

# Deep Learning II

Russ Salakhutdinov

Department of Computer Science  
Department of Statistics  
University of Toronto → CMU  
Canadian Institute for Advanced Research



**CIFAR**  
CANADIAN INSTITUTE  
for ADVANCED RESEARCH

# Talk Roadmap

- Zero-Shot Learning
  - Embedding text and images into the joint space
- Caption Generation
  - Multiplicative Neural Language Model
  - Visual Attention
- Learning Recurrent Attention Models
- Learning Skip-Thought Vectors

# Talk Roadmap

- Zero-Shot Learning
  - Embedding text and images into the joint space



Lei Jimmy Ba



Kevin Swersky



Sanja Fidler

Ba, Swersky, Fidler, Salakhutdinov, arXiv 2015

# Can you solve Zero-Shot Problem?

## Description [\[edit\]](#)

---

These birds have yellow underparts, blue-grey upperparts and pink legs; they also have yellow eye-rings and thin, pointed bills. Adult males have black foreheads and black necklaces. Females and immatures have faint grey necklaces. They have yellow “spectacles” round the eyes.

The Canada warbler is the host to the parasite *Apororhynchus amphistomi*.<sup>[2]</sup>





# Can you solve Zero-Shot Problem?

## Description [\[edit\]](#)

---

These birds have yellow underparts, blue-grey upperparts and pink legs; they also have yellow eye-rings and thin, pointed bills. Adult males have black foreheads and black necklaces. Females and immatures have faint grey necklaces. They have yellow “spectacles” round the eyes.

The Canada warbler is the host to the parasite *Apororhynchus amphistomi*.<sup>[2]</sup>



Canada Warbler



Yellow Warbler



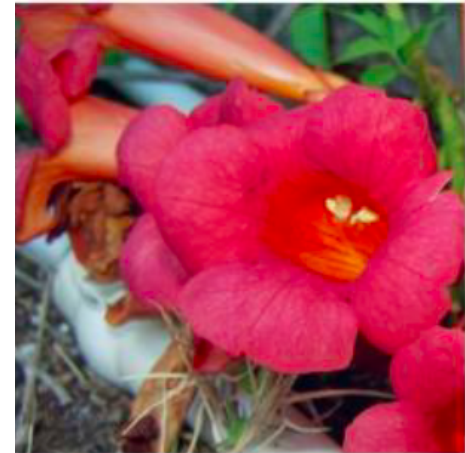
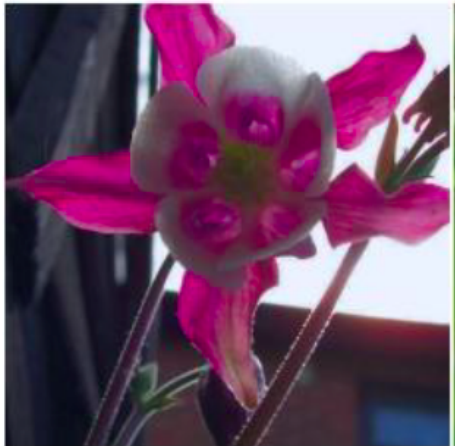
Sharpe-tailed  
Sparrow

# Can you solve Zero-Shot Problem?

## Description [\[edit\]](#)

---

Fritillaries often have nodding, bell- or cup-shaped flowers, and the majority are spring-flowering. Certain species have flowers that emit disagreeable odors. The scent of *Fritillaria imperialis* has been called "rather nasty", while that of *F. agrestis*, known commonly as stink bells, is reminiscent of [dog droppings](#).<sup>[6]</sup> On the other hand, *F. striata* has a sweet fragrance.<sup>[6]</sup>



# The Model

- Consider binary one vs. all classifier for class  $c$ :

$$\hat{y}_c = w_c^\top x$$



Weight vector for a  
particular class  $c$

- How can we deal with previously unseen classes using this standard formulation?

# The Model

- Consider binary one vs. all classifier for class  $c$ :

$$\hat{y}_c = w_c^\top x$$

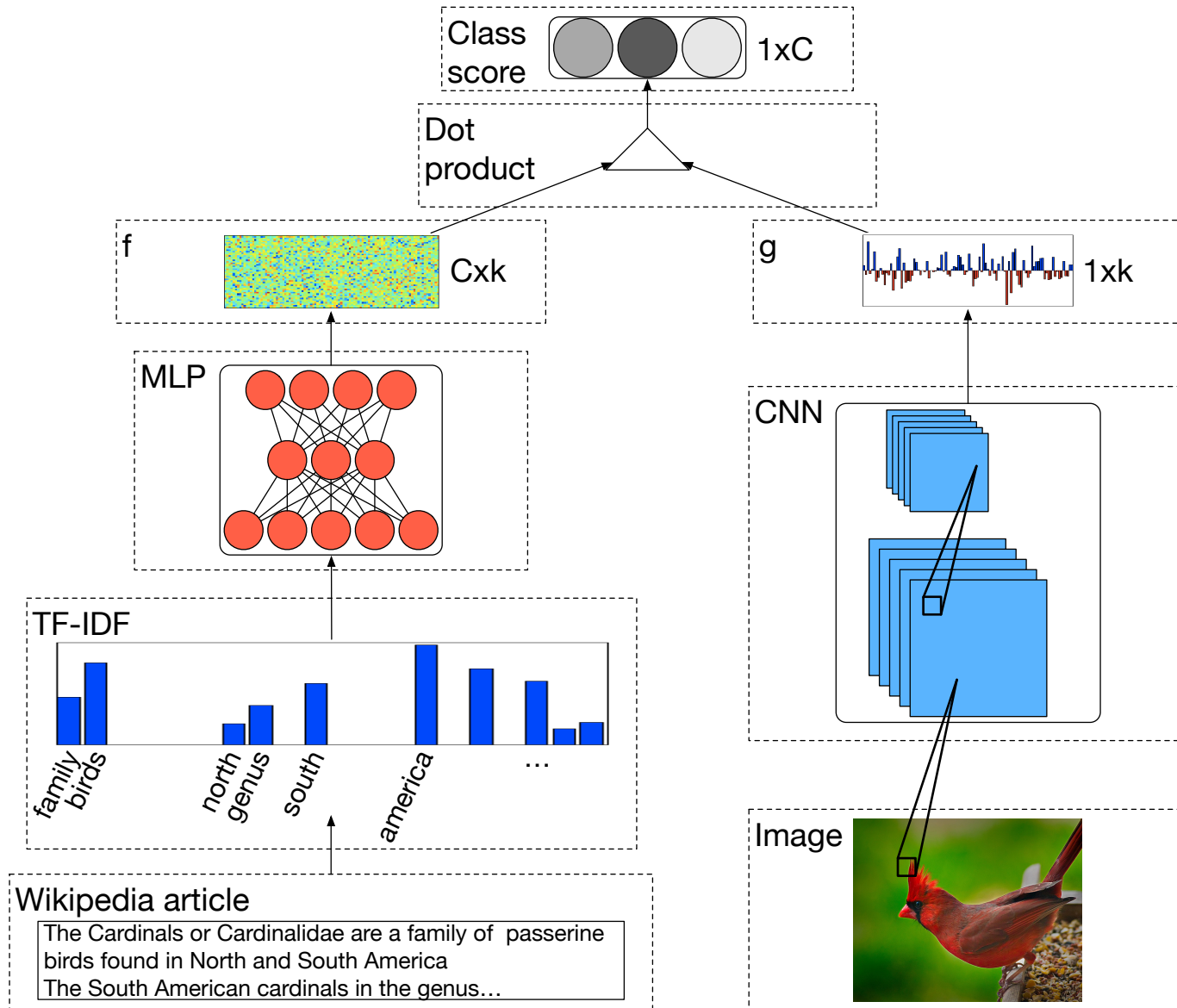
- Assume that we are given an additional text feature  $t_c \in \mathbb{R}^p$ .
- **Simple idea**: Instead of learning a static weight vector  $w_c$ , the text feature can be used to predict the weights:

$$w_c = f_t(t_c),$$

where  $f_t : \mathbb{R}^p \mapsto \mathbb{R}^d$  is a mapping that transforms the text features to the visual image feature space.

- We can use this idea to predict the output weights of a classifier (both the fully connected and convolutional layers of a CNN).

# Model Architecture



# Alternative View

Joint Semantic  
Feature space



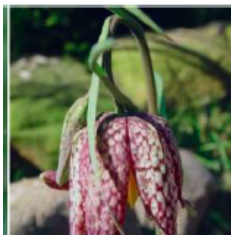
**Fritillary**

Fritillaria is a genus of about 100 species of bulbous plants in the family Liliaceae, native to temperate regions of the Northern Hemisphere. The name is derived from the Latin term for a dice-box (fritillus), and probably refers to the checkered pattern, frequently of chocolate-brown and greenish yellow, that is common to many species' flowers. Collectively, the genus is known in English as fritillaries; some North American species are called mission bells.

They often have nodding, bell- or cup-shaped flowers, and the majority are spring-flowering. Most species' flowers have a rather disagreeable scent, often referred to as "fory," like fomes or wet fur. The Scarlet Lily Beetle (Lilicoccis lili) eats fritillaries, and may become a pest where these plants are grown in gardens.

Several species (such as *F. cirrhosa* and *F. verticillata*) are used in traditional Chinese cough remedies. They are listed as chuan bei (Chinese: 川贝) or zhe bei (Chinese: 浙贝), respectively, and are often in formulations combined with extracts of Luqian (Eriobotrya japonica). *F. verticillata* bulbs are also traded as bei mu or, in Kambo, balmo (Chinese/Kanji: 贝母, Kabakana: ???). *F. thunbergii* is contained in the standardized Chinese herbal preparation HealthGuard T15, taken against hyperthyroidism.

.....



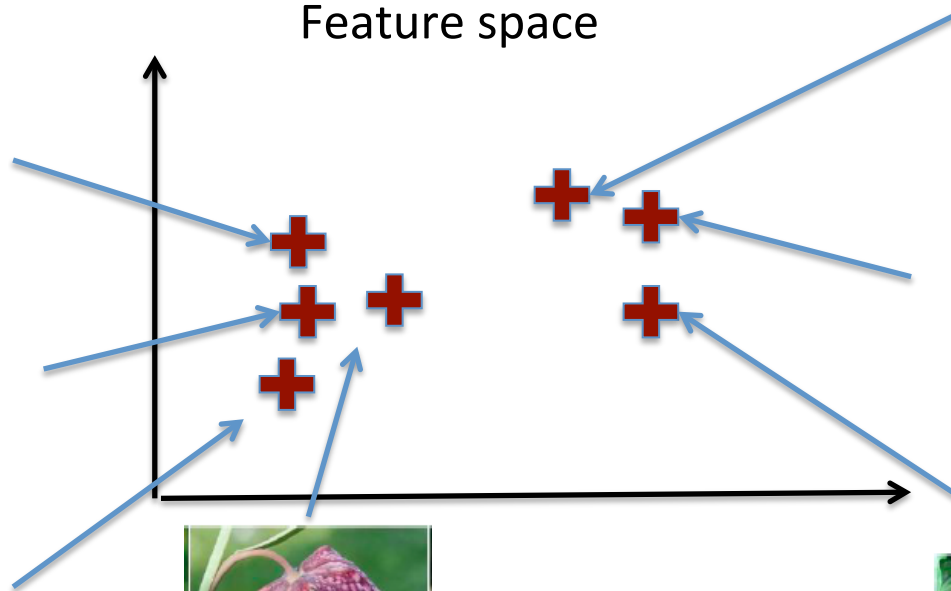
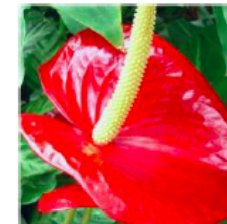
**Corn Poppy**

*Papaver rhoeas* (common names include corn poppy, corn rose, field poppy, Flanders poppy, red poppy, red weed, coquelicot, and, due to its odour, which is said to cause them, as headache and headwork) is a species of flowering plant in the poppy family, Papaveraceae. This poppy, a native of Europe, is notable as an agricultural weed (hence the "corn" and "field") and as a symbol of fallen soldiers.

*P. rhoeas* sometimes is so abundant in agricultural fields that it may be mistaken for a crop. The only species of Papaveraceae grown as a field crop on a large scale is *Papaver somniferum*, the opium poppy.

The plant is a variable annual, forming a long-lived soil seed bank that can germinate when the soil is disturbed. In the northern hemisphere it generally flowers in late spring, but if the weather is warm enough other flowers frequently appear at the beginning of autumn. The flower is large and showy, with four petals that are vivid red, most commonly with a black spot at their base. Like many other species of Papaver, it exudes a white latex when the tissues are broken.

.....



- Minimize the cross-entropy or hinge-loss objective.



# Problem Setup

- The training set contains  $N$  images  $x \in R^d$  and their associated class labels  $l \in \{1, \dots, C\}$ . There are  $C$  distinct class labels.
- During test time, we are given additional  $n_0$  number of the previously unseen classes, such that  $l_{test} \in \{1, \dots, C, \dots, C+n_0\}$ .
- Our goal is to predict previously unseen classes and perform well on the previously seen classes.
- **Interpretation:** Learning a good similarity kernel between images and encyclopedia articles .



# Caltech UCSD Bird and Oxford Flower Datasets

- The CUB200-2010 contains 6,033 images from 200 bird species (about 30 images per class).
  - There is one Wikipedia article associated with each bird class (200 articles in total). The average number of words per articles is 400.
  - Out of 200 classes, 40 classes are defined as unseen and the remaining 160 classes are defined as seen.
- 

- The Oxford Flower-102 dataset contains 102 classes with a total of 8189 images.
- Each class contains 40 to 260 images. 82 flower classes are used for training and 20 classes are used as unseen during testing.

# Results: Area Under ROC

The CUB200-2010 Dataset

Learning Algorithm	Unseen	Seen	Mean
DA [Elhoseiny, et.al., 2013]	0.59	-	-
DA (VGG features)	0.62	-	-
Ours (fc)	<b>0.82</b>	0.96	0.934
Ours (conv)	0.73	0.96	0.91
Ours (fc+conv)	0.80	<b>0.987</b>	<b>0.95</b>

# Results: Area Under ROC

The Oxford Flower Dataset

Learning Algorithm	Unseen	Seen	Mean
DA [Elhoseiny, et.al.]	0.62	-	-
GPR+DA [Elhoseiny, et.al.]	0.68	-	-
Ours (fc)	0.70	0.987	0.90
Ours (conv)	0.65	0.97	0.85
Ours (fc+conv)	<b>0.71</b>	<b>0.989</b>	<b>0.93</b>

# Attribute Discovery

## Scarlet Tanager

The Scarlet Tanager (*Piranga olivacea*) is a medium-sized American songbird. Formerly placed in the tanager family (Thraupidae), it and other members of its genus are now classified in the cardinal family (Cardinalidae). The species's plumage and vocalizations are similar to other members of the cardinal family.

Adults have pale stout smooth bills. Adult males are bright red with black wings and tail; females are yellowish on the underparts and olive on top, with olive-brown wings and tail. The adult male's winter plumage is similar to the female's, but the wings and tail remain darker. Young males briefly show a more complex variegated plumage intermediate between adult males and females. It apparently was such a specimen that was first scientifically described. {Citation needed|date=July 2010} Hence the older though somewhat confusing specific epithet *olivacea* ("the olive-colored one") is used rather than *erythromelas* ("the red-and-black one"), as had been common throughout the 19th century.

