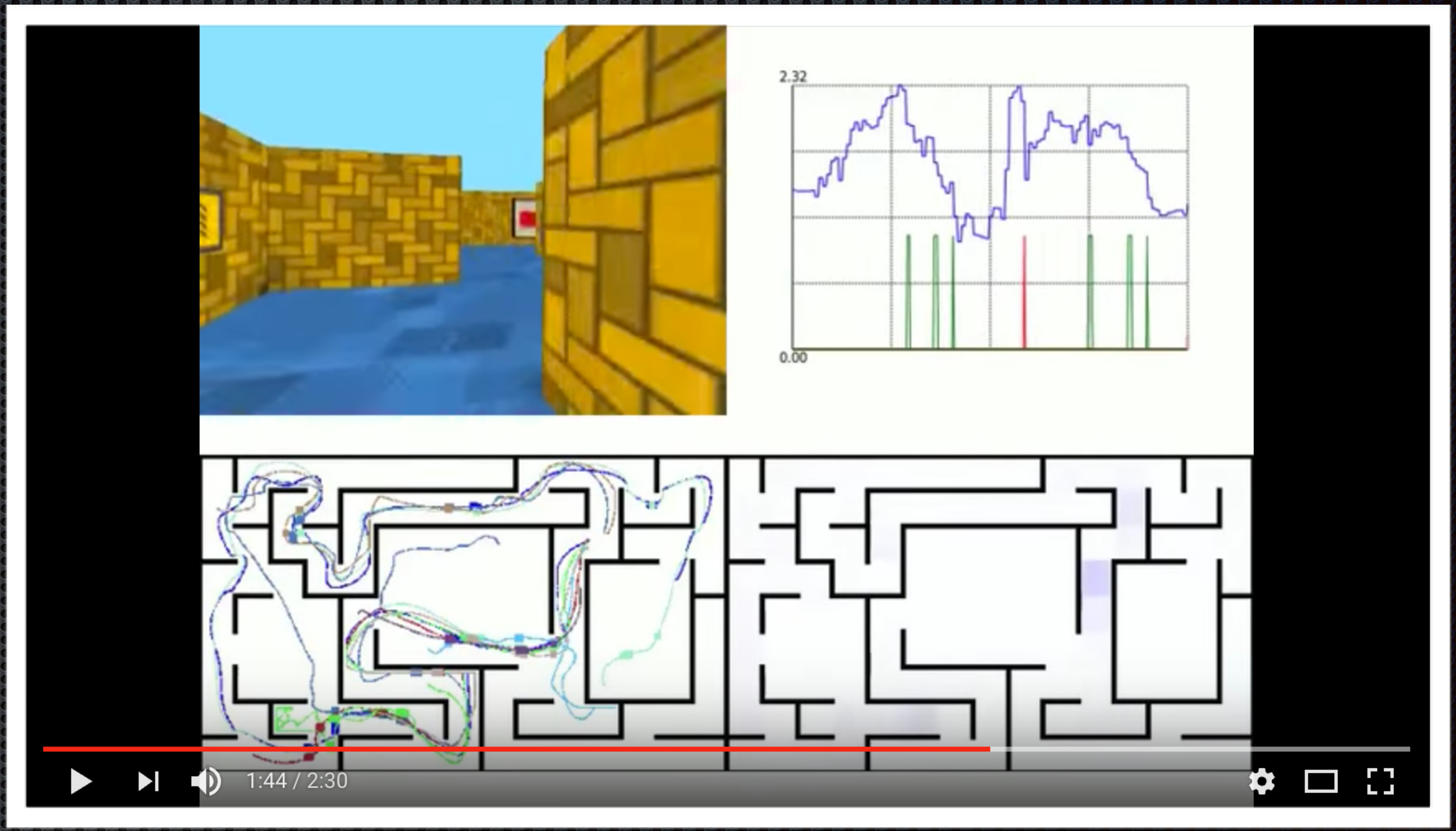
(Deep) Learning and Playing with Sequences

Piotr Mirowski, DeepMind



7 November 2016
London ML Meetup

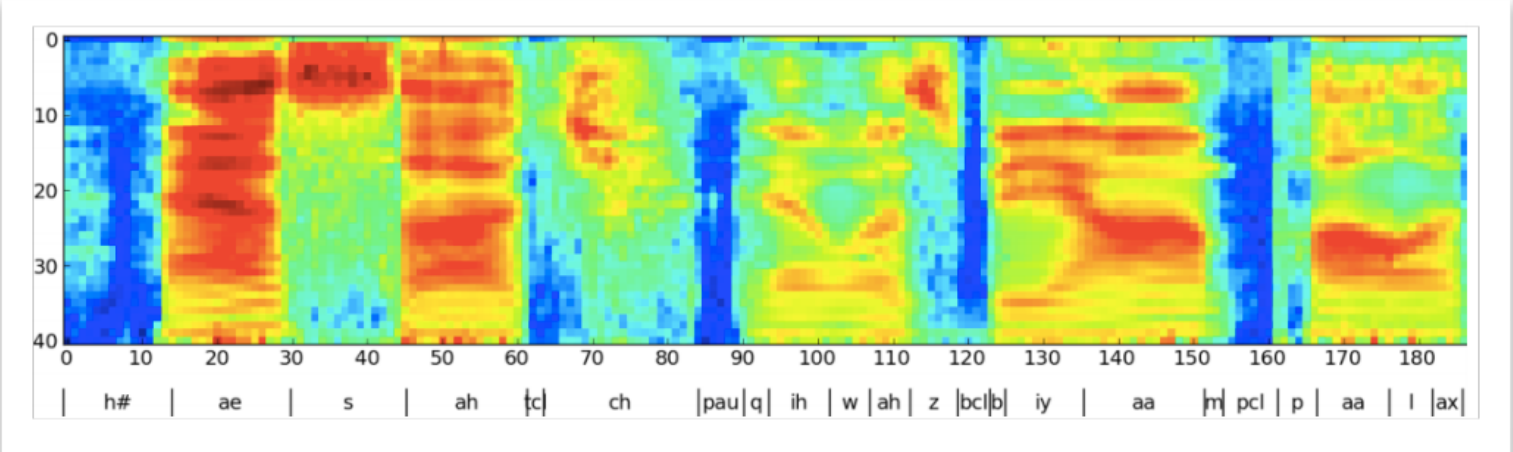
Quiz

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

Who wrote these lines?

A. William Shakespeare
B. William Shakespeare's ghostwriter
C. Ben Johnson
D. Molière (translation)
E. Andrej Karpathy's recurrent neural network

Why? (examples of applications)



[Graves et al. (2013b) "Speech recognition with deep recurrent neural networks", *ICASSP*]

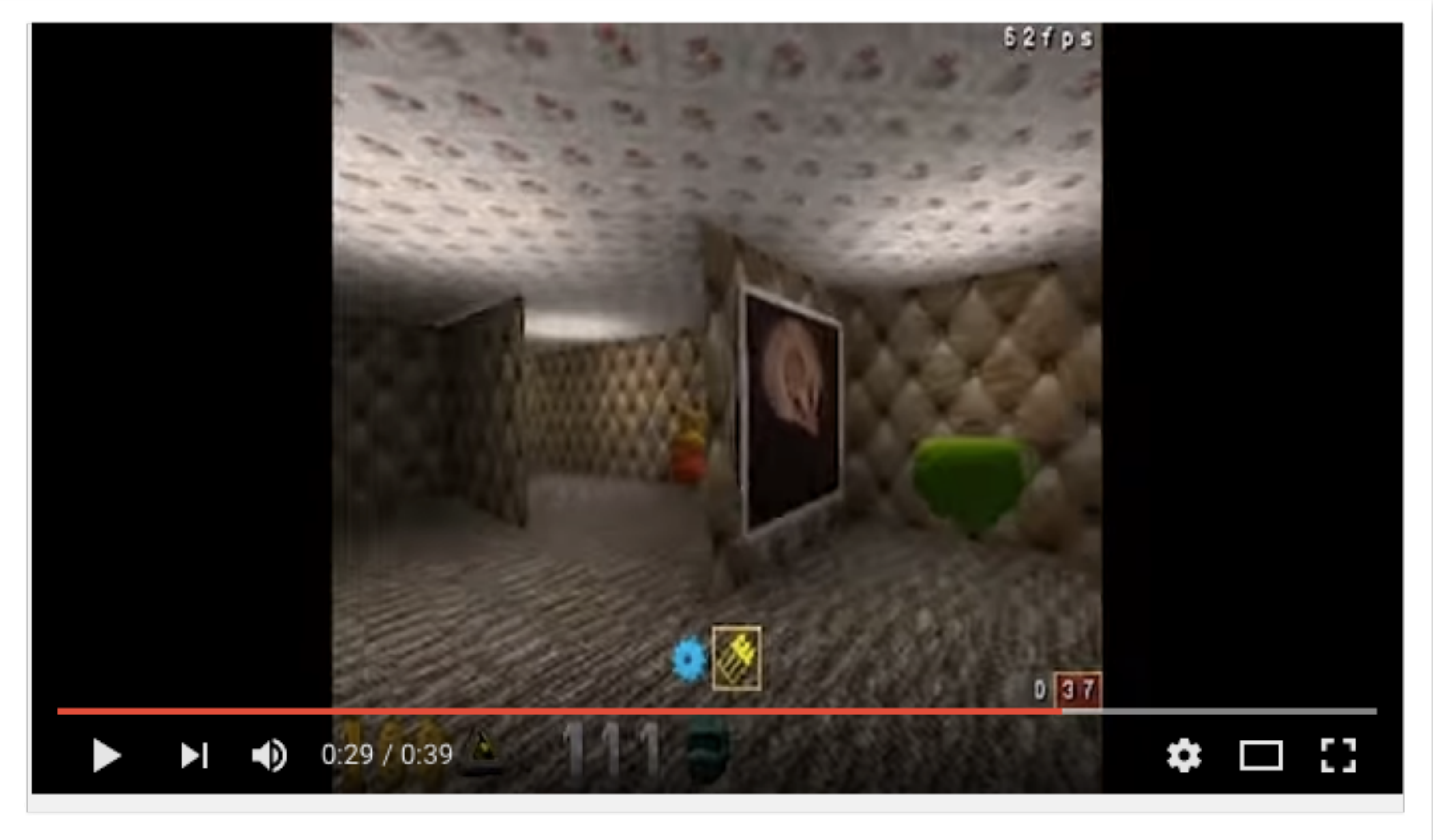
Language modeling

- Sentence completion
- Sentence-to-sentence machine translation
- Speech recognition
- Image captioning
- Text generation
- Query answering

[Image credits: Vinyals et al (2014)]

Control in 3D games

- Learning to navigate



How? (what this talk will cover)

Fixed-memory language models

n-grams and Markov chains

Learning representations

Word embeddings

Maximum likelihood learning

Neural language models

Recurrent Neural Networks (RNNs)

Long Short-Term Memory RNNs

Attention and memory models

Control through Reinforcement Learning

Language modeling

Sentence completion

Machine translation

Text generation

Speech recognition

Image captioning

Query answering

Playing 3D games

Learning to navigate

How? (what this talk will cover)

Fixed-memory language models

n-grams and Markov chains

Learning representations

Word embeddings

Maximum likelihood learning

Neural language models

Recurrent Neural Networks (RNNs)

Long Short-Term Memory RNNs

Attention and memory models

Control through Reinforcement Learning

Language modeling

Sentence completion

Machine translation

Text generation

Speech recognition

Image captioning

Query answering

Playing 3D games

Learning to navigate

Language models

Quantify, word by word, how likely is a sequence of words

$$P(w_1, w_2, \ldots, w_{T-1}, w_T) \approx \prod_{t=1}^{T} P(w_t | w_{t-1}, \ldots, w_{t-n+1})$$

Applications:

Speech recognition

Sentence completion

Sentence translation

Search query formulation

Question answering

what to cook with broccoli and _

what to cook with broccoli and **beef**

what to cook with broccoli and **butter**

what to cook with broccoli and **blenders**

what to cook with broccoli and **boomboxes**

```
the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...
```

Chain rule of probability

$$P(w_1, w_2, \ldots, w_{T-1}, w_T) = \prod_{t=1}^{T} P(w_t | w_{t-1}, w_{t-2}, \ldots, w_1)$$

the	cat	sat	on	the	mat	$P(w_1)$
the	**cat**	sat	on	the	mat	$P(w_2 \| w_1)$
the	cat	**sat**	on	the	mat	$P(w_3 \| w_2, w_1)$
the	cat	sat	**on**	the	mat	$P(w_4 \| w_3, w_2, w_1)$
the	cat	sat	on	**the**	mat	$P(w_5 \| w_4, w_3, w_2, w_1)$
the	cat	sat	on	the	**mat**	$P(w_6 \| w_5, w_4, w_3, w_2, w_1)$

n-grams and Markov chains

$$P(w_1, w_2, \ldots, w_{T-1}, w_T) \approx \prod_{t=1}^{T} P(w_t | w_{t-1}, \ldots, w_{t-n+1})$$

the	cat	sat	on	the	mat	$P(w_1)$
the	**cat**	sat	on	the	mat	$P(w_2 \| w_1)$
the	cat	**sat**	on	the	mat	$P(w_3 \| w_2, w_1)$
the	cat	sat	**on**	the	mat	$P(w_4 \| w_3, w_2)$
the	cat	sat	on	**the**	mat	$P(w_5 \| w_4, w_3)$
the	cat	sat	on	the	**mat**	$P(w_6 \| w_5, w_4)$

n-grams and conditional word probability

context					*target*	$P(w_t \| w_{t-1}, w_{t-2}, \ldots w_{t-5})$
the	cat	sat	on	the	**mat**	0.15
$w_{t\text{-}5}$	$w_{t\text{-}4}$	$w_{t\text{-}3}$	$w_{t\text{-}2}$	$w_{t\text{-}1}$	w_t	
the	cat	sat	on	the	**rug**	0.12
the	cat	sat	on	the	**hat**	0.09
the	cat	sat	on	the	**dog**	0.01
the	cat	sat	on	the	**the**	0
the	cat	sat	on	the	**sat**	0
the	cat	sat	on	the	**robot**	?
the	cat	sat	on	the	**printer**	?

Limitations of *n*-gram language models

No memory **beyond *n* words** (e.g., this sentence generated by Claude Shannon):

"The head and frontal attack on an English writer that the character of this point is therefore another method for the letters that the time of whoever told the problem for an unexpected…"

Curse of dimensionality:
n-grams need exponential number of examples for a vocabulary of V words:
V^n possible n-grams

No notion of word similarity

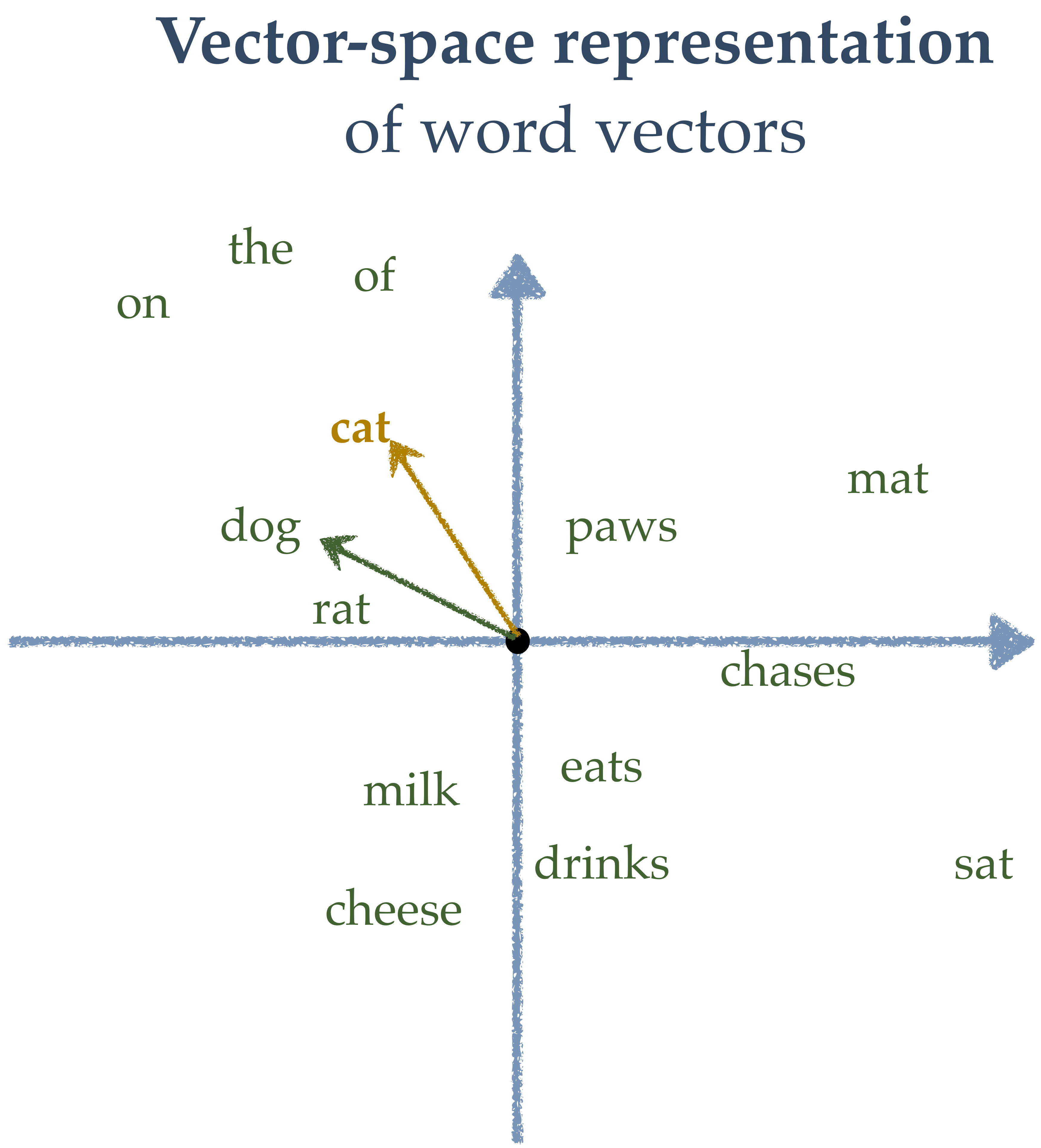
Solution: **word embeddings-based n-grams**

How? (what this talk will cover)

Fixed-memory language models
n-grams and Markov chains
Learning representations
Word embeddings
Maximum likelihood learning
Neural language models
Recurrent Neural Networks (RNNs)
Long Short-Term Memory RNNs
Attention and memory models
Control through Reinforcement Learning

Language modeling
Sentence completion
Machine translation
Text generation
Speech recognition
Image captioning
Query answering
Playing 3D games
Learning to navigate

