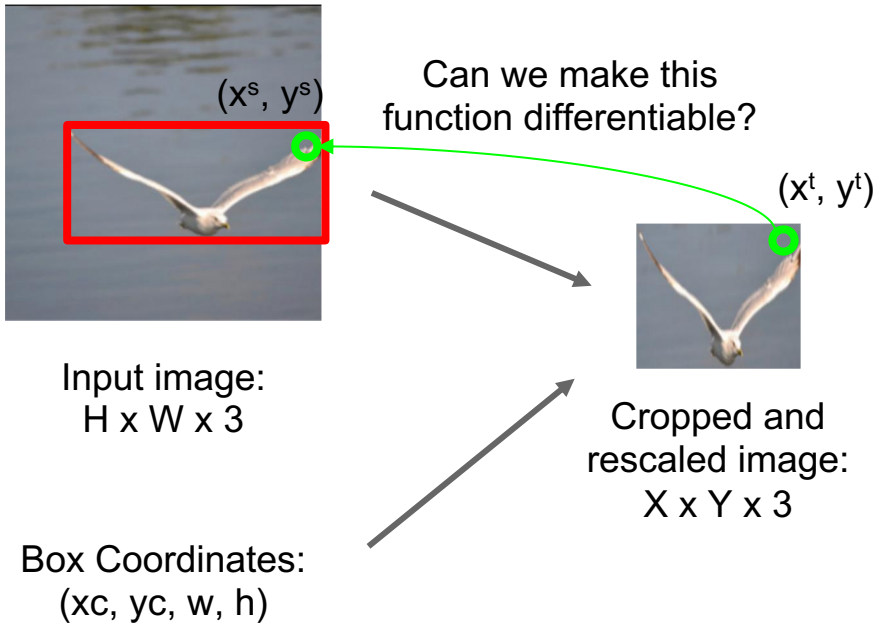
Box Coordinates:
(xc, yc, w, h)

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

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

Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks

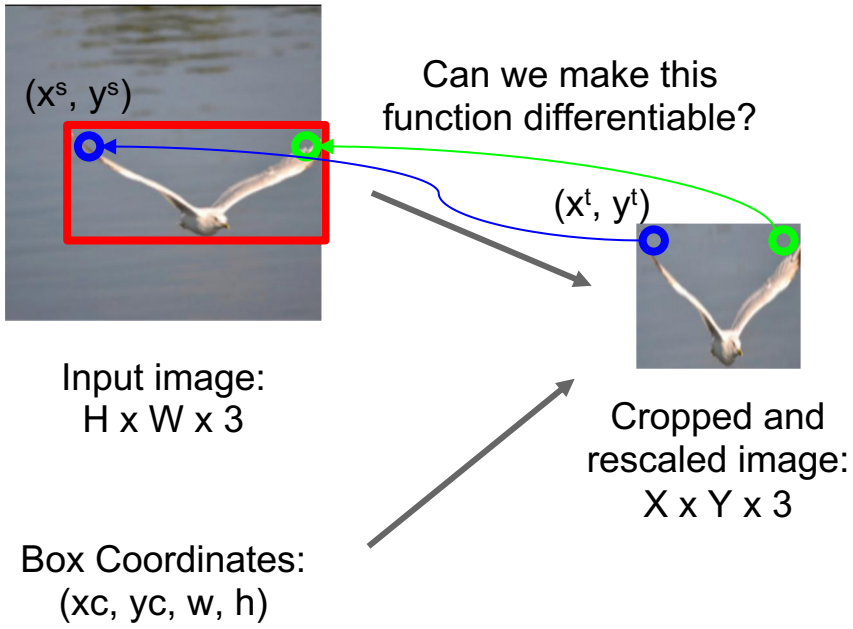


Idea: Function mapping *pixel coordinates* (xt, yt) of output to *pixel coordinates* (xs, ys) of input

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

Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks

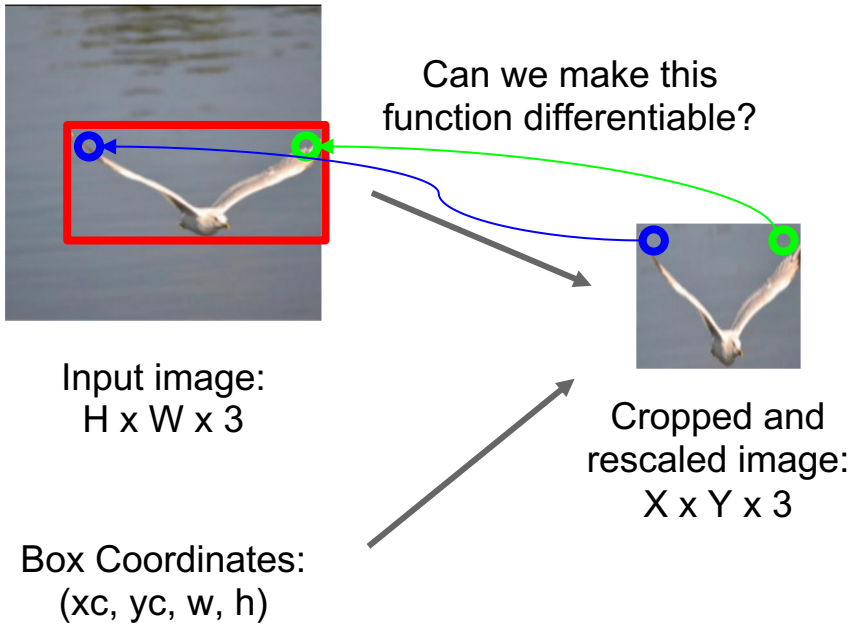


Idea: Function mapping *pixel coordinates* (xt, yt) of output to *pixel coordinates* (xs, ys) of input

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

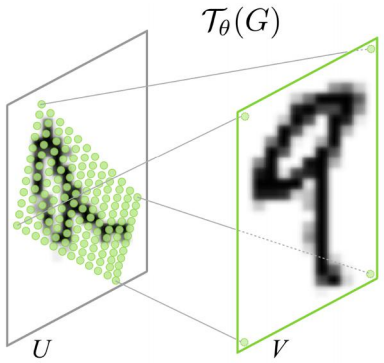
Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks



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

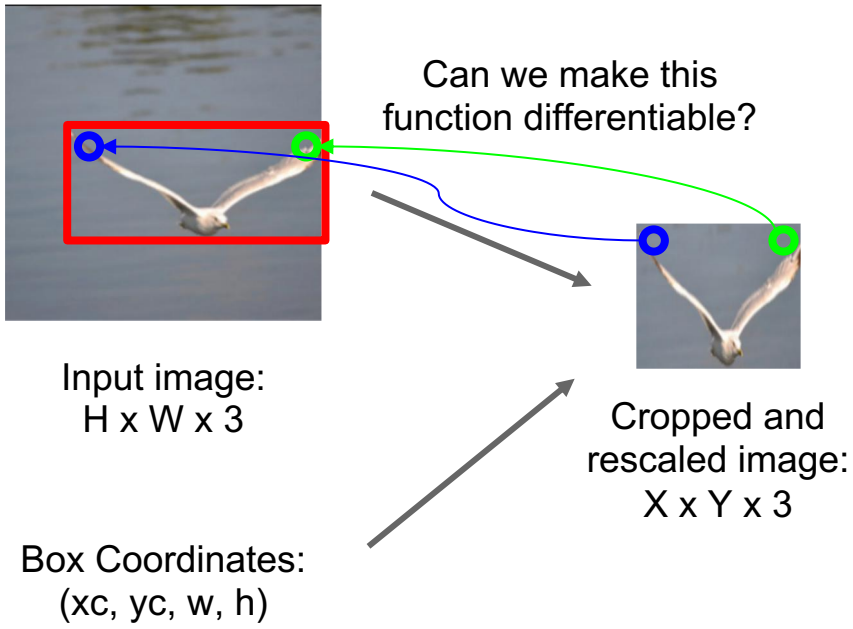
$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



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

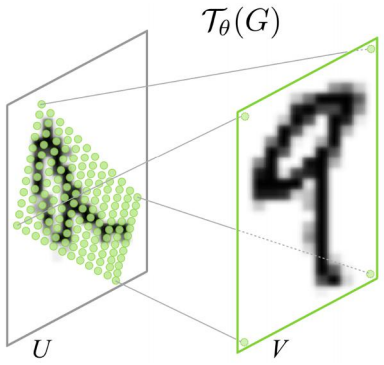
Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Spatial Transformer Networks



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

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

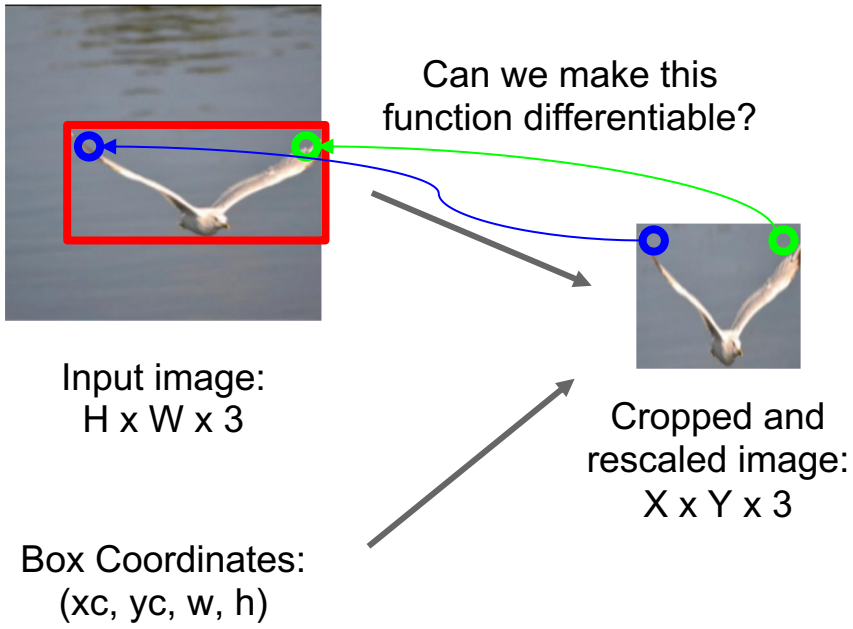


Repeat for all pixels in *output* to get a **sampling grid**

Then use **bilinear interpolation** to compute output

Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

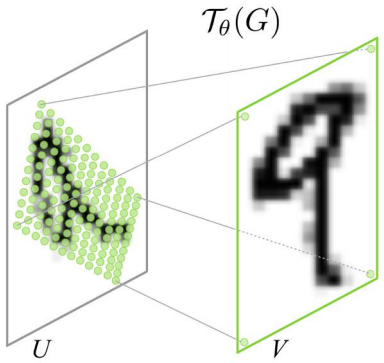
Spatial Transformer Networks



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

Network attends to input by predicting θ

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

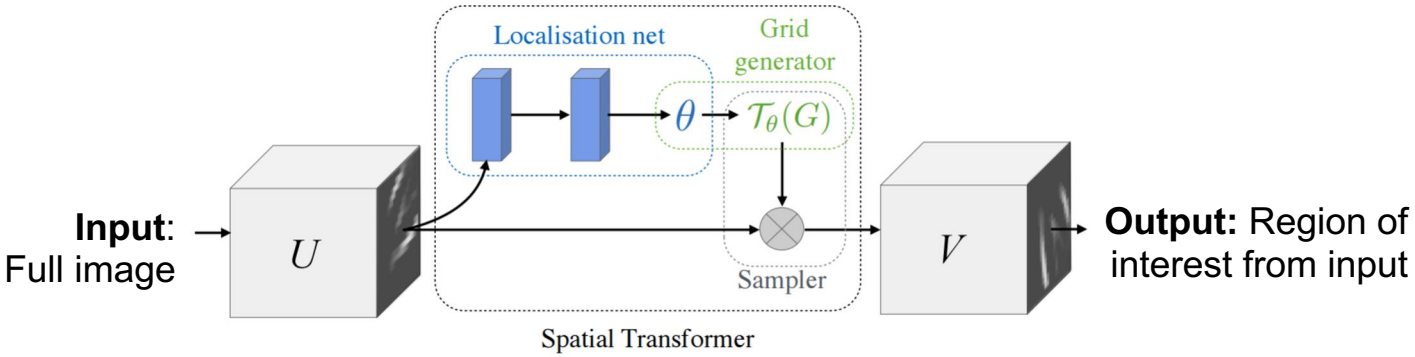


Repeat for all pixels in *output* to get a **sampling grid**

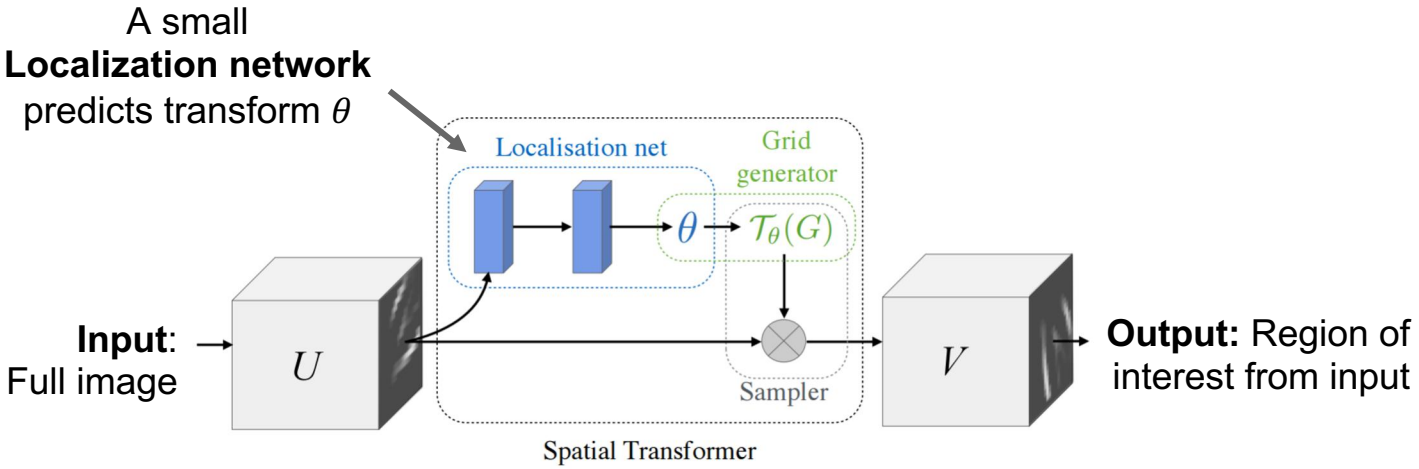
Then use **bilinear interpolation** to compute output

Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

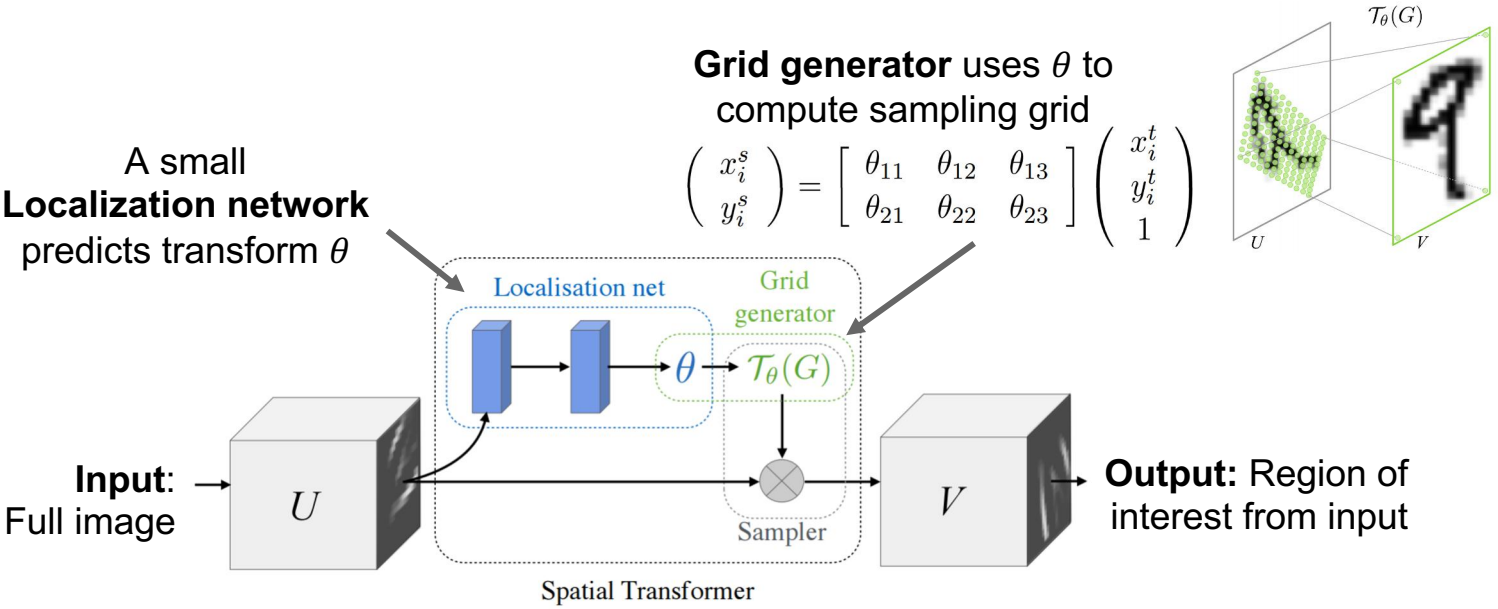
Spatial Transformer Networks



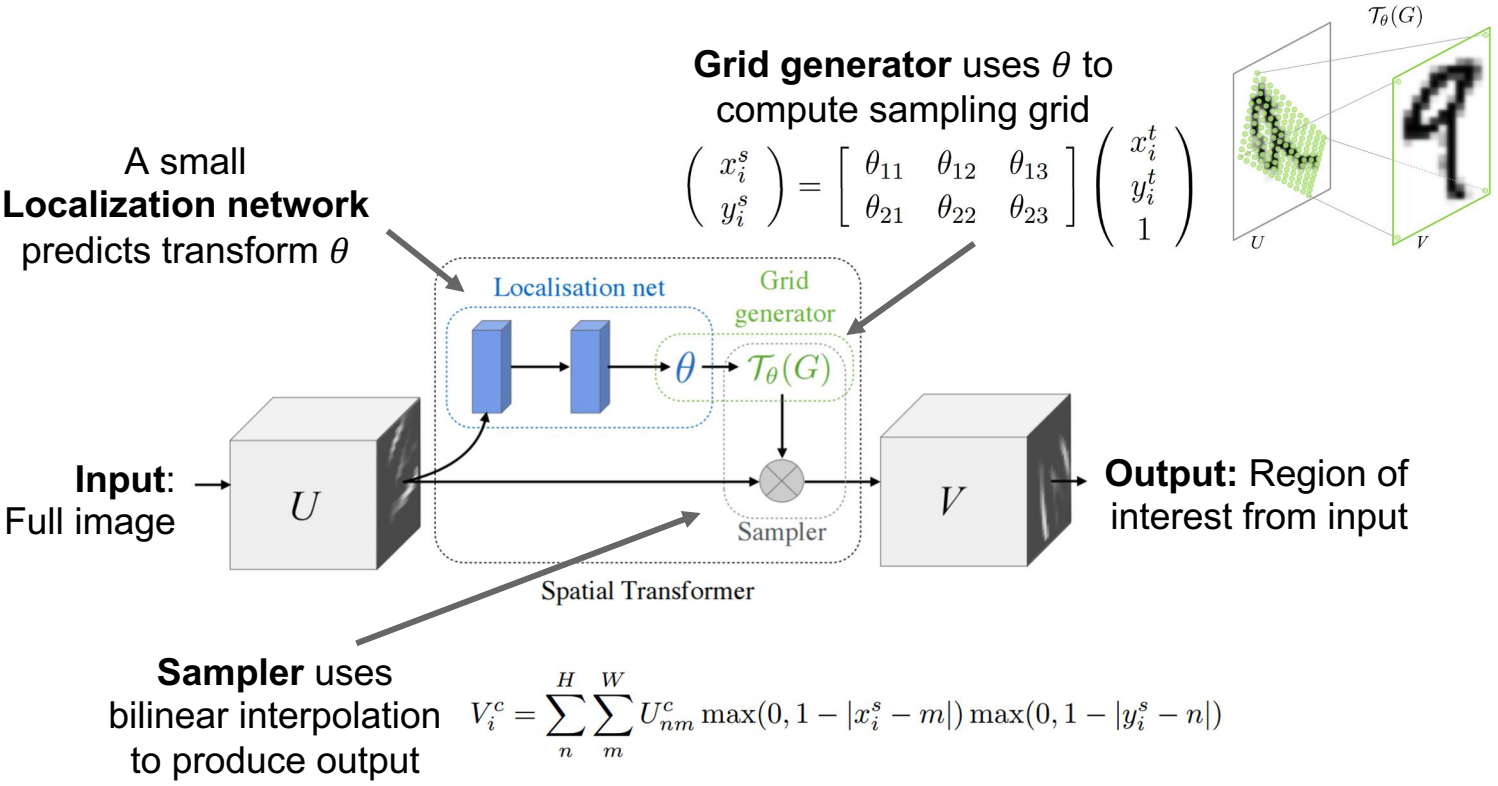
Spatial Transformer Networks



Spatial Transformer Networks

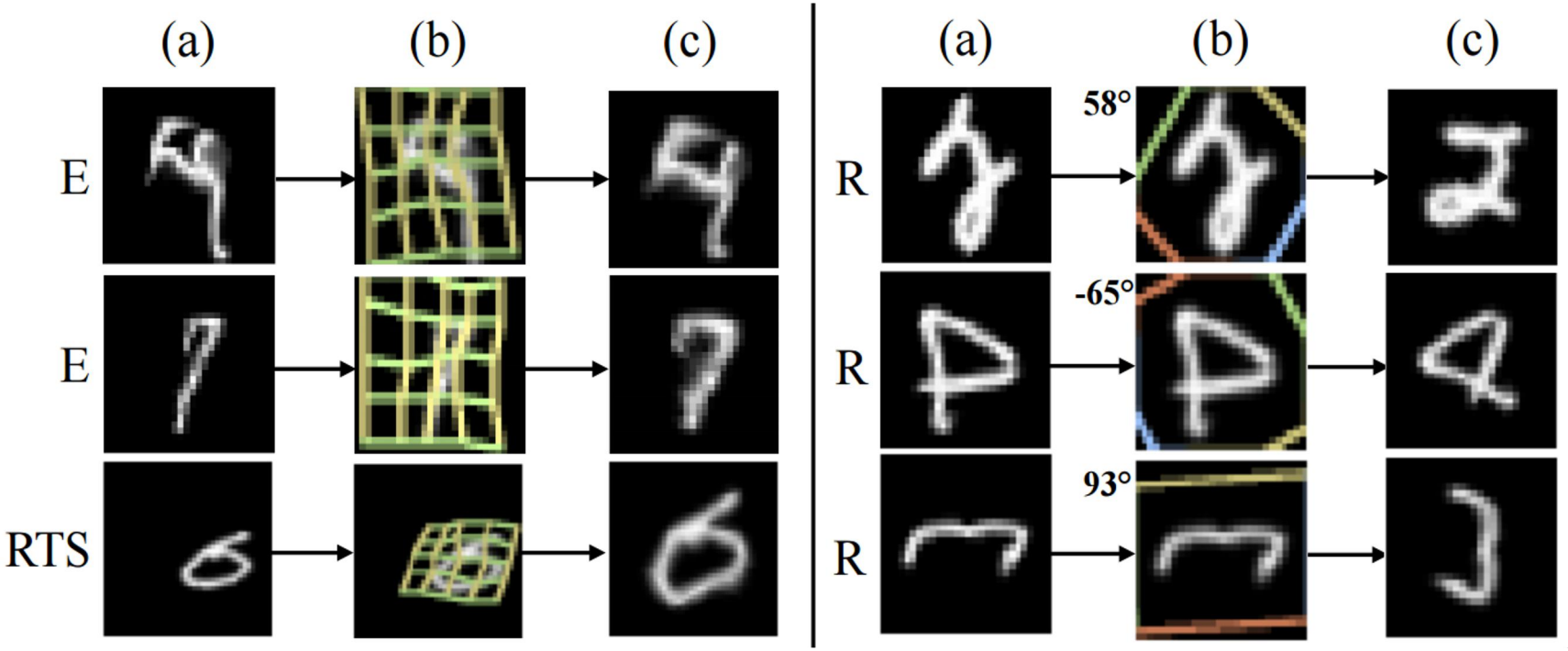


Spatial Transformer Networks



Spatial Transformer Networks

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



Attention Recap

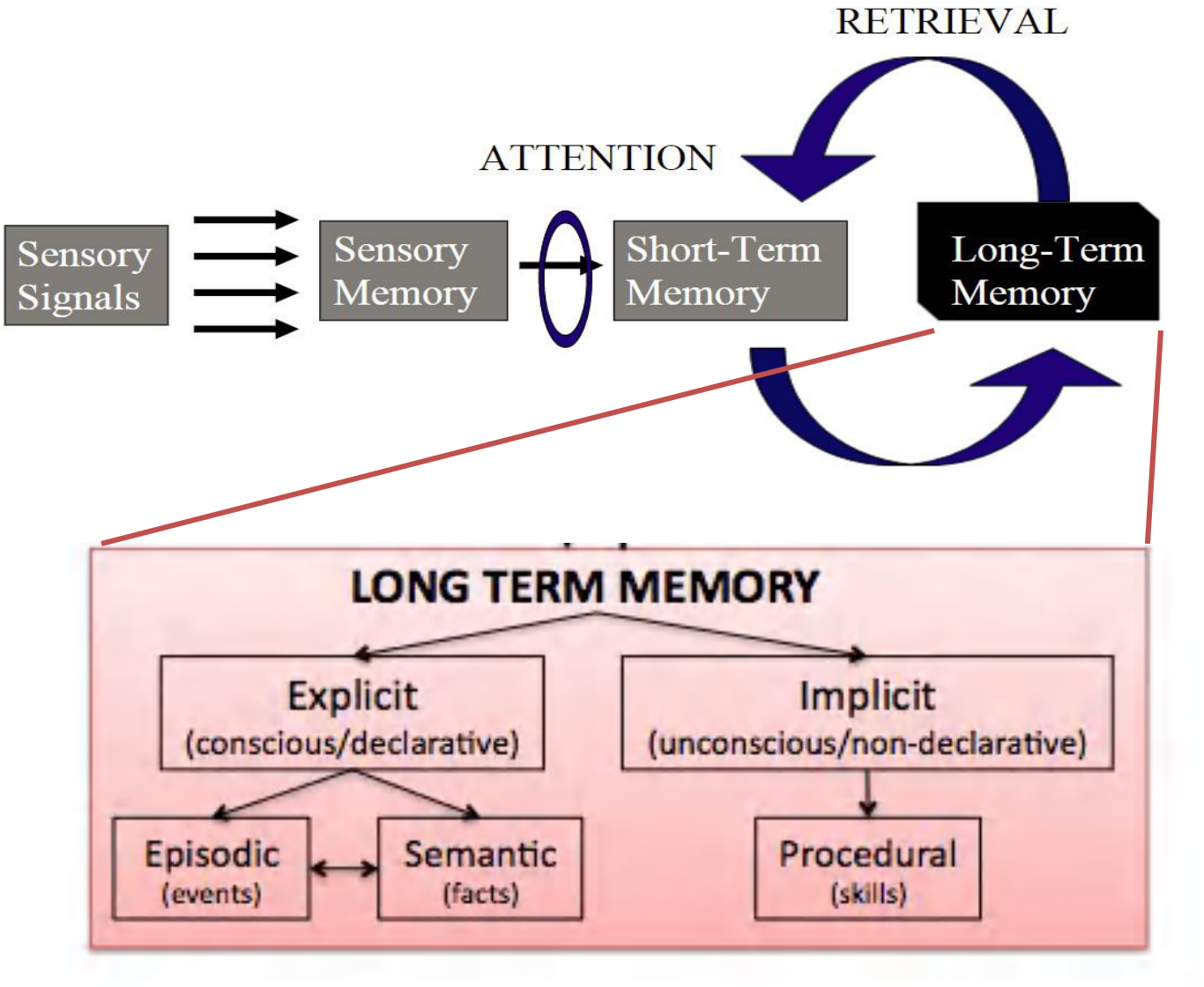
- Soft attention:
  - Easy to implement: produce distribution over input locations, reweight features and feed as input
  - Attend to arbitrary input locations using spatial transformer networks

- 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

Attention and Memory



https://memorisingmusic.files.wordpress.com/2013/03/long-termmemory.png?w=640

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?

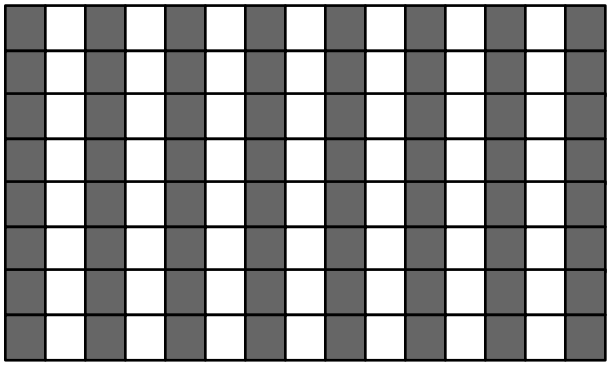
Why Need Memory

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



m_i

Each memory here is a column vector

Four Components :

I - Input Feature Map : Input manipulation

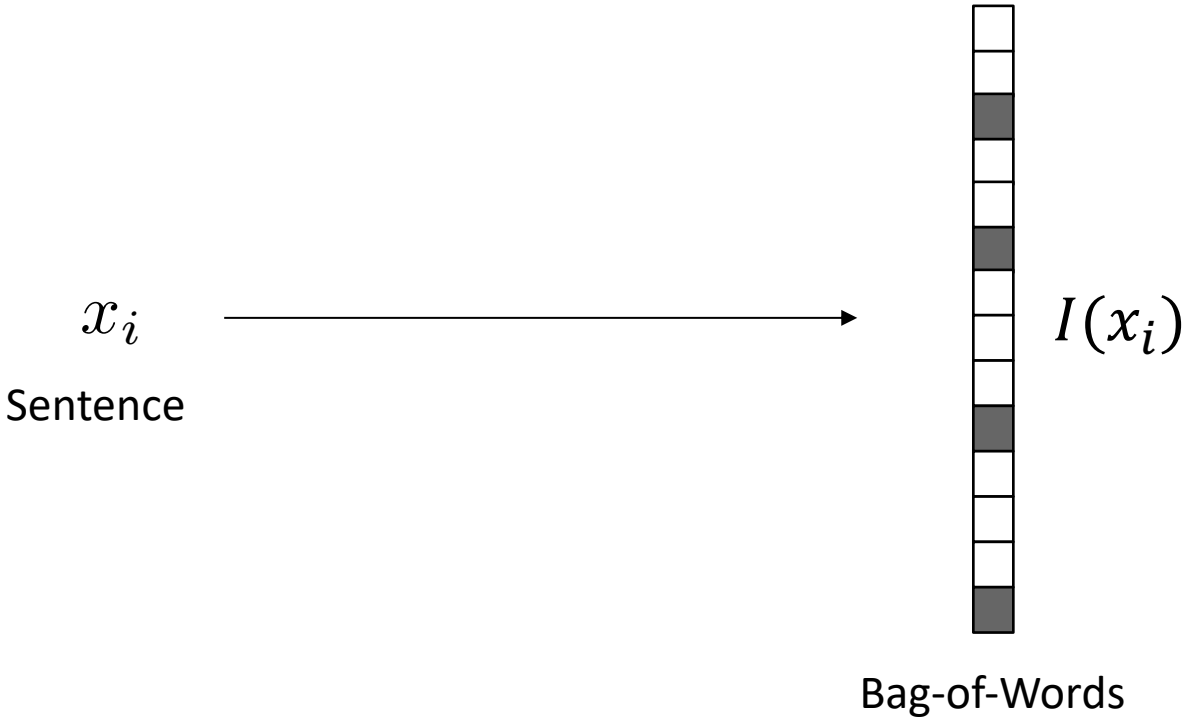
G - Generalization : Memory manipulation