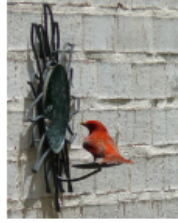
## Word sensitivities of unseen classes

- Tanagers
- Thraupidae
- Scarlet
- Cardinalidae
- Tanagers

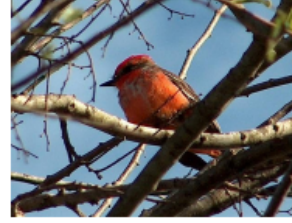
## Nearest Neighbors



Scarlet Tanager



Summer Tanager



Vermillion Flycatcher



Brown Thrasher

- Wikipedia article for each class is projected onto its feature space and the nearest image neighbors from the test-set are shown.

# Attribute Discovery

## Bearded Iris

Irises are **perennial plants**, growing from creeping **rhizomes** (rhizomatous irises) or, in drier climates, from **bulbs** (bulbous irises). They have long, erect flowering **stems** which may be simple or branched, solid or hollow, and flattened or have a circular cross-section. The rhizomatous species usually have 3–10 basal sword-shaped leaves growing in dense clumps. The bulbous species have cylindrical, basal leaves.

## Word sensitivities of unseen classes

- rhizome
- freezing
- iris
- depth
- compost

## Nearest Neighbors



Bearded Iris



Corn Poppy



Grape Hyacinth



Giant White  
Arum Lily

- Wikipedia article for each class is projected onto its feature space and the nearest image neighbors from the test-set are shown.

# Talk Roadmap

- Zero-Shot Learning
- Caption Generation



Ryan Kiros



Rich Zemel

Kiros, Salakhutdinov, Zemel,  
ICML 2014, TACL 2015

- Learning Recurrent Attention Models
- Learning Skip-Thought Vectors

# Generating Sentences

- More challenging problem.
- How can we generate complete descriptions of images?

Input

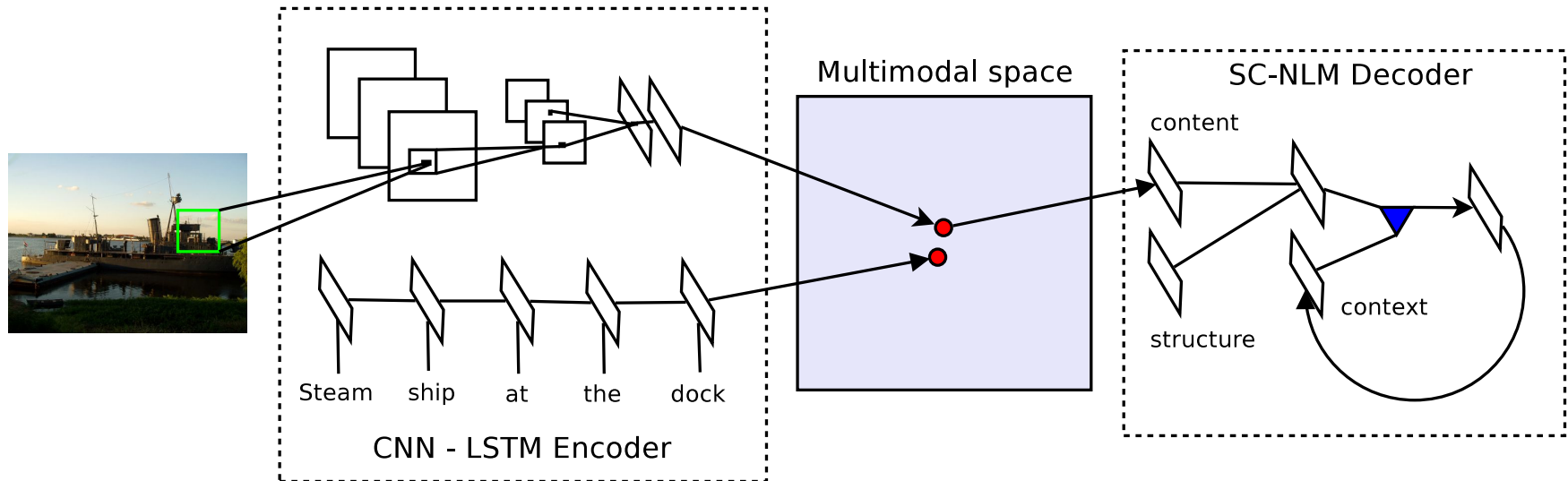


Output

A man skiing down the snow covered mountain with a dark sky in the background.



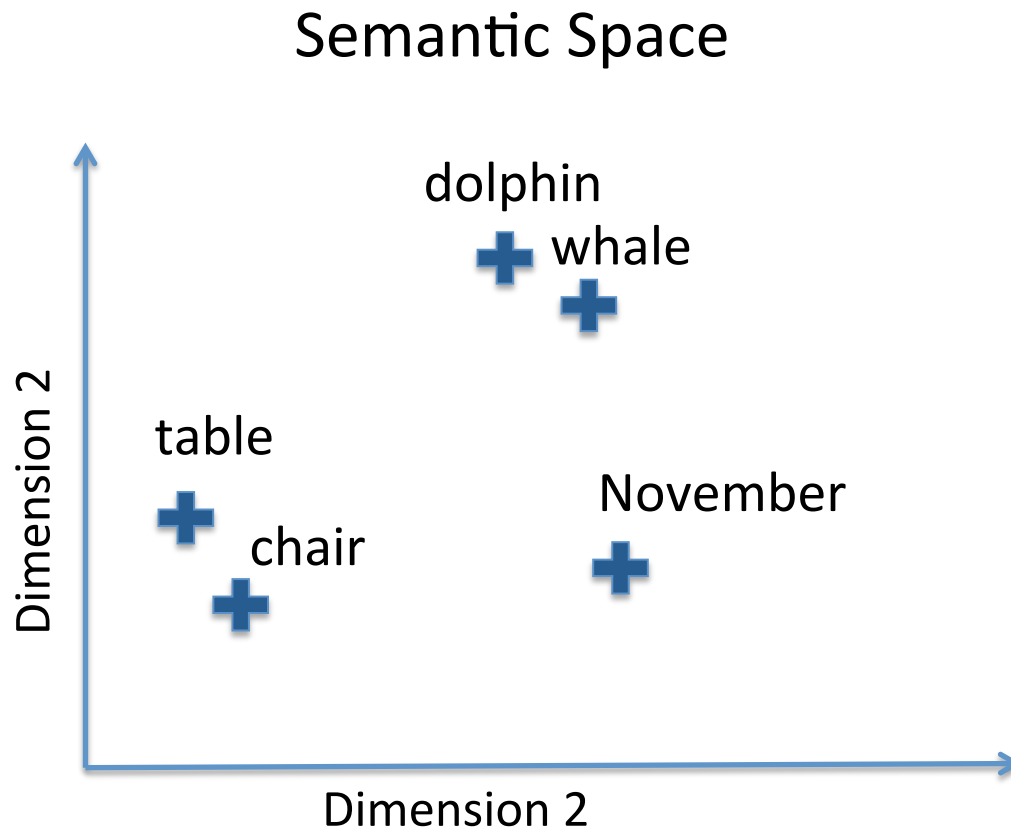
# Encode-Decode Framework



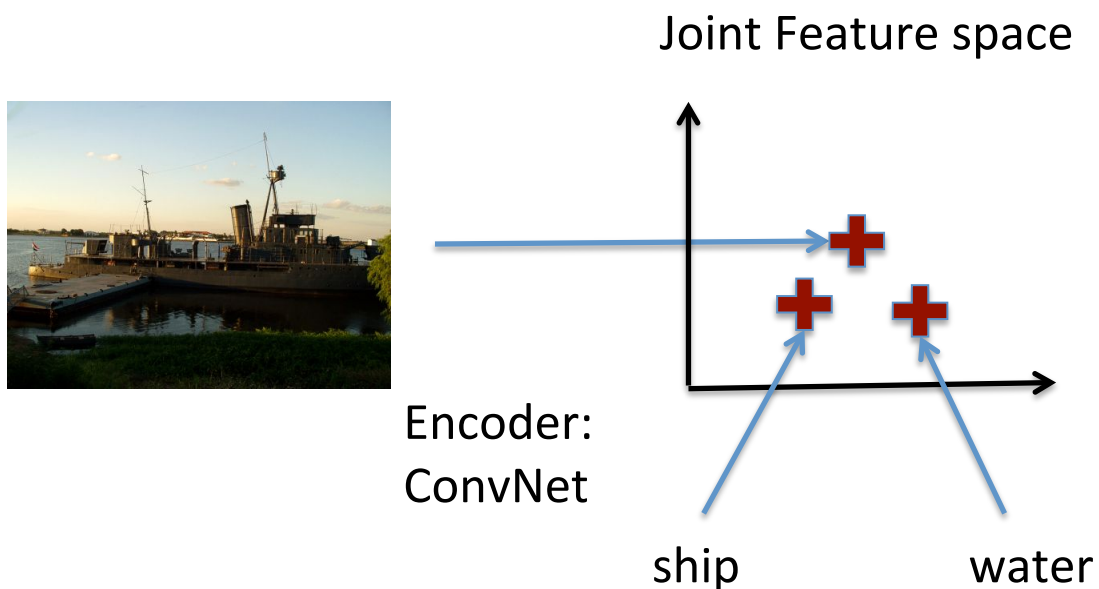
- Encoder: CNN and Recurrent Neural Net for a joint image-sentence embedding.
- Decoder: A neural language model that combines structure and content vectors for generating a sequence of words

# Representation of Words

- **Key Idea:** Each word  $w$  is represented as a  $D$ -dimensional real-valued vector  $r_w \in \mathbb{R}^D$ .

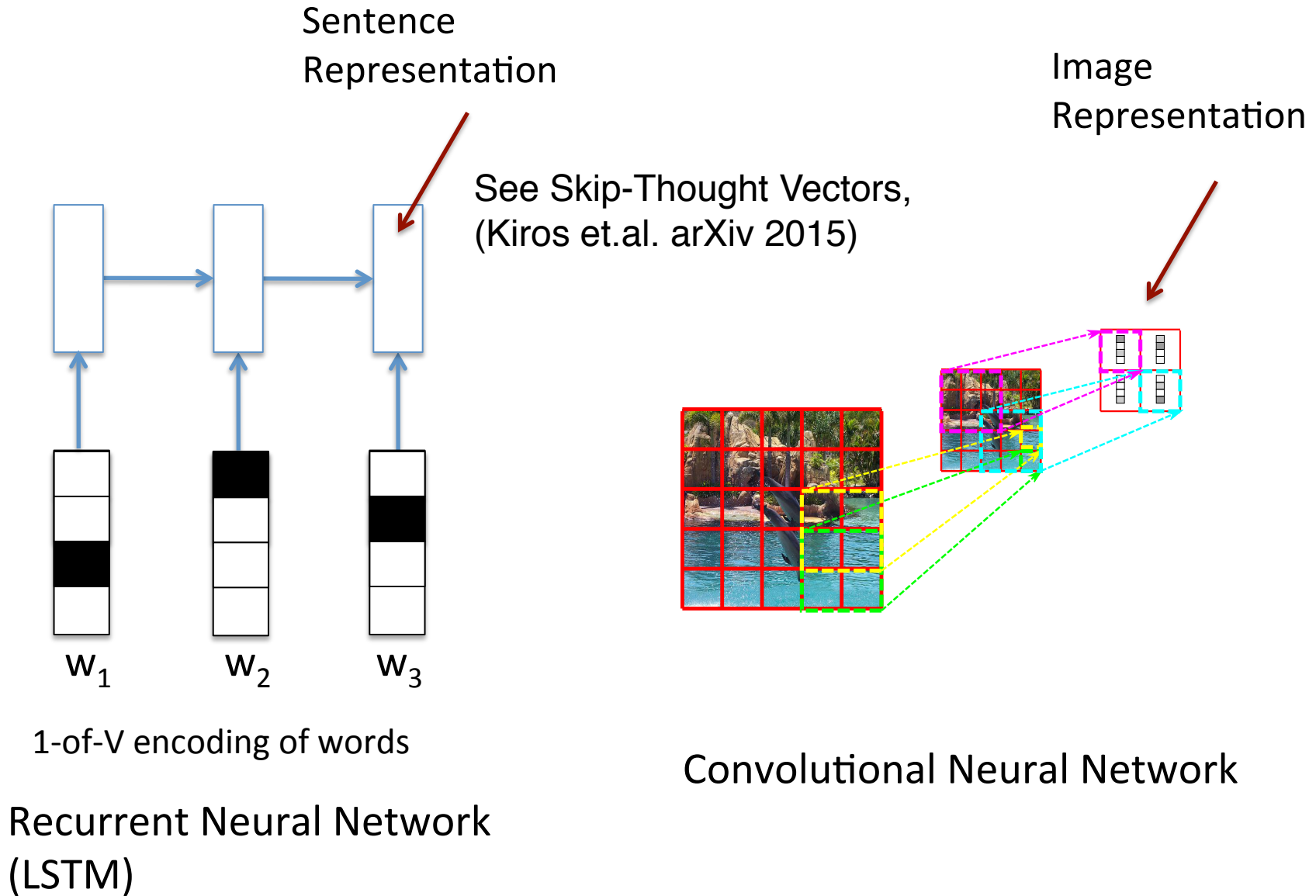


# An Image-Text Encoder

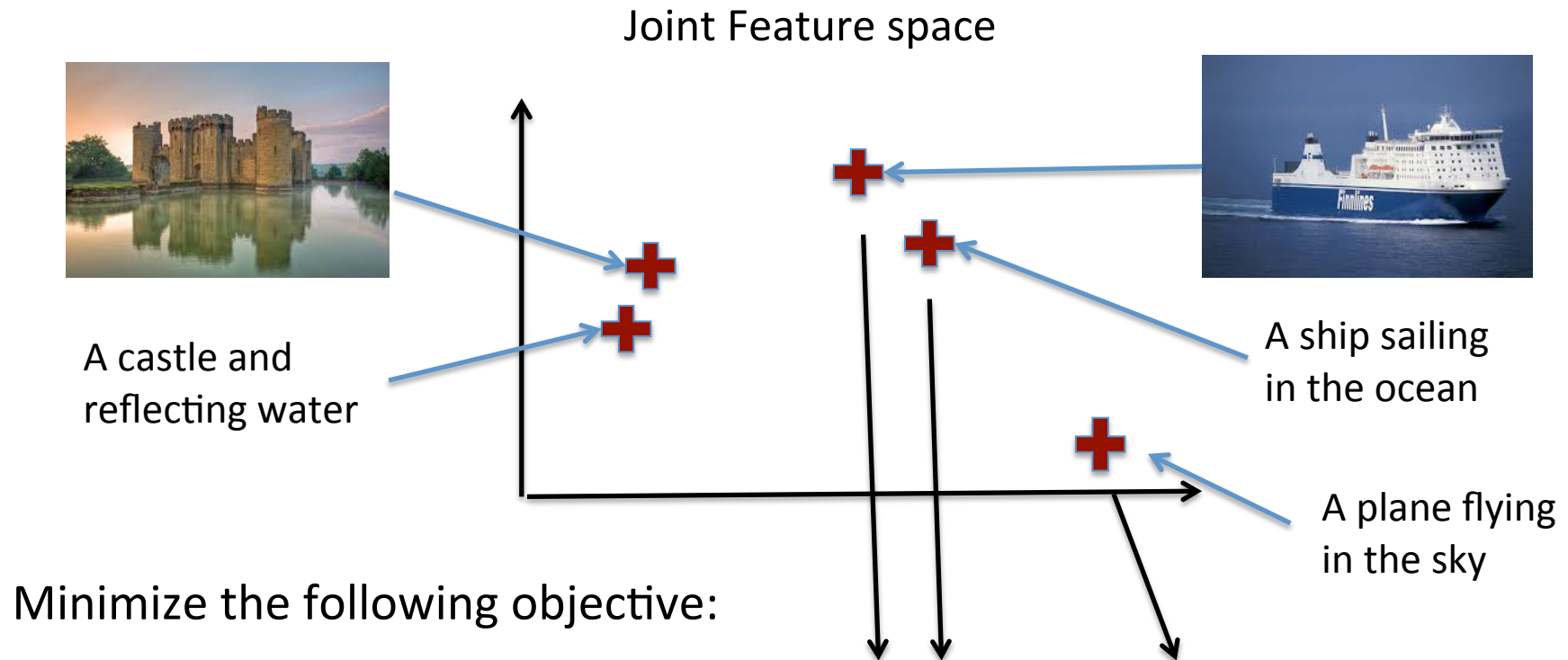


- Learn a joint embedding space of images and text:
  - Can condition on anything (images, words, phrases, etc)
  - Natural definition of a scoring function (inner products in the joint space).