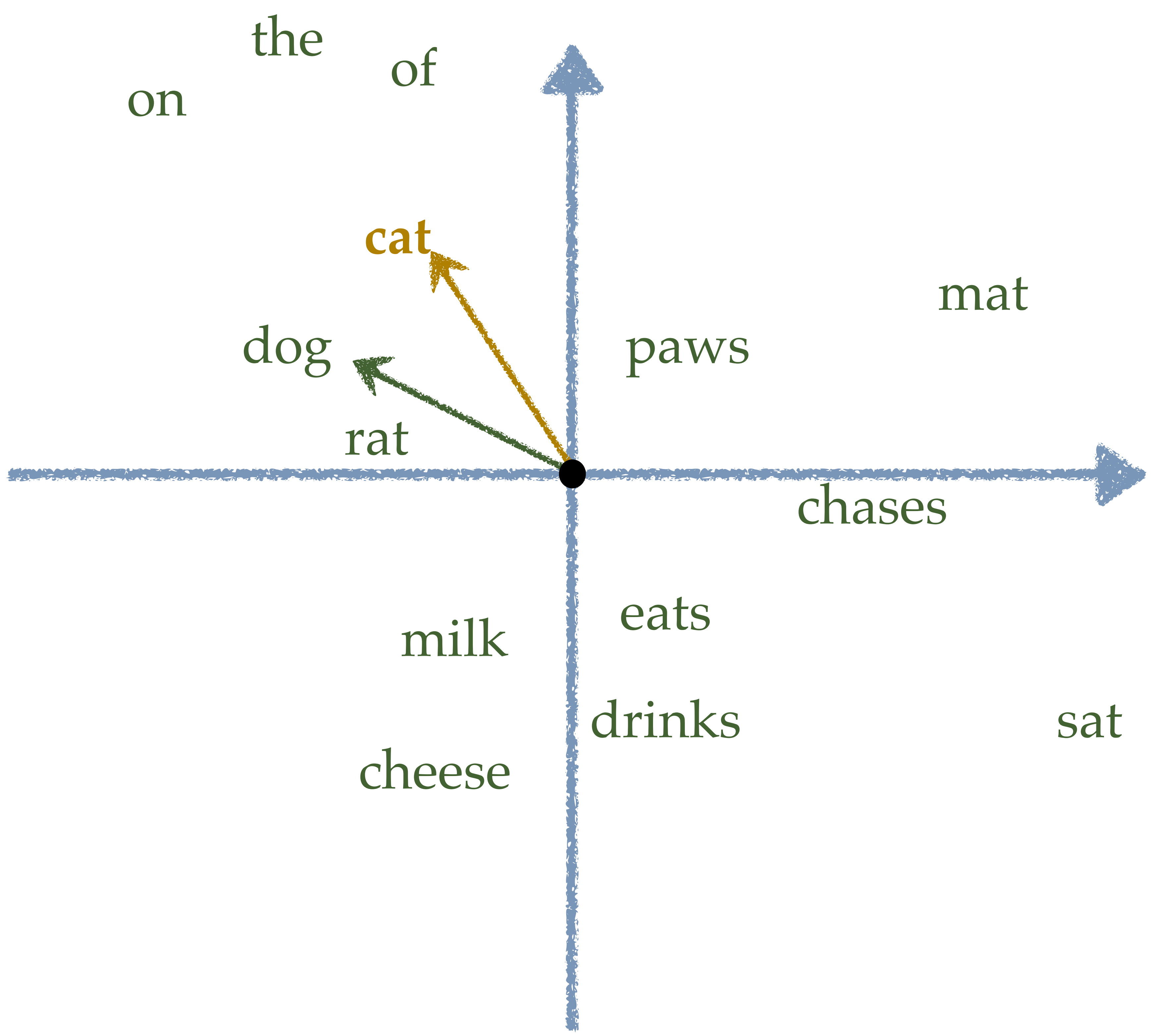
Words as vectors



We will learn these word vector representations from data

Similarity between word vectors

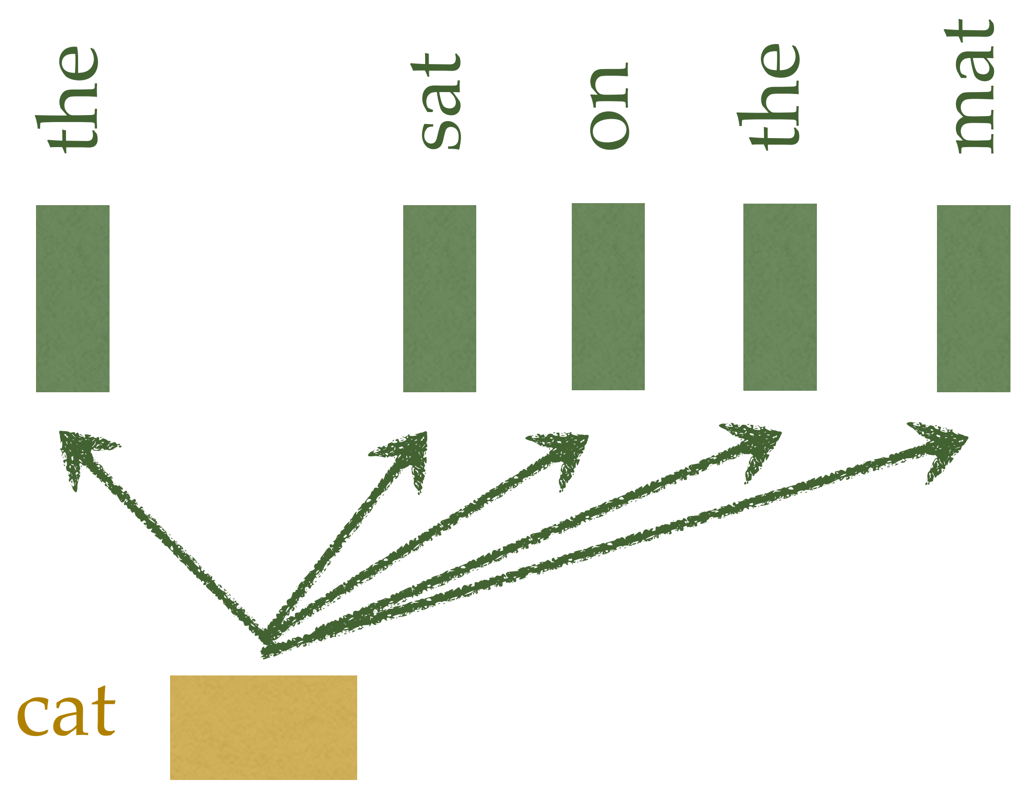
Vector-space representation
of word vectors



Vector-space **cosine similarity**
between words w and v

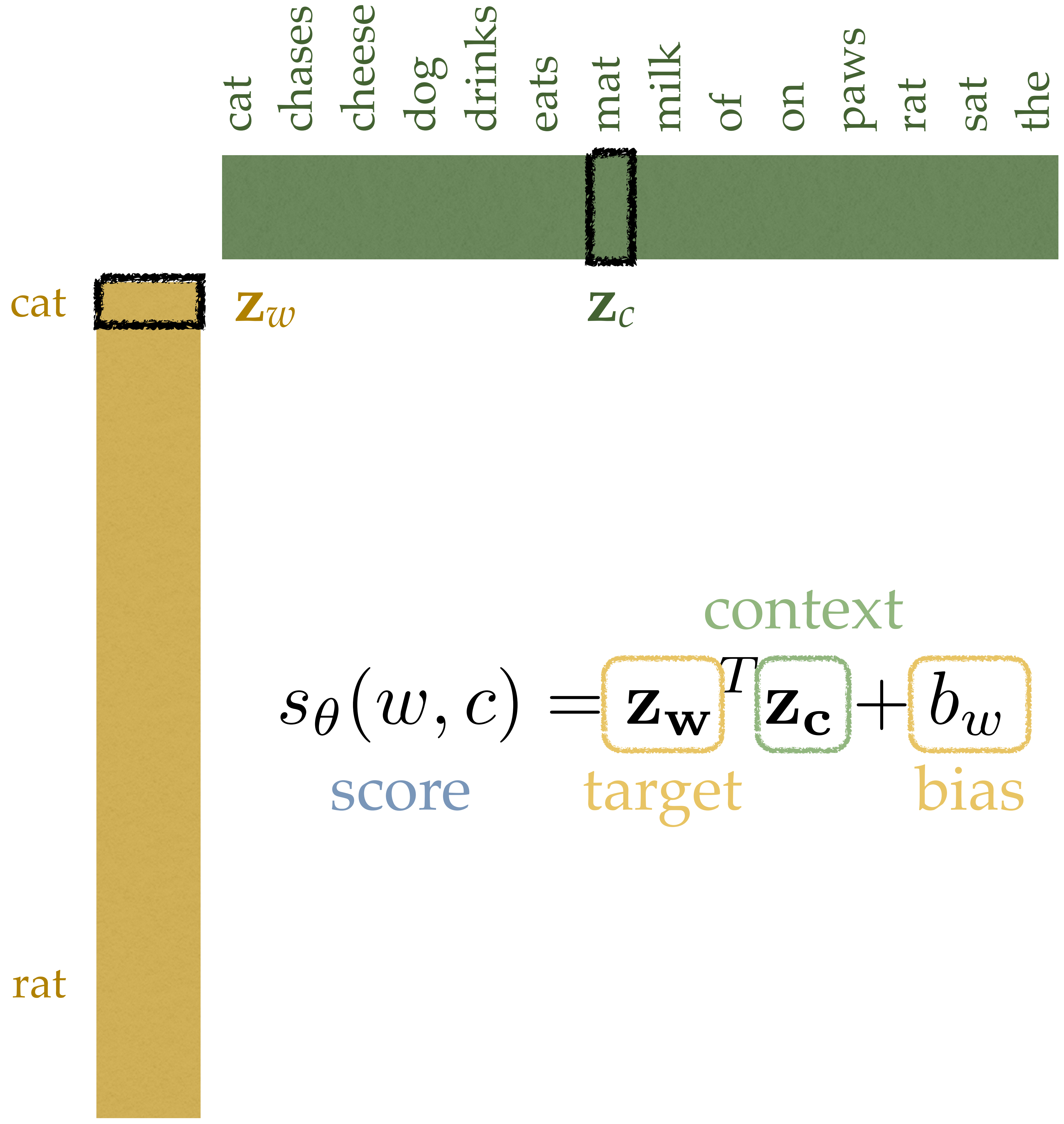
$$\cos(w, v) = \frac{\mathbf{z_w}^T \mathbf{z_v}}{||\mathbf{z_w}||_2 ||\mathbf{z_v}||_2}$$

the cat sat on the mat



Word embedding

[Andriy Mnih and Koray Kavukcuoglu (2013) “Learning word embeddings efficiently with noise-contrastive estimation“, *NIPS*; Tomas Mikolov et al. (2013a) “Efficient Estimation of Word Representation in Vector Space“, *arXiv*; Tomas Mikolov et al. (2013b) “Distributed Representation of Words and Phrases and Their Compositionality“, *NIPS*]



the cat sat on the mat

Word embedding vectors have size D much smaller than vocabulary V

$$\underbrace{s_\theta(w, c)}_{\text{score}} = \underbrace{\boxed{\mathbf{z_w}}}_{\text{target}}^T \overbrace{\boxed{\mathbf{z_c}}}^{\text{context}} + \underbrace{\boxed{b_w}}_{\text{bias}}$$

Learn word embeddings **iteratively** using pairs (w, c)

Learn context-dependent word probability

Learn model (e.g., word embeddings)
parameterized by θ, so that:

word context

correct answer

"softmax" $$P(w|c) = \frac{e^{\boxed{s_\theta(w,c)}}}{\boxed{\sum_{v=1}^{V} e^{s_\theta(v,c)}}}$$

normalization term

Maximum likelihood learning

Stochastic gradient ascent (or descent):
after showing each pair (word w, context c),
update the parameters θ

$$\theta \leftarrow \theta + \eta \frac{\partial L(w, c; \theta)}{\partial \theta}$$

maximize $\log P(w|c) = \underbrace{\boxed{s_\theta(w, c)}}_{\text{correct answer}} - \boxed{\log \sum_{v=1}^{V} e^{s_\theta(v,c)}}$

Learn context-dependent word probability

high-dimensional
normalization term
(e.g., *V*>100k words)

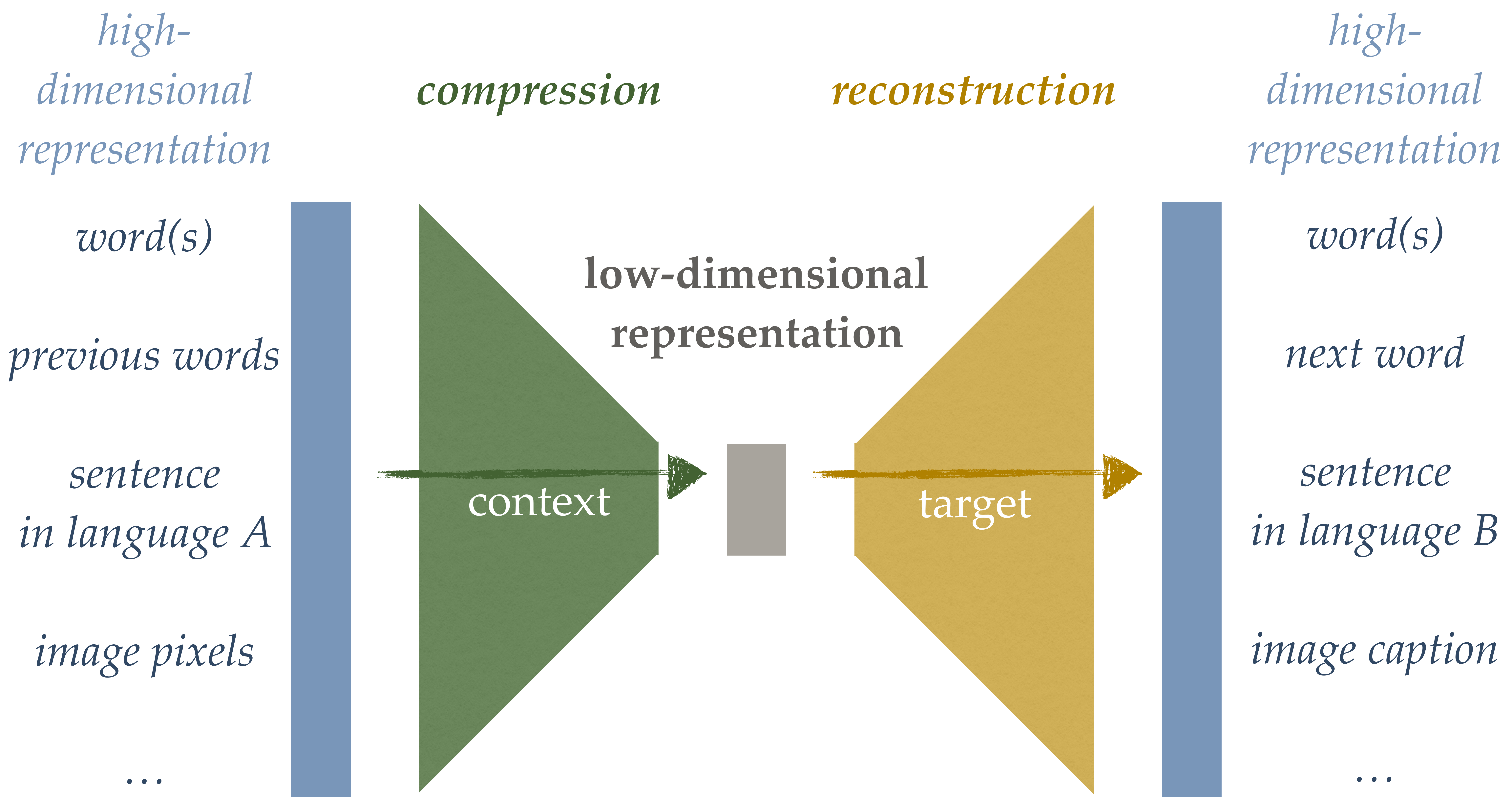
$$P(w|c) = \frac{e^{s_\theta(w,c)}}{\boxed{\sum_{v=1}^{V} e^{s_\theta(v,c)}}}$$