O - Output Feature Map : Output representation generator

R - Response : Response generator

Simple MemNN for Text

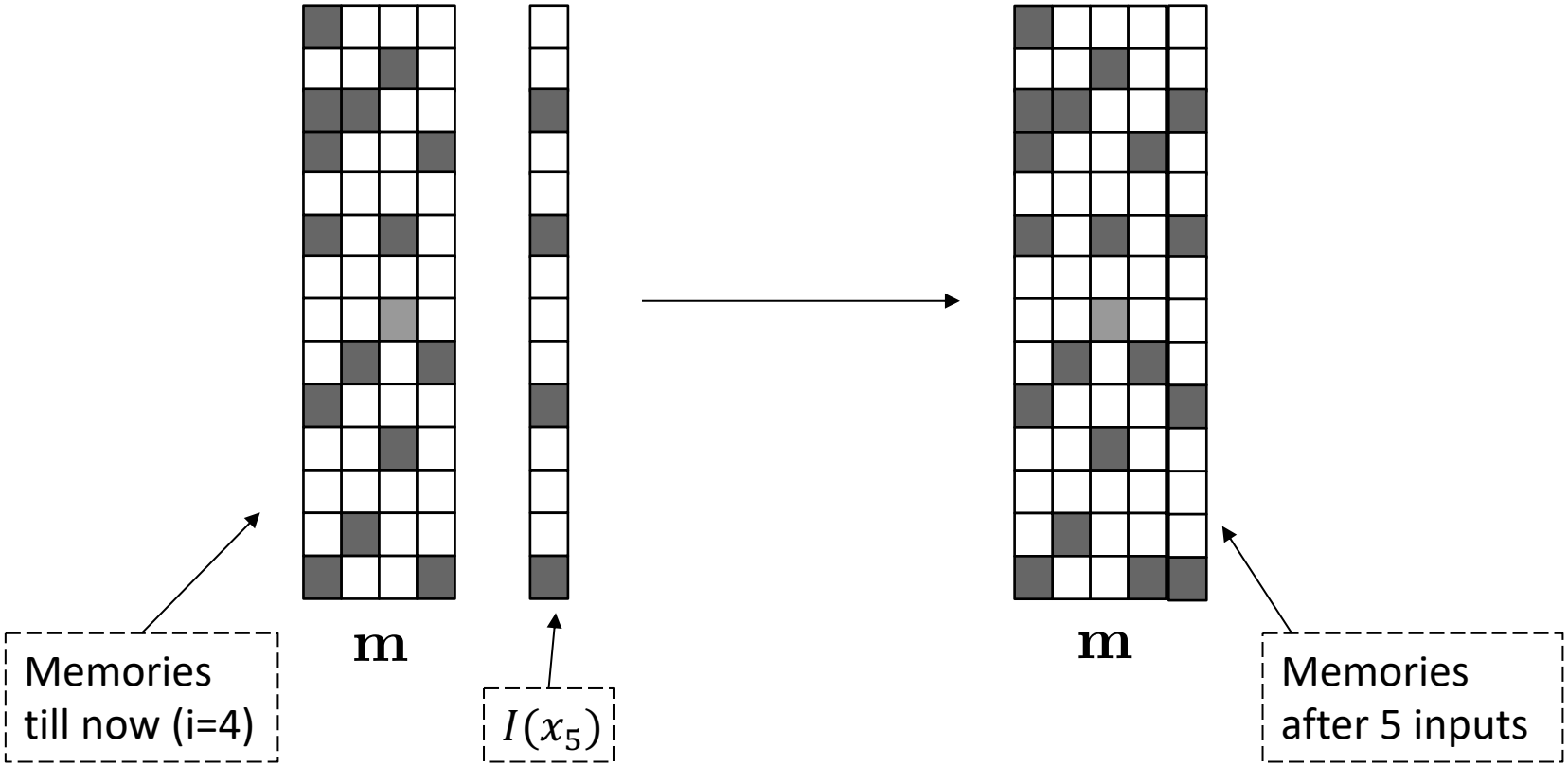
1. **Input Feature Map** - Bag-of-Words representation



Simple MemNN for Text

2. Generalization : Store input in new memory

$$m_i = I(x_i)$$



Memory Networks, Weston et. al., ICLR 2015

Simple MemNN for Text

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

1st Max scoring memory index

$$o_1 = O_1(x, \mathbf{m}) = \underset{i=1,\ldots,N}{argmax}\ s_O(I(x), m_i)$$

Score all memories against input

$$o_2 = O_2(x, \mathbf{m}) = \underset{i=1,\ldots,N}{argmax}\ s_O([I(x), m_{o_1}], m_i)$$

2nd Max scoring memory index

Score all memories against input & o_1

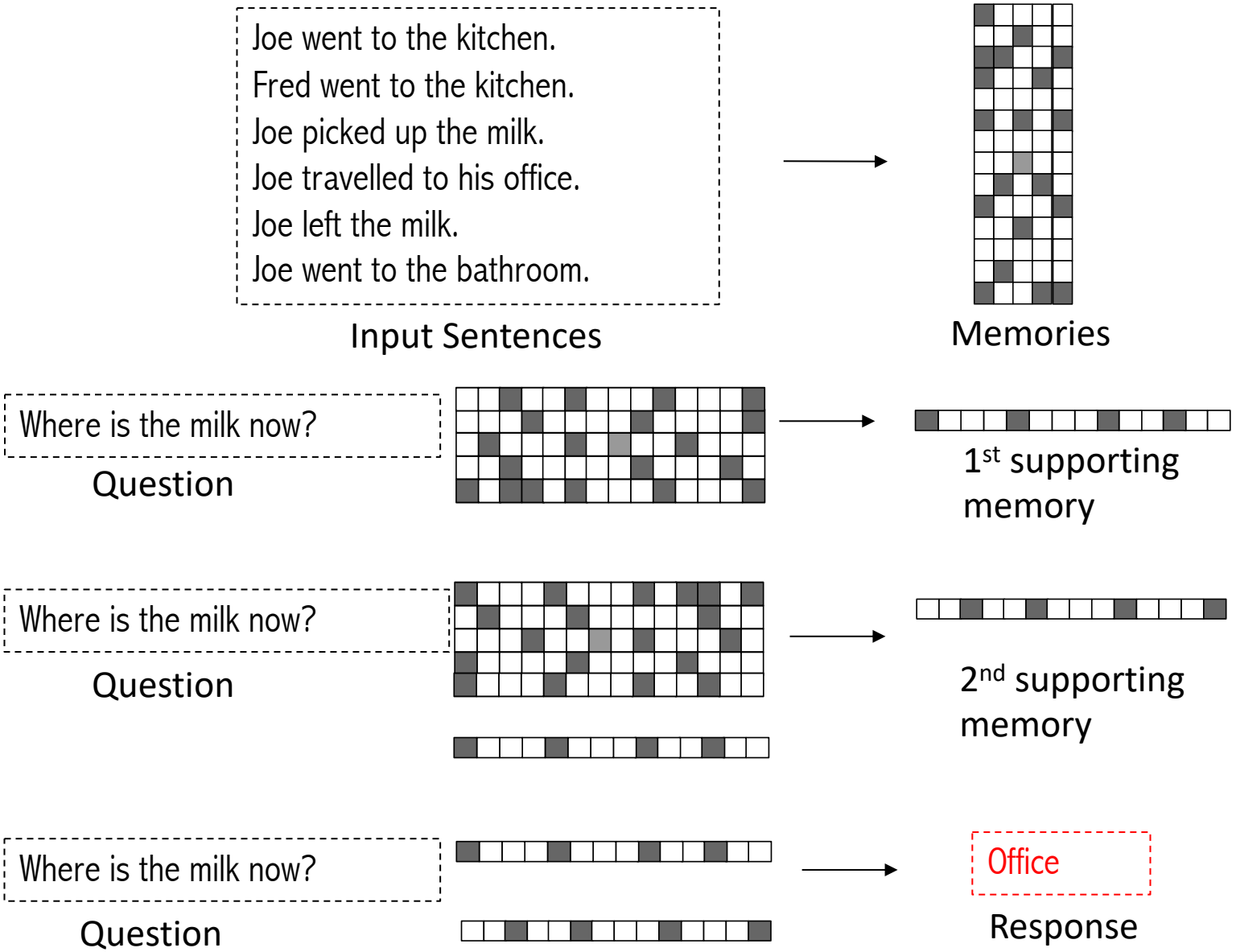
4. Response - Single Word Answer

Max scoring word

$$r = argmax_{w \in W}\ s_R([I(x), m_{o_1}, m_{o_2}], w)$$

Score all words against query and 2 supporting memories

Whole Pipeline



Memory Networks, Weston et. al., ICLR 2015

Limitations

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

Memory Networks, Weston et. al., ICLR 2015

End-to-End Memory Networks (MemN2N)

- What if the annotation is:
  - Input sentences $x_1, x_2, \ldots, x_n$
  - Query q
  - Answer a

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.

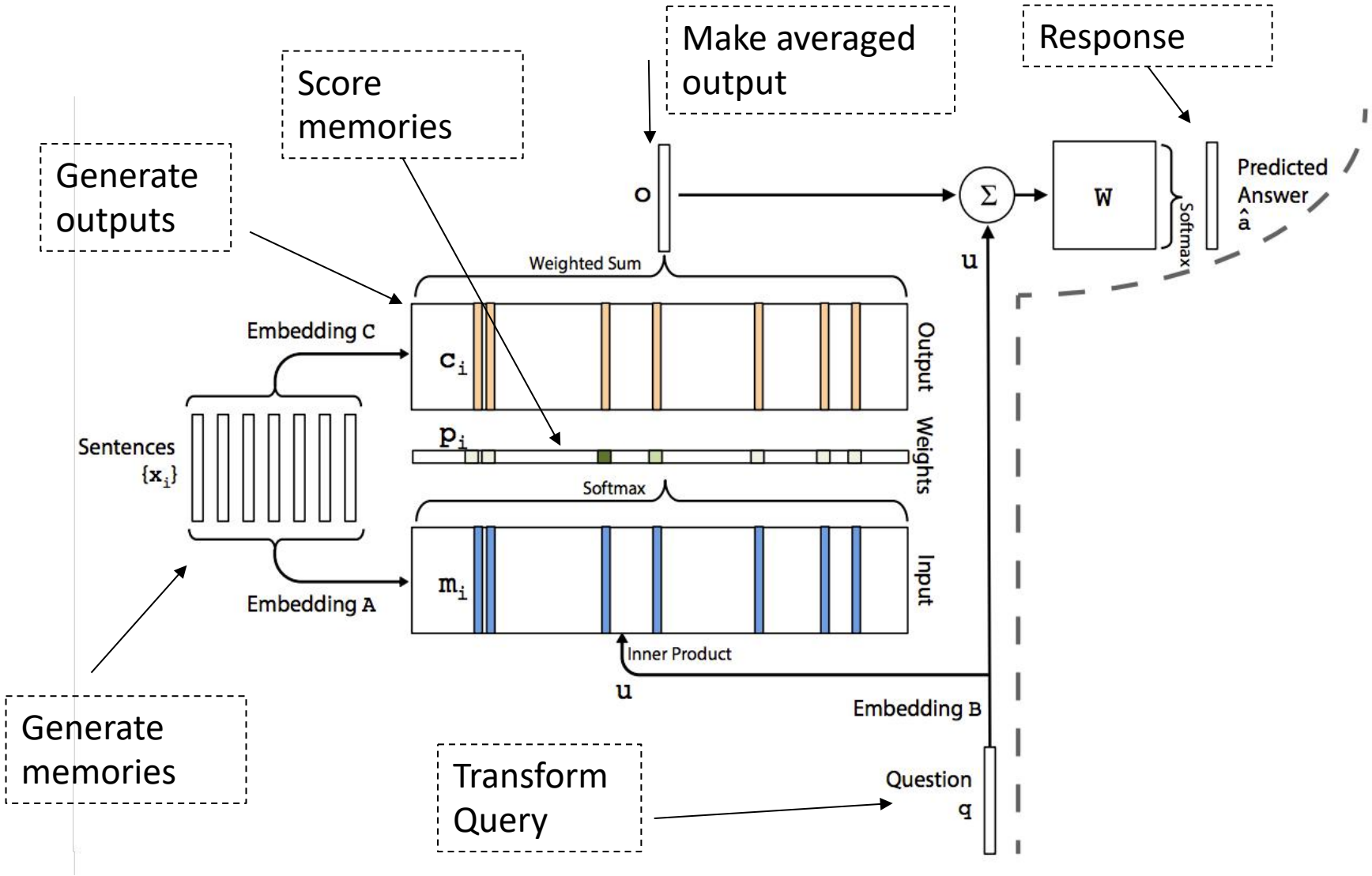
Where is the milk now?

Office

- 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

End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

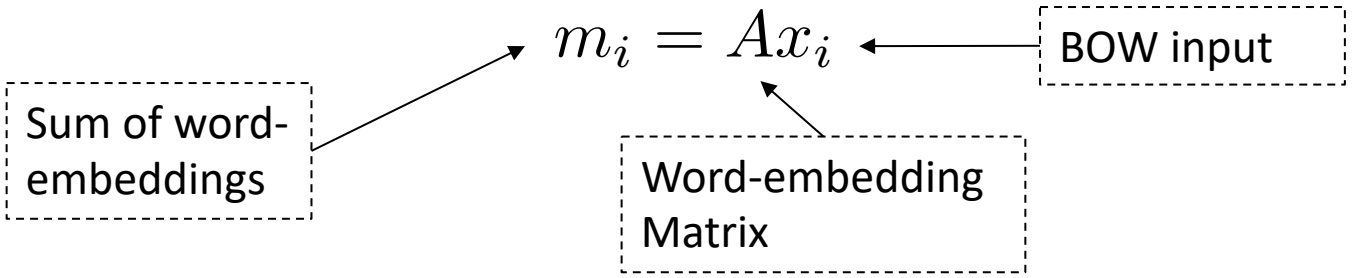
End-to-End Memory Networks (MemN2N)



End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

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, Sukhbaatar et. al., NIPS 2015

End-to-End Memory Networks (MemN2N)

3. Scoring memories against query

$$p_i = \text{Softmax}(u^T m_i)$$

Score for input/memory

Memories

Query (transformed)

4. Generate output

$$o = \sum_i p_i c_i$$

Weighted average of all inputs (transformed)

End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

End-to-End Memory Networks (MemN2N)

5. Generating Response

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

Distribution over response words

Query

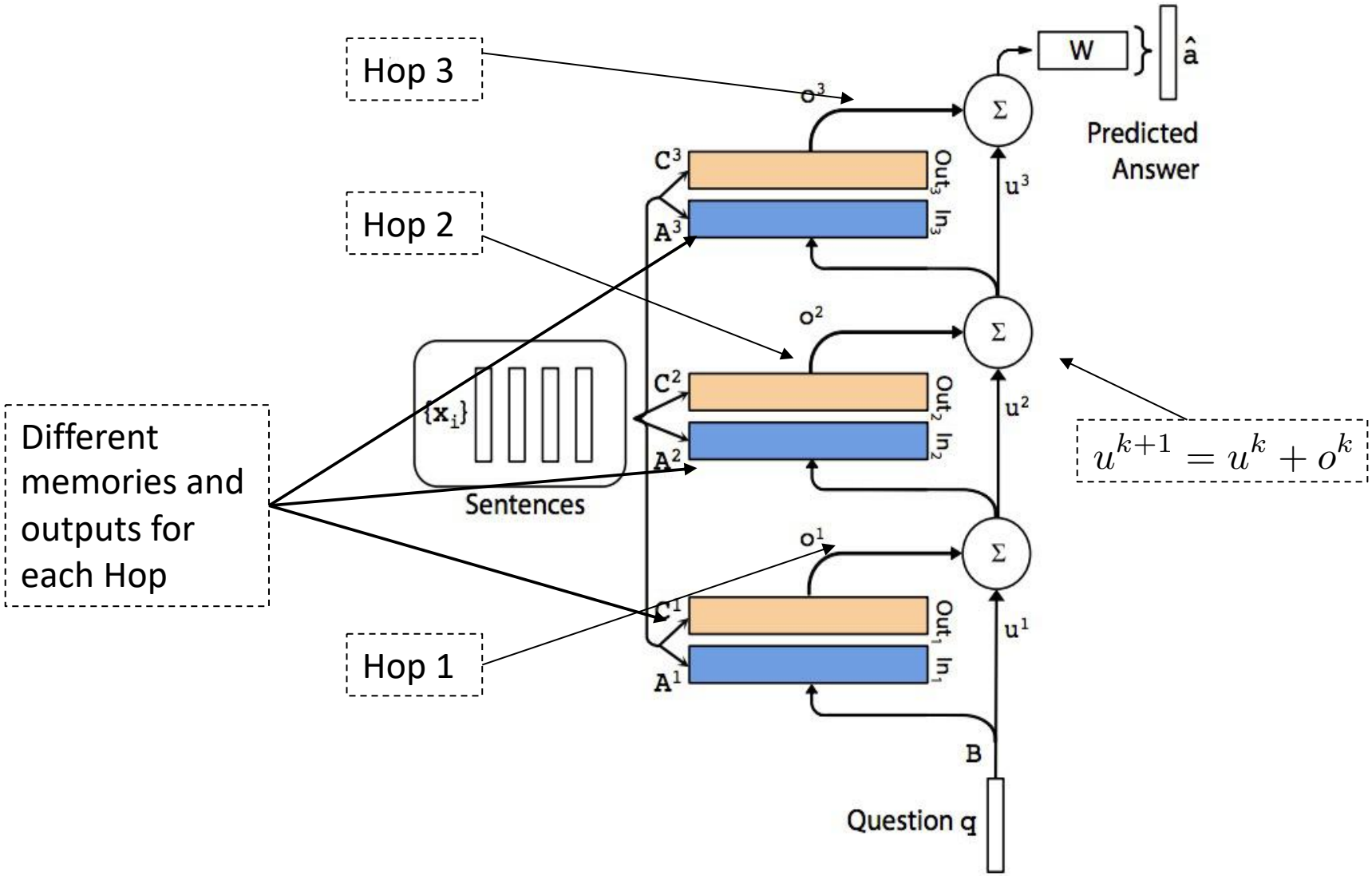
Averaged-output

Training Objective – Maximum Likelihood / Cross Entropy

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

End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

Multi-hop MemN2N



End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

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"*

End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

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**		

End-To-End Memory Networks, Sukhbaatar et. al., NIPS 2015

Movie Trivia Time

- Which was *Stanley Kubricks*'s first movie?