# An Image-Text Encoder



# An Image-Text Encoder



Images: 
$$\sum_{\mathbf{x}} \sum_k \max\{0, \alpha - s(\mathbf{x}, \mathbf{v}) + s(\mathbf{x}, \mathbf{v}_k)\} +$$

Text: 
$$\sum_{\mathbf{v}} \sum_k \max\{0, \alpha - s(\mathbf{v}, \mathbf{x}) + s(\mathbf{v}, \mathbf{x}_k)\}$$

# Retrieving Sentences for Images



The dogs are in the snow in front of a fence .



Four men playing basketball , two from each team .



A boy skateboarding



Two men and a woman smile at the camera .



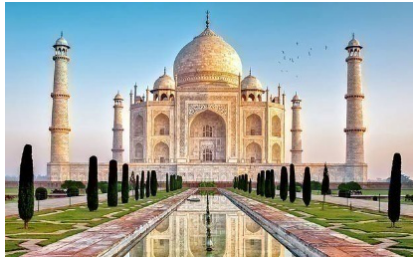
Women participate in a skit onstage .



A man is doing tricks on a bicycle on ramps in front of a crowd .



# Tagging and Retrieval



mosque, tower,  
building, cathedral,  
dome, castle



ski, skiing,  
skiers, skiers,  
snowmobile

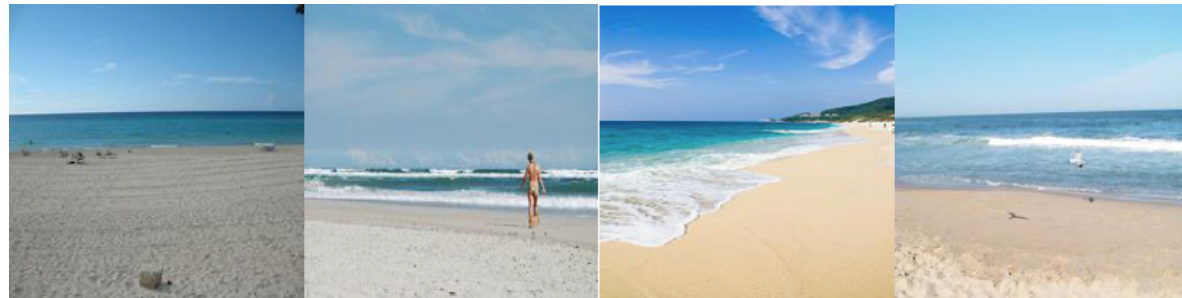


kitchen, stove, oven,  
refrigerator,  
microwave



bowl, cup,  
soup, cups,  
coffee

beach



snow





# Retrieval with Adjectives

fluffy



delicious



# Multimodal Linguistic Regularities

## Nearest Images



- blue + red =



- blue + yellow =



- yellow + red =





# Multimodal Linguistic Regularities

## Nearest Images



- day + night =



- flying + sailing =



- bowl + box =



- box + bowl =



(Kiros, Salakhutdinov, Zemel, TACL 2015)

# How About Generating Sentences!

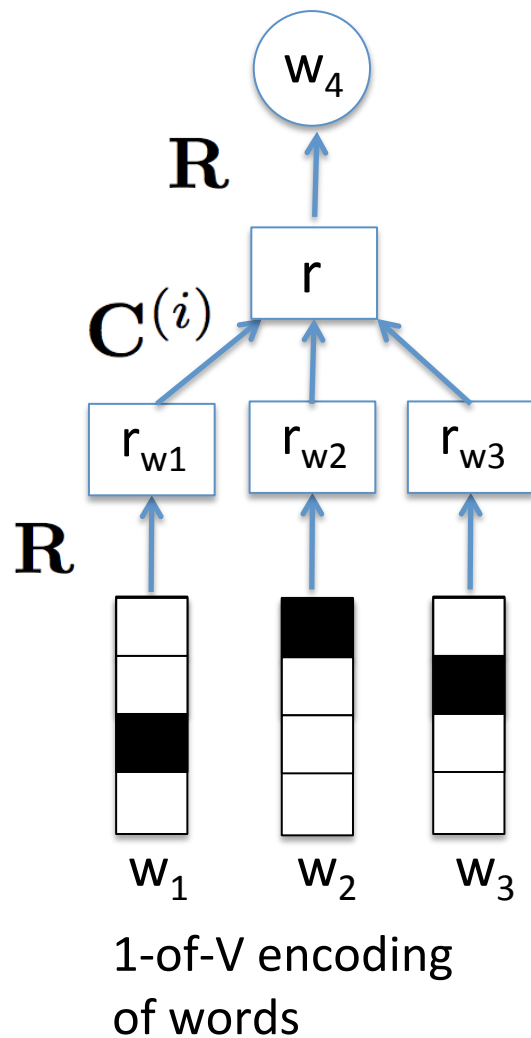
Input



Output

A man skiing down the snow covered mountain with a dark sky in the background.

# Log-bilinear Neural Language Model

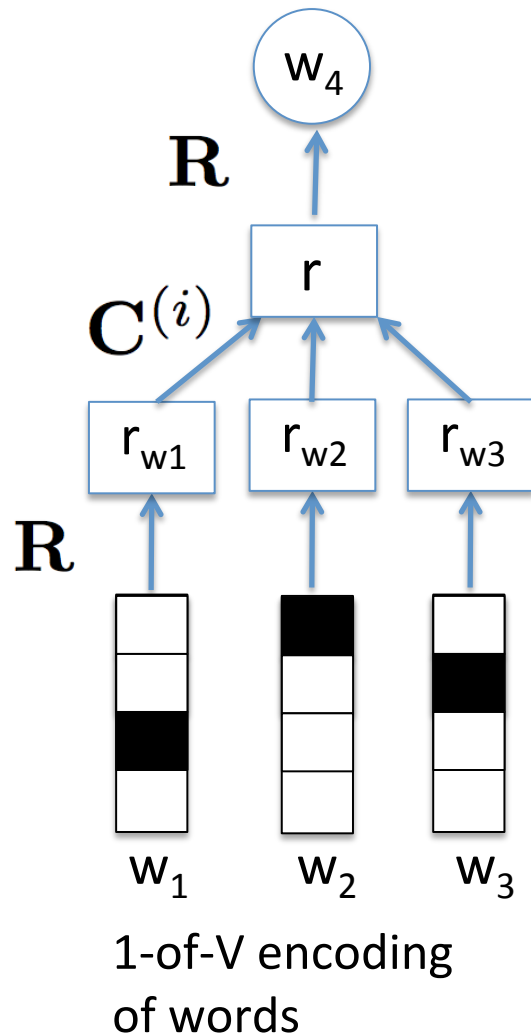


- Feedforward neural network with a single linear hidden layer.
- Each word  $w$  is represented as a  $K$ -dim real-valued vector  $\mathbf{r}_w \in \mathbb{R}^K$ .
- $\mathbf{R}$  denote the  $V \times K$  matrix of word representation vectors, where  $V$  is the vocabulary size.
- $(w_1, \dots, w_{n-1})$  is tuple of  $n-1$  words, where  $n-1$  is the context size. The next word representation becomes:

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \underbrace{\mathbf{C}^{(i)}}_{K \times K \text{ context parameter matrices}} \mathbf{r}_{w_i},$$

$K \times K$  context parameter matrices

# Log-bilinear Neural Language Model



$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{r}_{w_i},$$

Predicted representation of  $r_{w_n}$ .

- The conditional probability of the next word given by:

$$P(w_n = i | w_{1:n-1}) = \frac{\exp(\hat{\mathbf{r}}^T \mathbf{r}_i + b_i)}{\sum_{j=1}^V \exp(\hat{\mathbf{r}}^T \mathbf{r}_j + b_j)}$$

Can be expensive to compute

# Multiplicative Model

- We represent words as a tensor:

$$\mathcal{T} \in \mathbb{R}^{V \times K \times G}$$

where  $G$  is the number of tensor slices.

- Given an attribute vector  $\mathbf{u} \in \mathbb{R}^G$  (e.g. image features), we can compute attribute-gated word representations as:

$$\mathcal{T}^u = \sum_{i=1}^G u_i \mathcal{T}^{(i)}$$

- Re-represent Tensor in terms of 3 lower-rank matrices (where  $F$  is the number of pre-chosen factors):

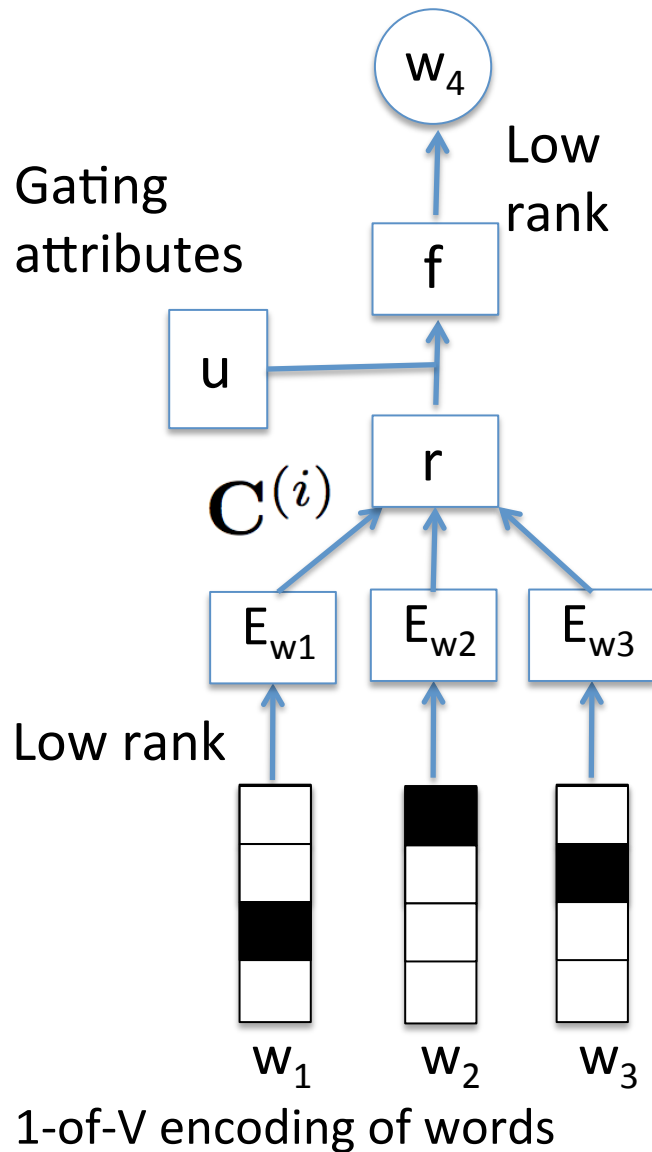
$$\mathbf{W}^{fk} \in \mathbb{R}^{F \times K}, \mathbf{W}^{fd} \in \mathbb{R}^{F \times G}, \mathbf{W}^{fv} \in \mathbb{R}^{F \times V}$$

$$\mathcal{T}^u = (\mathbf{W}^{fv})^\top \cdot \text{diag}(\mathbf{W}^{fd} \mathbf{u}) \cdot \mathbf{W}^{fk}$$

(Kiros, Zemel, Salakhutdinov, NIPS 2014)



# Multiplicative Log-bilinear Model



- Let  $\mathbf{E} = (\mathbf{W}^{fk})^\top \mathbf{W}^{fv}$  denote a **folded**  $K \times V$  matrix of word embeddings.

- Then the predicted next word representation is:

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{E}(:, w_i)$$

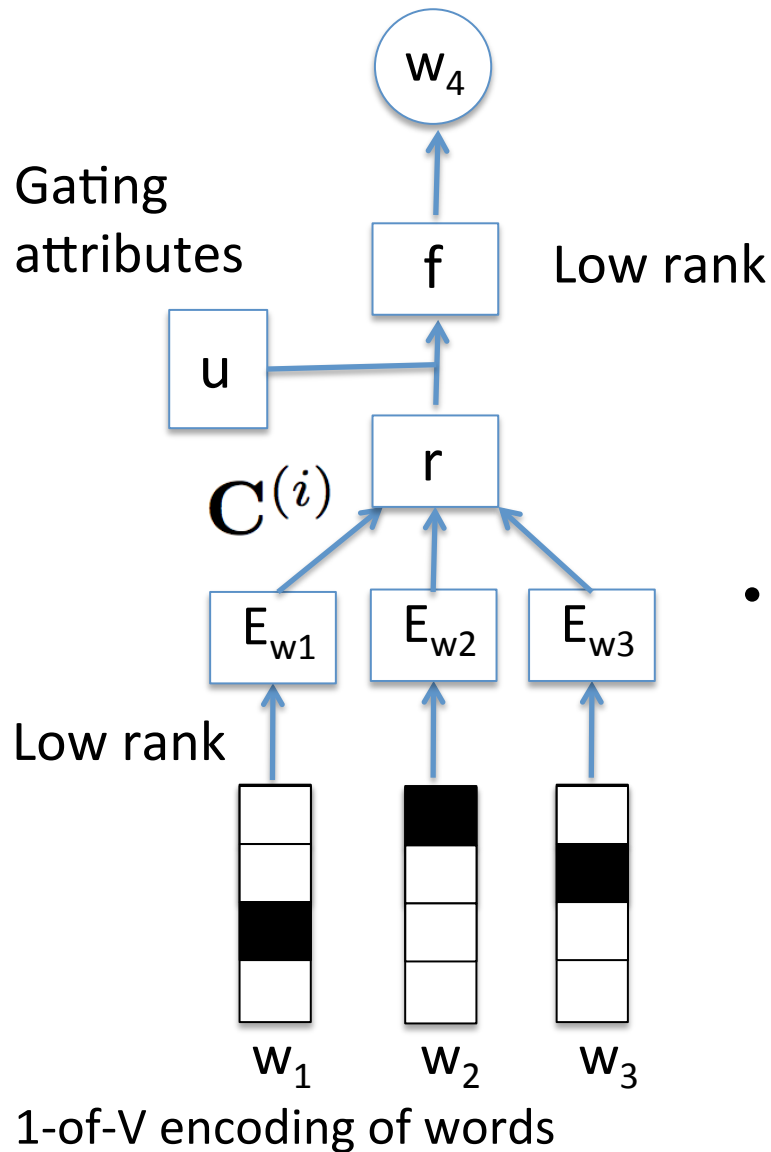
- Given next word representation  $\mathbf{r}$ , the factor outputs are:

$$\mathbf{f} = (\mathbf{W}^{fk} \hat{\mathbf{r}}) \bullet (\mathbf{W}^{fd} \mathbf{x})$$

Component-wise product

(Kiros, Zemel, Salakhutdinov, NIPS 2014)