normalization term

Solution #1:
approximate
normalisation term

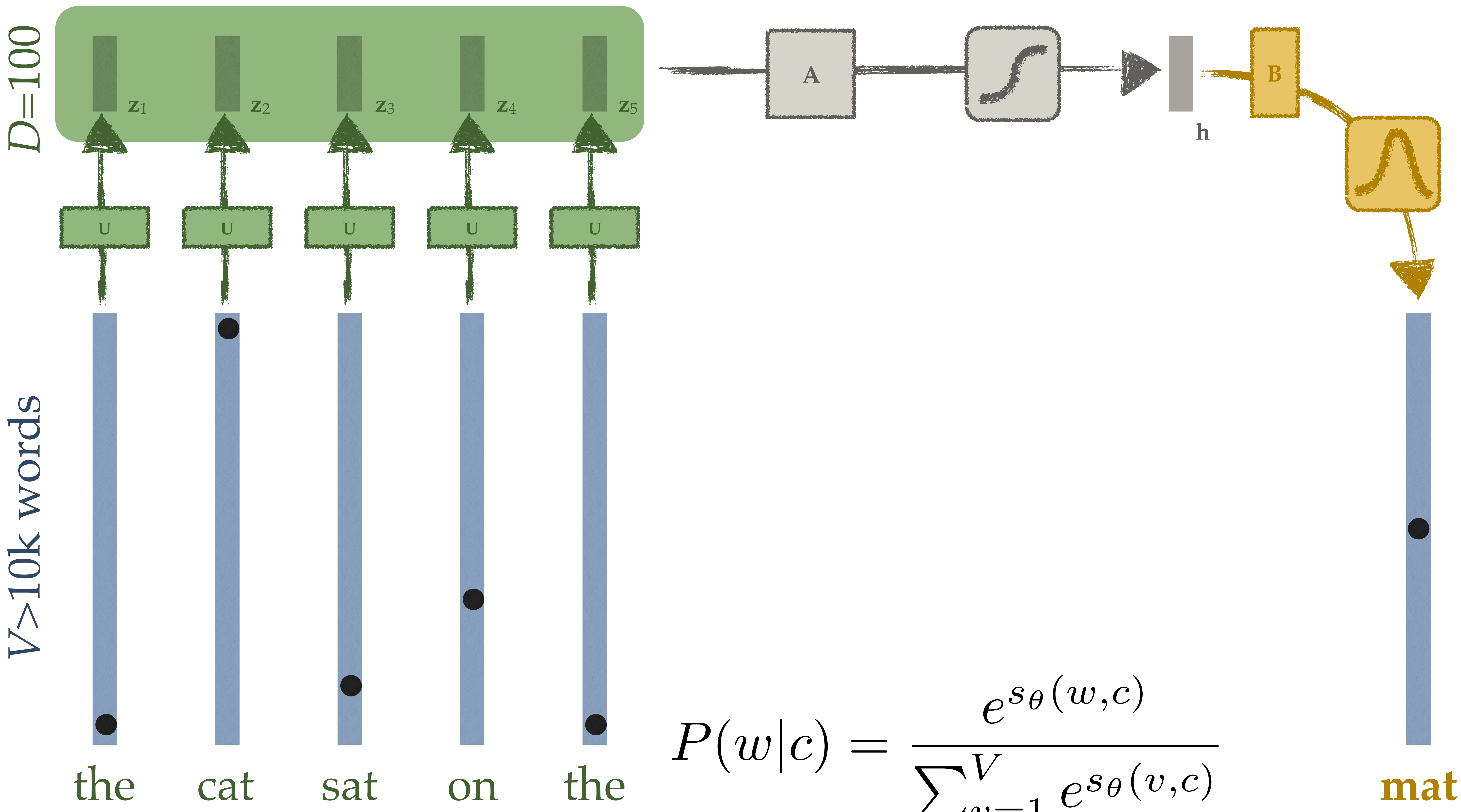
Solution #2:
parallelise
on a GPU

Dimensionality reduction



Neural Probabilistic Language Models

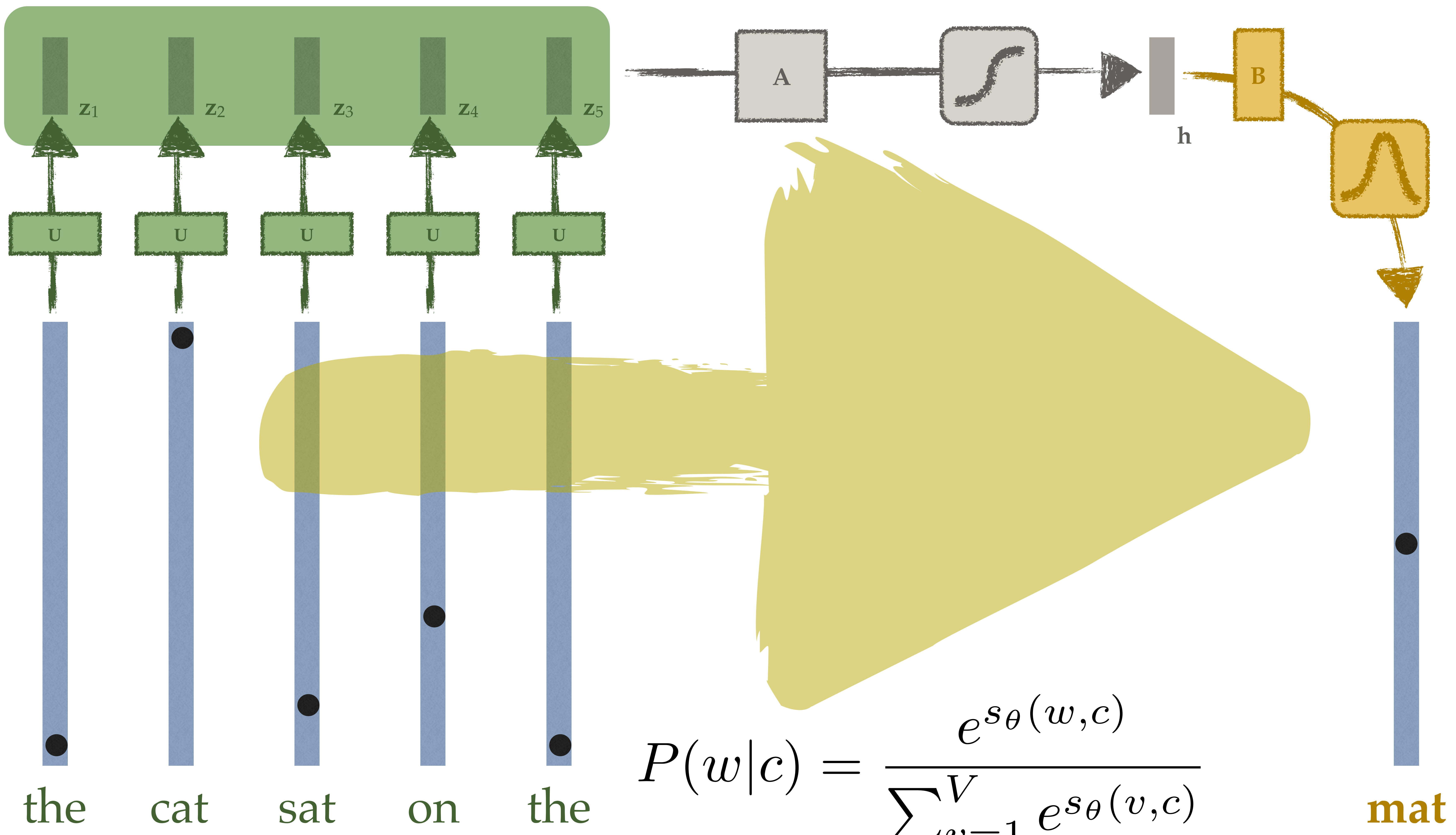
[Yoshua Bengio et al. (2001, 2003), "A Neural Probabilistic Language Model", *JMLR*;
Andriy Mnih and Geoff Hinton, "Three new graphical models for statistical language modeling", *ICML*]



$$P(w|c) = \frac{e^{s_\theta(w,c)}}{\sum_{v=1}^{V} e^{s_\theta(v,c)}}$$

Learning LMs: forward propagation

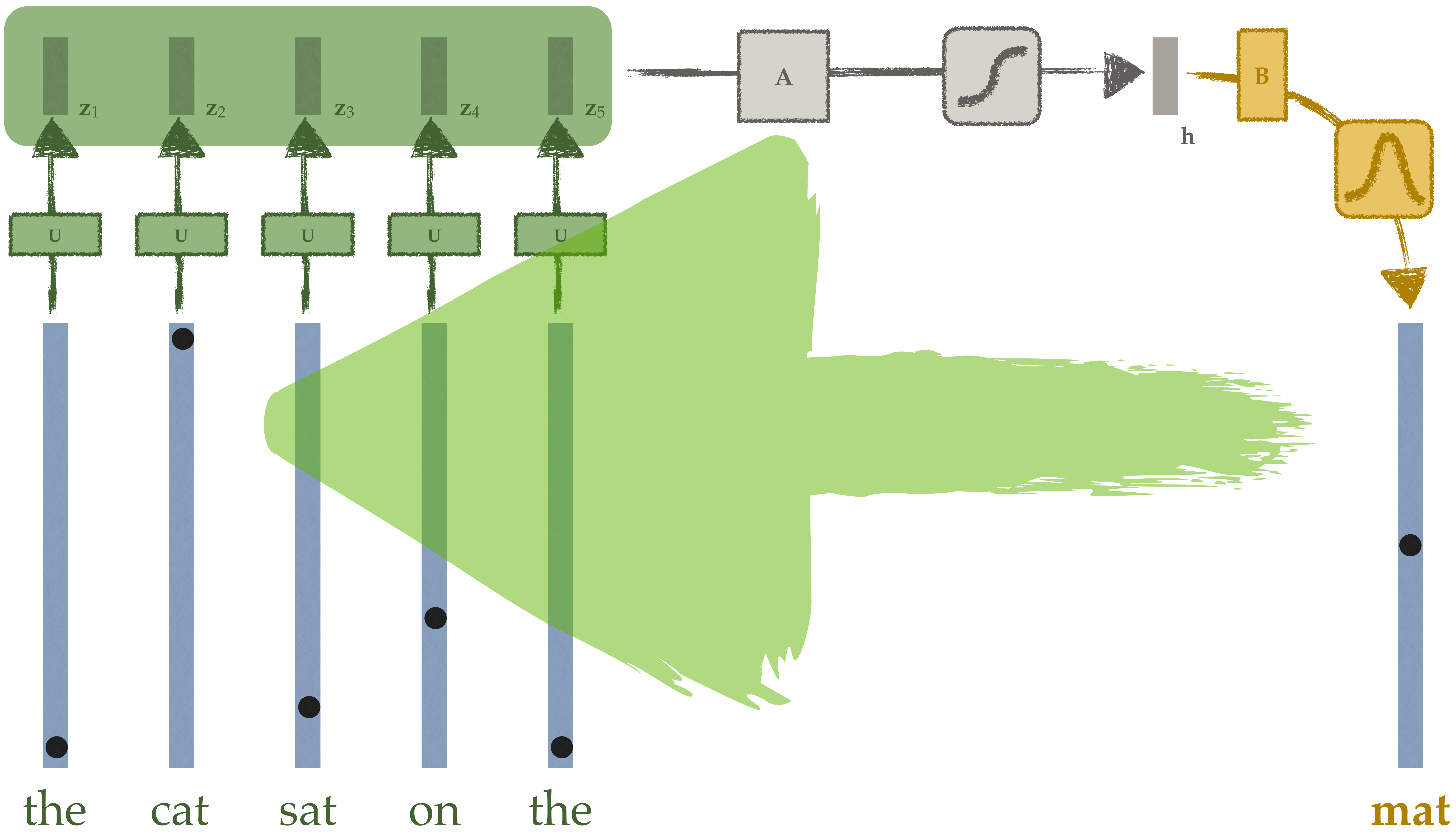
[Yoshua Bengio et al. (2001, 2003), "A Neural Probabilistic Language Model", *JMLR*;
Andriy Mnih and Geoff Hinton, "Three new graphical models for statistical language modeling", *ICML*]



$$P(w|c) = \frac{e^{s_\theta(w,c)}}{\sum_{v=1}^{V} e^{s_\theta(v,c)}}$$

Learning LMs: gradient back-propagation

[Yoshua Bengio et al. (2001, 2003), “A Neural Probabilistic Language Model”, *JMLR*;
Andriy Mnih and Geoff Hinton, “Three new graphical models for statistical language modeling”, *ICML*]



How? (what this talk will cover)

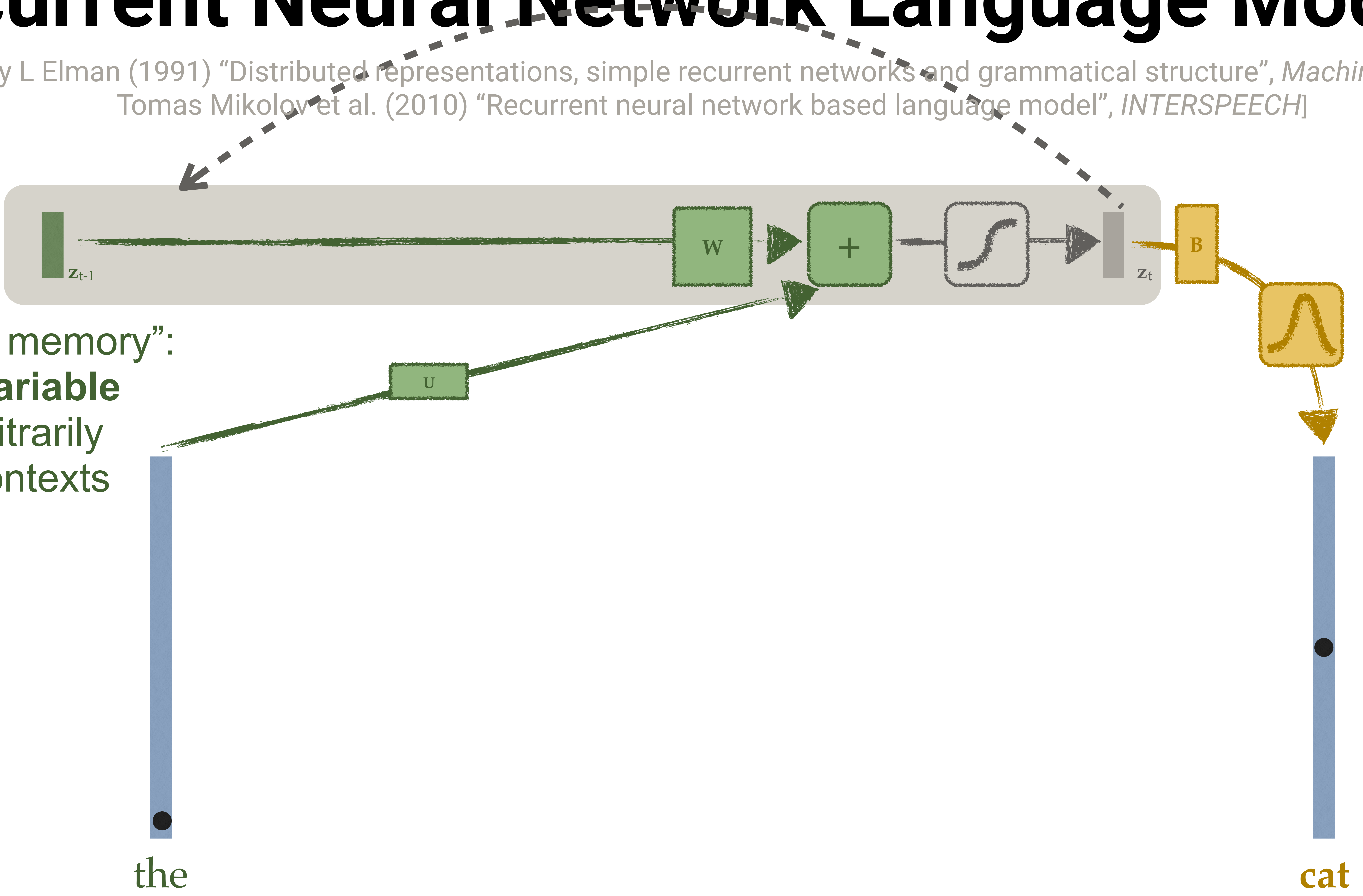
Fixed-memory language models
n-grams and Markov chains
Learning representations
Word embeddings
Maximum likelihood learning
Neural language models
Recurrent Neural Networks (RNNs)
Long Short-Term Memory RNNs
Attention and memory models
Control through Reinforcement Learning

Language modeling
Sentence completion
Machine translation
Text generation
Speech recognition
Image captioning
Query answering
Playing 3D games
Learning to navigate

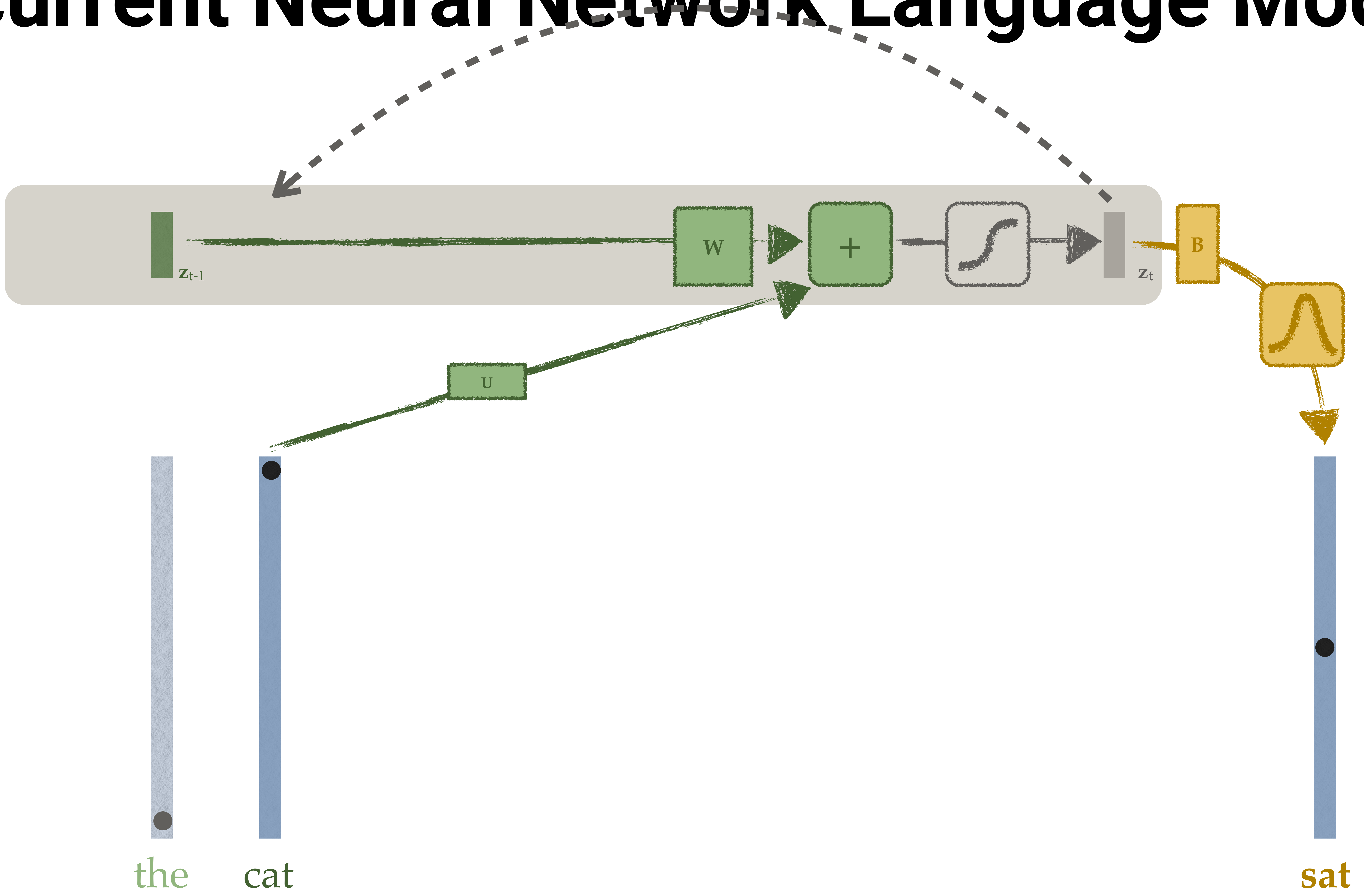
Recurrent Neural Network Language Models

[Jeffrey L Elman (1991) "Distributed representations, simple recurrent networks and grammatical structure", *Machine Learning*; Tomas Mikolov et al. (2010) "Recurrent neural network based language model", *INTERSPEECH*]

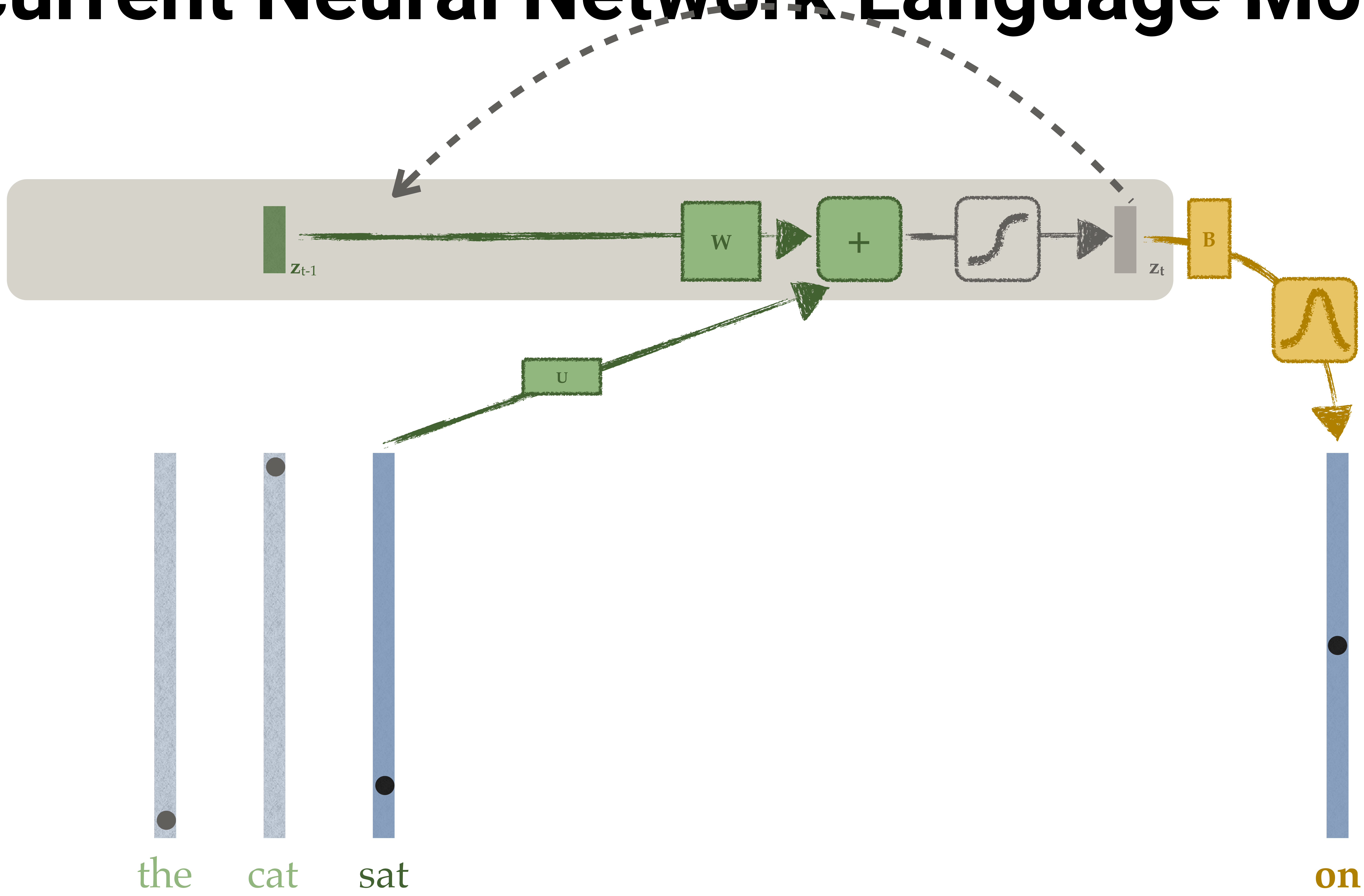
"persistent memory": **state variable** for arbitrarily long contexts