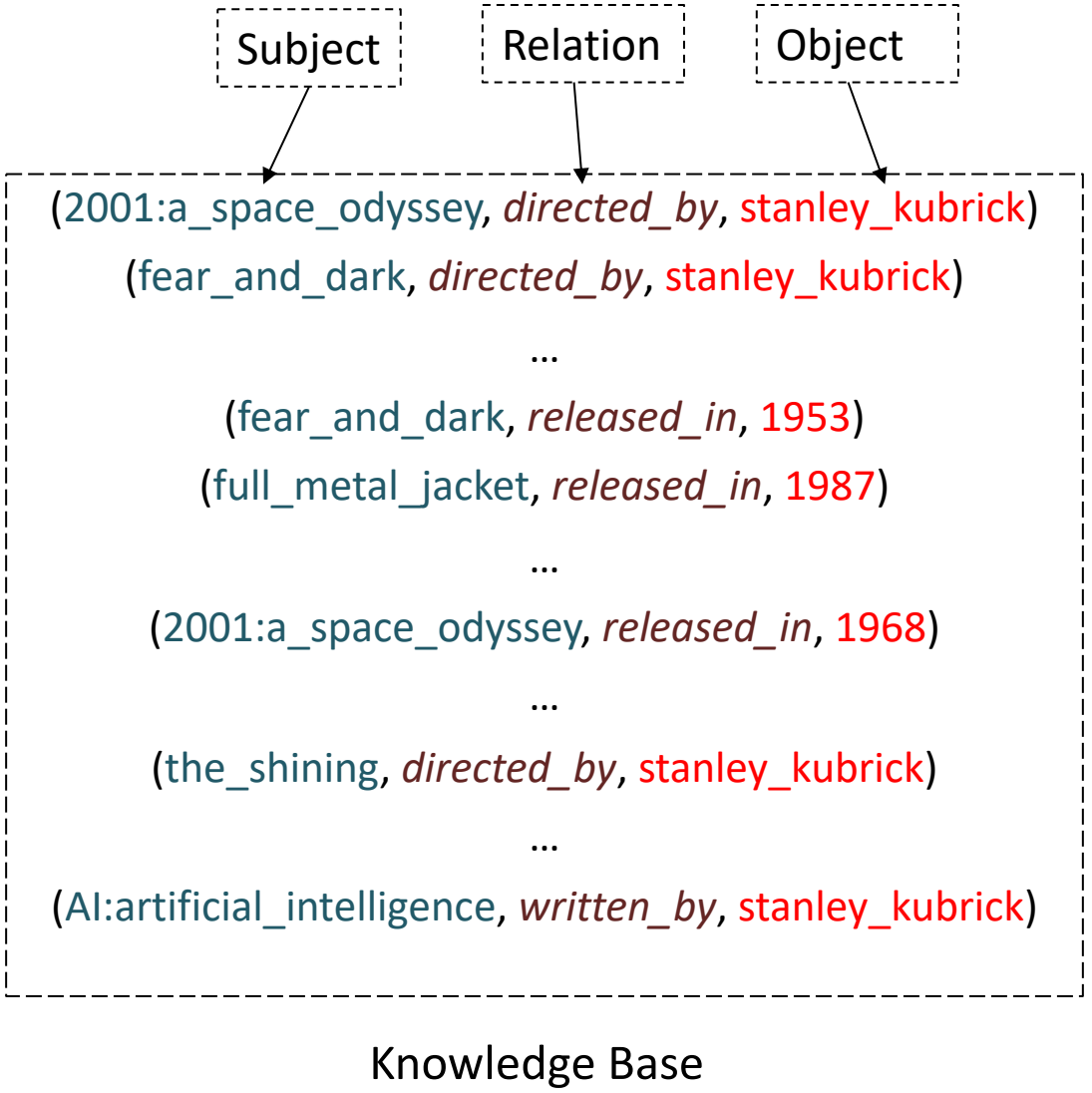
  Fear and Dark

- When did *2001:A Space Odyssey* release?

  1968

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

  Full Metal Jacket



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

$$\text{Memories} = (k_1, v_1),\ (k_2, v_2),\ \ldots,\ (k_N, v_N)$$

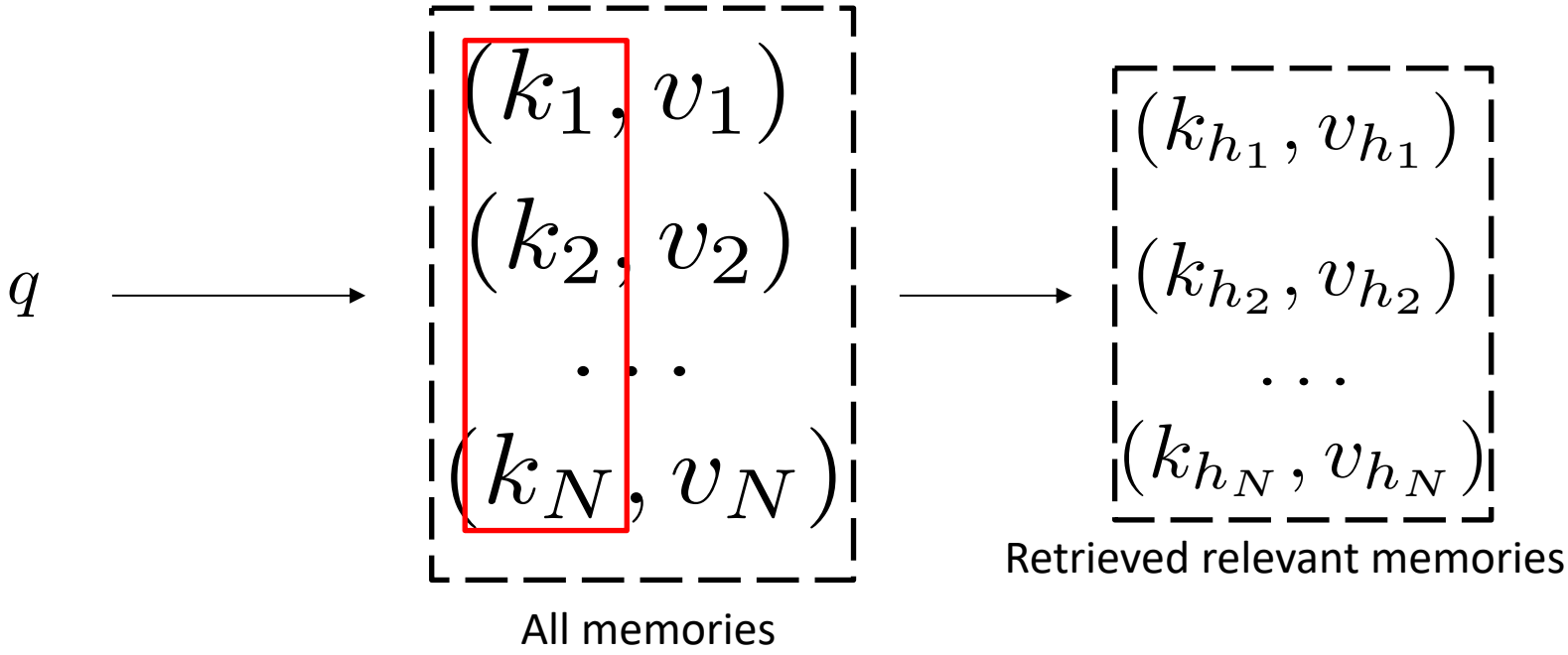
(*k: Kubrick's first movie was, v: Fear and Dark*)

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

Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

Key-Value MemNNs

1. Retrieve relevant memories using keys



Use inverted index, locality sensitive hashing, something sensible

Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

Key-Value MemNNs

2. Score Memory-Keys

$$p_i = \text{Softmax}(u^T m_i)$$

$$p_{h_i} = \text{Softmax}(A\Phi_Q(q) \cdot A\Phi_K(k_{h_i}))$$

Distribution over memory-keys

Dot-prod

Transformation of embeddings

BOW

Key

3. Generate Output

$$o = \sum_i p_{h_i} A\Phi_V(v_{h_i})$$

$$o = \sum_i p_i c_i$$

Weighted average of memory-values

Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

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

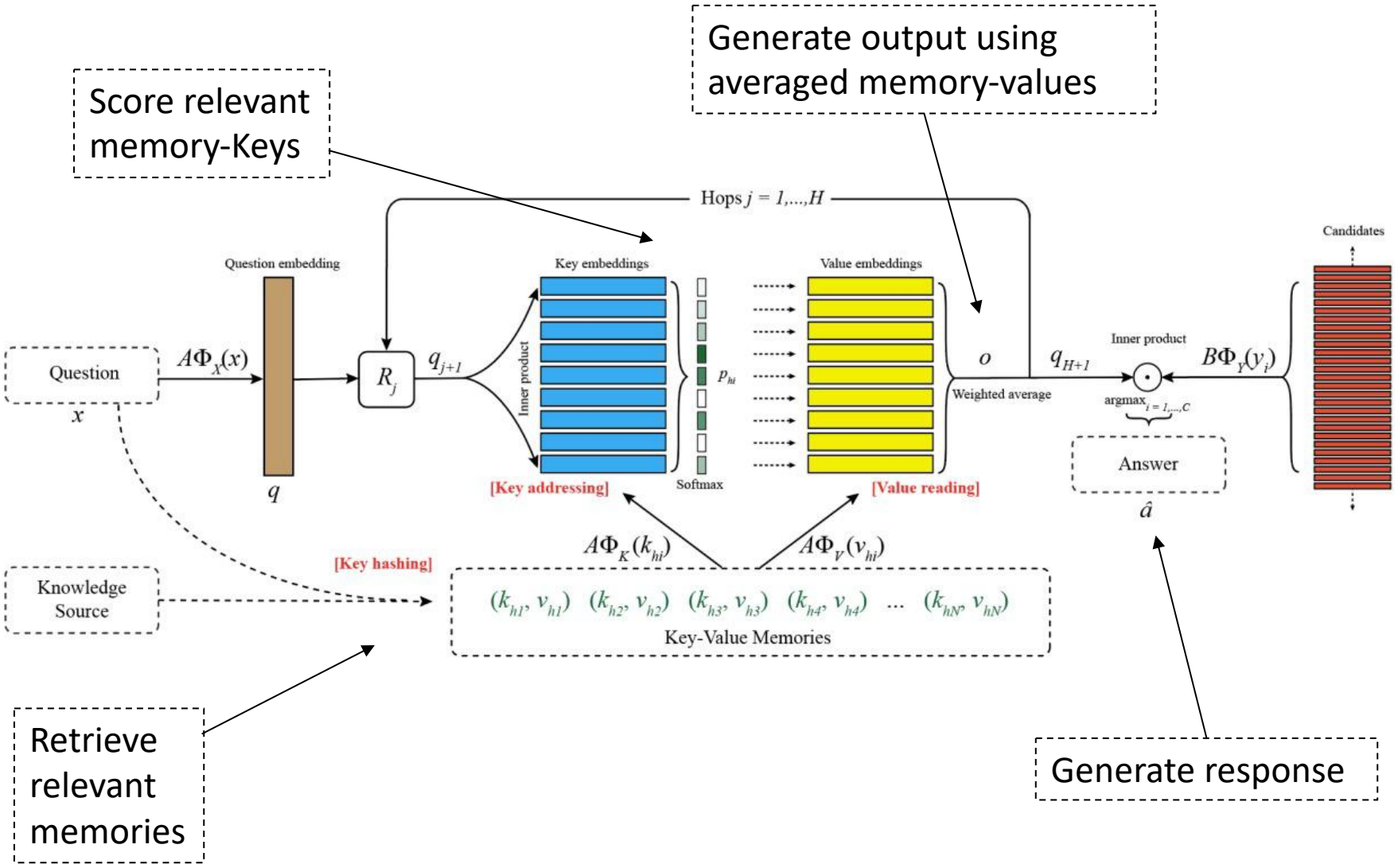
For each entity in document, extract 5-word window around it

Key: window; Value: Entity

K: screenplay written by and; V: Hampton

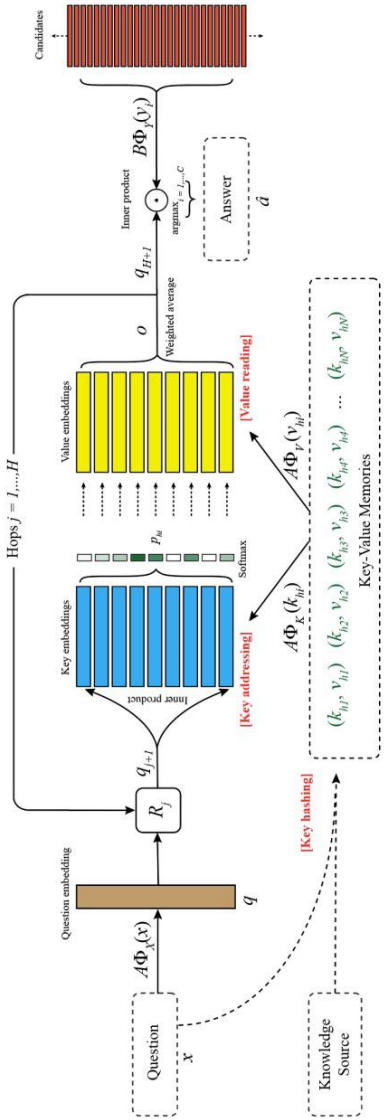
Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

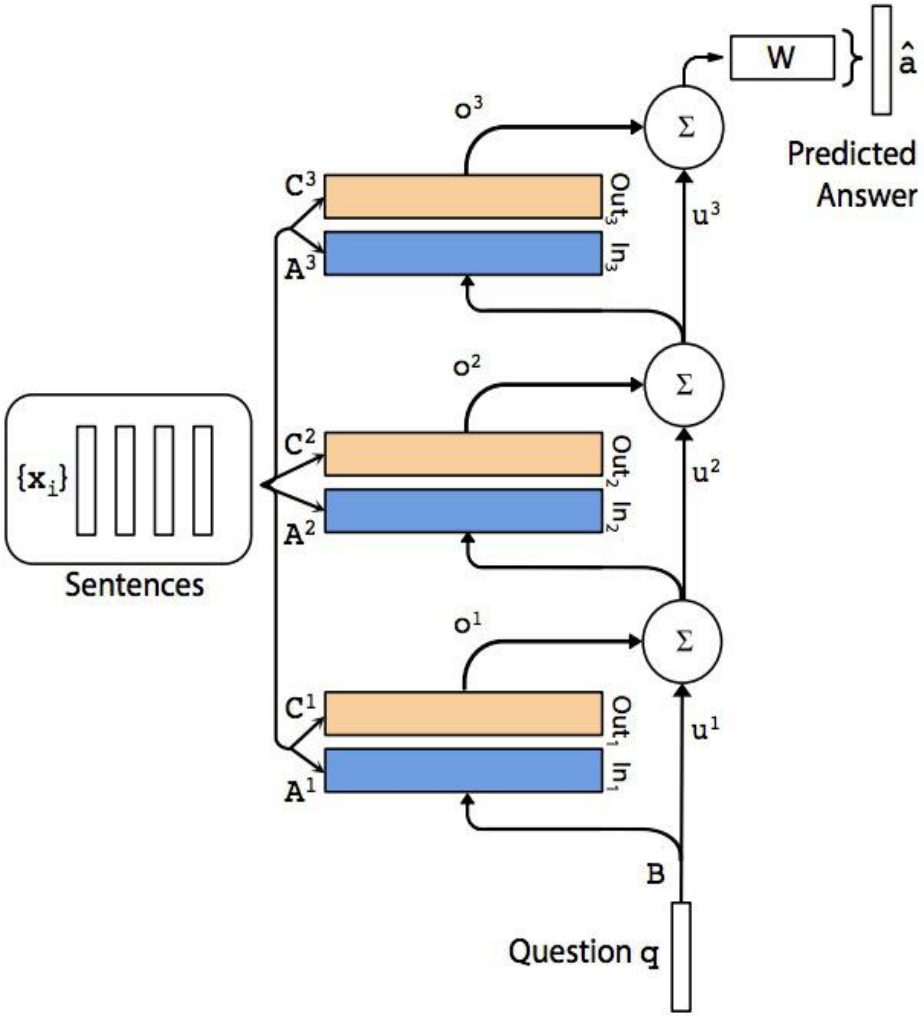
KV-MemNN



Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

KV-MemNN





Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

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

Key-Value Memory Networks for Directly Reading Documents, Miller et. al., EMNLP 2016