# Multiplicative Log-bilinear Model



$$\mathbf{E} = (\mathbf{W}^{fk})^\top \mathbf{W}^{fv}$$

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{E}(:, w_i)$$

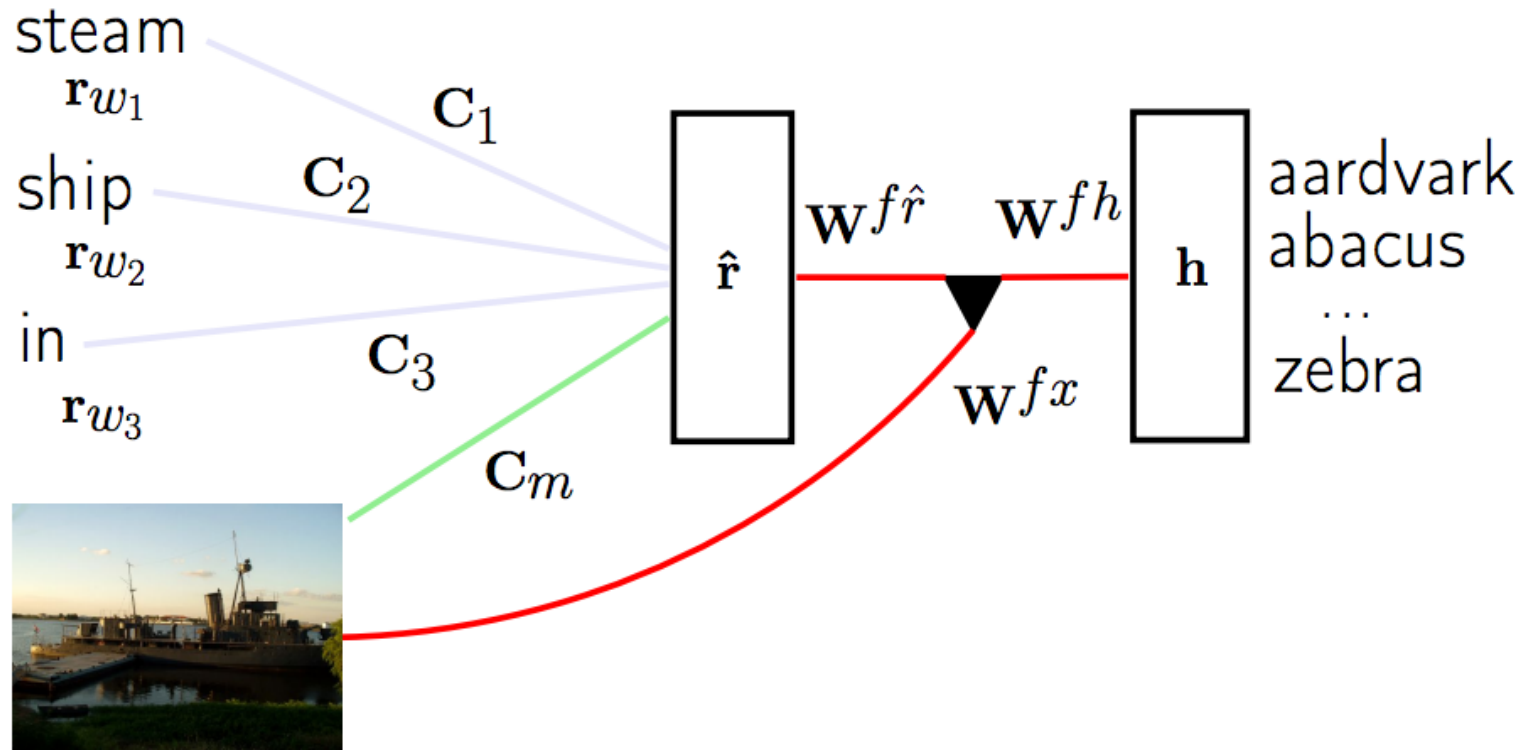
$$\mathbf{f} = (\mathbf{W}^{fk} \hat{\mathbf{r}}) \bullet (\mathbf{W}^{fd} \mathbf{x})$$

- The conditional probability of the next word given by:

$$P(w_n = i | w_{1:n-1}, \mathbf{u}) =$$

$$\frac{\exp((\mathbf{W}^{fv}(:, i))^\top \mathbf{f} + b_i)}{\sum_{j=1}^V \exp((\mathbf{W}^{fv}(:, j))^\top \mathbf{f} + b_j)}$$

# Decoding: Neural Language Model



- Image features are gating the hidden-to-output connections when predicting the next word!
- We can also condition on POS tags when generating a sentence.

# Decoding: Structured NLM



\_\_\_\_\_ (NN VBN IN DT NN)  
DT

$$P(w_n | w_{1:n-1}, t_{n:n+k}, \mathbf{x})$$

n-th word

word context

POS context

# Decoding: Structured NLM



\_\_\_\_\_ (NN VBN IN DT NN)  
DT

A \_\_\_\_\_ (VBN IN DT NN -)  
NN

$$P(w_n | w_{1:n-1}, t_{n:n+k}, \mathbf{x})$$

n-th word

word context

POS context

# Decoding: Structured NLM



\_\_\_\_\_ (NN VBN IN DT NN)  
DT

A \_\_\_\_\_ (VBN IN DT NN -)  
NN

A bicycle \_\_\_\_\_ (IN DT NN - -)  
VBN

$$P(w_n | w_{1:n-1}, t_{n:n+k}, \mathbf{x})$$

n-th word

word context

POS context



# Decoding: Structured NLM



\_\_\_\_\_ (NN VBN IN DT NN)  
DT

A \_\_\_\_\_ (VBN IN DT NN -)  
NN

A bicycle \_\_\_\_\_ (IN DT NN - -)  
VBN

A bicycle parked \_\_\_\_\_ (DT NN - - -)  
IN

$$P(w_n | w_{1:n-1}, t_{n:n+k}, \mathbf{x})$$

n-th word

word context

POS context

# Decoding: Structured NLM



$$P(w_n | w_{1:n-1}, t_{n:n+k}, \mathbf{x})$$

↑                      ↑                      ↑  
 n-th word    word context    POS context

\_\_\_\_\_ (NN VBN IN DT NN)  
DT

A \_\_\_\_\_ (VBN IN DT NN -)  
NN

A bicycle \_\_\_\_\_ (IN DT NN - -)  
VBN

A bicycle parked \_\_\_\_\_ (DT NN - - -)  
IN

A bicycle parked on \_\_\_\_\_ (NN - - - -)  
DT

A bicycle parked on the \_\_\_\_\_ (- - - - -)  
NN

# Caption Generation



LZ  
a car is parked in  
the middle of nowhere .



a wooden table and chairs  
arranged in a room .



there is a cat sitting on a shelf .



a ferry boat on a marina  
with a group of people .



a little boy with a bunch  
of friends on the street .

# Caption Generation



the two birds are trying  
to be seen in the water .  
(can't count)



a giraffe is standing next  
to a fence in a field .  
(hallucination)



a parked car while  
driving down the road .  
(contradiction)



# Caption Generation



the two birds are trying  
to be seen in the water .  
(can't count)



a giraffe is standing next  
to a fence in a field .  
(hallucination)



a parked car while  
driving down the road .  
(contradiction)



the handlebars are trying  
to ride a bike rack .  
(nonsensical)



a woman and a bottle of wine  
in a garden . (gender)

# Filling in the Blanks



The (cat) is in the box.  
NN

The cat is in the (box) .  
NN

The cat is (sitting) in the box .  
VBG

The (cute) cat is in the box .  
JJ



# Filling in the Blanks



The (cat) is in the box.  
NN

The cat is in the (box).  
NN

The cat is (sitting) in the box.  
VBG

The (cute) cat is in the box.  
JJ



This is a (bus).  
NN

The bus is (parked).  
JJ

There is a (car) behind the bus.  
NN

The tree is (on) the bus.  
IN

# Caption Generation



TAGS:

colleagues waiters waiter  
entrepreneurs busboy

Model Samples

- Two men in a room talking on a table .
- Two men are sitting next to each other .
- Two men are having a conversation at a table .
- Two men sitting at a desk next to each other .

# Caption Generation



TAGS:

speaker techie typist laptop  
computers

Top-5 Model Samples

- A man working on his computer .
- A man is sitting down with a laptop typing on a computer .
- A man wearing a blue shirt is using a laptop .
- A man in a blue shirt is working on a laptop .
- A man wearing headphones is working on something on the desk .

# Caption Generation



TAGS:

speakers panelist presenters  
newscasters audience

Top-5 Model Samples

- Two people are sitting and looking at their laptops at a cafe .
- A man and a woman are sitting at a desk with their feet .
- Two people sitting at a desk with their laptops .
- Several young people sitting in front of a laptop .
- A couple of people are sitting in front of tables .

# Caption Generation



TAGS:

balcony woman skylit pinny  
flowerpots

Top-5 Model Samples

- A woman stands in the middle of a small art gallery .
- A woman is standing in front of a wall .
- A painting of a woman leaning up against a wall in a room .
- A young woman painting a picture of a wall in the middle of an art gallery .
- A woman in a blue shirt sits in the middle of a wall .

# Results

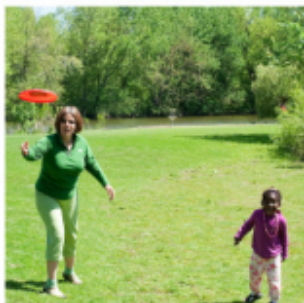
Flickr30K								
Model	Image Annotation				Image Search			
	R@1	R@5	R@10	Med $r$	R@1	R@5	R@10	Med $r$
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
† DeViSE [5]	4.5	18.1	29.2	26	6.7	21.9	32.7	25
† SDT-RNN [6]	9.6	29.8	41.1	16	8.9	29.8	41.1	16
† DeFrag [15]	14.2	37.7	51.3	10	10.2	30.8	44.2	14
† DeFrag + Finetune CNN [15]	16.4	<u>40.2</u>	<u>54.7</u>	<u>8</u>	10.3	31.4	44.5	<u>13</u>
m-RNN [7]	<u>18.4</u>	<u>40.2</u>	<u>50.9</u>	10	<u>12.6</u>	31.2	41.5	16
Our model	14.8	39.2	50.9	10	11.8	<u>34.0</u>	<u>46.3</u>	<u>13</u>
Our model (OxfordNet)	<b>23.0</b>	<b>50.7</b>	<b>62.9</b>	<b>5</b>	<b>16.8</b>	<b>42.0</b>	<b>56.5</b>	<b>8</b>

- R@K is Recall@K (high is good).
- Med  $r$  is the median rank (low is good).



# Caption Generation with Visual Attention

A woman is throwing a frisbee in a park.



# Caption Generation with Visual Attention

A woman is throwing a frisbee in a park.



# Caption Generation with Visual Attention


- Montreal/Toronto team takes 3<sup>rd</sup> place on Microsoft COCO caption generation competition, finishing slightly behind Google and Microsoft. This is based on the human evaluation results.

[Table-C5](#)

[Table-C40](#)

[Table-human](#)

Last update: June 8, 2015. Visit [CodaLab](#) for the latest results.

	<b>M1</b>	 <b>M2</b>	<b>M3</b>	<b>M4</b>	<b>M5</b>
Human <sup>[5]</sup>	0.638	0.675	4.836	3.428	0.352
Google <sup>[4]</sup>	0.273	0.317	4.107	2.742	0.233
MSR <sup>[8]</sup>	0.268	0.322	4.137	2.662	0.234
Montreal/Toronto <sup>[10]</sup>	0.262	0.272	3.932	2.832	0.197
MSR Captivator <sup>[9]</sup>	0.250	0.301	4.149	2.565	0.233
Berkeley LRCN <sup>[2]</sup>	0.246	0.268	3.924	2.786	0.204
m-RNN <sup>[15]</sup>	0.223	0.252	3.897	2.595	0.202
Nearest Neighbor <sup>[11]</sup>	0.216	0.255	3.801	2.716	0.196

# Improving Action Recognition

- Consider performing action recognition in a video:



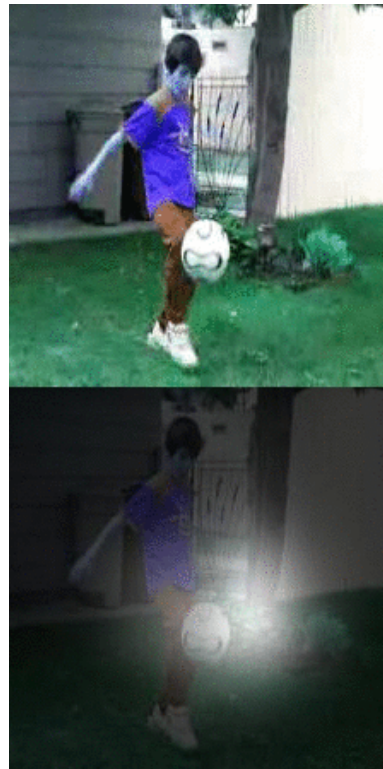
- Instead of processing each frame, we can process only a small piece of each frame.

# Improving Action Recognition

Cycling



Soccer juggling



Horse back riding



Basketball Shooting





# Talk Roadmap

- Zero-Shot Learning
- Caption Generation
- Learning Recurrent Attention Models



Lei Jimmy Ba



Roger Grosse

Wake-Sleep Recurrent Attention Models  
Ba, Grosse, Frey, Salakhutdinov, 2015

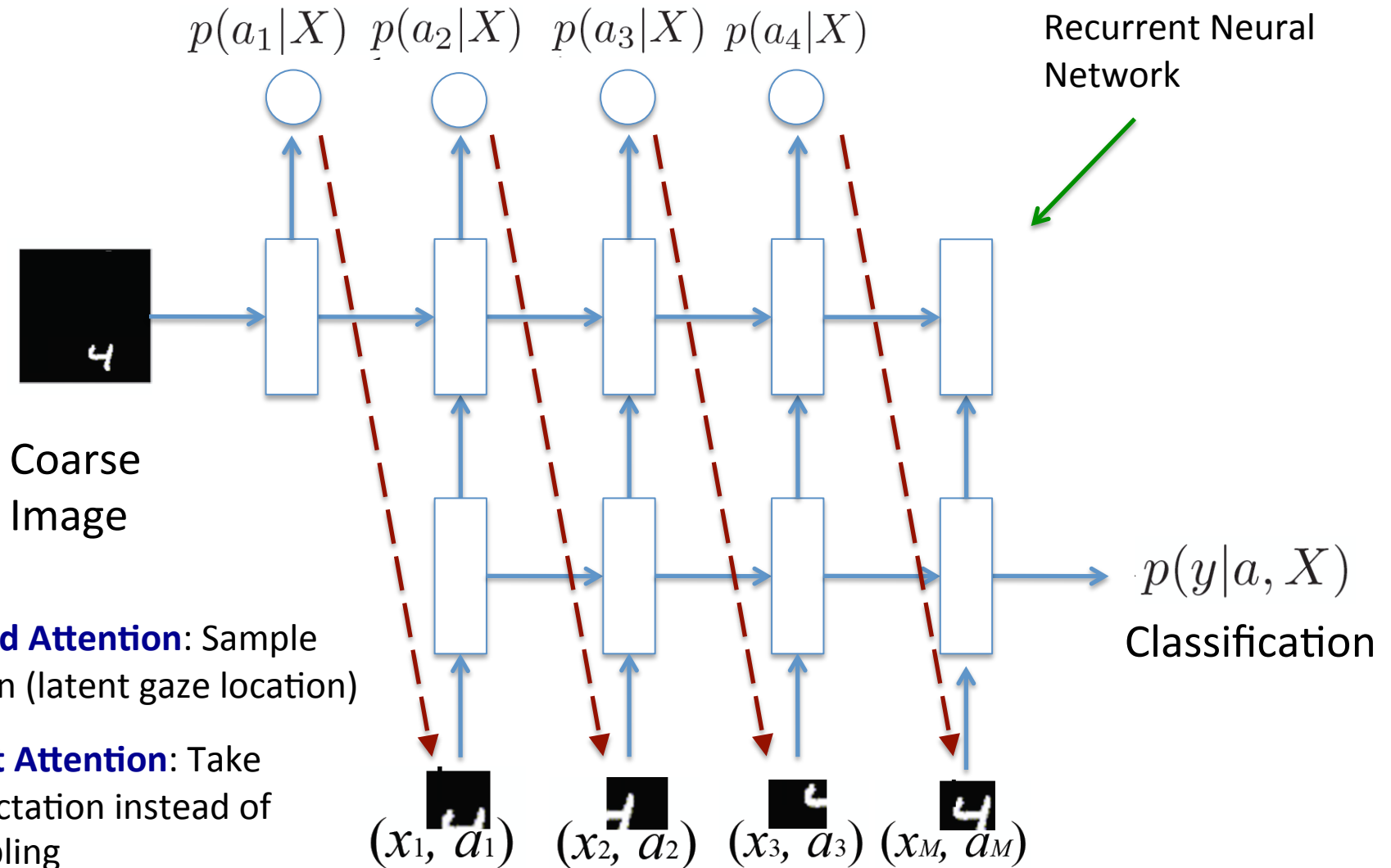
- Learning Skip-Thought Vectors



# Recurrent Attention Model

Sample action:

$$\tilde{a}_1 \sim p(a_1|X) \quad \tilde{a}_2 \sim p(a_2|X)$$



# Model Setup

- We assume that we have a dataset with labels  $y$  for the supervised prediction task (e.g. object category).
- **Goal:** Learn an attention policy: The best locations to attend to are the ones which lead the model to predict the correct class.