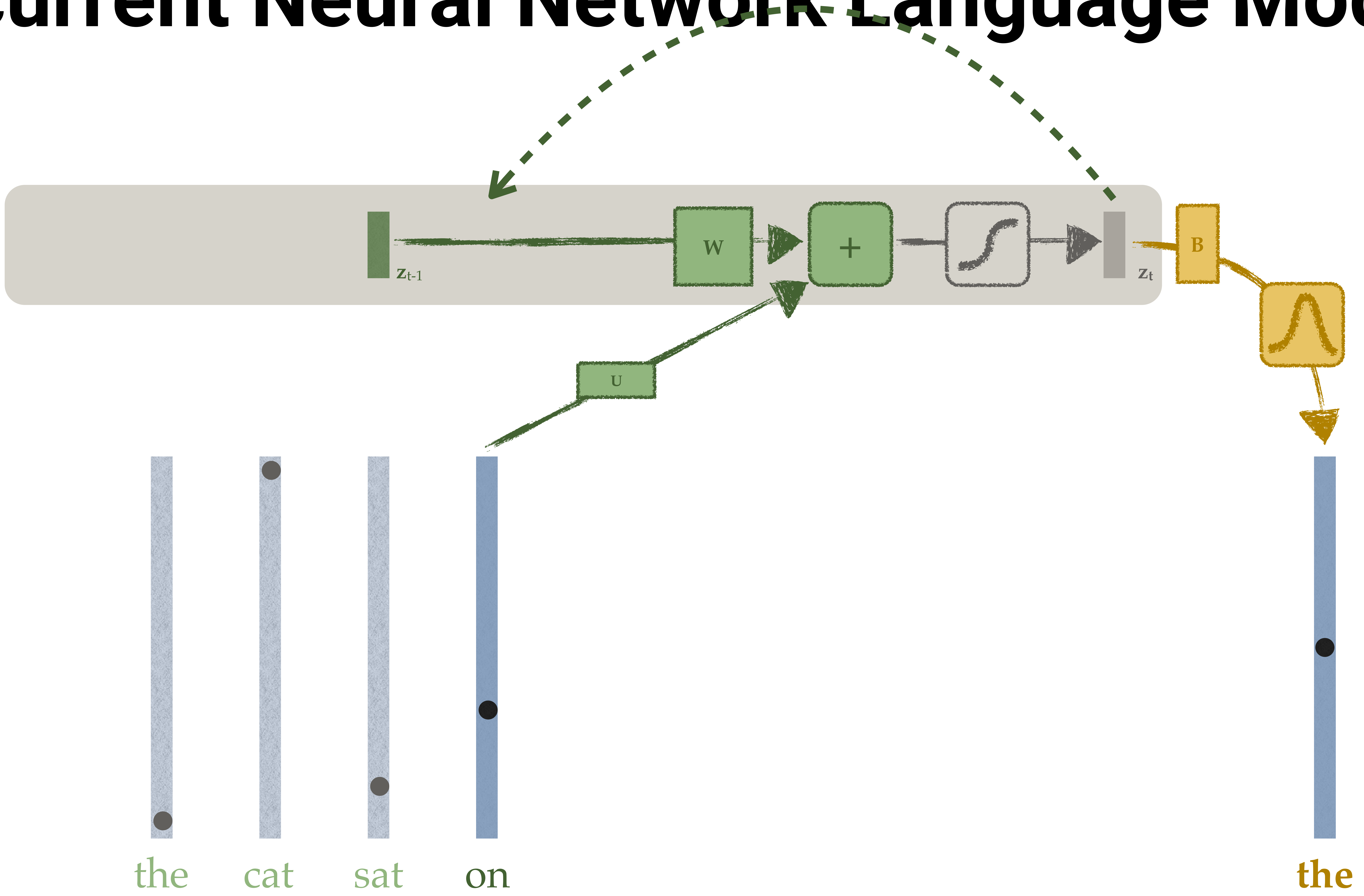
Recurrent Neural Network Language Models



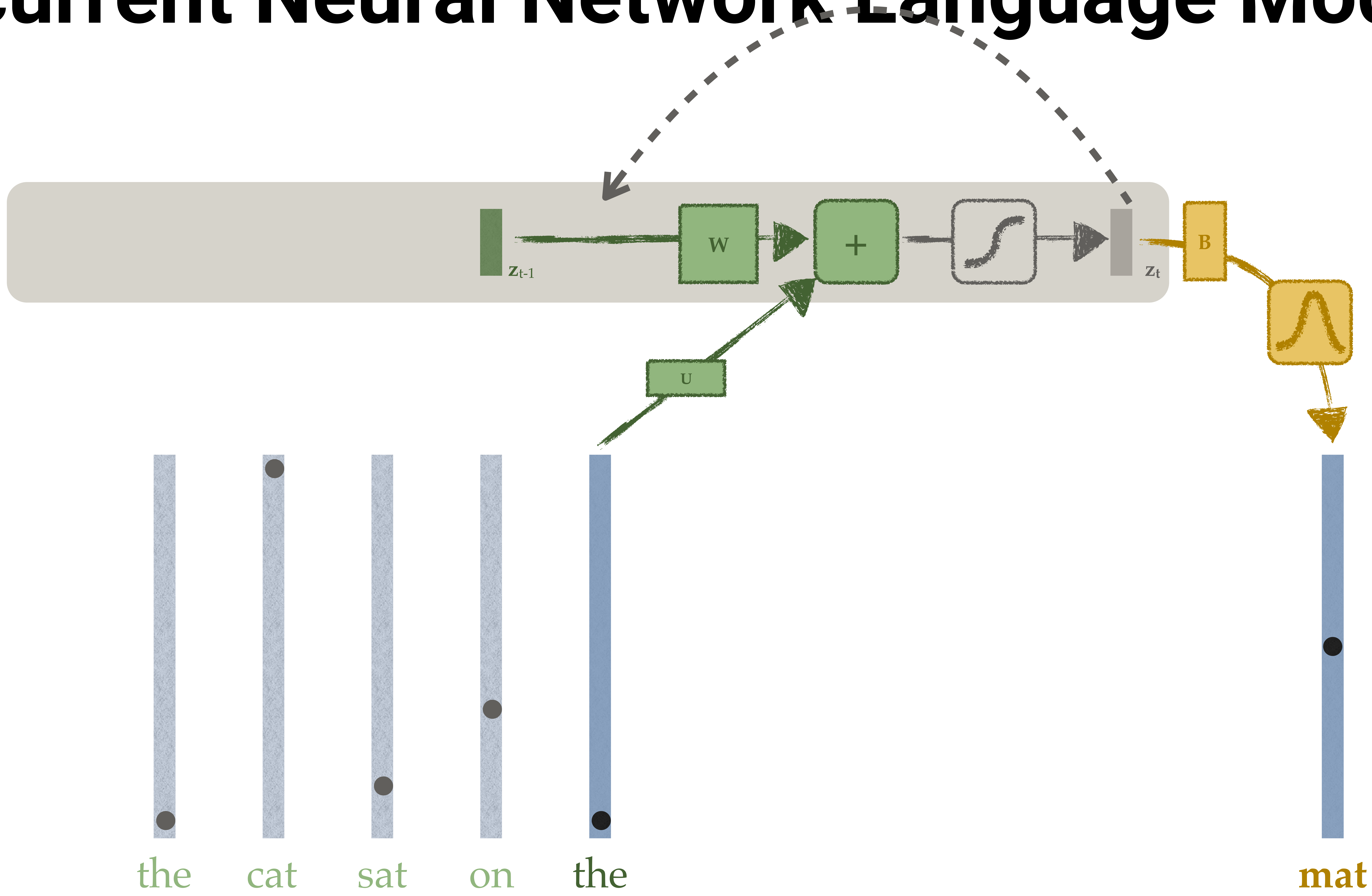
Recurrent Neural Network Language Models



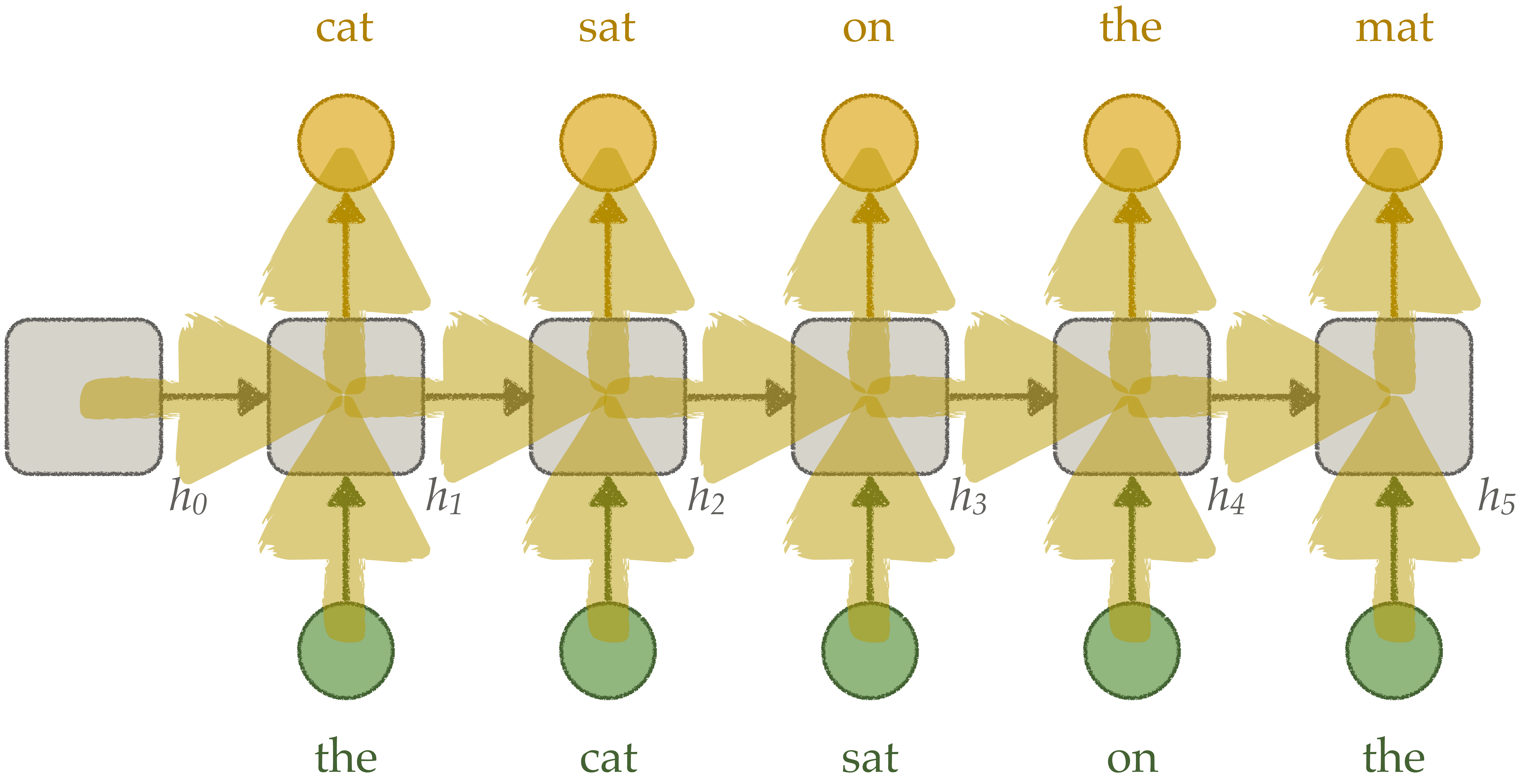
Recurrent Neural Network Language Models



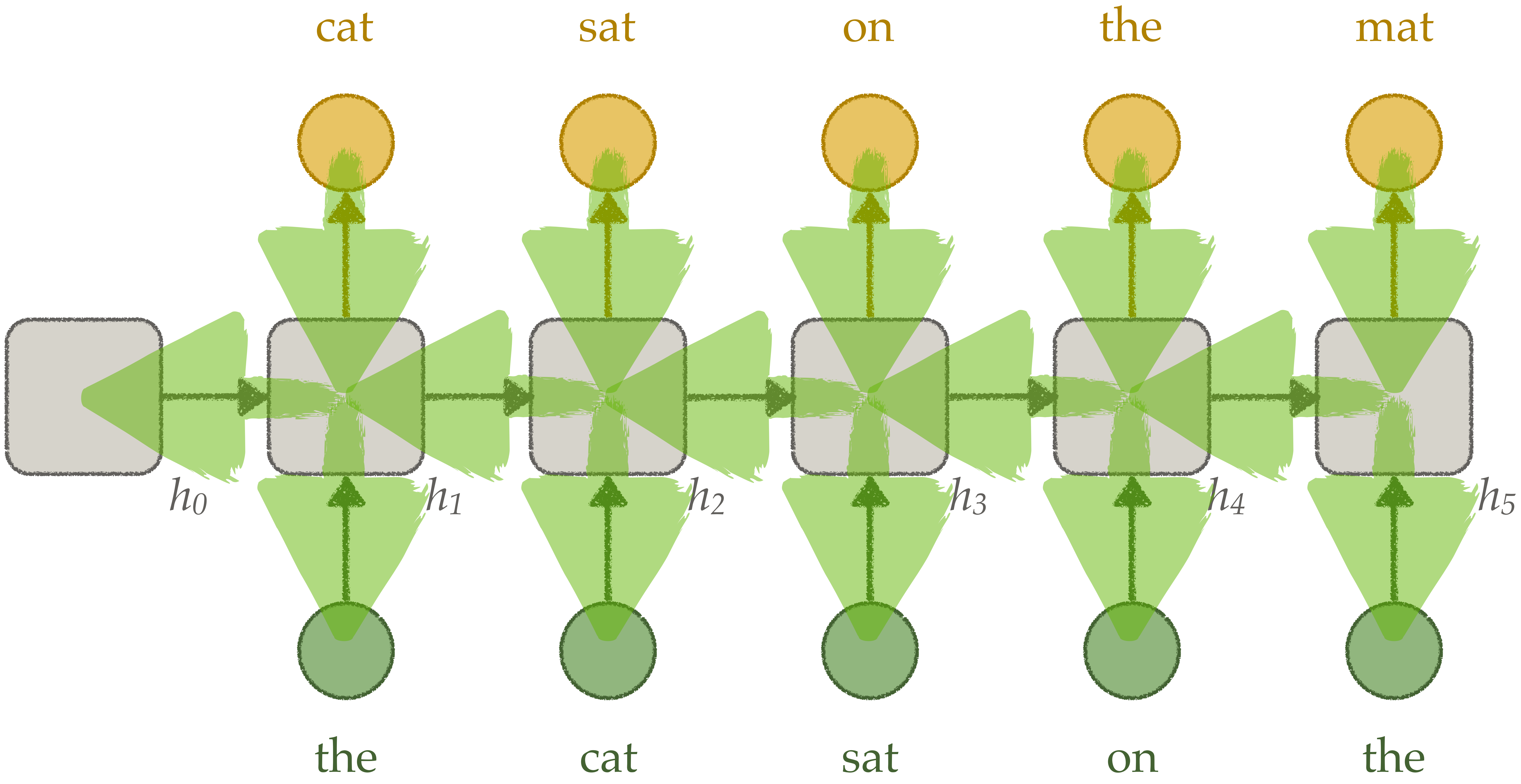
Recurrent Neural Network Language Models



Recurrent Neural Network Language Models



Back-Propagation Through Time



Sentence completion task

Microsoft Research Sentence Completion Task

[Geoff Zweig and Chris Burges (2011), "The Microsoft Research Sentence Completion Challenge", MSR Technical Report]

Training set:

~520 novels (19th century)

48M words

Evaluation on 1024 sentences

From 5 Sherlock Holmes novels

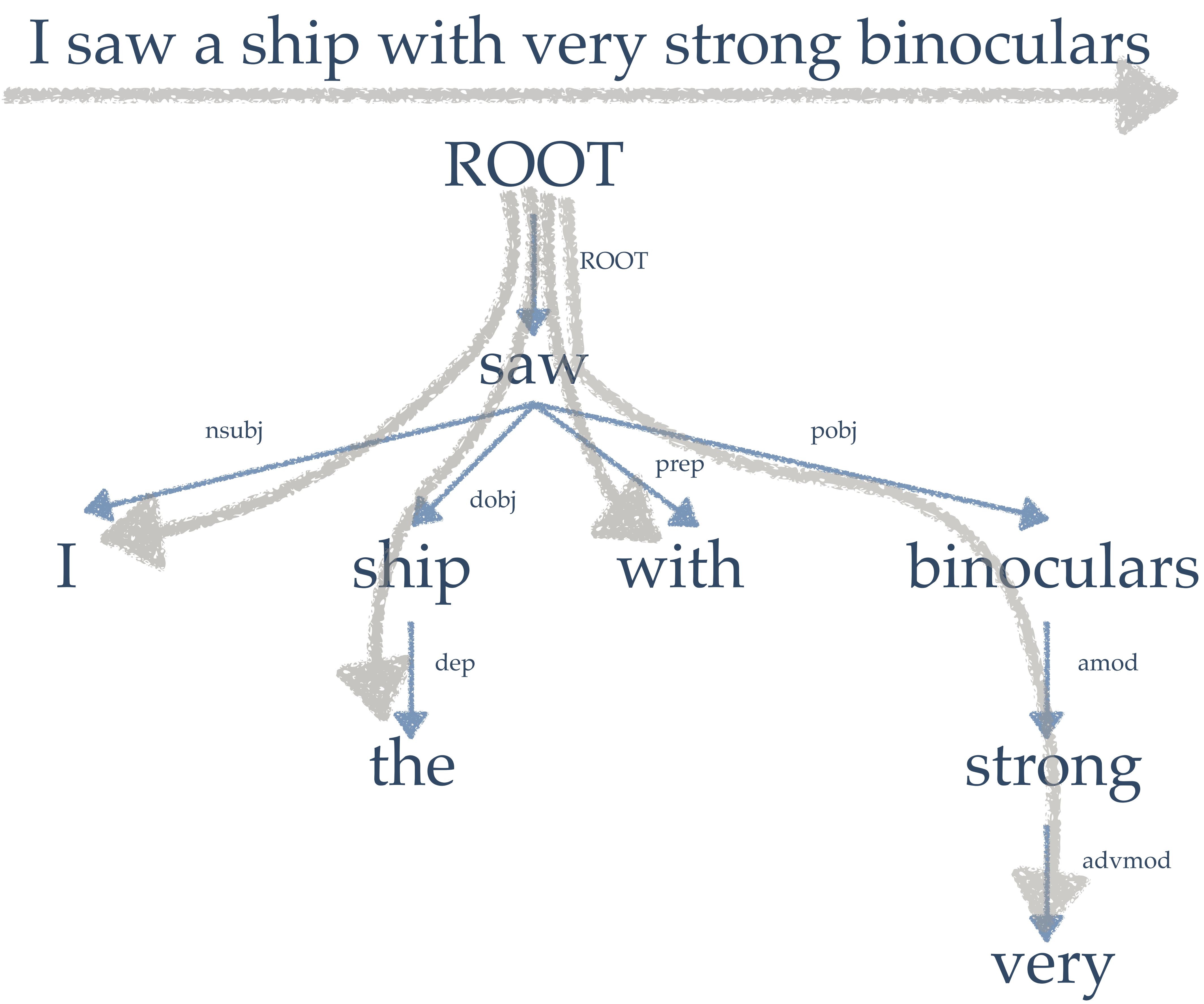
1 missing word, 5 choices:

1 ground truth

4 grammatically correct impostors

```
That is his generous   fault, but on the whole he's a good worker.
That is his mother's   fault, but on the whole he's a good worker.
That is his successful fault, but on the whole he's a good worker.
That is his main       fault, but on the whole he's a good worker.
That is his favourite  fault, but on the whole he's a good worker.
```

Beyond sequential: tree-based RNNs



Algorithm	Accuracy (test set)
random	20%
SVD (word-paragraph)	49%
skip-gram	48%
smoothed 4-gram	39%
RNN + 4-gram features	45%
RNN on dependency tree	**53%**
Long Short-Term Memory	63%
human	90%

[Piotr Mirowski and Andreas Vlachos (2015) “Dependency recurrent neural language models for sentence completion”, *ACL*;
Kai Sheng et al. (2015) “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks”, *ACL*;
Xiaodan Zhu et al. (2015) “Long Short-Term Memory Over Recursive Structures”, *ICML*]

How? (what this talk will cover)

Fixed-memory language models
n-grams and Markov chains
Learning representations
Word embeddings
Maximum likelihood learning
Neural language models
Recurrent Neural Networks (RNNs)
Long Short-Term Memory RNNs
Attention and memory models
Control through Reinforcement Learning

Language modeling
Sentence completion
Machine translation
Text generation
Speech recognition
Image captioning
Query answering
Playing 3D games
Learning to navigate

End-to-end natural language processing

One **integrated** algorithm for:

Speech recognition from **acoustic vectors** to **text**

Machine translation from **one language** to **another**

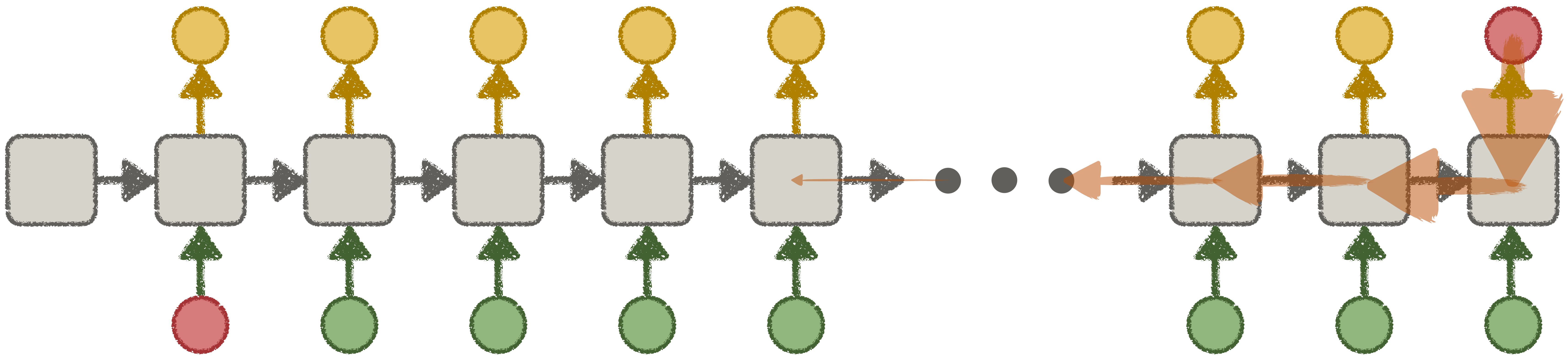
Image captioning from **image** to **text**

[Image credits: Vinyals et al (2014)]

Learning long-range dependencies…

… is difficult for Recurrent Neural Networks

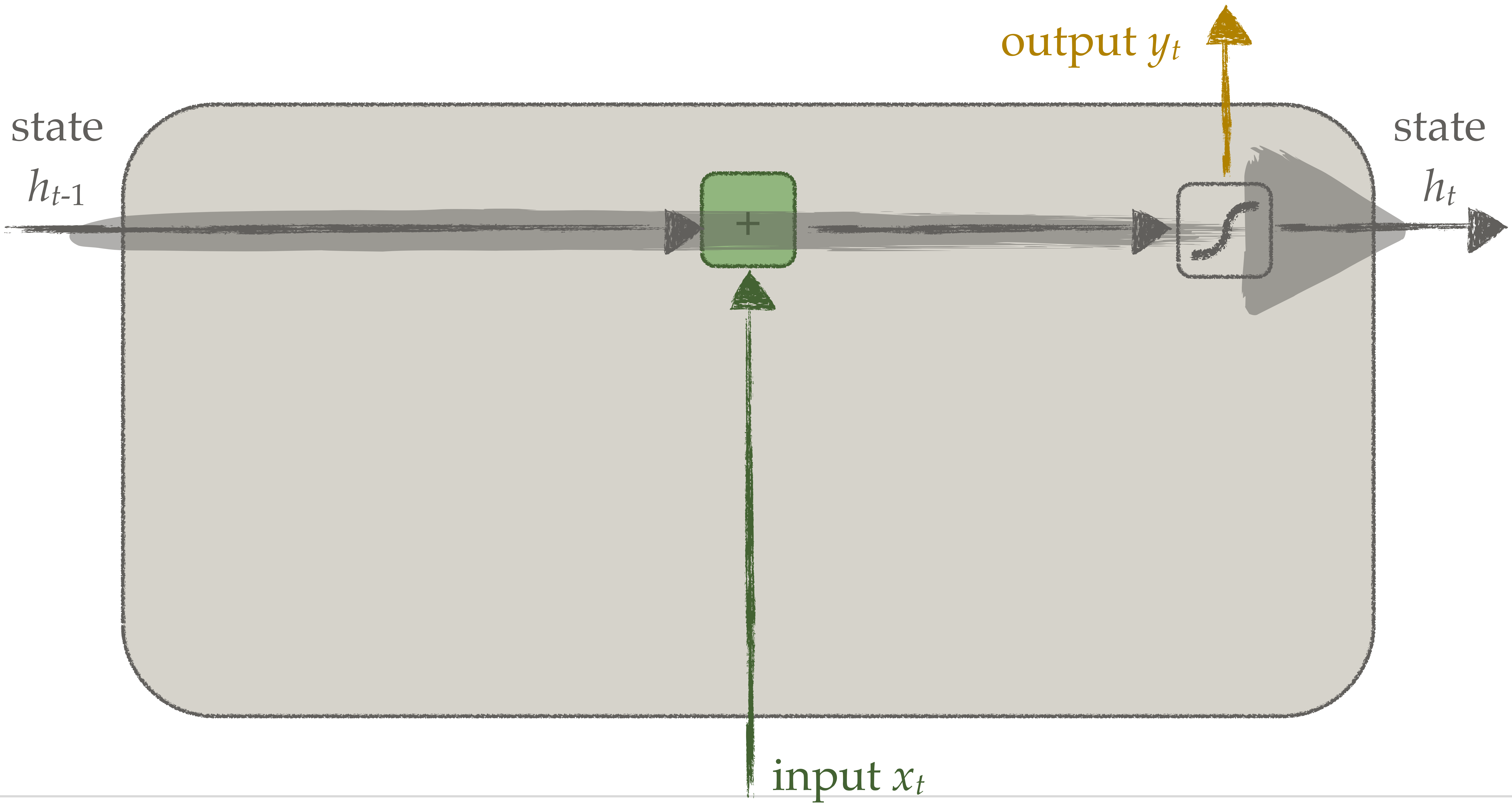
(and *n*-grams cannot retain information beyond *n* steps)



Because of the **non-linearity** in the hidden units,
gradients of the error during back-propagation
decay **exponentially** with the length of the sequence

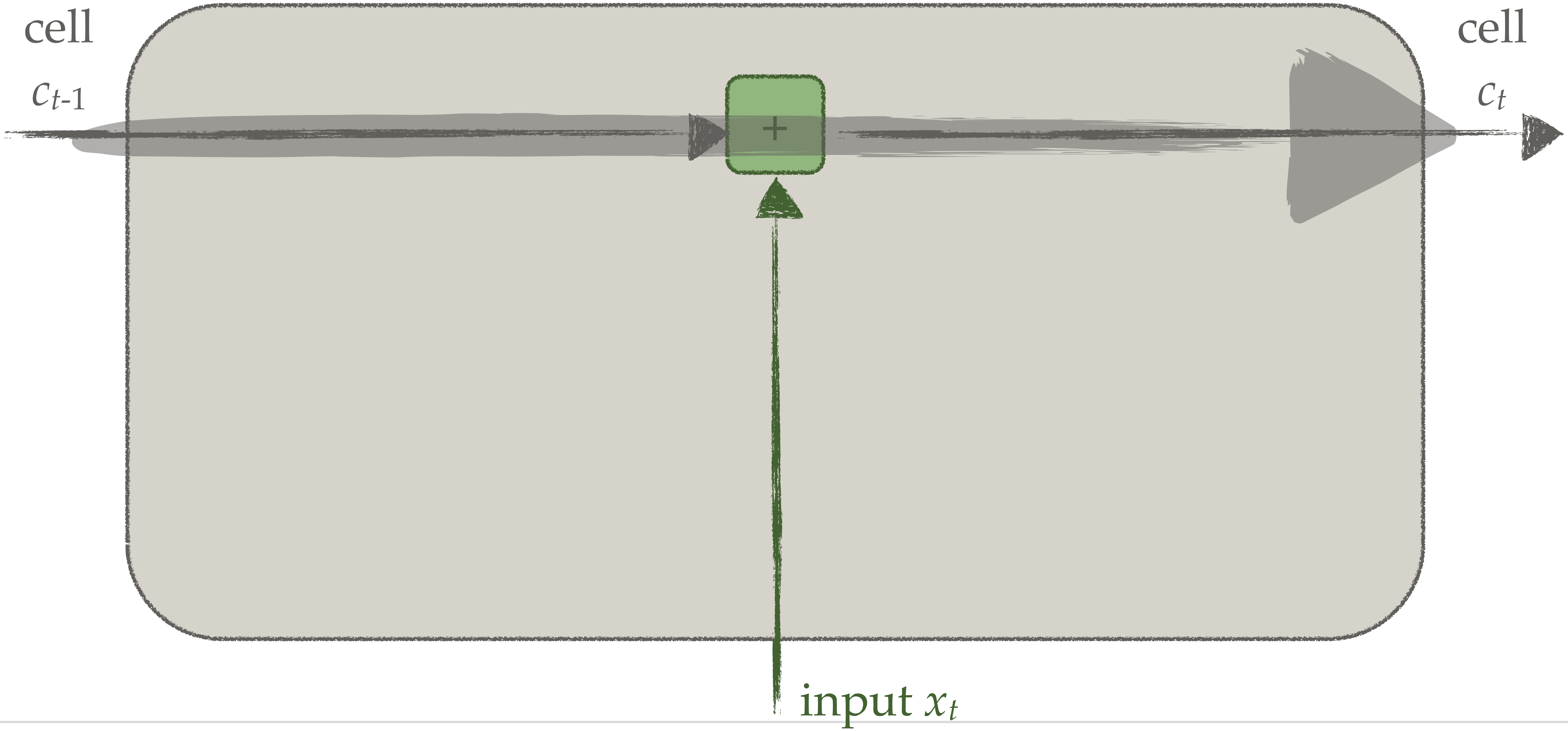
[Sepp Hochreiter (1991) "Untersuchungen zu dynamischen neuronalen Netzen", *Diploma TUM*;
Yoshua Bengio et al. (1994) "Learning Long-Term Dependencies with Gradient Descent is Difficult", *IEEE Transactions on Neural Networks*]

Recurrent Neural Networks



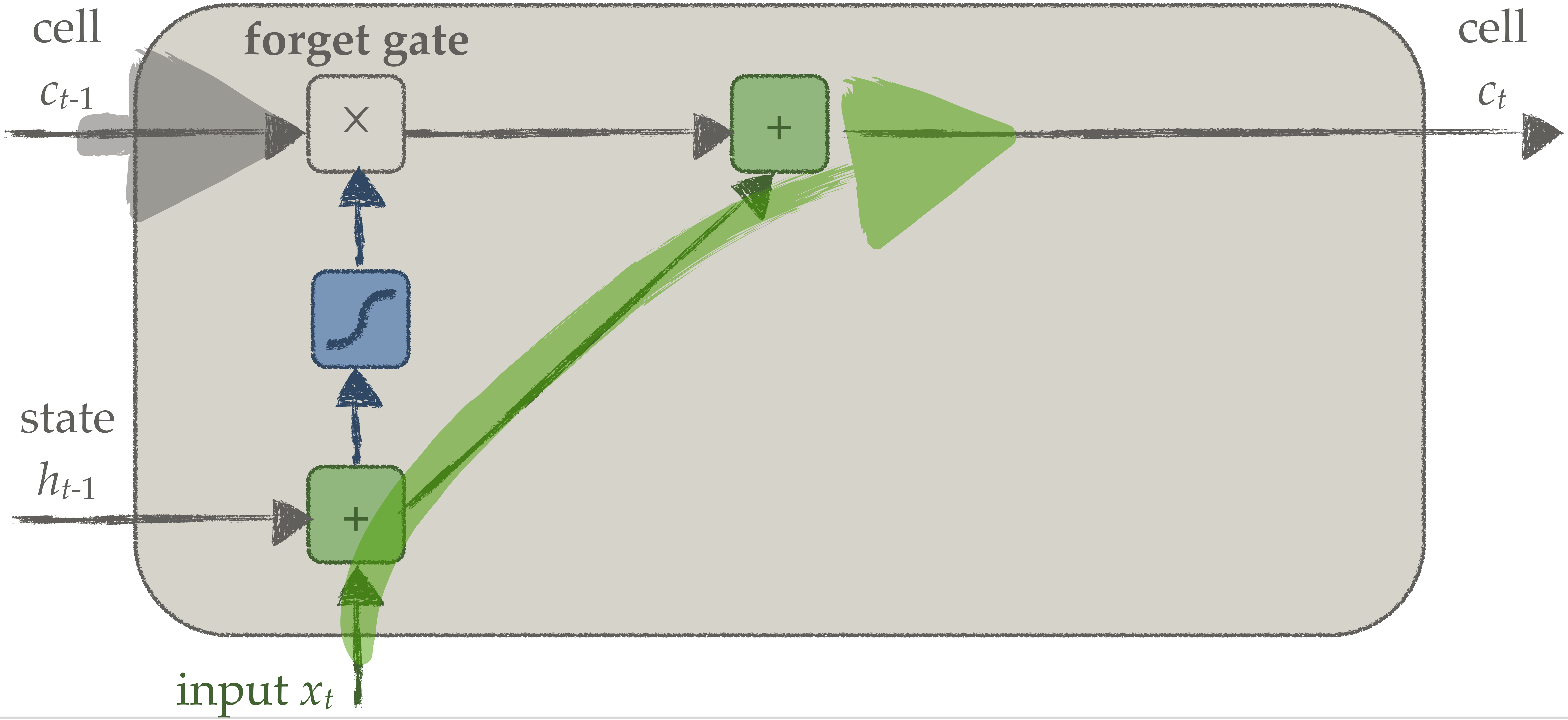
Requirement #1: linear cell

[Sepp Hochreiter and Jürgen Schmidhuber (1997) “Long Short-Term Memory”, *Neural Computation*;
Alex Graves (2013a) “Generating sequences with recurrent neural networks”, *arXiv* 1308.0850]