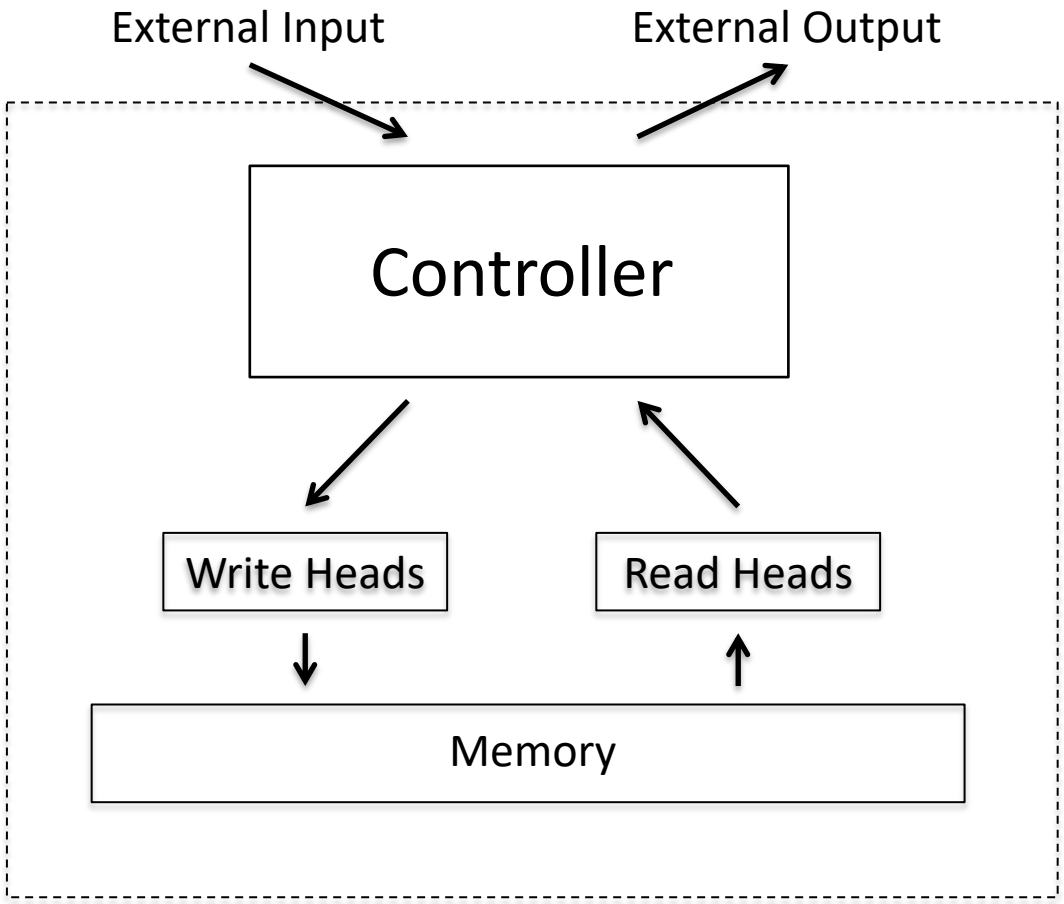
Neural Turing Machine

A more general memory

Neural Turing Machine learns:

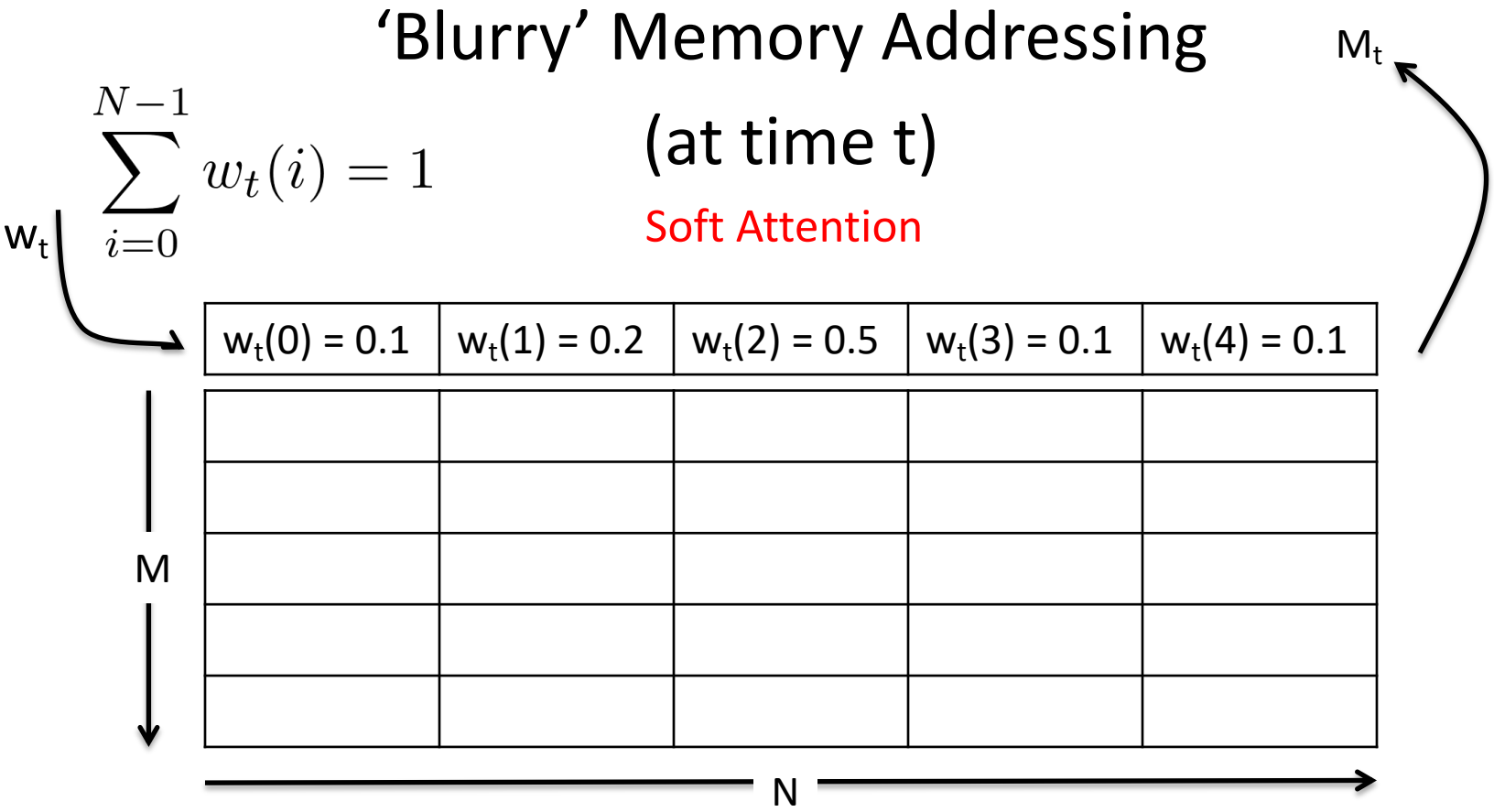
1. What to write to memory
2. When to write to memory
3. When to stop writing
4. Which memory cell to read from
5. How to convert result of read into final output

Neural Turing Machine



Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machine



Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machine

More formally,

Blurry Read Operation

Given: M_t (memory matrix) of size NxM

w_t (weight vector) of length N

t (time index)

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

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Blurry Writes

Blurry Write Operation

Decomposed into blurry erase + blurry add

Given: M_t (memory matrix) of size NxM

w_t (weight vector) of length N

t (time index)

e_t (erase vector) of length M

a_t (add vector) of length M

$$\mathbf{M}_t(i) = \underbrace{\mathbf{M}_{t-1}(i)(1 - w_t(i)\mathbf{e}_t)}_{\text{Erase Component}} + \underbrace{w_t(i)\mathbf{a}_t}_{\text{Add Component}}$$

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Erase

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

1xN

	$w_1(0) = 0.1$	$w_1(1) = 0.2$	$w_1(2) = 0.5$	$w_1(3) = 0.1$	$w_1(4) = 0.1$	e_1
M_0	5	7	9	2	12	1.0
	11	6	3	1	2	0.7
	3	7	3	10	6	0.2
	4	2	5	9	9	0.5
	3	5	12	8	4	0.0

Mx1

Neural Turing Machines, Graves et. al., arXiv:1410.5401

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, Graves et. al., arXiv:1410.5401

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, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Blurry Writes

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

M_1 ⇨

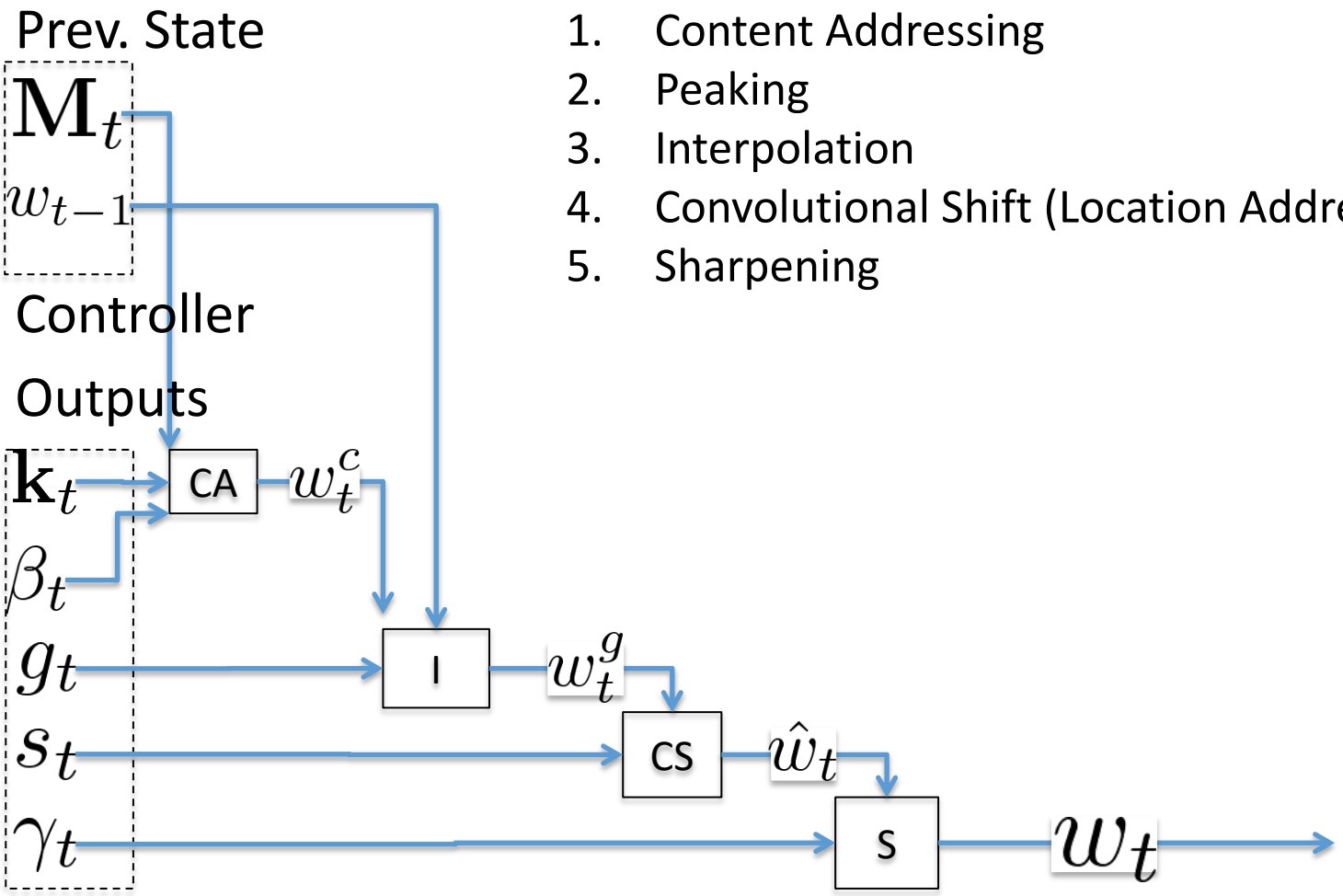
4.8	6.2	6	2.1	11.1
10.63	5.96	3.95	1.33	2.26
2.74	6.32	1.7	9.6	5.68
3.8	1.8	3.75	8.55	8.55
3.2	5.4	13	8.2	4.2

Neural Turing Machines, Graves et. al., arXiv:1410.5401

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, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Generate w_t

Prev. State

$\mathbf{M}_t$

Controller

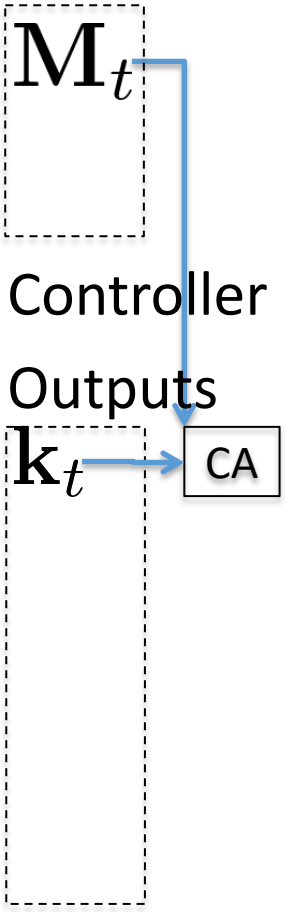
Outputs

$\mathbf{k}_t$

$\mathbf{k}_t$:Vector (length M) produced by Controller

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Generate w_t

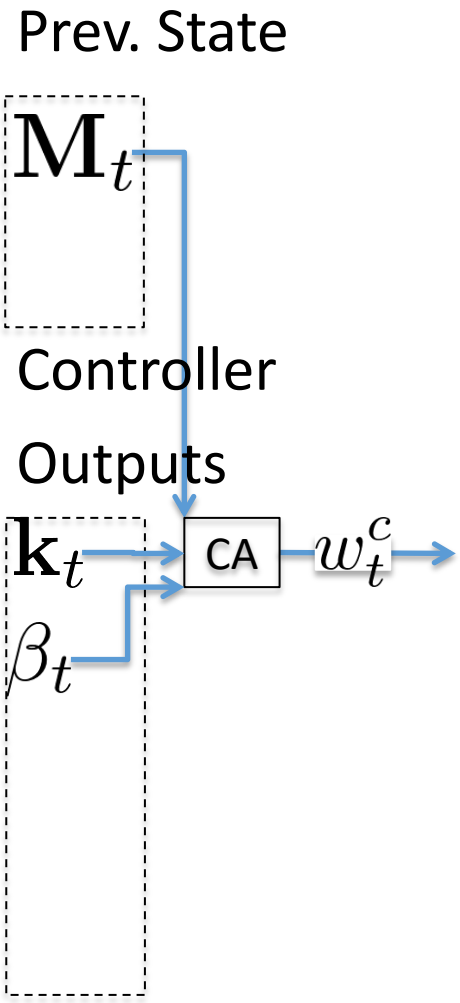


Step 1: Content Addressing (CA)