# Model Definition

- We aim to maximize the probability of correct class by marginalizing over the actions (or latent gaze locations):

$$\mathcal{LL} = \log p(y|X, W) = \log \sum_a p(a|X, W)p(y|a, X, W).$$

where

- $W$  is the set of parameters of the recurrent network.
- $a$  is a set of actions (latent gaze locations, scale).
- $X$ : is the input (e.g. image, video frame).

For clarity of presentation, I will sometimes omit conditioning on  $W$  or  $X$ . It should be obvious from the context.

# Variational Learning

- Previous approaches used variational lower bound:

$$\mathcal{L} = \log \sum_a p(a|X, W) p(y|a, X, W) \geq \sum_a q(a|y, X) \log p(y, a|X, W) + \mathcal{H}[q] = \mathcal{F}.$$

- Here  $q(a|y, X)$  is some approximation to posterior over the gaze locations.
- In the case where  $q$  is the prior,  $q(a|y, X) = p(a|X, W)$ , the variational bound becomes:

$$\mathcal{F} = \sum_a p(a|X, W) \log p(y|a, X, W).$$

Ba et.al., ICLR 2015  
Mnih et.al., NIPS 2014

# Variational Learning

$$\mathcal{F} = \sum_a p(a|X, W) \log p(y|a, X, W).$$

- Derivatives w.r.t model parameters:

$$\frac{\partial \mathcal{F}}{\partial W} = \sum_a p(a|X, W) \left[ \frac{\partial \log p(y|a, X, W)}{\partial W} + \underbrace{\log p(y|a, X, W)}_{\text{Very bad term as it is unbounded.}} \frac{\partial \log p(a|X, W)}{\partial W} \right].$$

Very bad term as it is unbounded.  
Introduces high variance in the estimator.

- Need to introduce heuristics (e.g. replacing this term with a 0/1 discrete indicator function, which leads to REINFORCE algorithm of Williams, 1992).

# Variational Learning

$$\mathcal{F} = \sum_a p(a|X, W) \log p(y|a, X, W).$$

- Derivatives w.r.t model parameters:

$$\frac{\partial \mathcal{F}}{\partial W} = \sum_a p(a|X, W) \left[ \frac{\partial \log p(y|a, X, W)}{\partial W} + \log p(y|a, X, W) \frac{\partial \log p(a|X, W)}{\partial W} \right].$$

- The stochastic estimator of the gradient is given by:

$$\frac{\partial \mathcal{F}}{\partial W} \approx \frac{1}{M} \sum_{m=1}^M \left[ \frac{\partial \log p(y|\tilde{a}^m, X, W)}{\partial W} + \log p(y|\tilde{a}^m, X, W) \frac{\partial \log p(\tilde{a}^m|X, W)}{\partial W} \right].$$

where we draw M samples from the prior:  $\tilde{a}^m \sim p(a|X, W)$ .

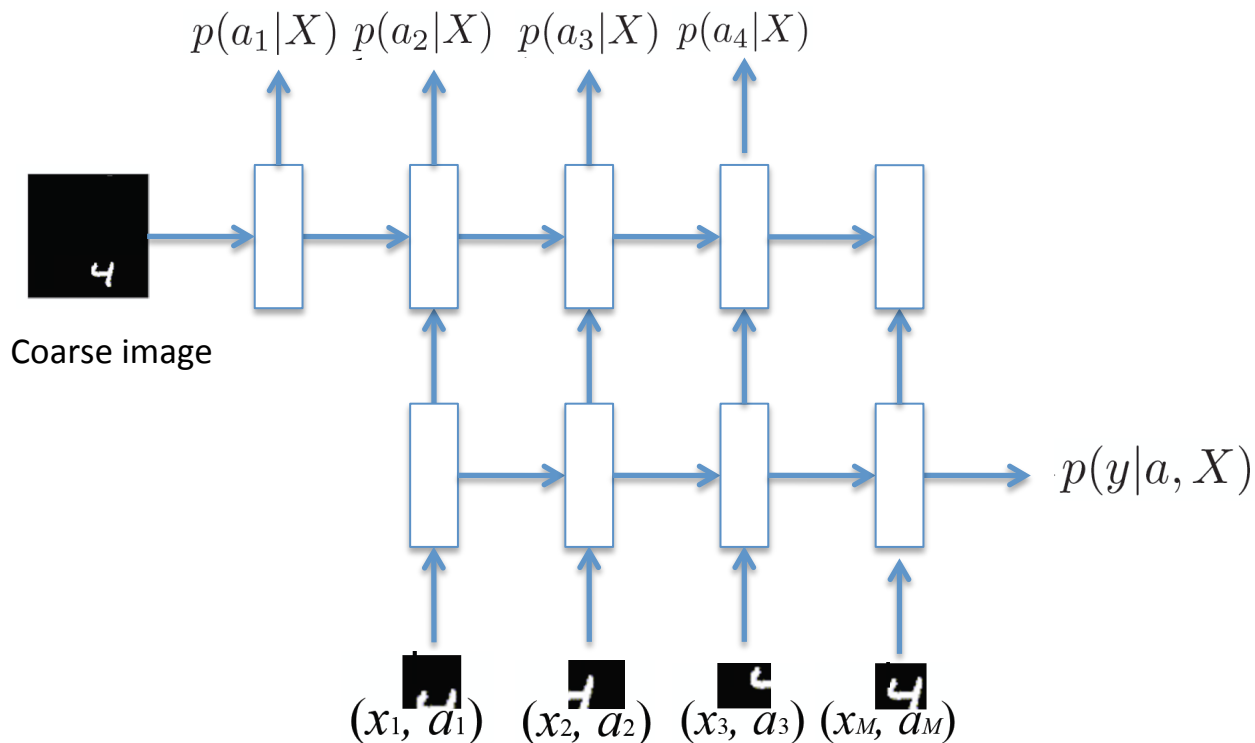


# Sampling from the Prior

- Generate M samples from the prior  $\tilde{a}^m \sim p(a|X, W)$ .

$$\frac{\partial \mathcal{F}}{\partial W} \approx \frac{1}{M} \sum_{m=1}^M \left[ \frac{\partial \log p(y|\tilde{a}^m, X, W)}{\partial W} + \log p(y|\tilde{a}^m, X, W) \frac{\partial \log p(\tilde{a}^m|X, W)}{\partial W} \right].$$

Run the network forward:



# Key Observation

- We can maximize the **marginal class log-probability directly** without adhering to the variational lower bound:

$$\mathcal{L} = \log p(y|X, W) = \log \sum_a p(a|X, W) p(y|a, X, W).$$

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{1}{\mathcal{Z}} \sum_a \underbrace{p(a|X, W) p(y|a, X, W)}_{\text{Posterior: } p(a|y, X, W)} \left[ \frac{\partial \log p(y|a, X, W)}{\partial W} + \frac{\partial \log p(a|X, W)}{\partial W} \right]$$

where

$$\mathcal{Z} = \sum_a p(a|X, W) p(y|a, X, W)$$

- We can use importance sampling to estimate required expectations.

# Maximizing Marginal Likelihood

- Need to estimate:

$$\frac{\partial \mathcal{L}\mathcal{L}}{\partial W} = \frac{1}{\mathcal{Z}} \sum_a p(a|X, W) p(y|a, X, W) \left[ \frac{\partial \log p(y|a, X, W)}{\partial W} + \frac{\partial \log p(a|X, W)}{\partial W} \right],$$

- Let  $q(a|y, X)$  be some approximation to the posterior:

$$q(a|y, X) \approx p(a|y, X, W).$$

- Using **Importance Sampling**, we obtain:

$$\tilde{w}^m = \frac{p(\tilde{a}^m|X, W) p(y|\tilde{a}^m, X, W)}{q(\tilde{a}^m|y, X)}, \quad \tilde{a}^m \sim q(a|y, X).$$

- The **stochastic estimator** of the gradient is given by:

$$\frac{\partial \mathcal{L}\mathcal{L}}{\partial W} \approx \frac{1}{\tilde{\mathcal{Z}}} \sum_{m=1}^M \tilde{w}^m \left[ \frac{\partial \log p(y|\tilde{a}^m, X, W)}{\partial W} + \frac{\partial \log p(\tilde{a}^m|X, W)}{\partial W} \right],$$

where  $\tilde{\mathcal{Z}} = \sum_m \tilde{w}^m$ .

# Comparing the Two Estimators

- Variational bound vs. Marginal likelihood:

$$\frac{\partial \mathcal{F}}{\partial W} \approx \frac{1}{M} \sum_{m=1}^M \left[ \frac{\partial \log p(y|\tilde{a}^m, X, W)}{\partial W} + \log p(y|\tilde{a}^m, X, W) \frac{\partial \log p(\tilde{a}^m|X, W)}{\partial W} \right]$$

Very bad term, as it is unbounded

$$\frac{\partial \mathcal{LL}}{\partial W} \approx \frac{1}{\tilde{Z}} \sum_{m=1}^M \tilde{w}^m \left[ \frac{\partial \log p(y|\tilde{a}^m, X, W)}{\partial W} + \frac{\partial \log p(\tilde{a}^m|X, W)}{\partial W} \right]$$

- The performance gain from importance sampling heavily relies on an appropriate choice of the proposal distribution  $q$ !

When approximate posterior  $q$  is equal to the prior, this approach is equivalent to Tang and Salakhutdinov for learning generative networks (NIPS 2013). It is also similar to the Reweighted Wake-Sleep of Bornschein and Bengio (ICLR 2015).

# Another Key Observation

- Using **finite number of samples M**, our importance sampling estimator can be viewed as the gradient ascent on the following objective:

$$\mathbb{E} \left[ \log \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right]$$

- Using Jensen's inequality we obtain:

$$\mathbb{E} \left[ \log \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right] \leq \log \mathbb{E} \left[ \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right] = \log \mathbb{E} [\tilde{w}^m] = \mathcal{L}\mathcal{L}$$

- Hence in expectation, we are optimizing a lower bound on the marginal likelihood (although the variance can be high).
- The bound becomes tighter as we increase M.

# Another Key Observation

- Using **finite number of samples  $M$** , our importance sampling estimator can be viewed as the gradient ascent on the following objective:

$$\mathbb{E} \left[ \log \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right]$$

- Using Jensen's inequality we obtain:

$$\mathbb{E} \left[ \log \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right] \leq \log \mathbb{E} \left[ \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right] = \log \mathbb{E} [\tilde{w}^m] = \mathcal{L}\mathcal{L}$$

- However, our proposed bound is at least as accurate as the variational bound:

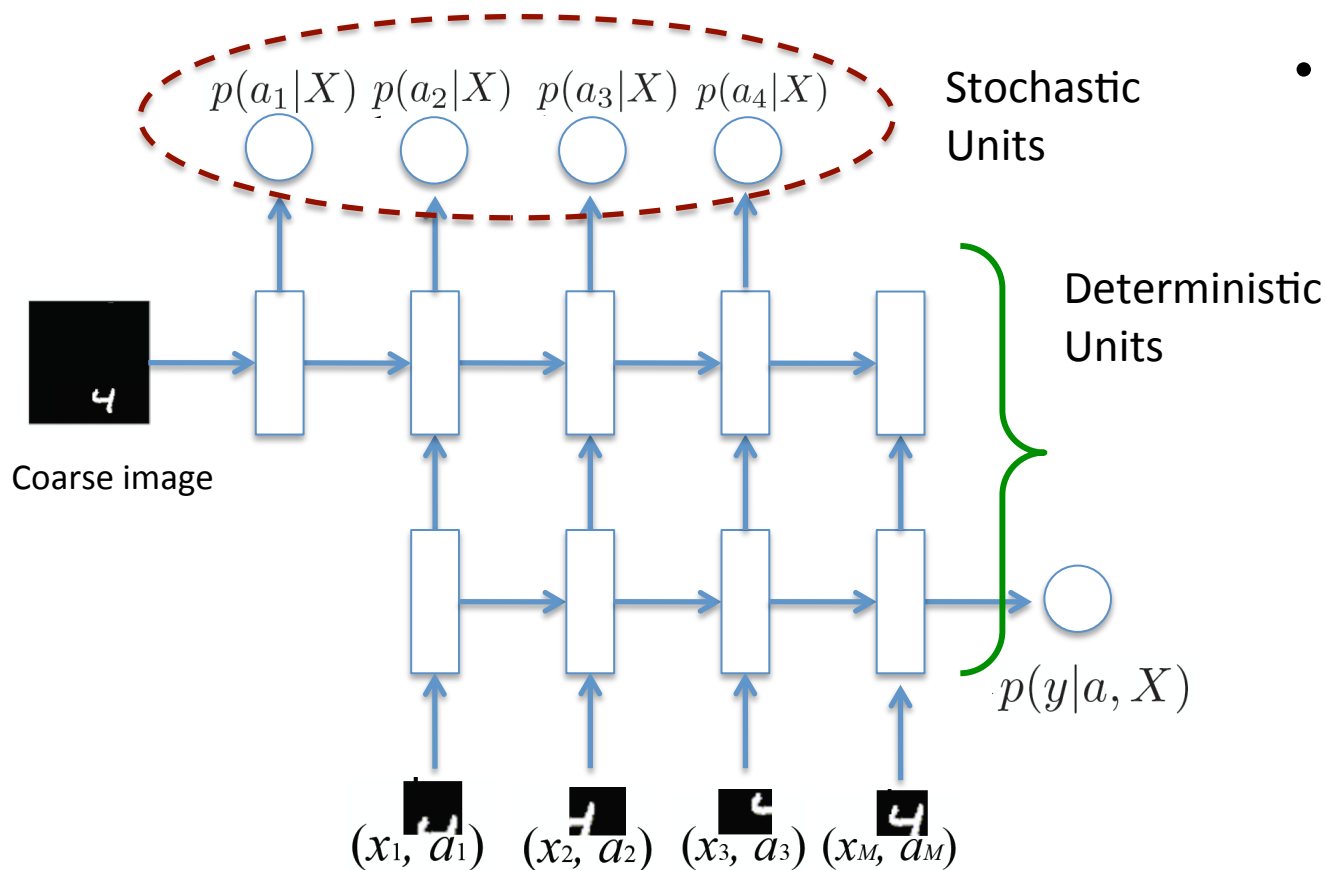
$$\mathcal{F} = \mathbb{E} [\log \tilde{w}^m] = \mathbb{E} \left[ \frac{1}{M} \sum_{m=1}^M \log \tilde{w}^m \right] \leq \mathbb{E} \left[ \log \frac{1}{M} \sum_{m=1}^M \tilde{w}^m \right]$$