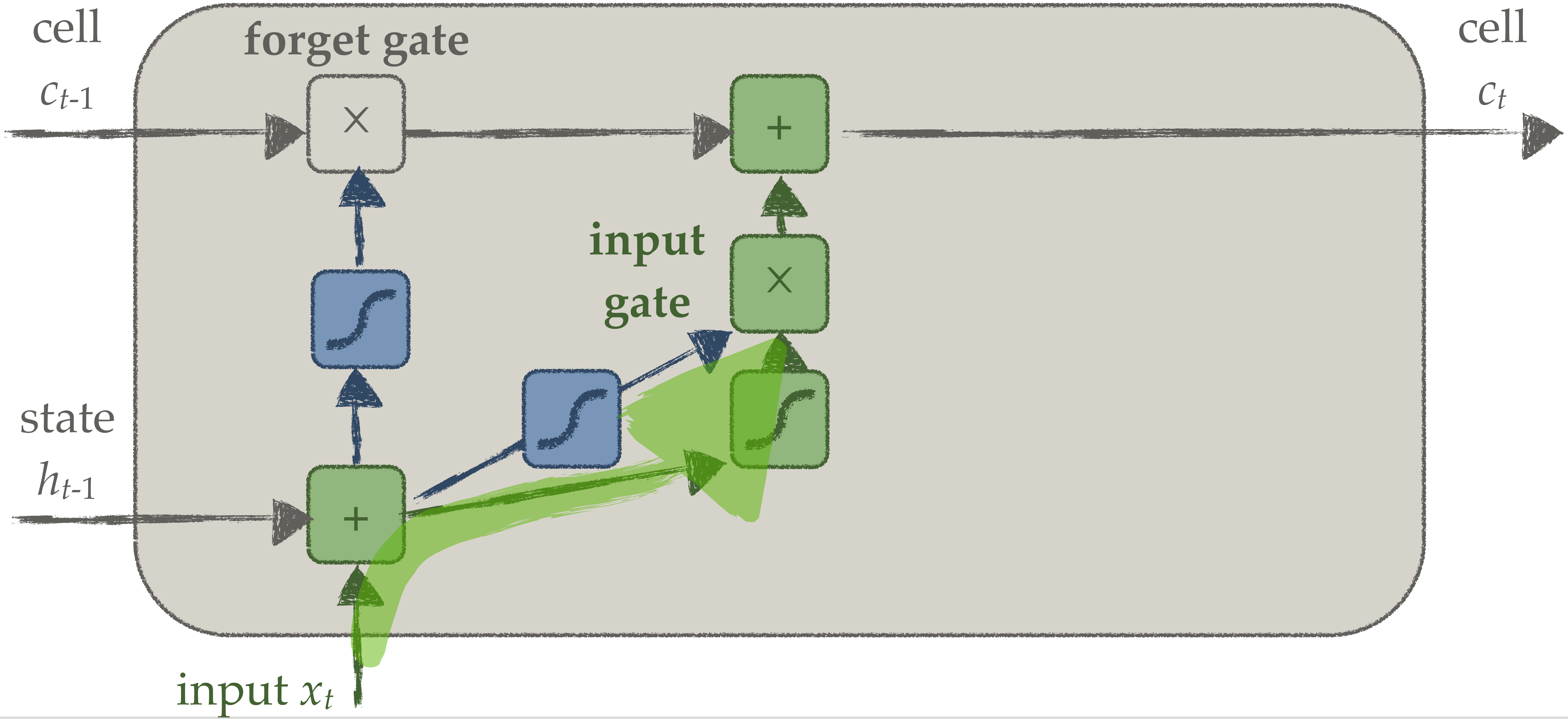


Requirement #2: forget information

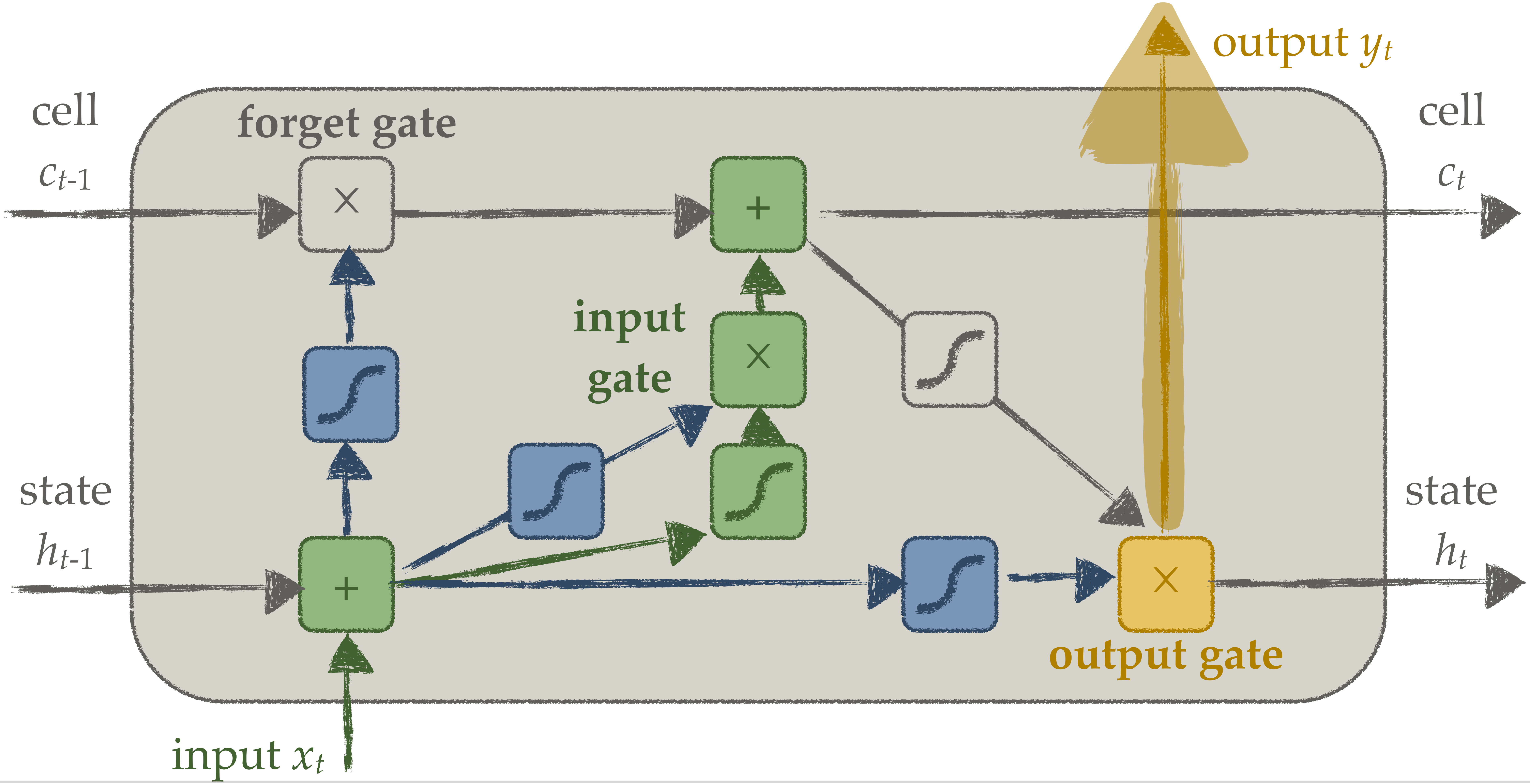
[Sepp Hochreiter and Jürgen Schmidhuber (1997) "Long Short-Term Memory", *Neural Computation*; Alex Graves (2013a) "Generating sequences with recurrent neural networks", *arXiv* 1308.0850]



Requirement #3: ignore inputs

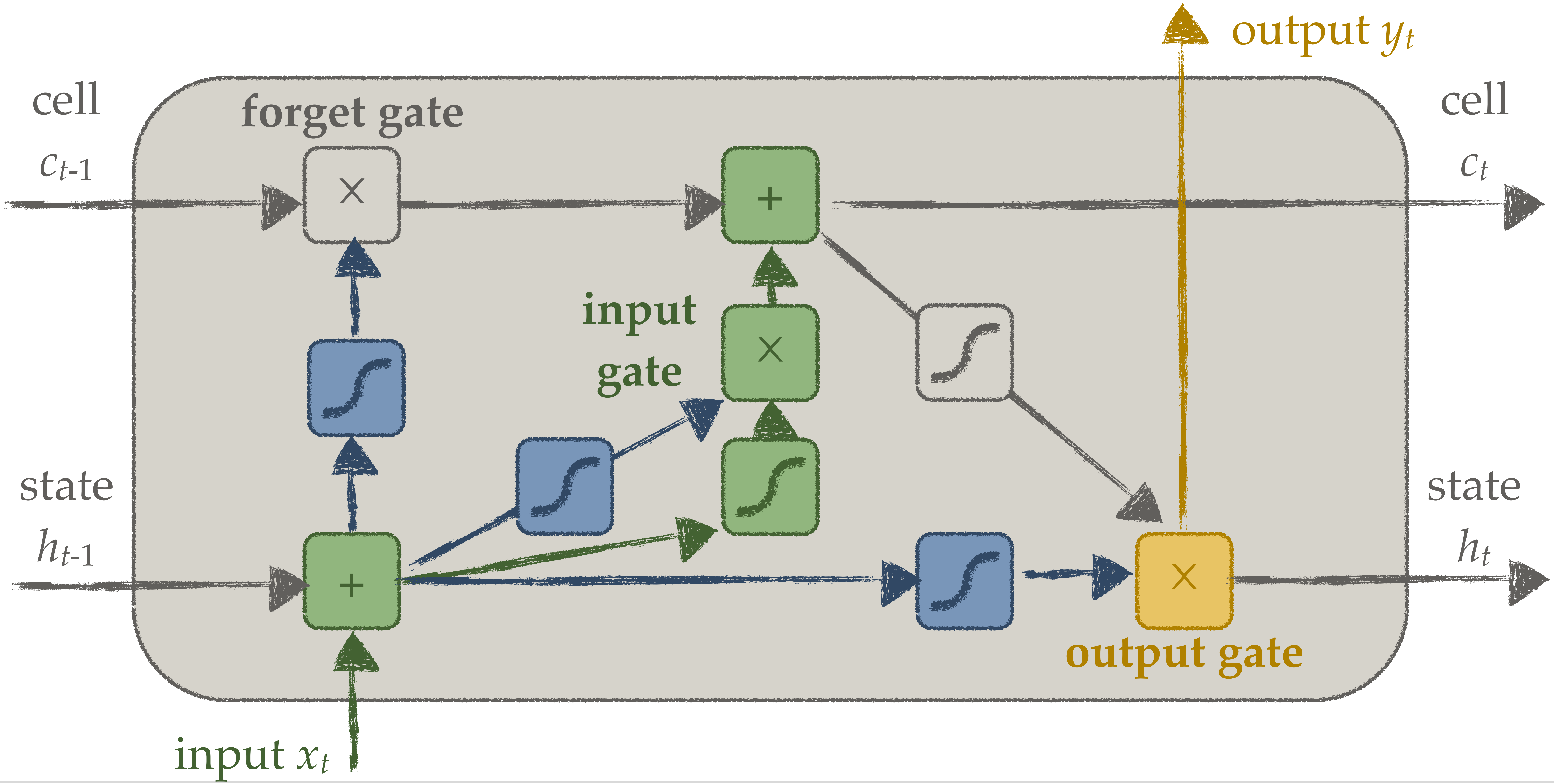


Requirement #4: control outputs



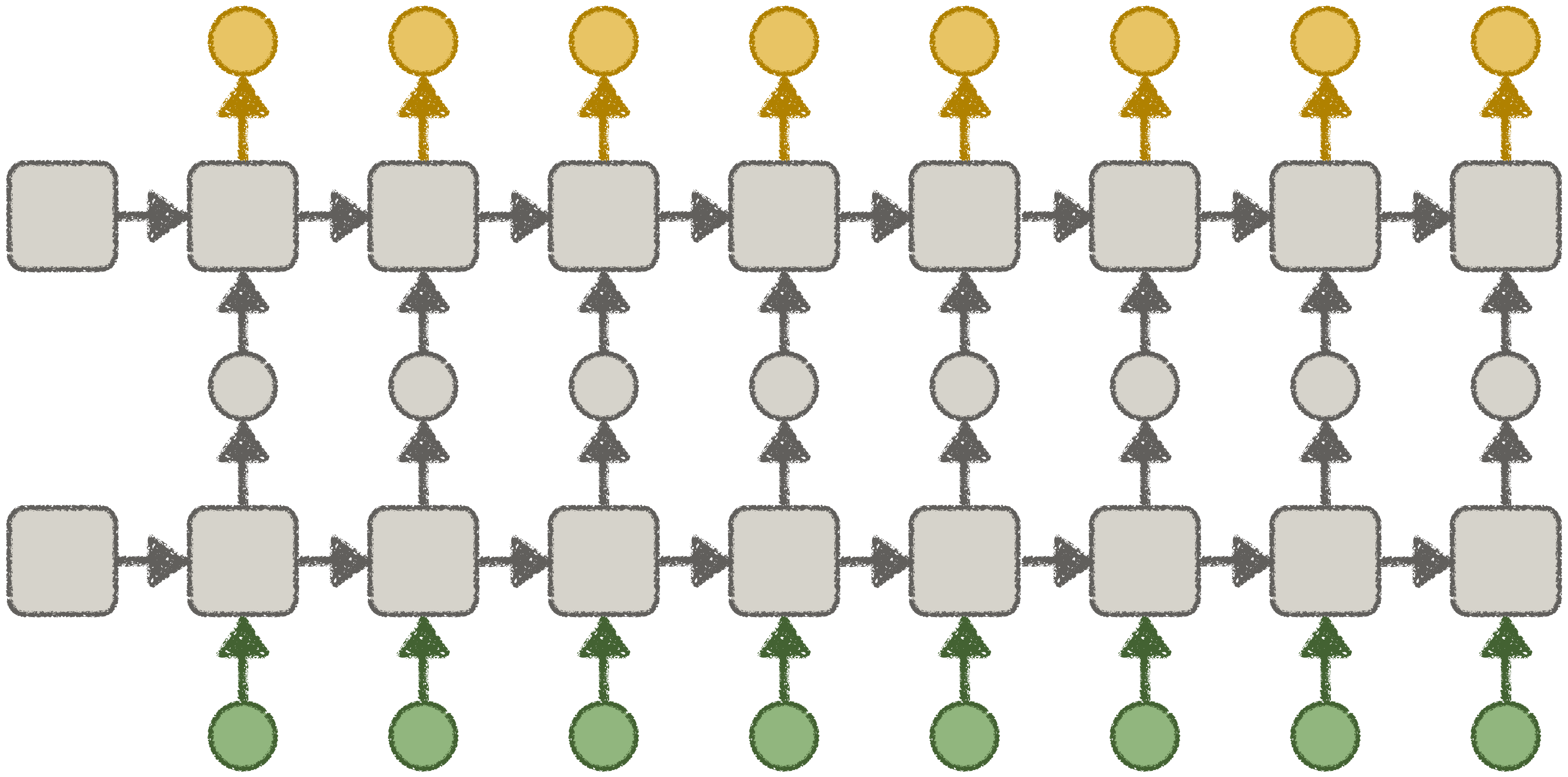
[Hochreiter and Schmidhuber (1997); Graves (2013a)]

Long Short-Term Memory (LSTM)



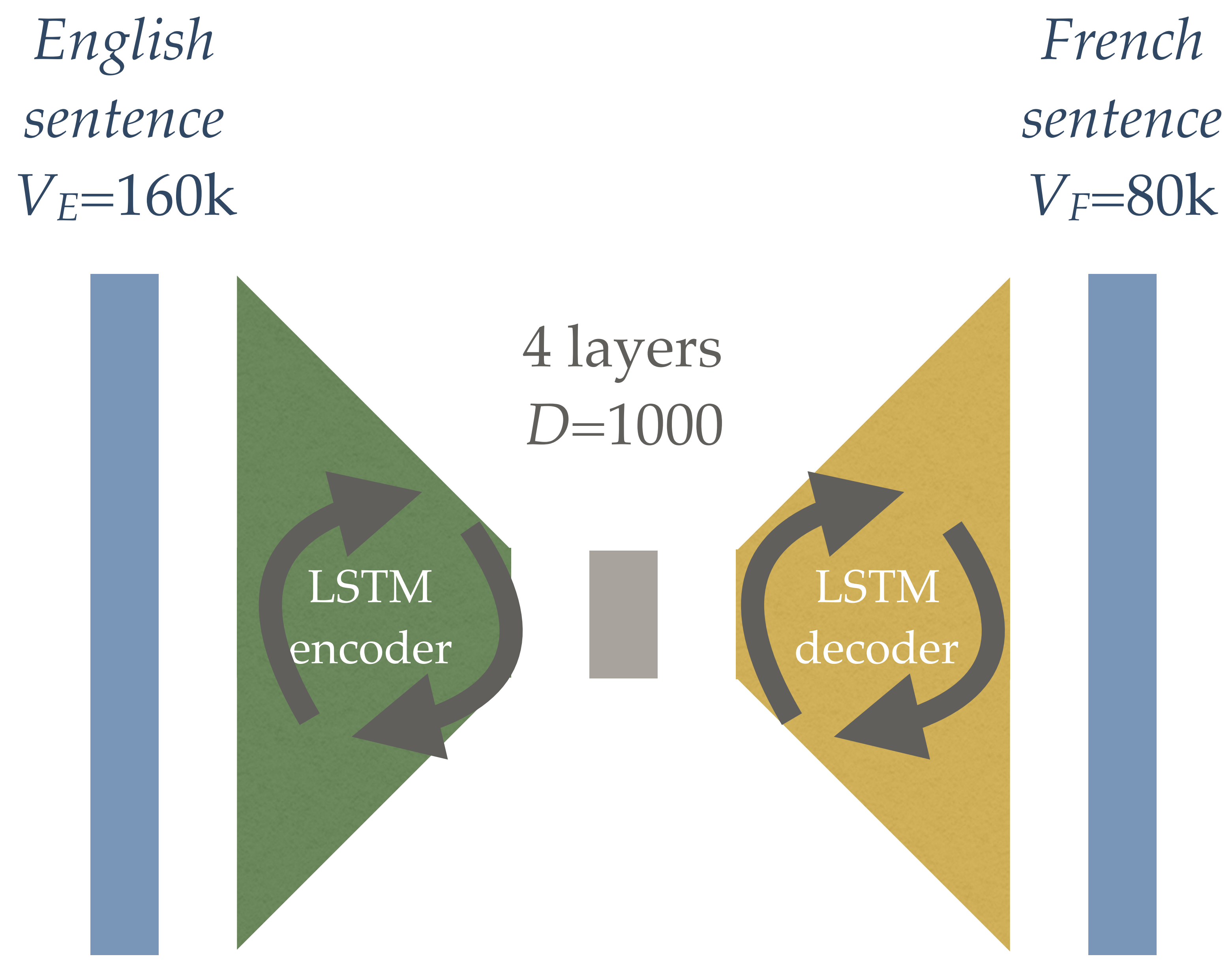
[Hochreiter and Schmidhuber (1997); Graves (2013a)]

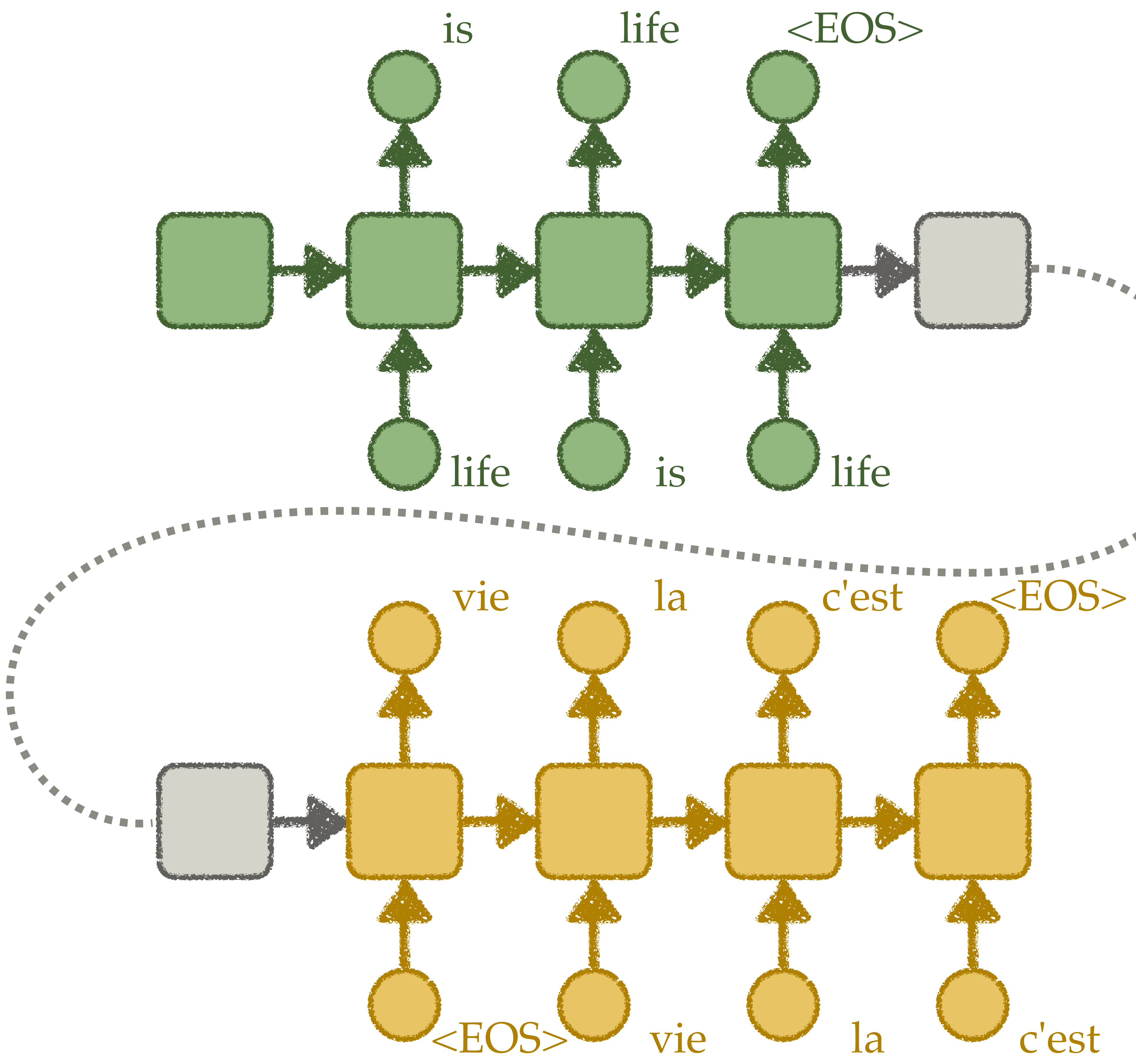
Deep LSTMs: stacking layers



[Graves (2013a)]