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

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Generate w_t

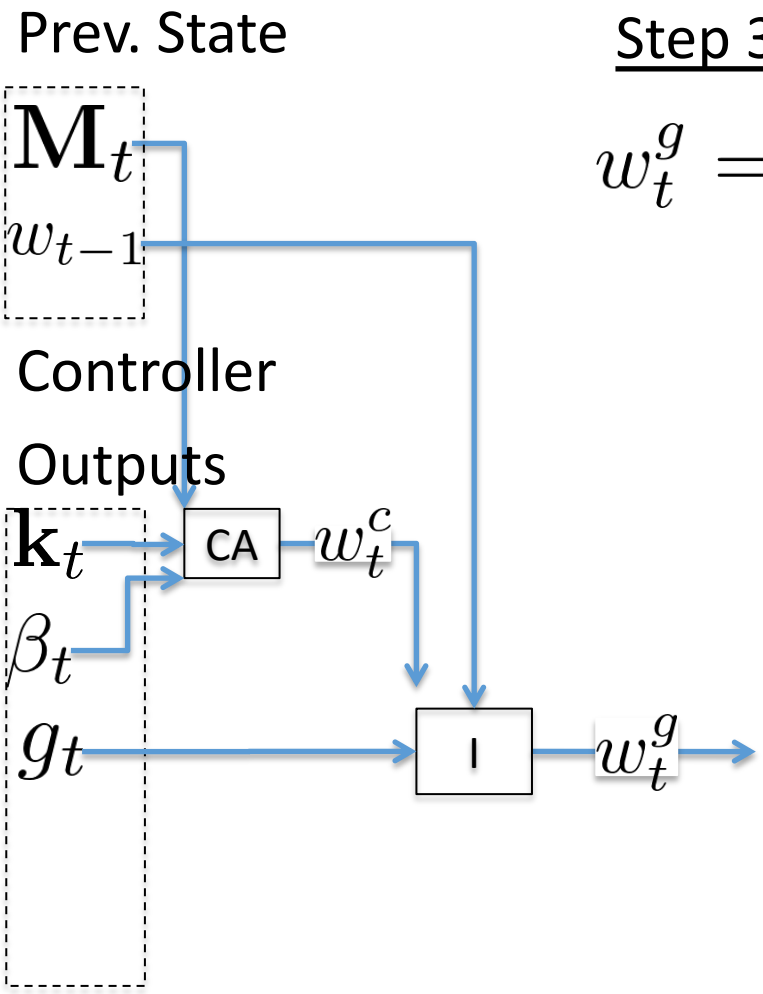


Step 2: Peaking

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

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Generate w_t

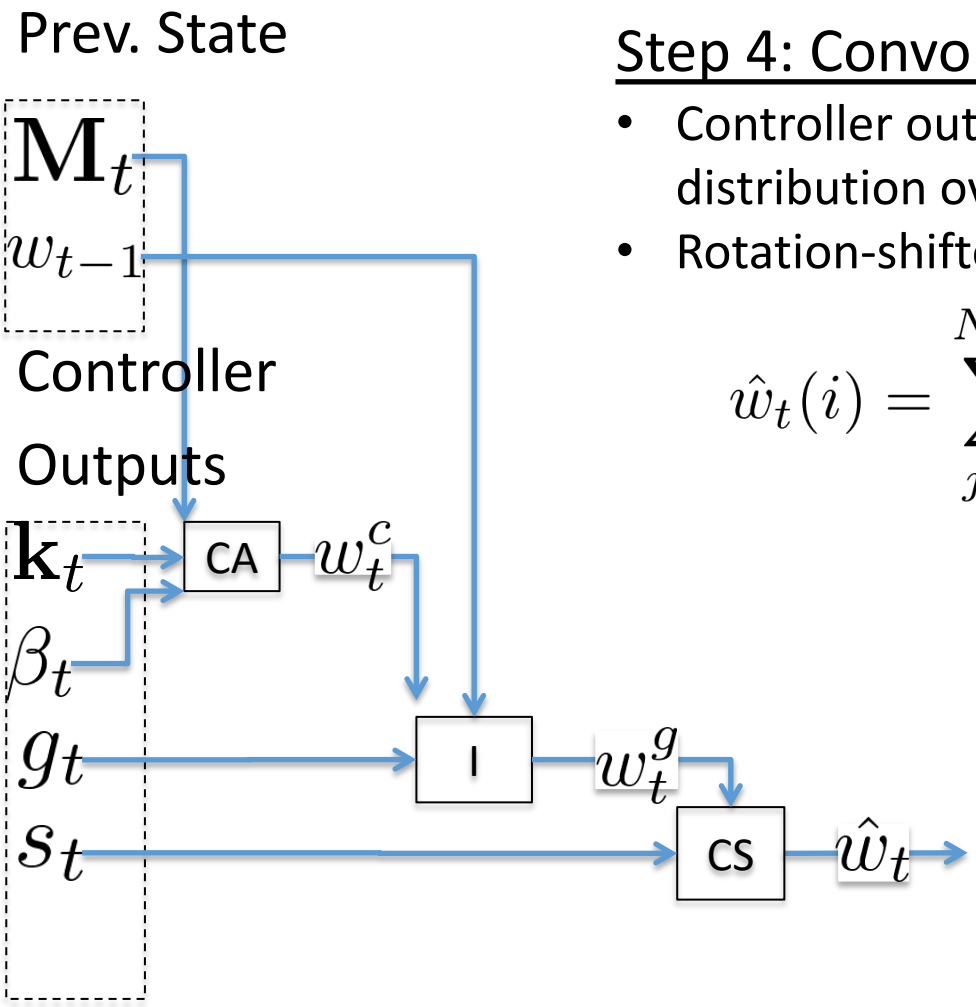


Step 3: Interpolation (I)

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

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines: Generate w_t

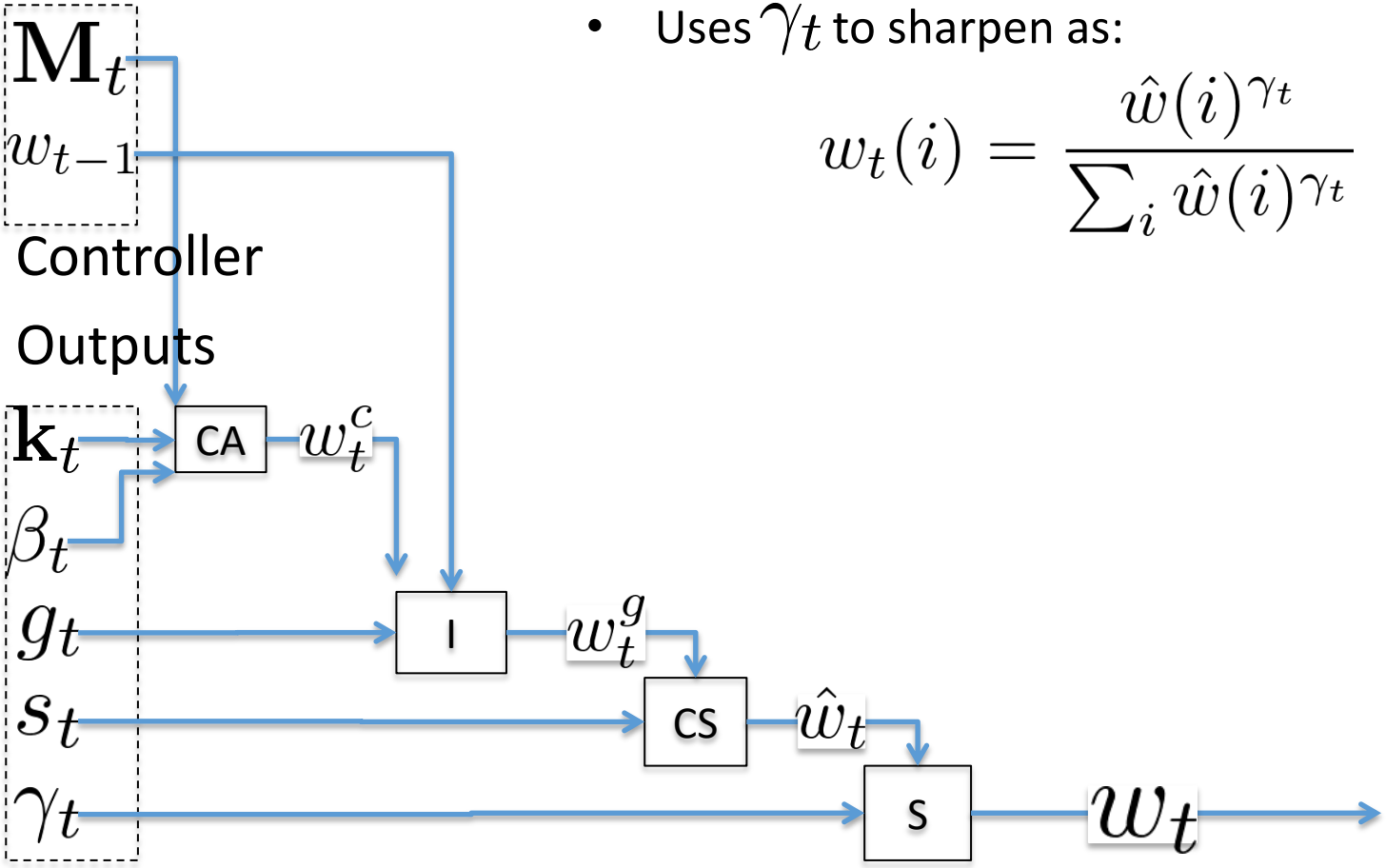


Step 4: Convolutional Shift (CS)

- 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



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 Machines, Graves et. al., arXiv:1410.5401

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 Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machine: Network Overview

Unrolled Feed-forward Controller

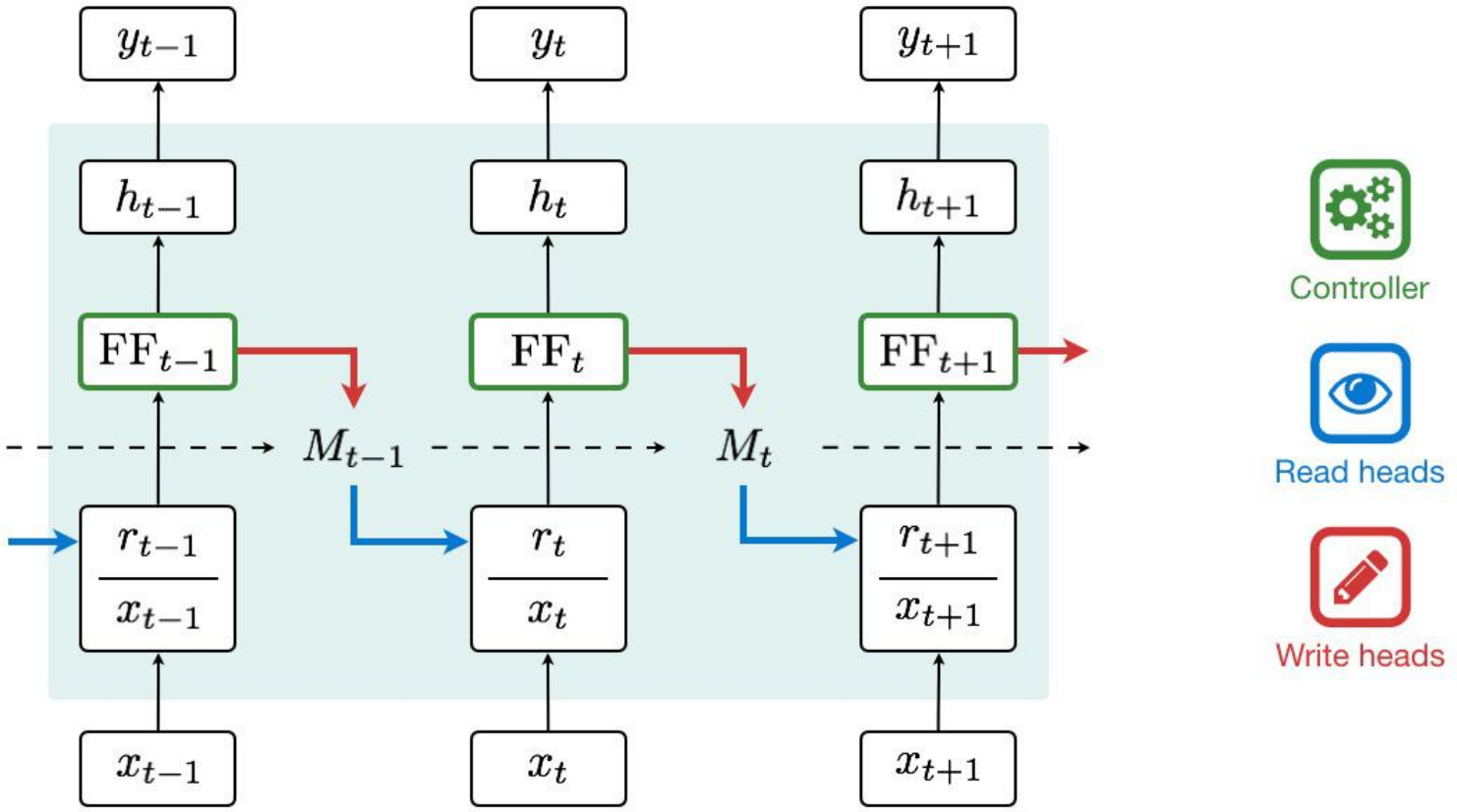


Figure from [Snips AI's Medium Post]

Neural Turing Machines, Graves et. al., arXiv:1410.5401

Neural Turing Machines vs. MemNNs

MemNNs

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

NTMs

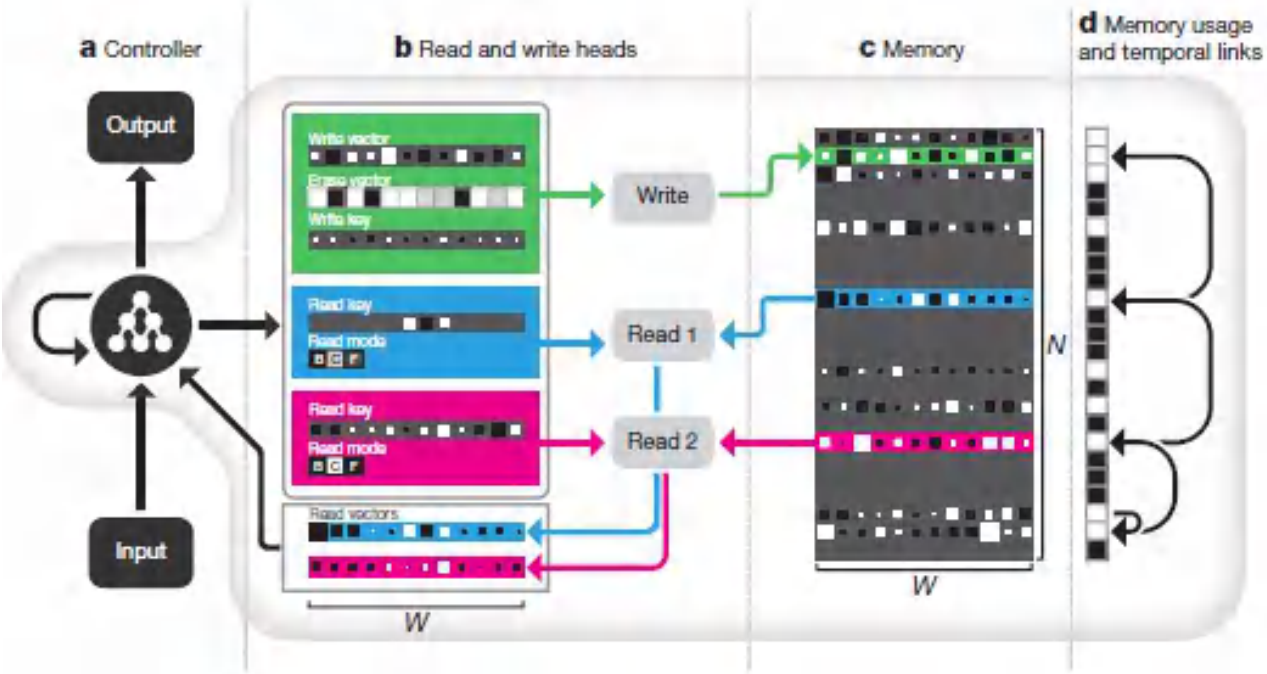
- Memory is continuously written to and read from, with network learning when to perform memory read and write

Differentiable Neural Computers

doi:10.1038/nature20101

Hybrid computing using a neural network with dynamic external memory

Alex Graves[1]*, Greg Wayne[1]*, Malcolm Reynolds[1], Tim Harley[1], Ivo Danihelka[1], Agnieszka Grabska-Barwińska[1], Sergio Gómez Colmenarejo[1], Edward Grefenstette[1], Tiago Ramalho[1], John Agapiou[1], Adrià Puigdomènech Badia[1], Karl Moritz Hermann[1], Yori Zwols[1], Georg Ostrovski[1], Adam Cain[1], Helen King[1], Christopher Summerfield[1], Phil Blunsom[1], Koray Kavukcuoglu[1] & Demis Hassabis[1]



Hybrid computing using a neural network with dynamic external memory, Graves et. al., Nature, 2016