# Relationship To Helmholtz Machines

- **Goal:** Maximize the probability of correct class (sequence of words) by marginalizing over the actions (or latent gaze locations):

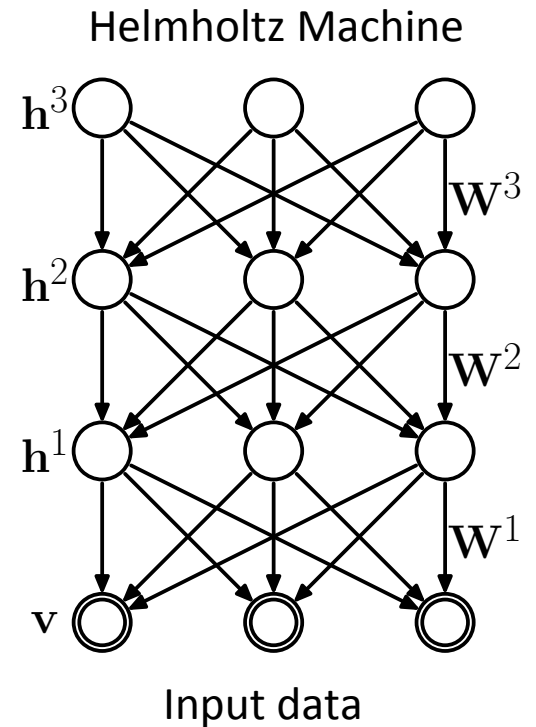
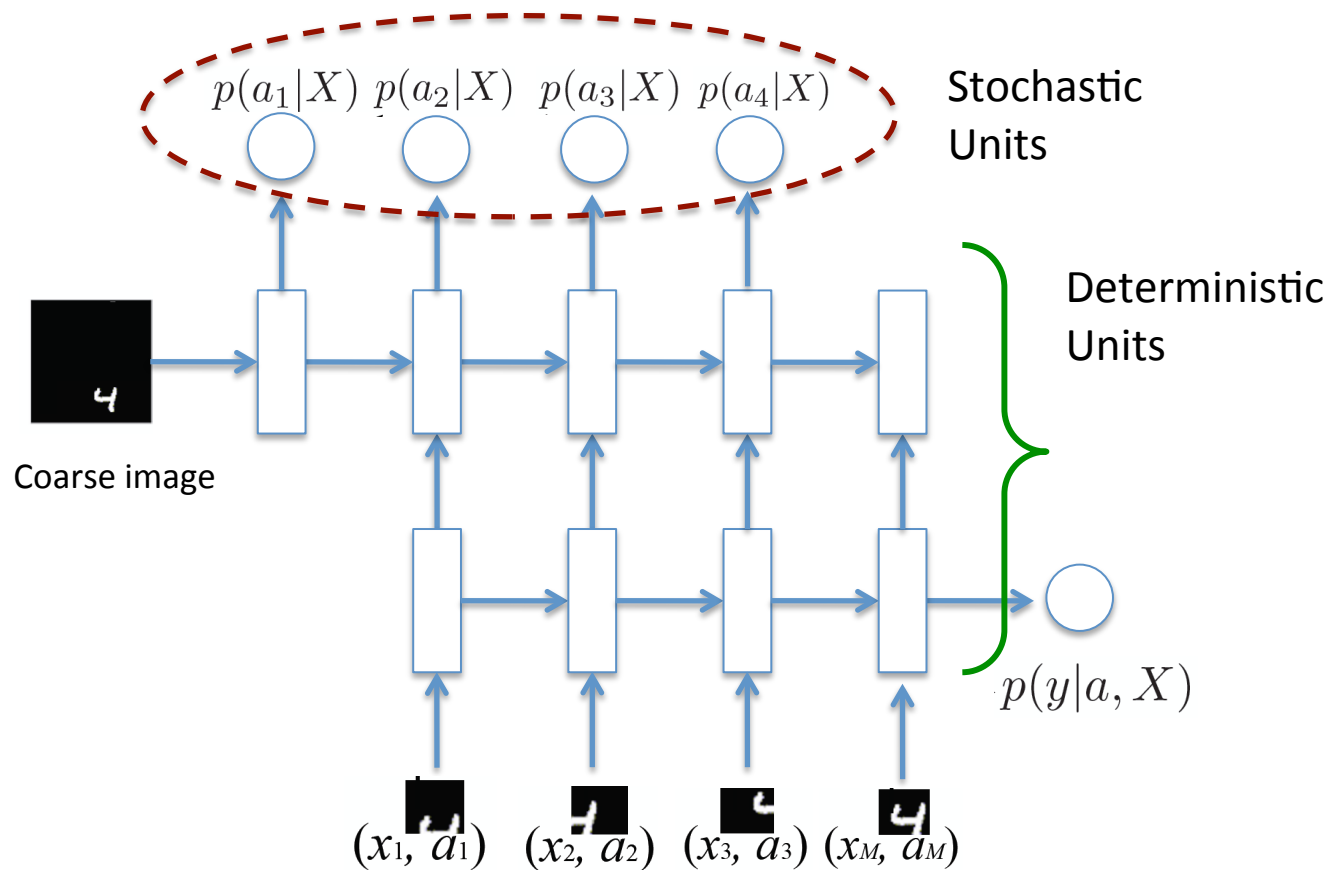
$$\mathcal{LL} = \log p(y|X, W) = \log \sum_a p(a|X, W)p(y|a, X, W).$$



- Neural Network with stochastic and deterministic units.

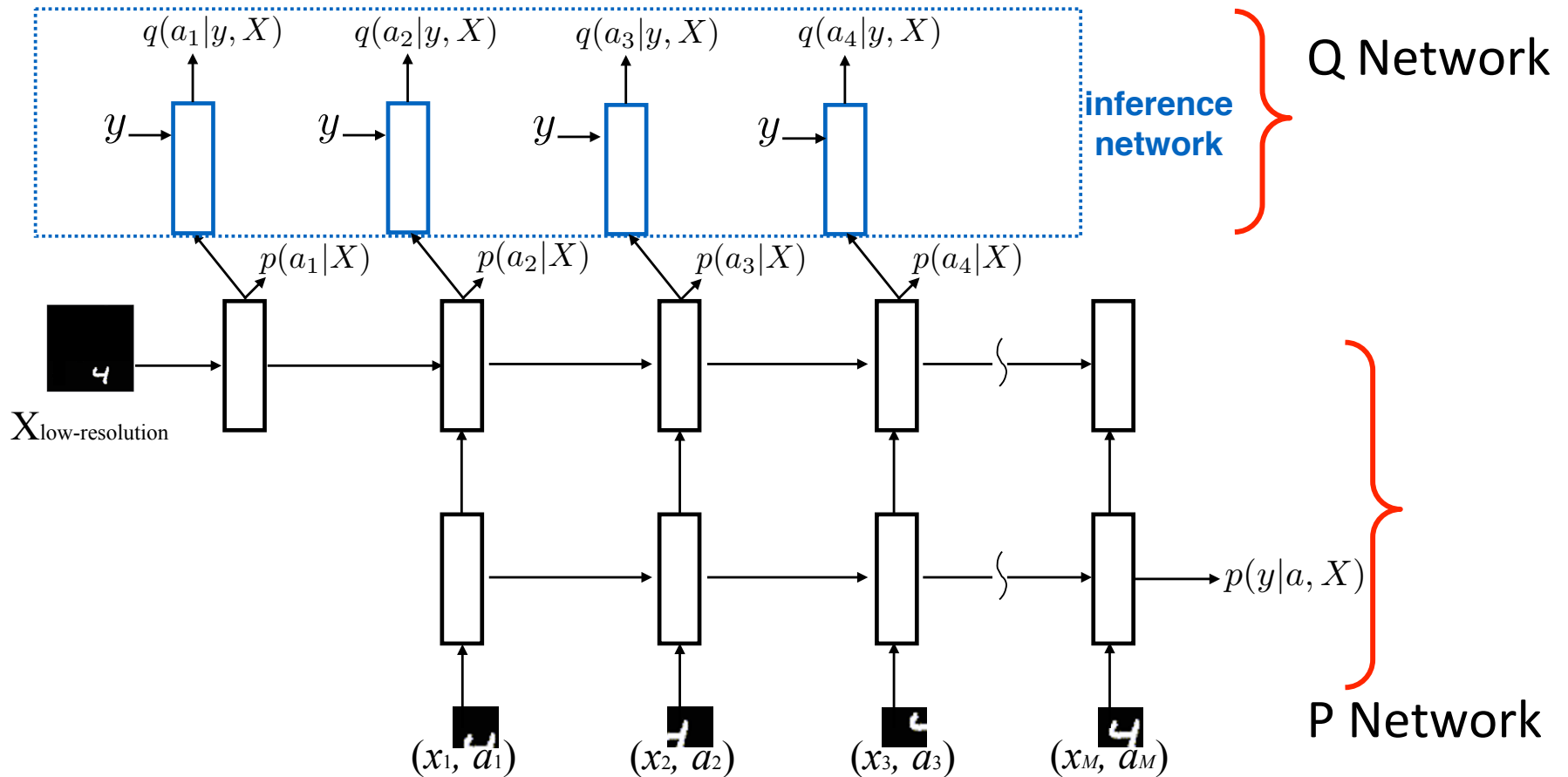
# Relationship To Helmholtz Machines

- View this model as a conditional Helmholtz Machine with stochastic and deterministic units.
- Can use Wake-Sleep, Re-weighted Wake Sleep, variational autoencoders, and their variants to learn good attention policy!



# The Wake-Sleep Recurrent Attention Model

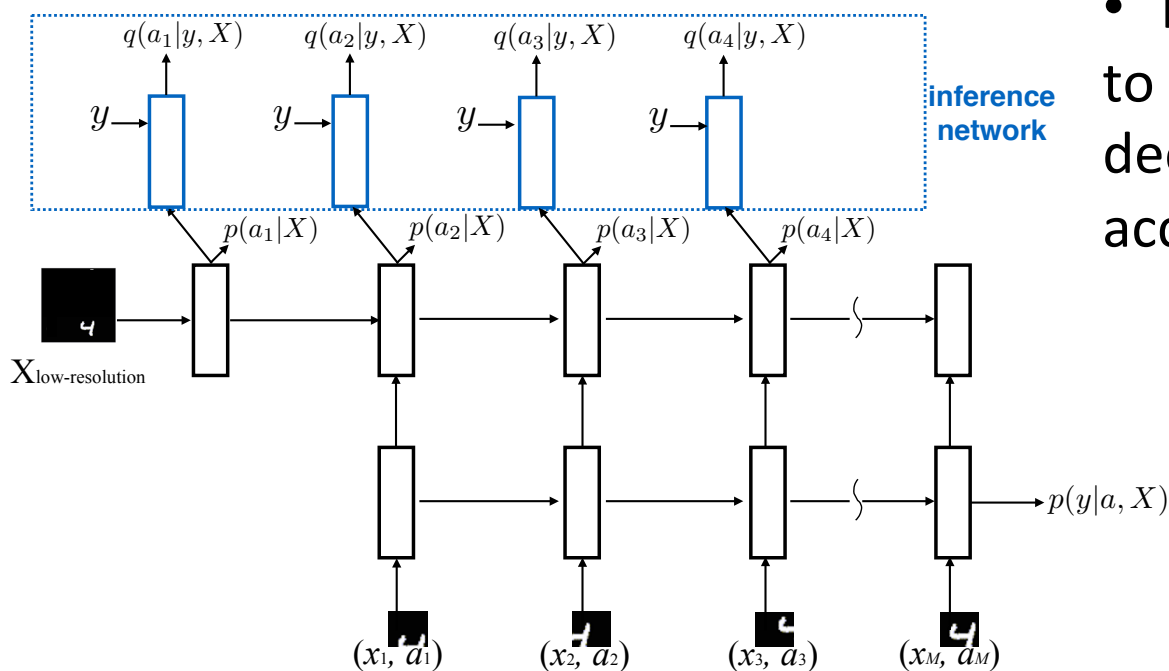
- We can learn both: generative model P and recognition model Q.



# Training Inference Network

- We train an inference network to predict glimpses, **given the observations as well as the class label**, where the network should look to correctly predict that class.

$$q(a|y, X, \theta) = \prod_{n=1}^N q(a_n | y, X, \theta, a_{1:n-1}).$$



- This distribution is analogous to the prior, except that each decision also takes into account the class label  $y$ .

# Training Inference Network

- To train  $q$  network we optimize:

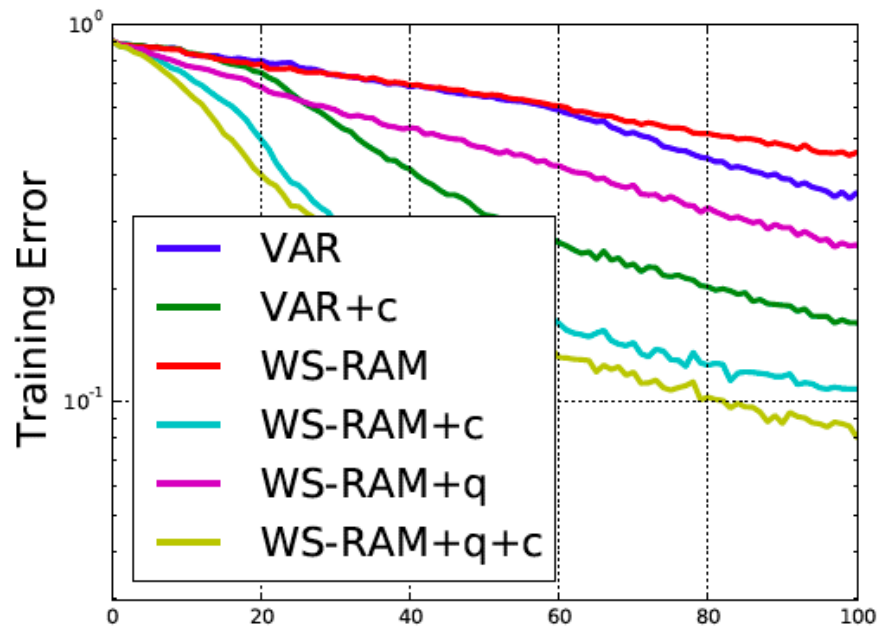
$$\frac{\partial D_{\text{KL}}(p||q)}{\partial \theta} = \mathbb{E}_{p(a|y, X, W)} \left[ \frac{\partial \log q(a|y, X, \theta)}{\partial \theta} \right]$$

- Using importance sampling, we get the following stochastic estimate of the gradient:

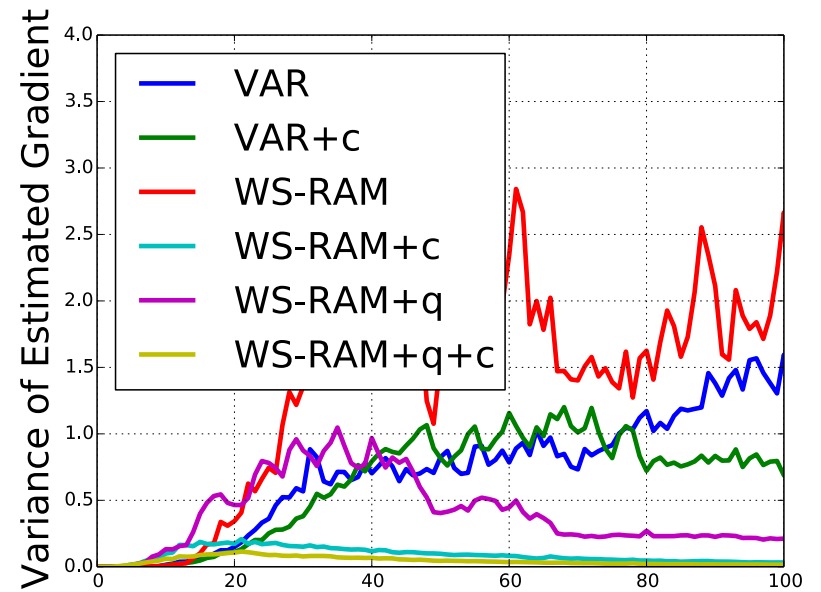
$$\frac{\partial D_{\text{KL}}(p||q)}{\partial \theta} \approx \frac{1}{Z} \sum_{m=1}^M \tilde{w}^m \frac{\partial \log q(\tilde{a}^m|y, X, \theta)}{\partial \theta}$$

- In fact we can reuse the importance weights computed for the attention model update.