Sentence-to-sentence machine translation



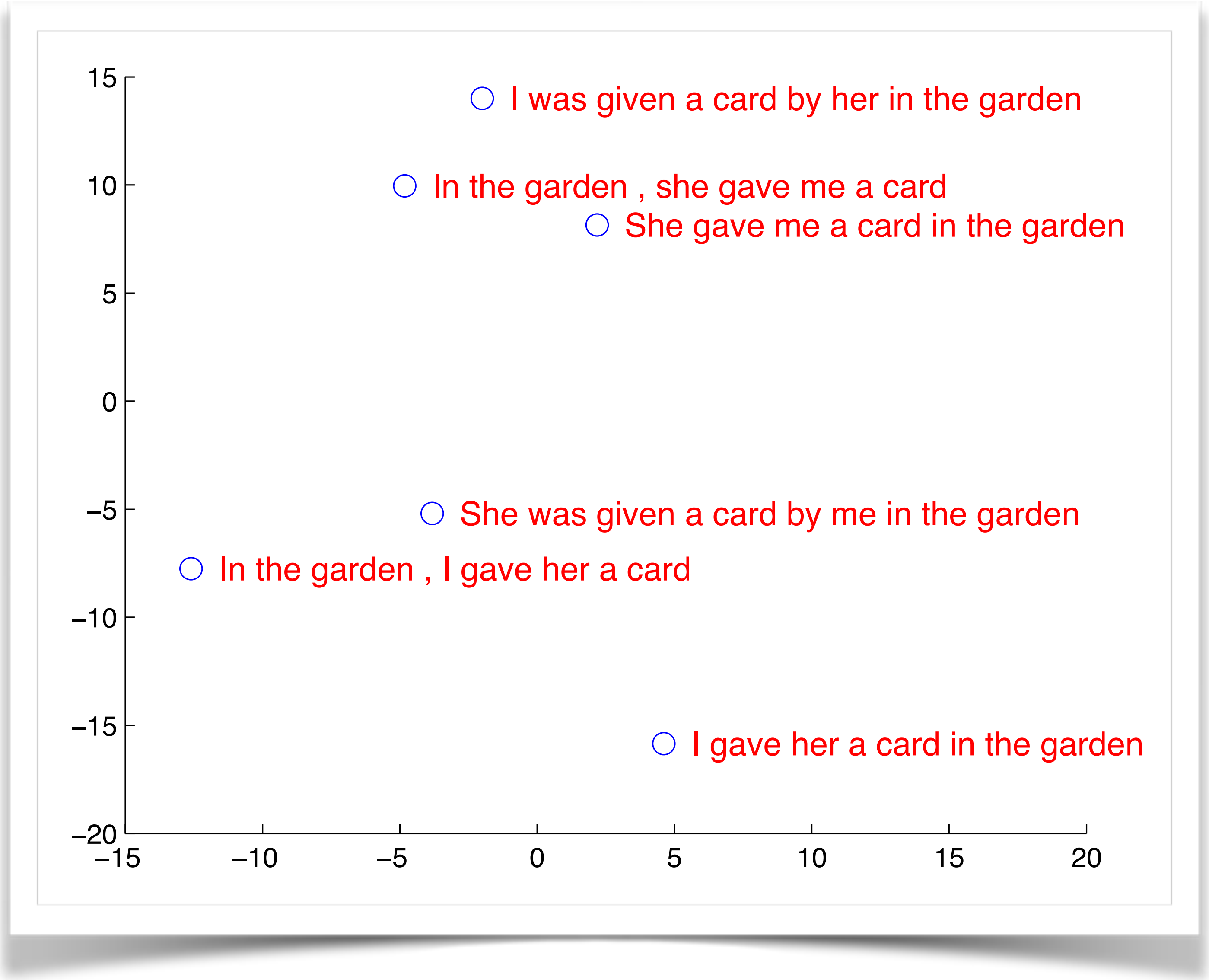


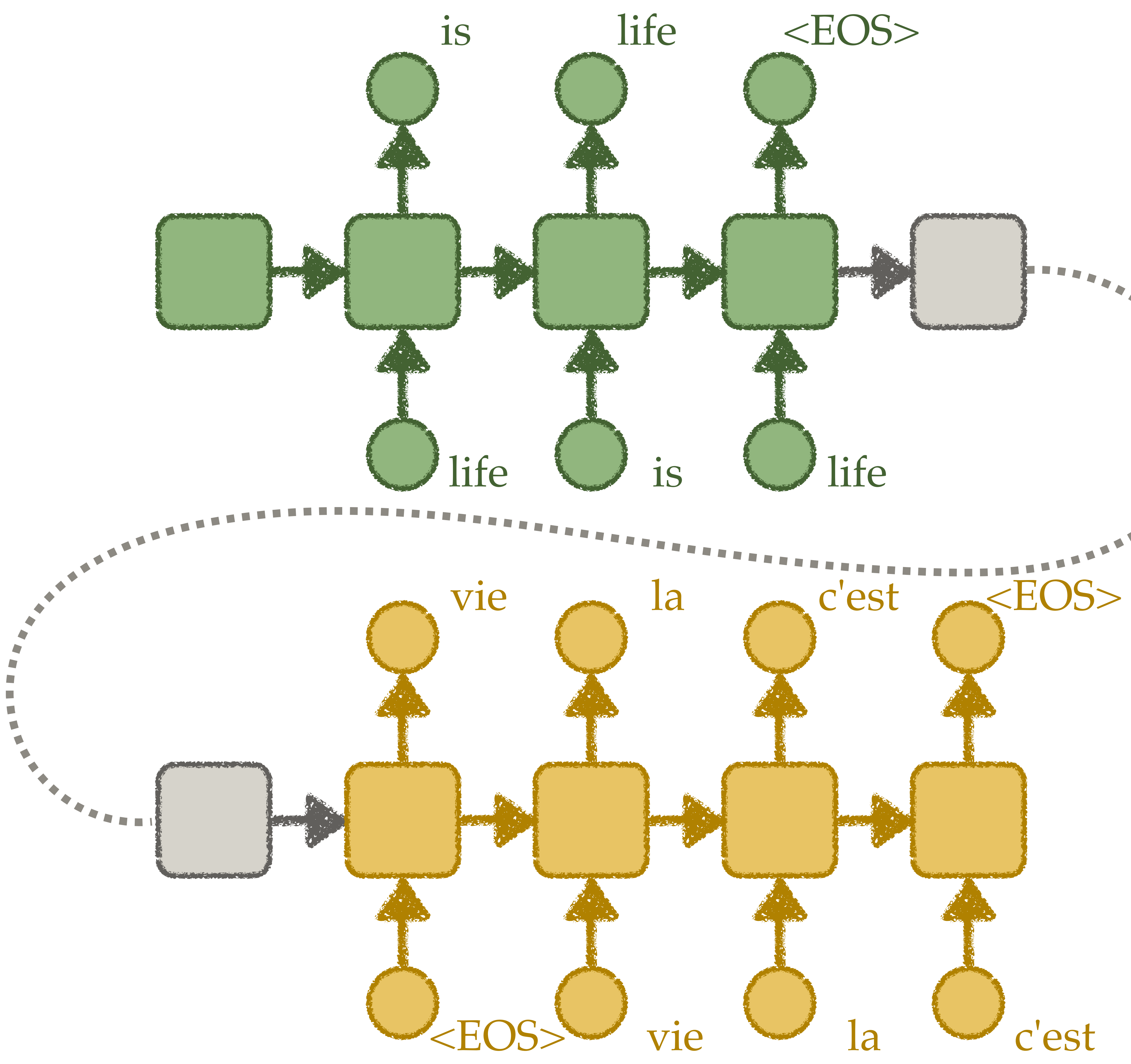
[Sutskever et al. (2014)
"Sequence to sequence learning
with neural networks", *NIPS*]

Sentence-to-sentence machine translation

" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsq' ils sont dans l' air " , dit <UNK> .

" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .





Character-based text generation

Wikipedia English (2006)

Train on 96MB of text

Vocabulary: V=205
(one-byte characters)

7-layer LSTM,
each layer with 700 hiddens

```
[http://www.sibeo.org/netspace/stainteology.htm The design of thinker Wikipedia
server-routed website]
*[http://balear.burned.edu/books/1978/tap/trpio.cs/cafa/c249ca.html Nation-state
 Player on the /1.3]
{{Linux}}
[[Category:Italo Succeeded bystables|General]]
[[Category:Italo-Saxon singers]]
[[Category:Aviation]]
[[de:Italo]]
[[es:Geotnia slago]]
[[ja:桛娯尔杏の洳]]
[[ko:혐슜불즷엉 유일]]
[[nl:Rodenbaueri]]
[[pl:Main Ages]]
[[pt:Ibanez Heights]]
[[ru:МлкракянѳелолↃуциянсьния агморелиа]]
[[simple:International Maritime Commission]]
[[sk:ICBM]]
[[su:Ol aal Mitnet]]
[[fi:Italo Colombonasto Warkō]]
[[sv:Opin den mail-climbere panczeur]]
[[zh:        ]]</text>
      </revision>
    </page>
    <page>
      <title>Italo-II article</title>
      <id>14848</id>
      <revision>
        <id>15912447</id>
        <timestamp>2004-11-19T19:47:39Z</timestamp>
        <contributor>
          <username>El andres</username>
          <id>509991</id>
        </contributor>
        <minor />
        <comment>disambiguation from [[List of ISBN]] newsletters, see [[ISO site]]</comment>
        <text xml:space="preserve">#REDIRECT [[Quartic war]]</text>
      </revision>
    </page>
    <page>
      <title>ICSM</title>
      <id>14939</id>
      <revision>
        <id>42109942</id>
        <timestamp>2006-02-28T17:22:02Z</timestamp>
        <contributor>
          <username>Dtelclan</username>
          <id>26</id>
        </contributor>
        <minor />
        <comment>/* Possible catheterman */</comment>
        <text xml:space="preserve">[[Image:Isaac.org/ice.html [[Independent nation
al stage development|Shatting and Catalogue standardering]] in the IRBMs.

Up-2000 they called the SC 4220 system: he was swalloped early in Calvino, or since each trial mentioned
based on [[Balbov's new single-jarget|bit-oriann guess]
```

[Alex Graves (2013) "Generating sequences with RNNs", *arXiv*]

LSTMs in popular culture

Lyrics generation

[The Guardian, 1 December 2015, "World's first computer-generated musical to debug in London"
https://www.theguardian.com/stage/2015/dec/01/beyond-the-fence-computer-generated-musical-greenham-common]

"World's first computer-generated musical to debut in London.
Beyond the Fence, the story of a family in Greenham Common, will incorporate machine-generated plot and music.
[…] But could a computer also generate a hit West End musical?
The answer may be provided next year with the announcement of the world's first computer musical, getting a run at the Arts Theatre […]"
[Courtesy of Guardian News & Media Ltd.]

Movie script generation
for short movie "Sunspring" by Ross Goodwin
[http://rossgoodwin.com]

Speech recognition

Google Research Blog

The latest news from Research at Google

[Google Research Blog, 11 August 2015, http://googleresearch.blogspot.co.uk/2015/08/the-neural-networks-behind-google-voice.html]

The neural networks behind Google Voice transcription

Tuesday, August 11, 2015

Posted by Françoise Beaufays, Research Scientist

Over the past several years, deep learning has shown remarkable success on some of the world's most difficult computer science challenges, from image classification and captioning to translation to model visualization techniques. Recently we announced improvements to Google Voice transcription using Long Short-term Memory Recurrent Neural Networks (LSTM RNNs)—yet another place neural networks are improving useful services. We thought we'd give a little more detail on how we did this.

Since it launched in 2009, Google Voice transcription had used Gaussian Mixture Model (GMM) acoustic models, the state of the art in speech recognition for 30+ years. Sophisticated techniques like adapting the models to the speaker's voice augmented this relatively simple modeling method.

Then around 2012, Deep Neural Networks (DNNs) revolutionized the field of speech recognition. These multi-layer networks distinguish sounds better than GMMs by using "discriminative training," differentiating phonetic units instead of modeling each one independently.

But things really improved rapidly with Recurrent Neural Networks (RNNs), and especially LSTM RNNs, first launched in Android's speech recognizer in May 2012. Compared to DNNs, LSTM RNNs have additional recurrent connections and memory cells that allow them to "remember" the data they've seen so far—much as you interpret the words you hear based on previous words in a sentence.

Search blog ...

Research at Google

google.com/+ResearchatGoogle

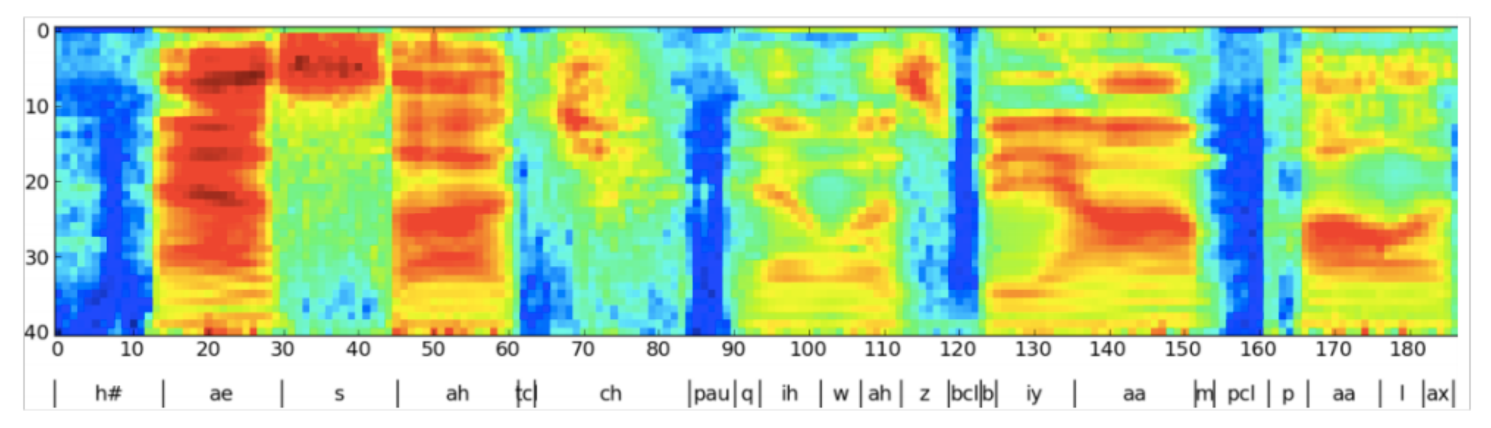
Follow +1

+ 1,179,560

Labels

Archive

Starting from acoustic vectors…



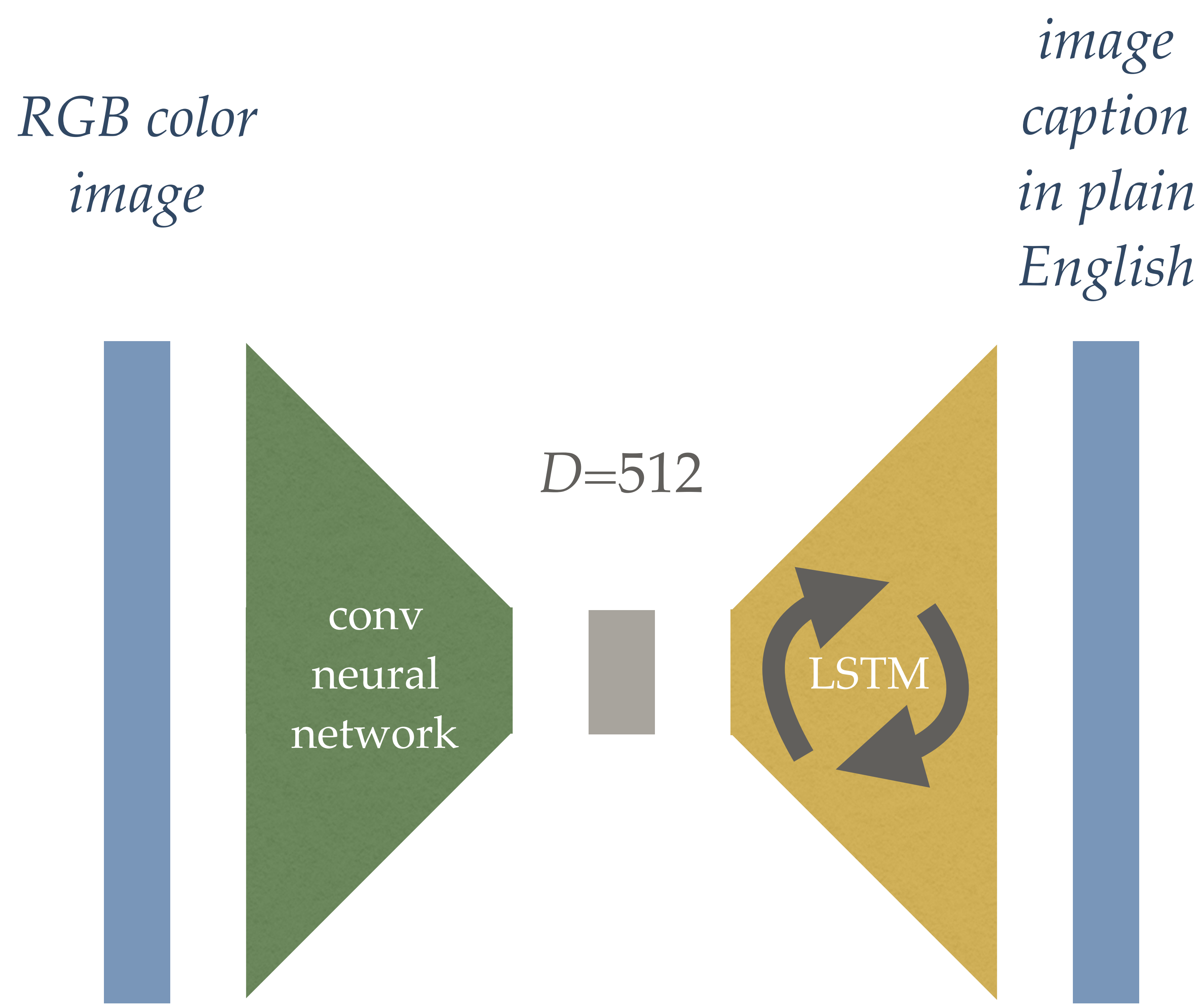
[Graves et al. (2013b) "Speech recognition with deep recurrent neural networks", *ICASSP*]