Differentiable Neural Computers

Advanced addressing mechanisms:

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

Hybrid computing using a neural network with dynamic external memory, Graves et. al., Nature, 2016

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

Hybrid computing using a neural network with dynamic external memory, Graves et. al., Nature, 2016

DNC: Improvements over NTMs

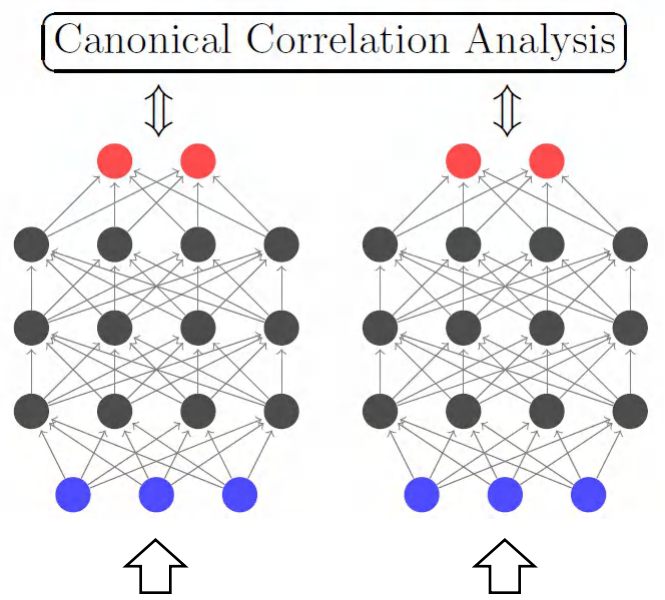
NTM	DNC
• Large contiguous blocks of memory needed	• Memory locations non-contiguous, usage-based
• No way to free up memory cells after writing	• Regular re-allocation based on usage-tracking

Hybrid computing using a neural network with dynamic external memory, Graves et. al., Nature, 2016

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



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

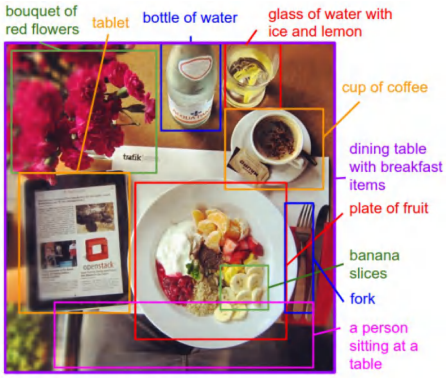


Image question answering

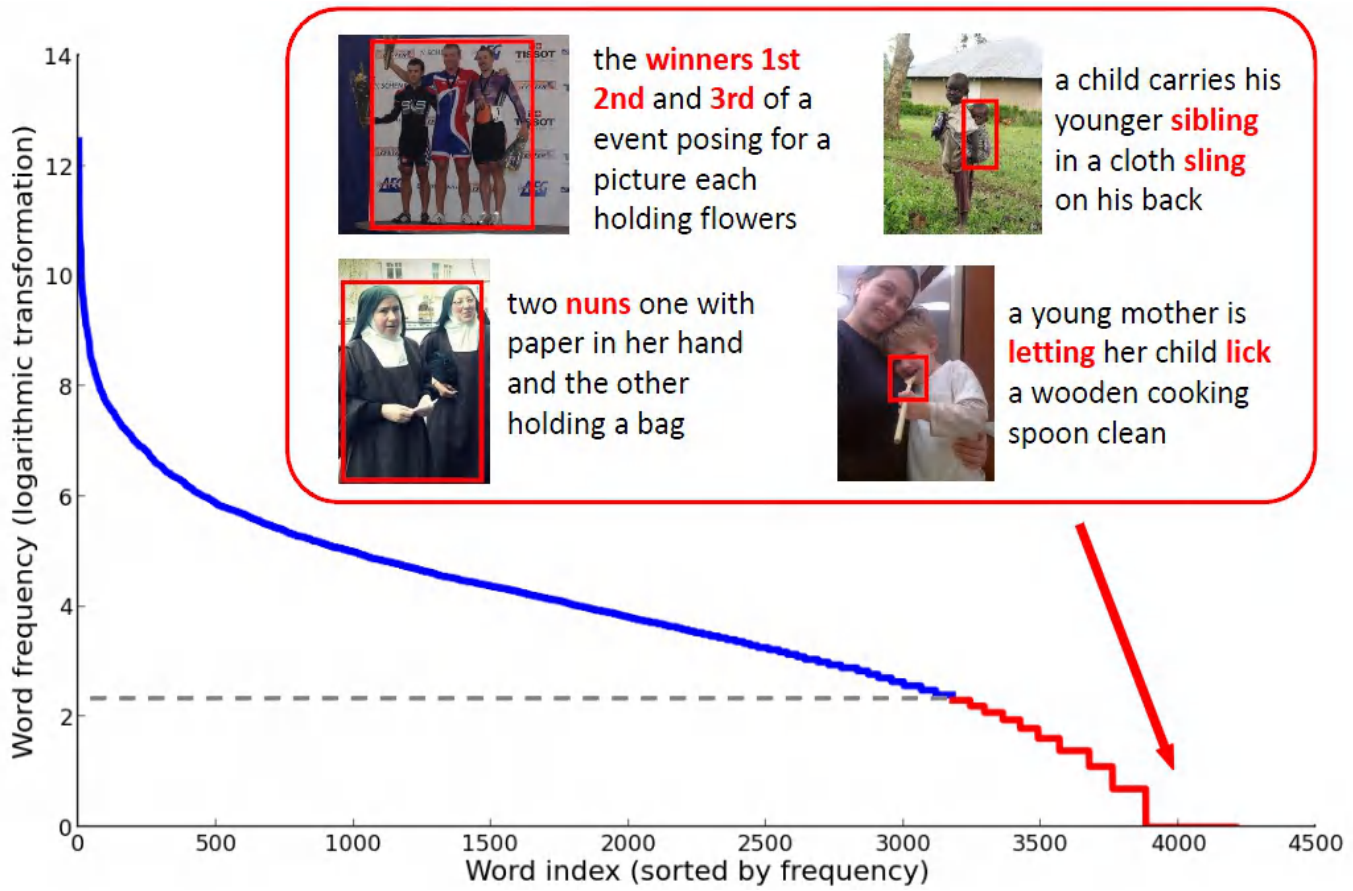
Image

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.

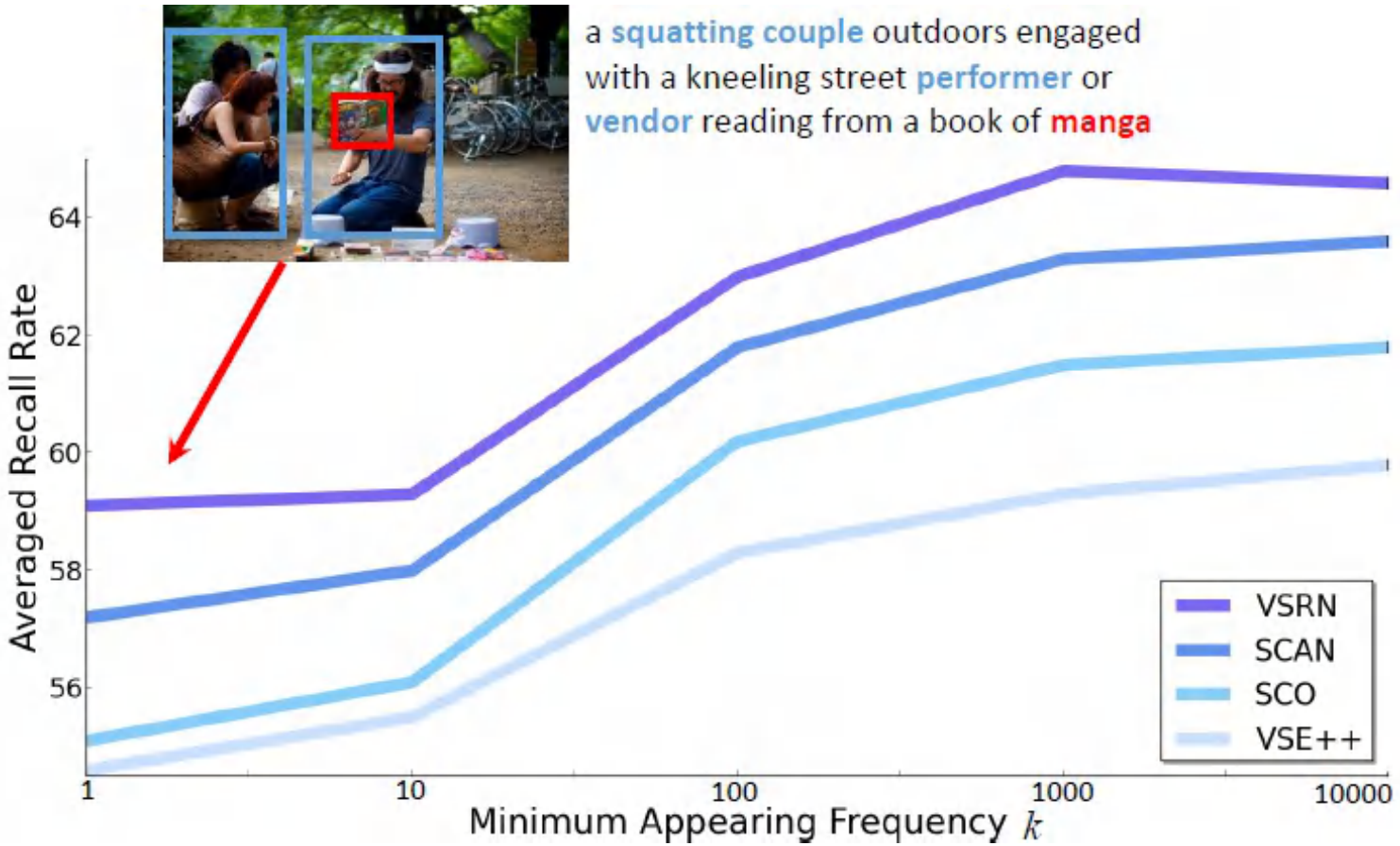
Few-shot Matching (1/2)

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



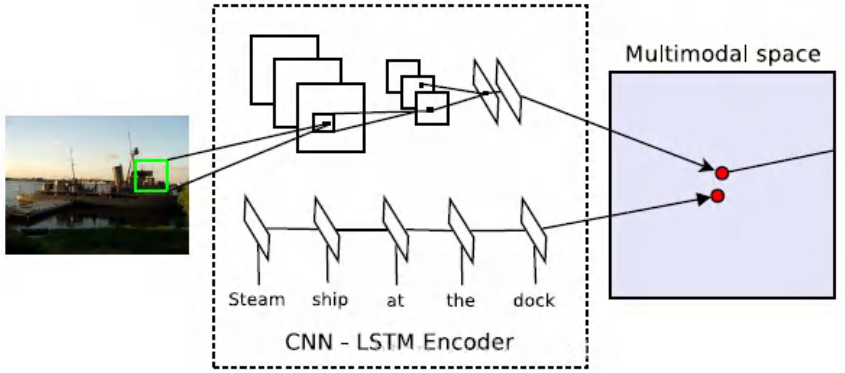
Few-shot Matching (2/2)

- A bottleneck for performance improvement, but commonly ignored

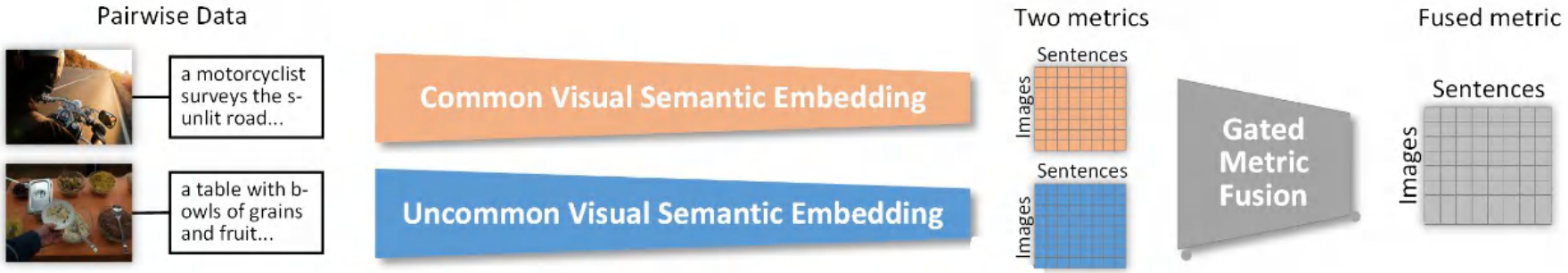


A Straightforward Idea: Two-Stream Framework

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

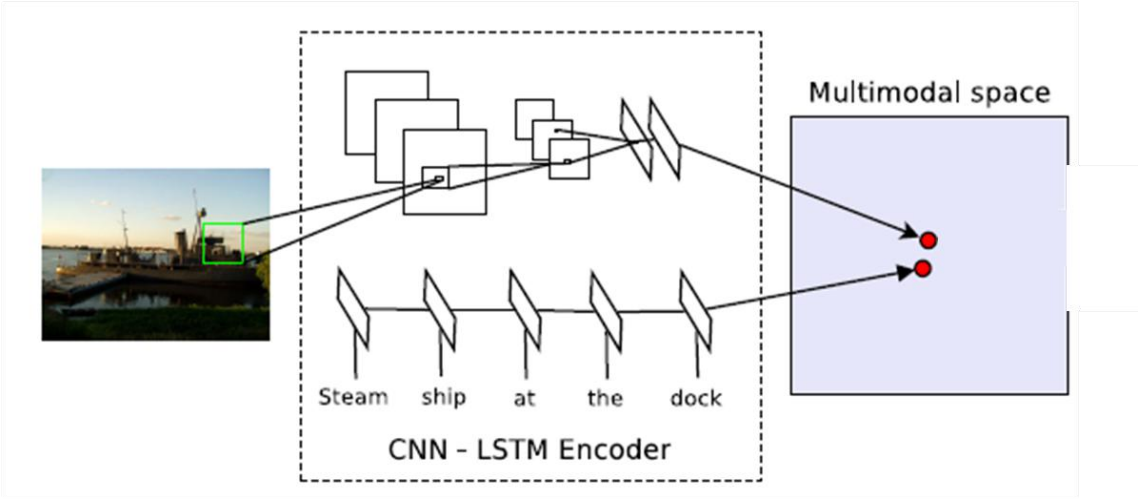


Visual-Semantic Embedding (VSE)



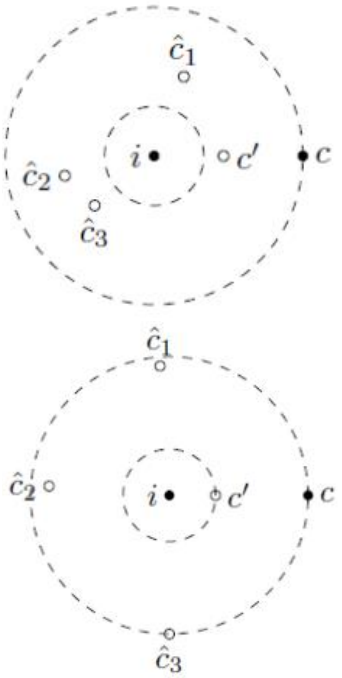
Common VSE

- Image and sentence features are directly learnt from scratch



Visual-Semantic Embedding (VSE)

sum loss

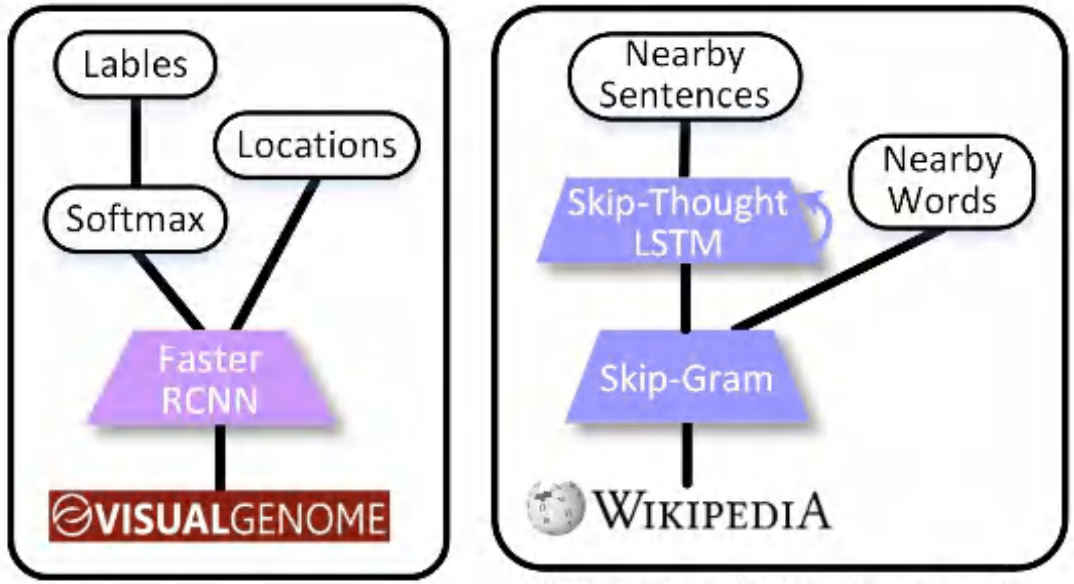


max loss

[Faghri et al., 2018]