# MNIST Example



Training error as a number of updates.



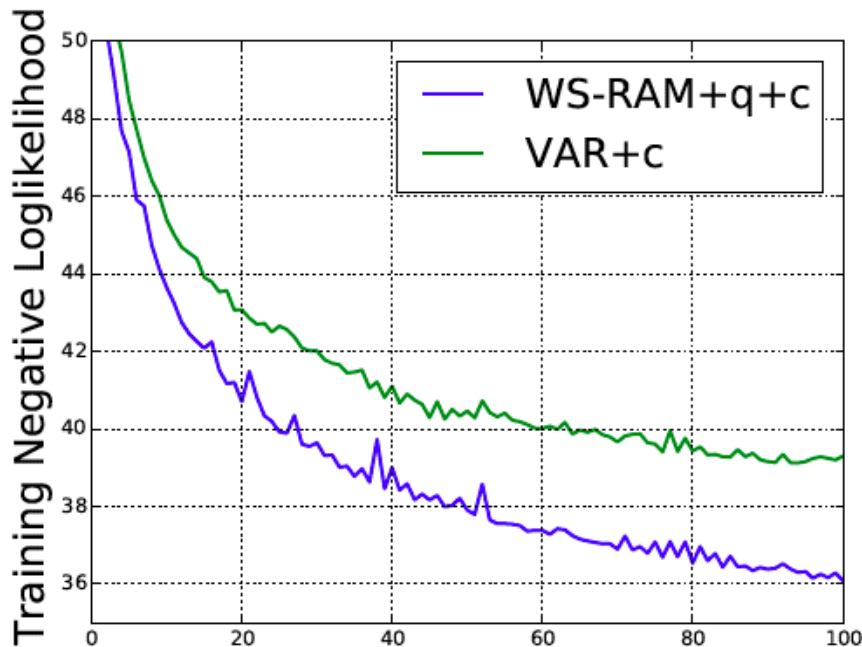
Variance of Estimated Gradients.



# Caption Generation: Flickr 8K

- Comparing the variational method with our wake-sleep based estimator using an inference network:

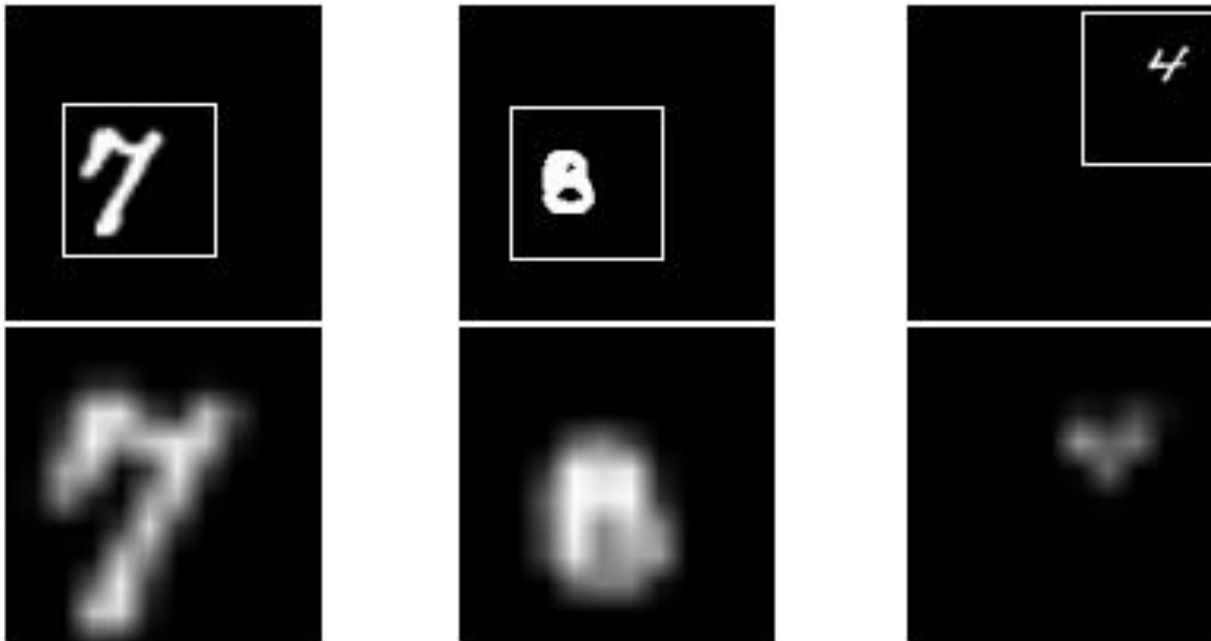
	BLEU1	BLEU2	BLEU3	BLEU4
VAR	62.3	41.6	26.9	17.2
WS-RAM+Qnet	61.1	40.4	26.9	17.8



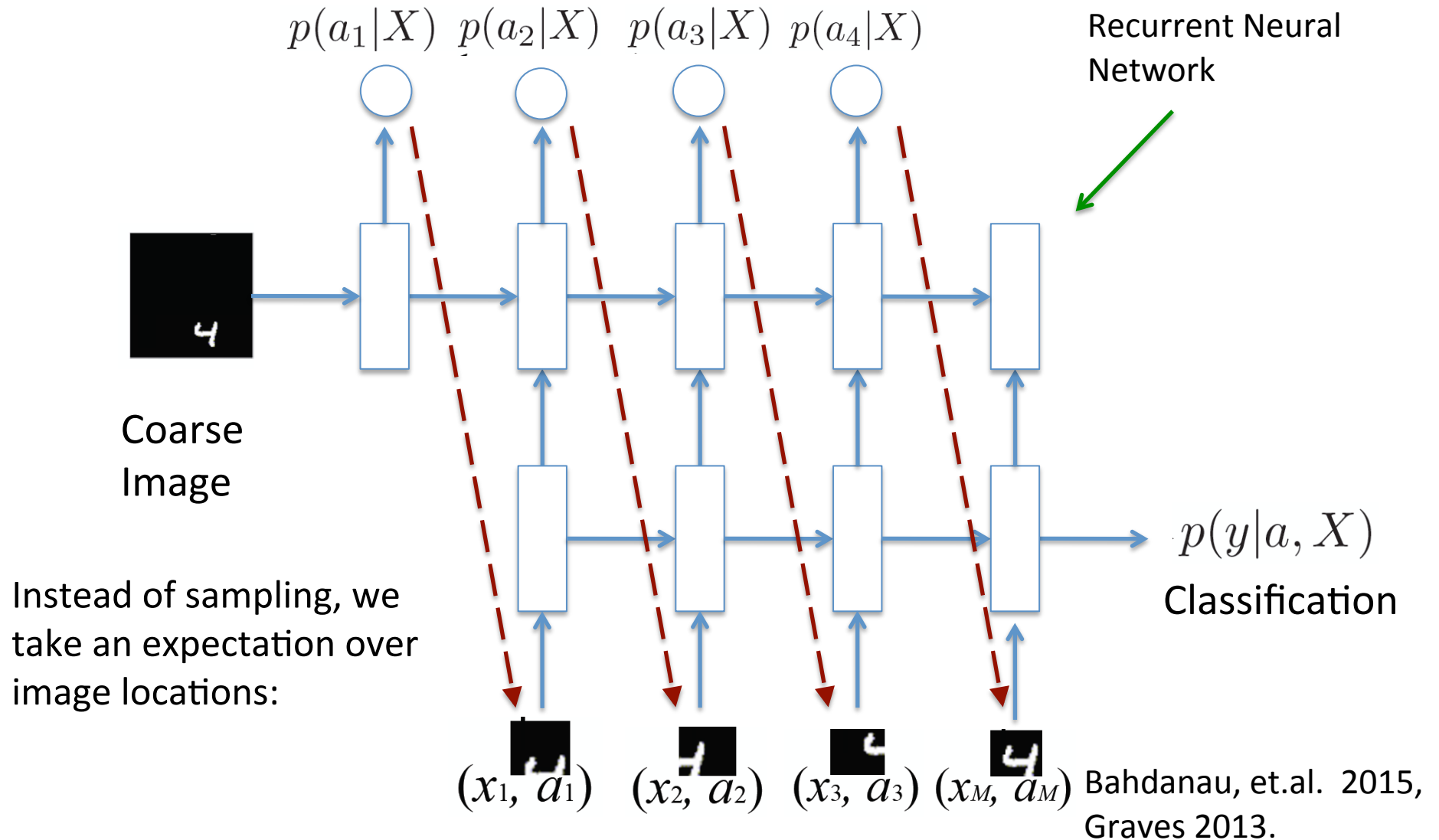
- Training negative log-likelihood on Flickr8K for the first 10,000 updates.

# MNIST Attention Demo

- Actions contain:
  - Location: 2-d Gaussian latent variable
  - Scale: 3-way softmax over 3 different scales



# Recurrent Attention Model



# Hard vs. Soft Attention

- Soft attention models:
  - **Computationally expensive**. They have to examine every image location. Hard to scale to large video datasets.
  - **Deterministic**. They can be trained by backprop.
- Hard attention models:
  - **Computationally more efficient**. They need to process only small part of each image frame.
  - **Stochastic**. Require some form of sampling, because they must make discrete choices.
- Research is taking place on both fronts!

# Talk Roadmap

- Zero-Shot Learning
- Caption Generation
- Learning Recurrent Attention Models
- Learning Skip-Thought Vectors



Ryan Kiros

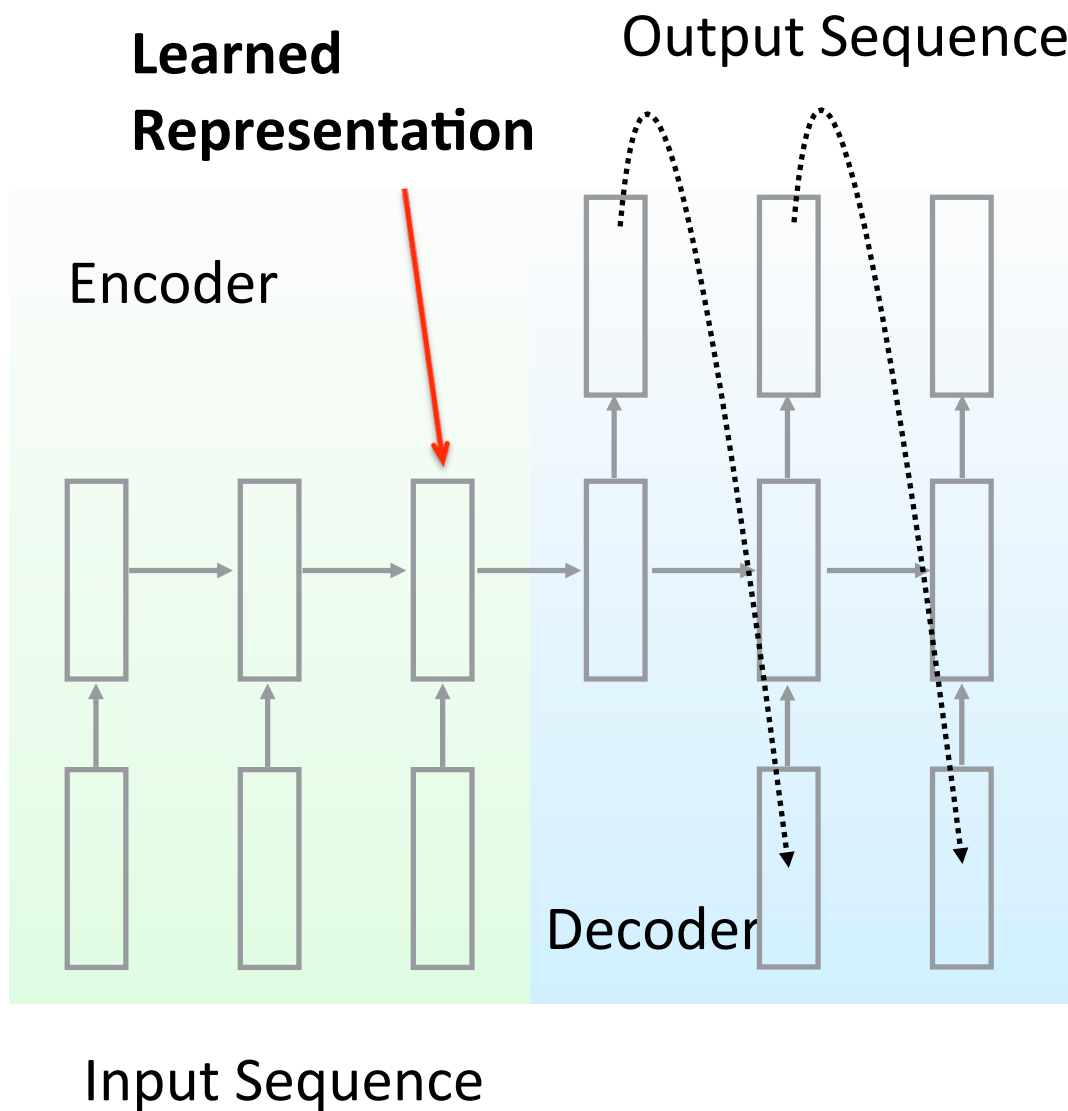


Yukun Zhu

Skip-Thought Vectors

Kiros, Zhu, Salakhutdinov, Zemel,  
Torralba, Urtasun, Fidler, arXiv 2015

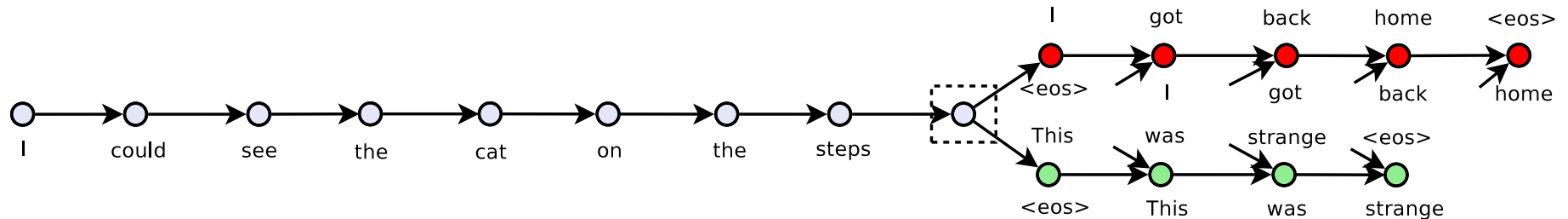
# Sequence to Sequence Learning



- RNN Encoder-Decoders for Machine Translation (Sutskever et al. 2014; Cho et al. 2014; Kalchbrenner et al. 2013, Srivastava et.al., 2015)