Uncommon VSE

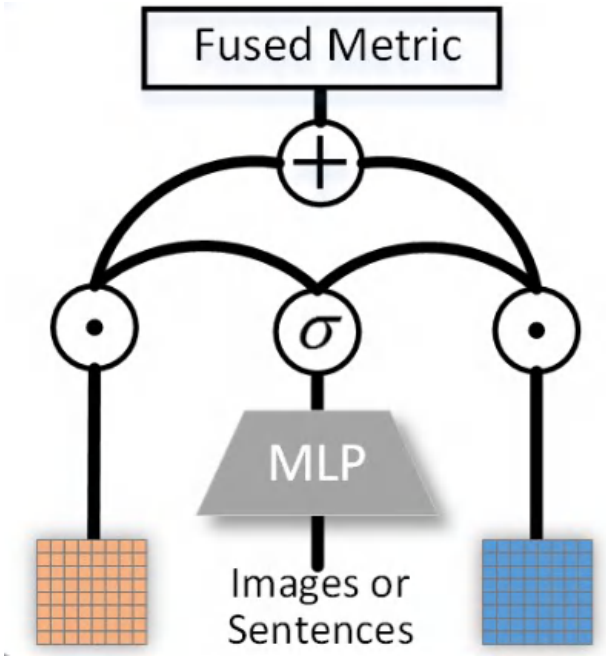
- Use a pretrained Faster RCNN to output region-level features
- Use a pretrained Skip-Gram to obtain word-level features
- Feed them into another VSE to correlate images and sentences



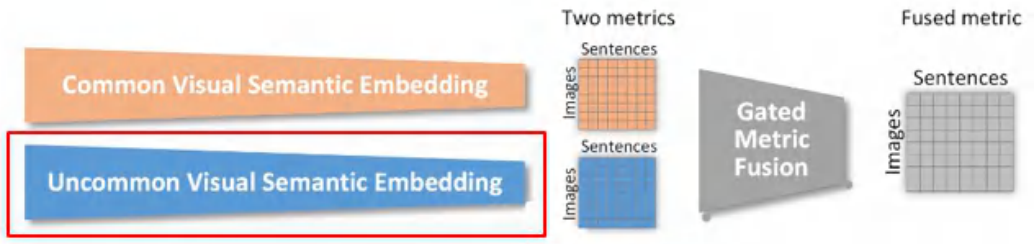
Pretrained Faster-RCNN and Skip-Gram

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



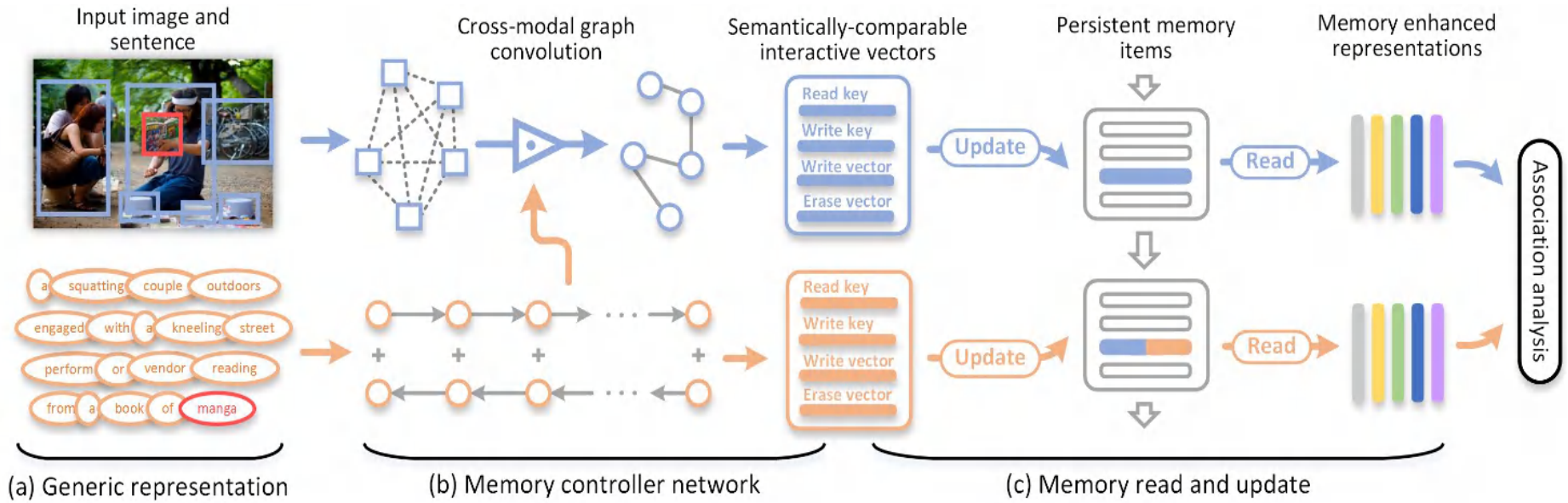
$$t = \sigma(f(\mathbf{x})),\ \hat{\mathbf{m}} = t \odot \mathbf{m}_0 + (1-t) \odot \mathbf{m}_1$$



Gated Metric Fusion

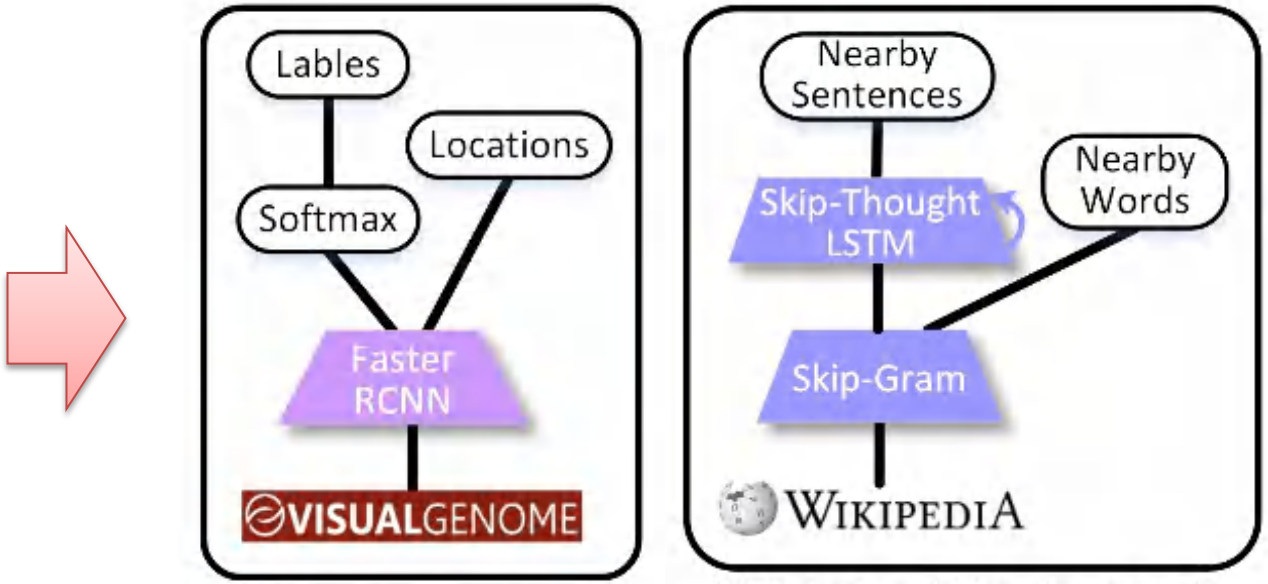
Uncommon VSE >>> Cross-Modal Memory

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



Generic Representation

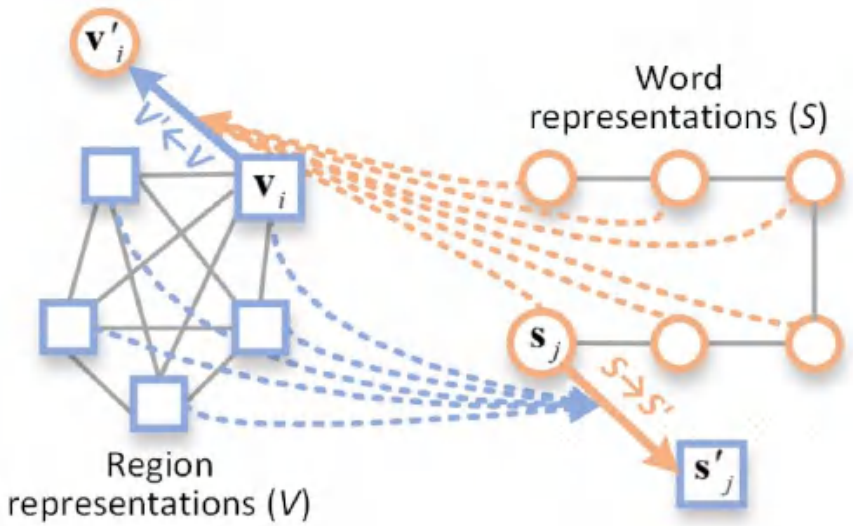
- Use a Faster RCNN pretrained on Visual Genome to output detected instances
- Use a Skip-Gram pretrained on Wikipedia to encode arbitrary words



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



(b) Cross-modal graph convolution

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

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

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

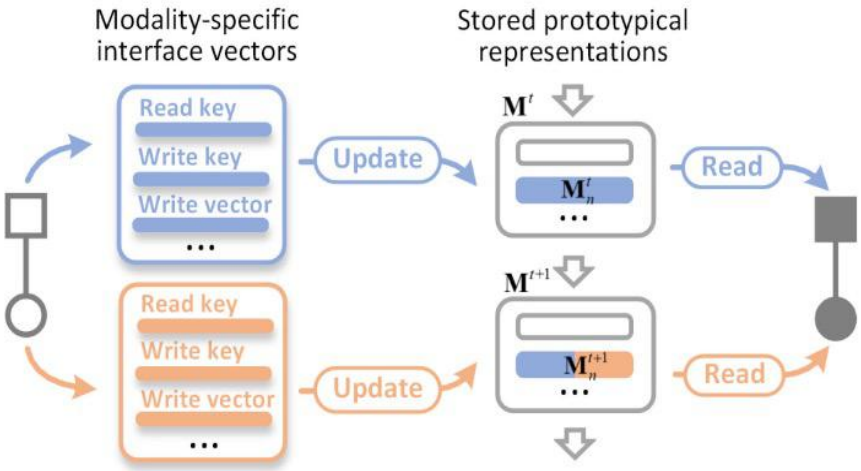
Cross-modal graph convolutional network

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

Interface Vectors

Memory Update and Read

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



Cross-modal shared memory

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

Memory update:

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

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

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. ……

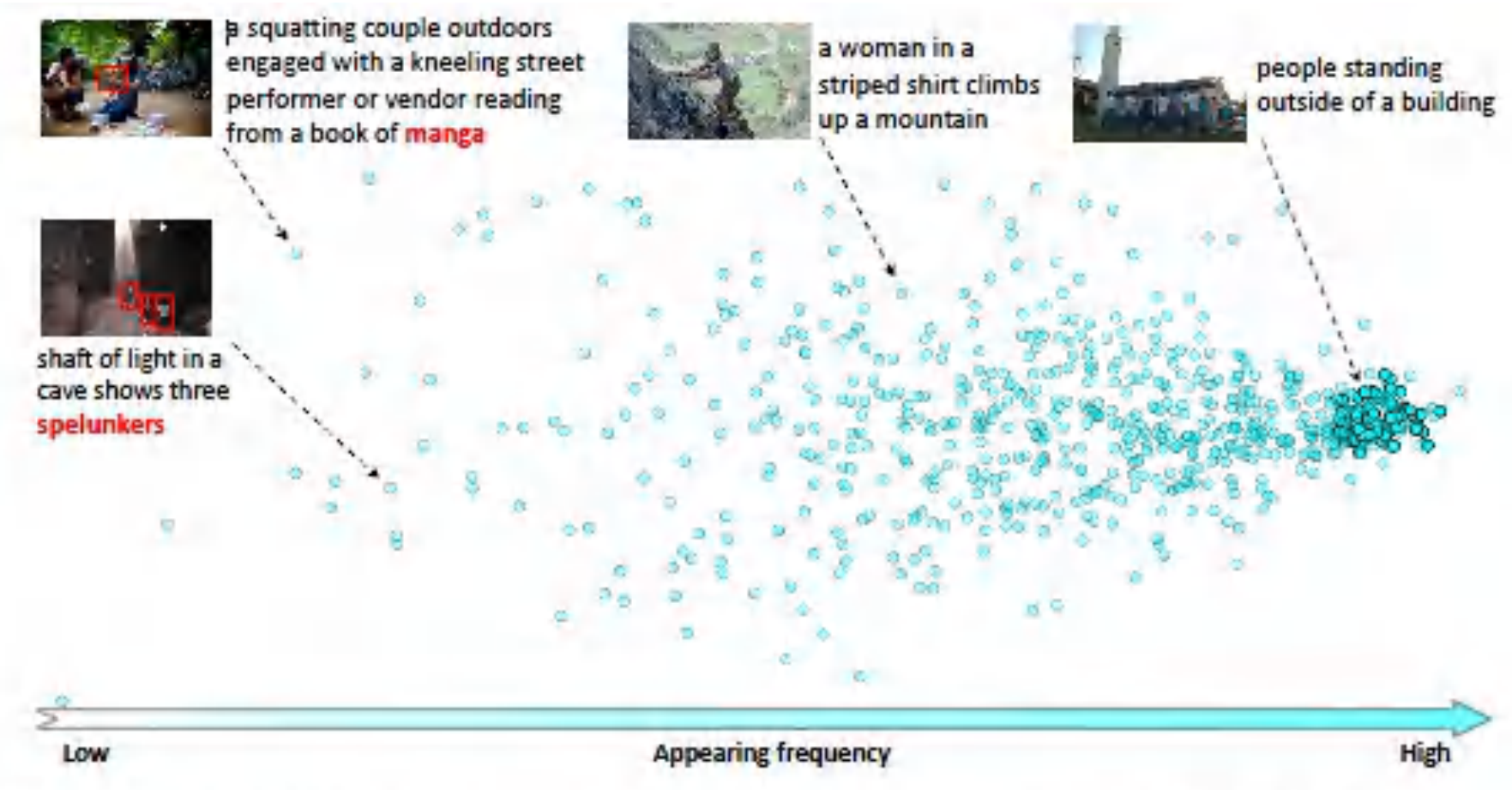
Ablation Study

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						
		Image Annotation			Image Retrieval			mR	Image Annotation			Image Retrieval			mR
		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
VGG	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
VGG	OEM [53]	-	-	-	-	-	-	-	46.7	78.6	88.9	37.9	73.7	85.9	68.6
VGG	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
VGG	RTP [43]	37.4	63.1	74.3	26.0	56.0	69.3	54.3	-	-	-	-	-	-	-
VGG	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
VGG	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
VGG	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
ResNet	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
ResNet	DAN [39]	55.0	81.8	89.0	39.4	69.2	79.1	68.9	-	-	-	-	-	-	-
ResNet	CHAIN [61]	-	-	-	-	-	-	-	59.4	88.0	94.2	43.5	79.8	90.2	75.9
ResNet	UVSE [64]	-	-	-	-	-	-	-	64.3	89.2	94.8	48.3	81.7	91.2	78.3
ResNet	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
ResNet	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
ResNet	PVSE [49]	-	-	-	-	-	-	-	69.2	91.6	96.6	55.2	86.5	93.7	82.1
ResNet	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
ResNet	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
Faster RCNN	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
Faster RCNN	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
Faster RCNN	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
Faster RCNN	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
Faster RCNN	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
Faster RCNN	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

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

a man wearing revolutionary period clothes is ringing a bell

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

an oriental traveler awaits his turn at the currency exchange

battling between the sexes who will win

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

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 !