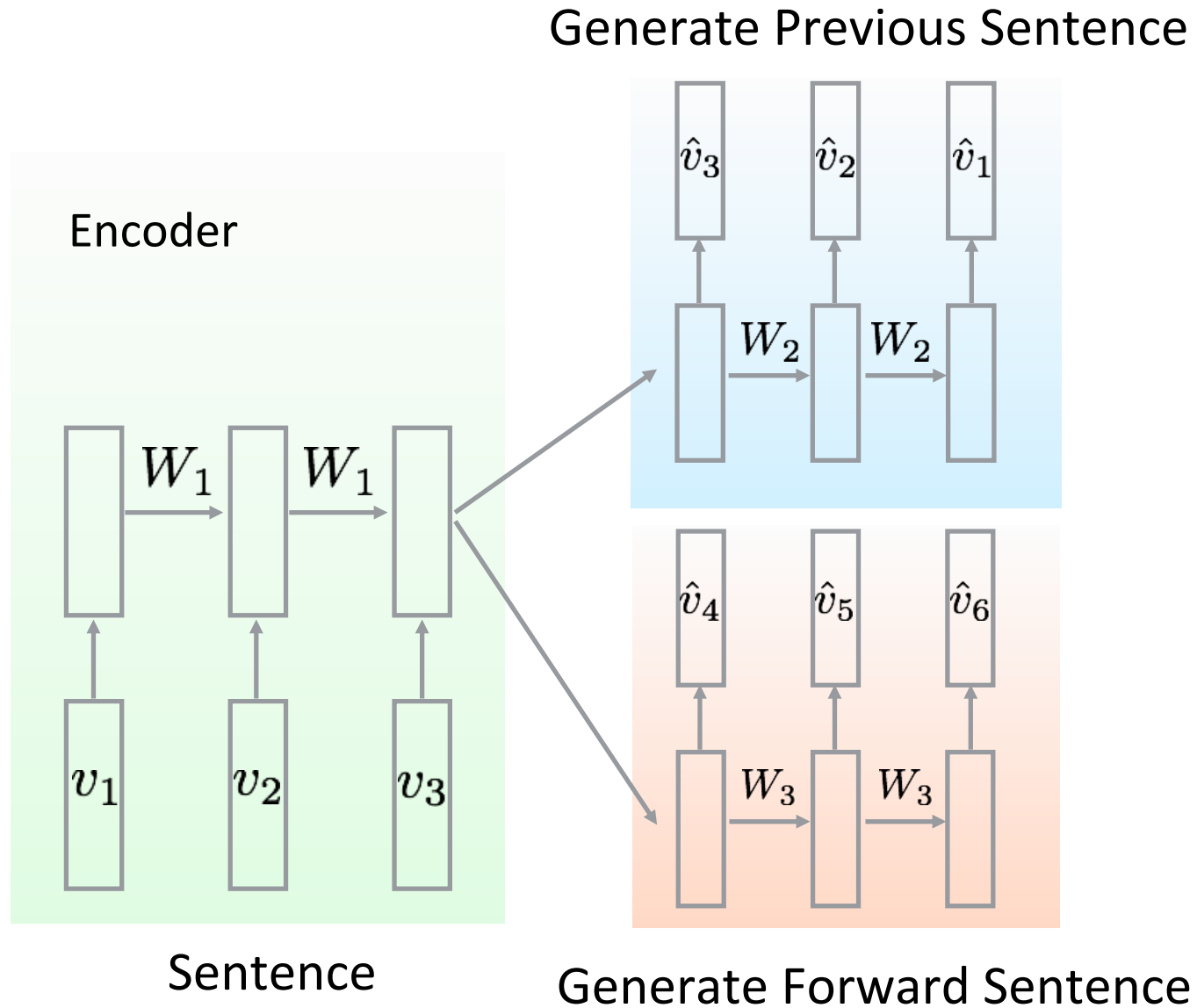


# Skip-Thought Model



- Given a tuple  $(s_{i-1}, s_i, s_{i+1})$  of contiguous sentences:
  - the sentence  $s_i$  is encoded using LSTM.
  - the sentence  $s_i$  attempts to reconstruct the previous sentence and next sentence  $s_{i+1}$ .
- The input is the sentence triplet:
  - I got back home.
  - I could see the cat on the steps.
  - This was strange.

# Skip-Thought Model



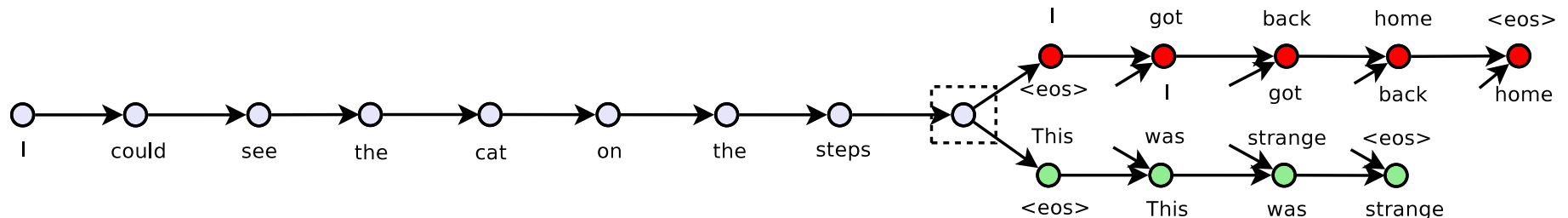
# Learning Objective

- We are given a tuple  $(s_{i-1}, s_i, s_{i+1})$  of contiguous sentences.
- **Objective:** The sum of the log-probabilities for the next and previous sentences conditioned on the encoder representation:

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$

representation of encoder  
↓

Forward sentence                      Previous sentence



# Book 11K corpus

# of books	# of sentences	# of words	# of unique words
11,038	74,004,228	984,846,357	1,316,420

- Query sentence along with its nearest neighbor from 500K sentences using cosine similarity:
  - He ran his hand inside his coat, double-checking that the unopened letter was still there.
  - He slipped his hand between his coat and his shirt, where the folded copies lay in a brown envelope.

# Book 11K corpus

# of books	# of sentences	# of words	# of unique words
11,038	74,004,228	984,846,357	1,316,420

- Query sentence along with its nearest neighbor from 500K sentences using cosine similarity:

---

## Query and nearest sentence

---

he ran his hand inside his coat , double-checking that the unopened letter was still there .  
he slipped his hand between his coat and his shirt , where the folded copies lay in a brown envelope .

---

im sure youll have a glamorous evening , she said , giving an exaggerated wink .  
im really glad you came to the party tonight , he said , turning to her .

---

although she could tell he had n't been too invested in any of their other chitchat , he seemed genuinely curious about this .  
although he had n't been following her career with a microscope , he 'd definitely taken notice of her appearances .

---

an annoying buzz started to ring in my ears , becoming louder and louder as my vision began to swim .  
a weighty pressure landed on my lungs and my vision blurred at the edges , threatening my consciousness altogether .

---

# Semantic Relatedness

- SemEval 2014 Task 1: semantic relatedness SICK dataset: Given two sentences, produce a score of how semantically related these sentences are based on human generated scores (1 to 5).
- The dataset comes with a predefined split of 4500 training pairs, 500 development pairs and 4927 testing pairs.
- Using skip-thought vectors for each sentence, we simply train a linear regression to predict semantic relatedness.
  - For pair of sentences, we compute component-wise features between pairs (e.g.  $|u-v|$ ).



# Semantic Relatedness

	Method	$r$	$\rho$	MSE
SemEval 2014 sub- missions	Illinois-LH [18]	0.7993	0.7538	0.3692
	UNAL-NLP [19]	0.8070	0.7489	0.3550
	Meaning Factory [20]	0.8268	0.7721	0.3224
	ECNU [21]	0.8414	–	–
Results reported by Tai et.al.	Mean vectors [22]	0.7577	0.6738	0.4557
	DT-RNN [23]	0.7923	0.7319	0.3822
	SDT-RNN [23]	0.7900	0.7304	0.3848
	LSTM [22]	0.8528	0.7911	0.2831
	Bidirectional LSTM [22]	0.8567	0.7966	0.2736
	Dependency Tree-LSTM [22]	<b>0.8676</b>	<b>0.8083</b>	<b>0.2532</b>
Ours	uni-skip	0.8477	0.7780	0.2872
	bi-skip	0.8405	0.7696	0.2995
	combine-skip	0.8584	0.7916	0.2687
	combine-skip+COCO	0.8655	0.7995	0.2561

- Our models outperform all previous systems from the SemEval 2014 competition. This is remarkable, given the simplicity of our approach and the lack of feature engineering.

# Semantic Relatedness

Sentence 1	Sentence 2	GT	pred
A little girl is looking at a woman in costume	A young girl is looking at a woman in costume	4.7	4.5
A little girl is looking at a woman in costume	The little girl is looking at a man in costume	3.8	4.0
A little girl is looking at a woman in costume	A little girl in costume looks like a woman	2.9	3.5
A sea turtle is hunting for fish	A sea turtle is hunting for food	4.5	4.5
A sea turtle is not hunting for fish	A sea turtle is hunting for fish	3.4	3.8
A man is driving a car	The car is being driven by a man	5	4.9
There is no man driving the car	A man is driving a car	3.6	3.5
A large duck is flying over a rocky stream	A duck, which is large, is flying over a rocky stream	4.8	4.9
A large duck is flying over a rocky stream	A large stream is full of rocks, ducks and flies	2.7	3.1
A person is performing acrobatics on a motorcycle	A person is performing tricks on a motorcycle	4.3	4.4
A person is performing tricks on a motorcycle	The performer is tricking a person on a motorcycle	2.6	4.4
Someone is pouring ingredients into a pot	Someone is adding ingredients to a pot	4.4	4.0
Nobody is pouring ingredients into a pot	Someone is pouring ingredients into a pot	3.5	4.2
Someone is pouring ingredients into a pot	A man is removing vegetables from a pot	2.4	3.6

- Example predictions from the SICK test set. GT is the ground truth relatedness, scored between 1 and 5.
- The last few results: slight changes in sentences result in large changes in relatedness that we are unable to score correctly.

# Paraphrase Detection

- Microsoft Research Paraphrase Corpus: For two sentences one must predict whether or not they are paraphrases.

	<b>Method</b>	<b>Acc</b>	<b>F1</b>	
Recursive Auto- encoders	feats [24]	73.2		The training set contains 4076 sentence pairs (2753 are positive)
	RAE+DP [24]	72.6		
	RAE+feats [24]	74.2		
	RAE+DP+feats [24]	76.8	83.6	
Best published results	FHS [25]	75.0	82.7	The test set contains 1725 pairs (1147 are positive).
	PE [26]	76.1	82.7	
	WDDP [27]	75.6	83.0	
	MTMETRICS [28]	<b>77.4</b>	<b>84.1</b>	
Ours	uni-skip	73.0	81.9	
	bi-skip	71.2	81.2	
	combine-skip	73.0	82.0	
	combine-skip + feats	75.8	83.0	

# Classification Benchmarks

- 5 datasets: movie review sentiment (MR), customer product reviews (CR), subjectivity/objectivity classification (SUBJ), opinion polarity (MPQA) and question-type classification (TREC).

	Method	MR	CR	SUBJ	MPQA	TREC
Bag-of-words	NB-SVM [41]	79.4	<u>81.8</u>	93.2	86.3	
	MNB [41]	79.0	80.0	<u>93.6</u>	86.3	
	cBoW [6]	77.2	79.9	91.3	86.4	87.3
Super-vised	GrConv [6]	76.3	81.3	89.5	84.5	88.4
	RNN [6]	77.2	82.3	93.7	90.1	90.2
	BRNN [6]	82.3	82.6	94.2	90.3	91.0
	CNN [4]	81.5	85.0	93.4	89.6	<b>93.6</b>
	AdaSent [6]	<b>83.1</b>	<b>86.3</b>	<b>95.5</b>	<b>93.3</b>	92.4
	Paragraph-vector [7]	74.8	78.1	90.5	74.2	91.8
Ours	uni-skip	75.5	79.3	92.1	86.9	91.4
	bi-skip	73.9	77.9	92.5	83.3	89.4
	combine-skip	76.5	80.1	<u>93.6</u>	87.1	<u>92.2</u>
	combine-skip + NB	<u>80.4</u>	81.3	<u>93.6</u>	<u>87.5</u>	

# Summary

- This model for learning skip-thought vectors only scratches the surface of possible objectives.
- Many variations have yet to be explored, including
  - deep encoders and decoders
  - larger context windows
  - encoding and decoding paragraphs
  - other encoders
- It is likely the case that more exploration of this space will result in even higher quality sentence representations.
- **Code and Data are available online**  
<http://www.cs.toronto.edu/~mbweb/>

Thank you

# References

- J. Ba, V. Mnih, K. Kavukcuoglu Multiple Object Recognition with Visual Attention, ICLR, 2015.
- L. Ba, K. Swersky, S. Fidler, Salakhutdinov, Predicting Deep Zero-Shot Convolutional Neural Networks using Textual Descriptions, arXiv 2015
- Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A Neural Probabilistic Language Model, JMLR 2003
- Y. Bengio, Y. LeCun, Scaling learning algorithms towards AI, Large-Scale Kernel Machines, MIT Press, 2007
- Y. Bengio, Learning Deep Architectures for AI, Foundations and Trends in Machine Learning, 2009
- Y. Burda, R. Grosse, R. Salakhutdinov, Accurate and Conservative Estimates of MRF Log-likelihood using Reverse Annealing, In AI and Statistics (AISTATS), 2015
- K Cho, B van Merriënboer, C Gulcehre, F Bougares, H Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, EMNLP 2014
- A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, T. Mikolov, DeViSE: A Deep Visual-Semantic Embedding Model, NIPS 2013
- G. Hinton, R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 2006.
- G. Hinton, S. Osindero, Y. The, A fast learning algorithm for deep belief nets, Neural Computation, 18, 2006G.
- Hinton, Training Products of Experts by Minimizing Contrastive Divergence, Neural Computation, 2002.



# References

- H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, NIPS 2007
- H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representation, .ICML 2009N.
- Kalchbrenner, P. Blunsom, Recurrent Continuous Translation Models, EMNLP 2013
- R. Kiros, R. Salakhutdinov, R. Zemel, Multimodal Neural Language Models, ICML 2014
- R. Kiros, R. Salakhutdinov, R. Zemel, Unifying, Visual-Semantic Embeddings with Multimodal Neural Language Models, TACL 2015.
- R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, S. Fidler, Skip-Thought Vectors, arXiv 2015Y.
- LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 1998
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, NIPS 2013
- R. Salakhutdinov, J. Tenenbaum, A. Torralba, Learning with Hierarchical-Deep Models, IEEE PAMI, vol. 35, no. 8, Aug. 2013,
- R. Salakhutdinov, Learning Deep Generative Models, Annual Review of Statistics and Its Application, Vol. 2, 2015
- R. Salakhutdinov, G. Hinton, An Efficient Learning Procedure for Deep Boltzmann Machines, Neural Computation, 2012, Vol. 24, No. 8.
- R. Salakhutdinov, H. Larochelle, Efficient Learning of Deep Boltzmann Machines, AI and Statistics, 2010

# References

- R. Salakhutdinov, G. Hinton, Replicated Softmax: an Undirected Topic Model, NIPS 2010
- R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann Machines for Collaborative Filtering, ICML 2007
- N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised Learning of Video Representations using LSTMs, ICML, 2015
- N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR, 2014
- I. Sutskever, O. Vinyals, Q. Le,, Sequence to Sequence Learning with Neural Networks, NIPS 2014
- R. Socher, M. Ganjoo, C. Manning, A. Ng, Zero-Shot Learning Through Cross-Modal Transfer, NIPS 2013
- Y. Tang, R. Salakhutdinov, G. Hinton, Deep Lambertian Networks, ICM 2012
- Y. Tang, R. Salakhutdinov, G. Hinton, Robust Boltzmann Machines for Recognition and Denoising, CVPR 2012
- K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML, 2015