… choose the "most likely" sentence

the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
…

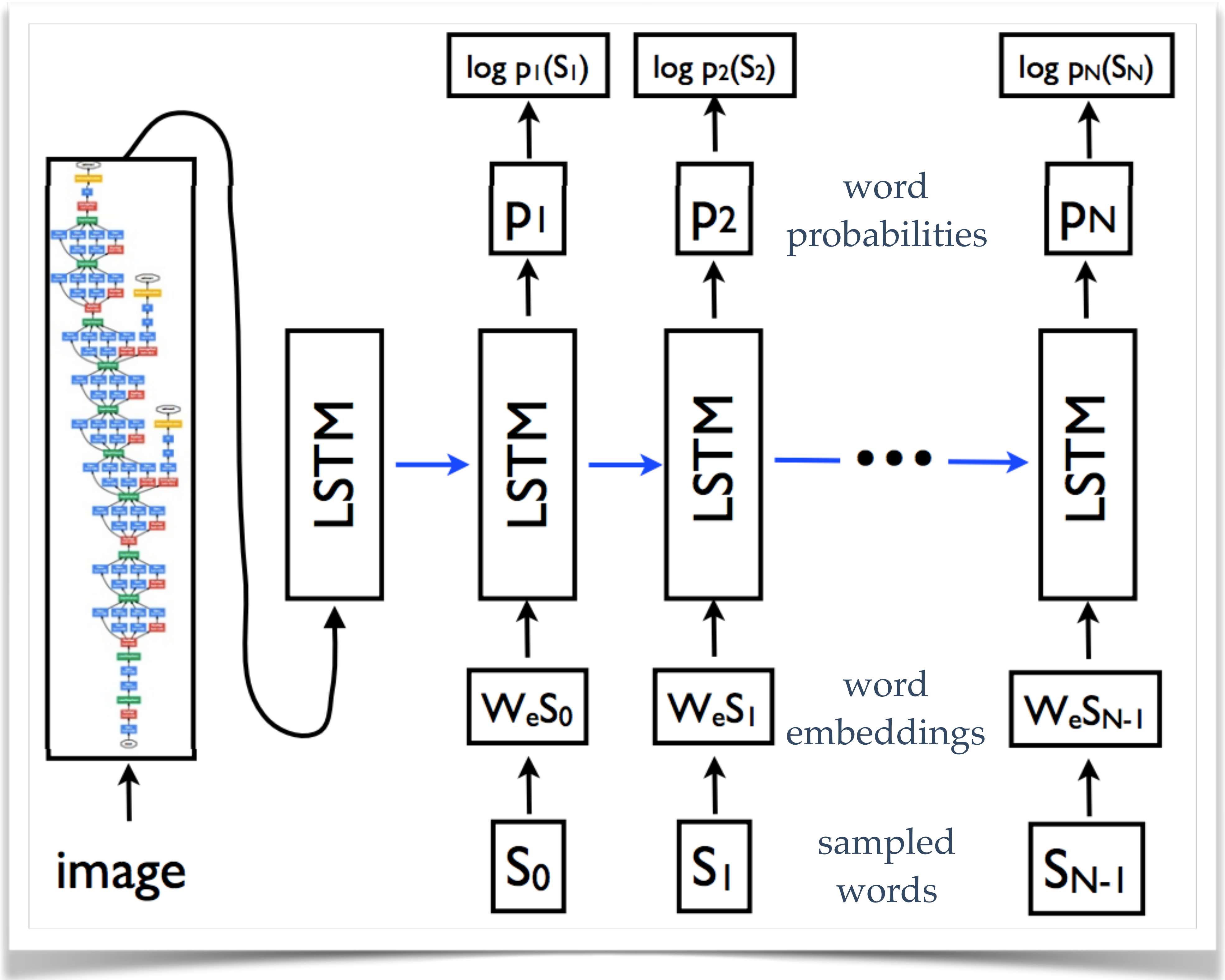
Image captioning

[Vinyals et al. (2014) "Show and Tell: Neural Image Caption Generation";
Karpathy et al. (2014) "Deep Visual-Semantic Alignments for Generating Image Descriptions";
Kiros et al. (2014) "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models"]



convolutional network
pre-trained on 2.5M ImageNet images

end-to-end system trained on 100k to 1M
image - caption pairs



[Image credits: Vinyals et al. (2014) "Show and Tell:
Neural Image Caption Generation"

Image captioning

[Vinyals et al. (2014) “Show and Tell: Neural Image Caption Generation”;
Karpathy et al. (2014) “Deep Visual-Semantic Alignments for Generating Image Descriptions”;
Kiros et al. (2014) “Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models"]

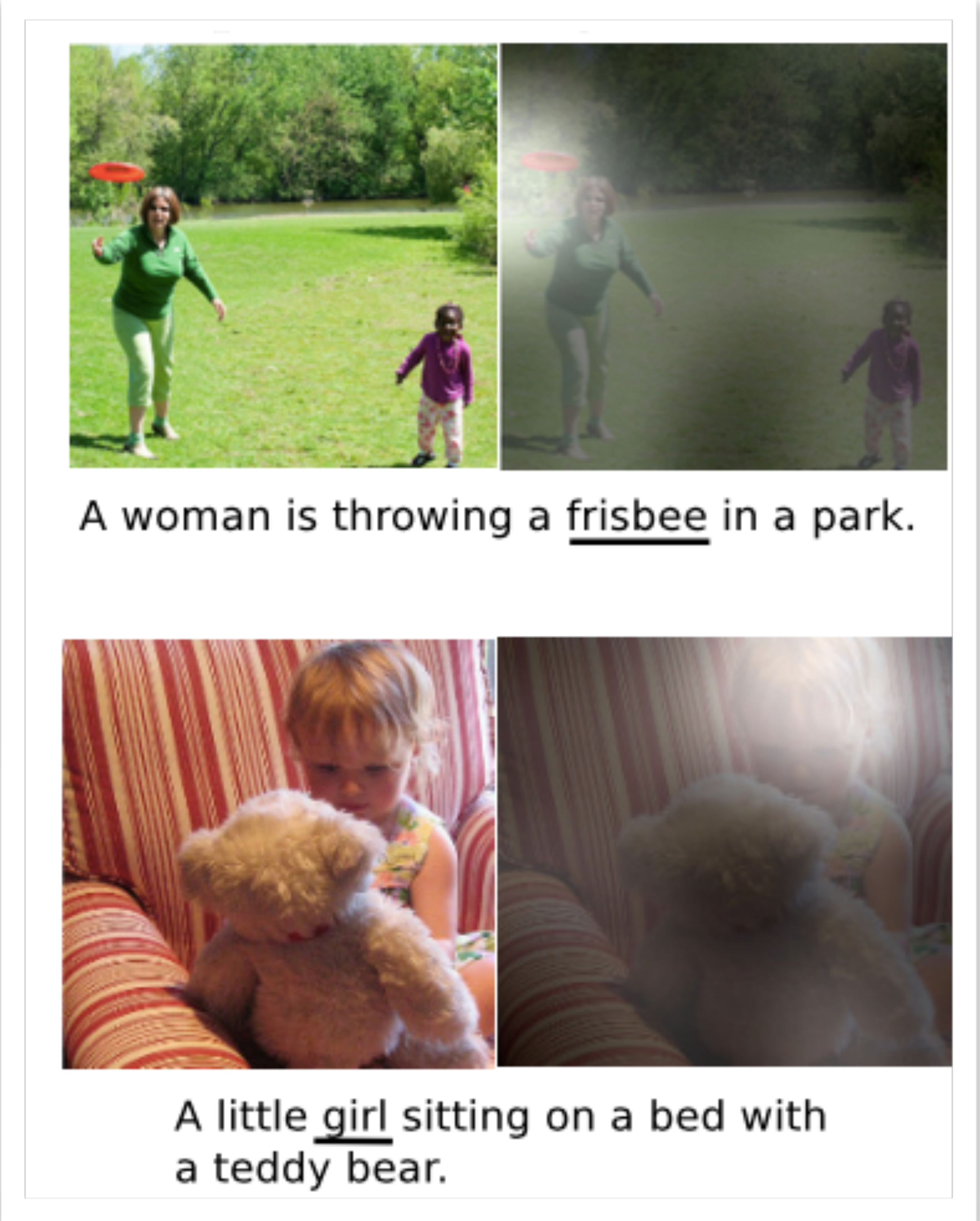
[Image credits: Vinyals et al. (2014) “Show and Tell: Neural Image Caption Generation”]

How? (what this talk will cover)

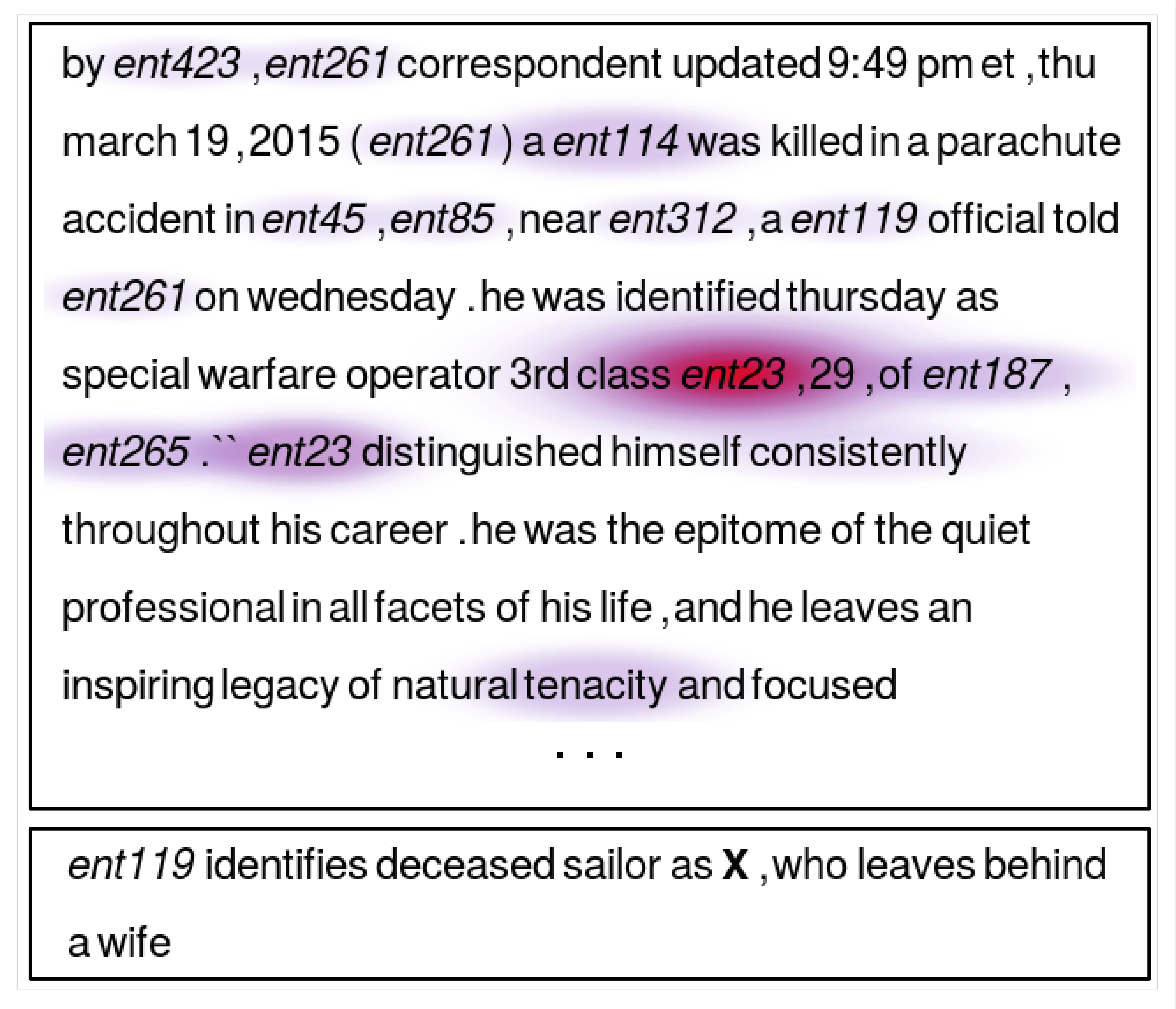
Fixed-memory language models
n-grams and Markov chains
Learning representations
Word embeddings
Maximum likelihood learning
Neural language models
Recurrent Neural Networks (RNNs)
Long Short-Term Memory RNNs
Attention and memory models
Control through Reinforcement Learning

Language modeling
Sentence completion
Machine translation
Text generation
Speech recognition
Image captioning
Query answering
Playing 3D games
Learning to navigate

Content-based attention

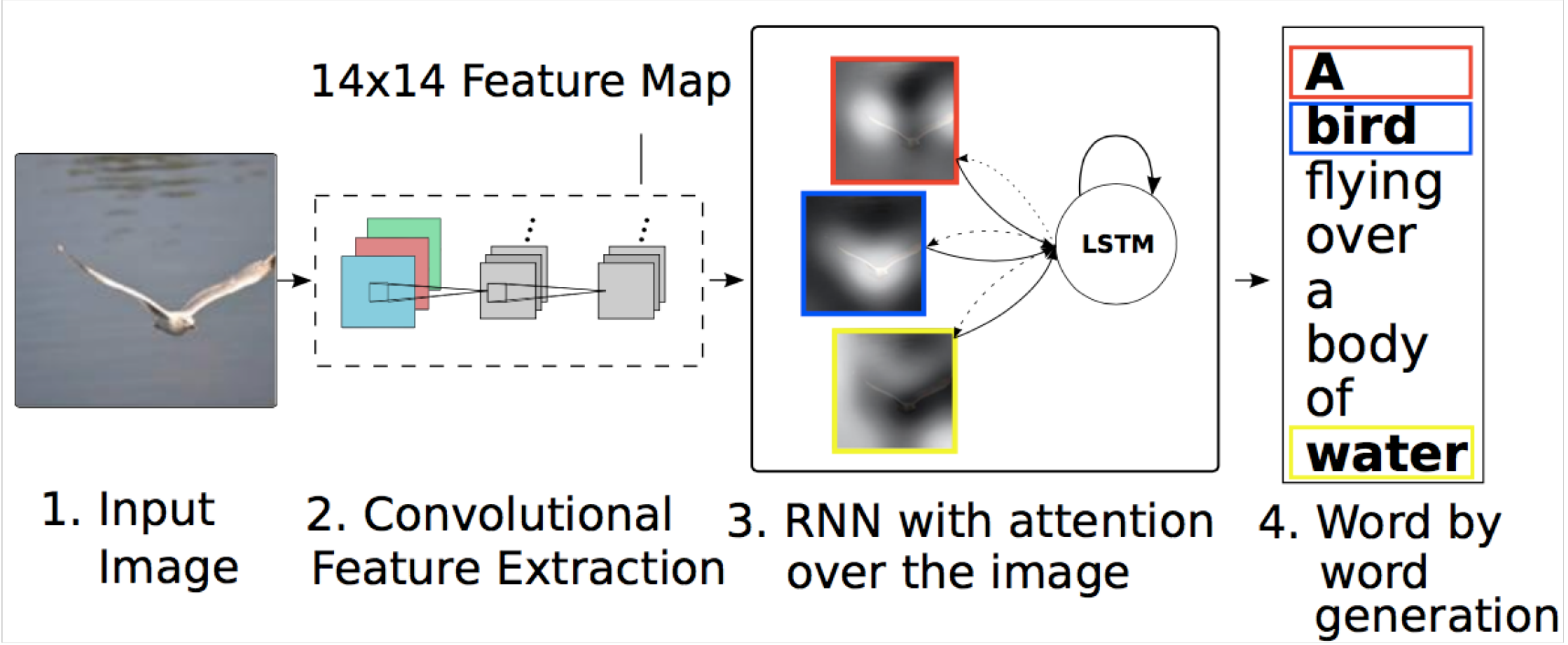


[Kelvin Xu et al. (2015)
"Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", *ICML*]



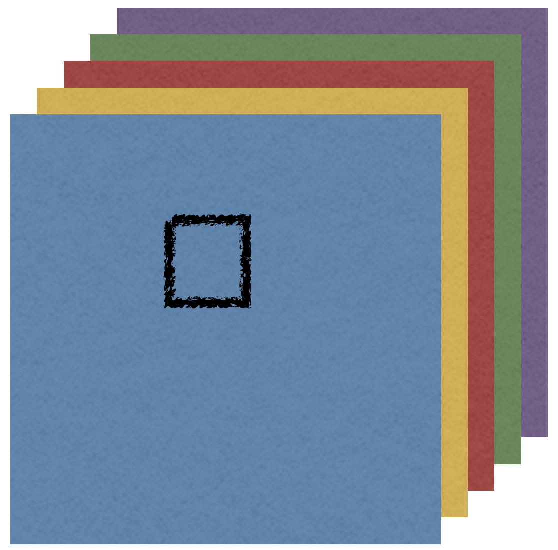
[Karl M Hermann et al. (2015)
"Teaching Machines to Read and to Comprehend", *NIPS*]

Image captioning with visual attention



[Kelvin Xu et al. (2015)
"Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", *ICML*]

last conv-net layer



512 maps of size 14x14

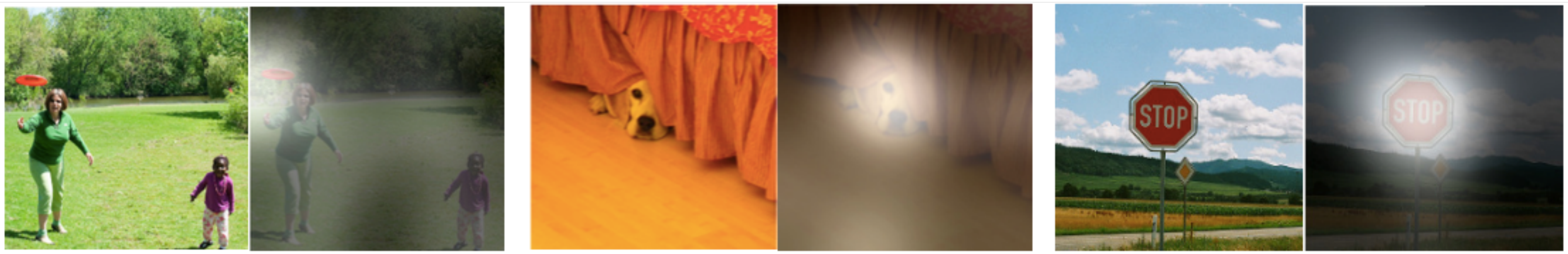
$\mathbf{f}_i$ *vector of image features D=512*

$\mathbf{h}_{t-1}$ *previous LSTM hiddens*

$$s_{i,t} = f_{att}(\mathbf{f}_i, \mathbf{h}_{t-1})$$ *attention score*

$$\alpha_{i,t} = \frac{e^{s_{i,t}}}{\sum_k e^{s_{k,t}}}$$ *attention distribution*

$$\mathbf{z}_t = \sum_i \alpha_{i,t} \mathbf{f}_i$$ *soft attention mechanism*



A woman is throwing a frisbee in a park.

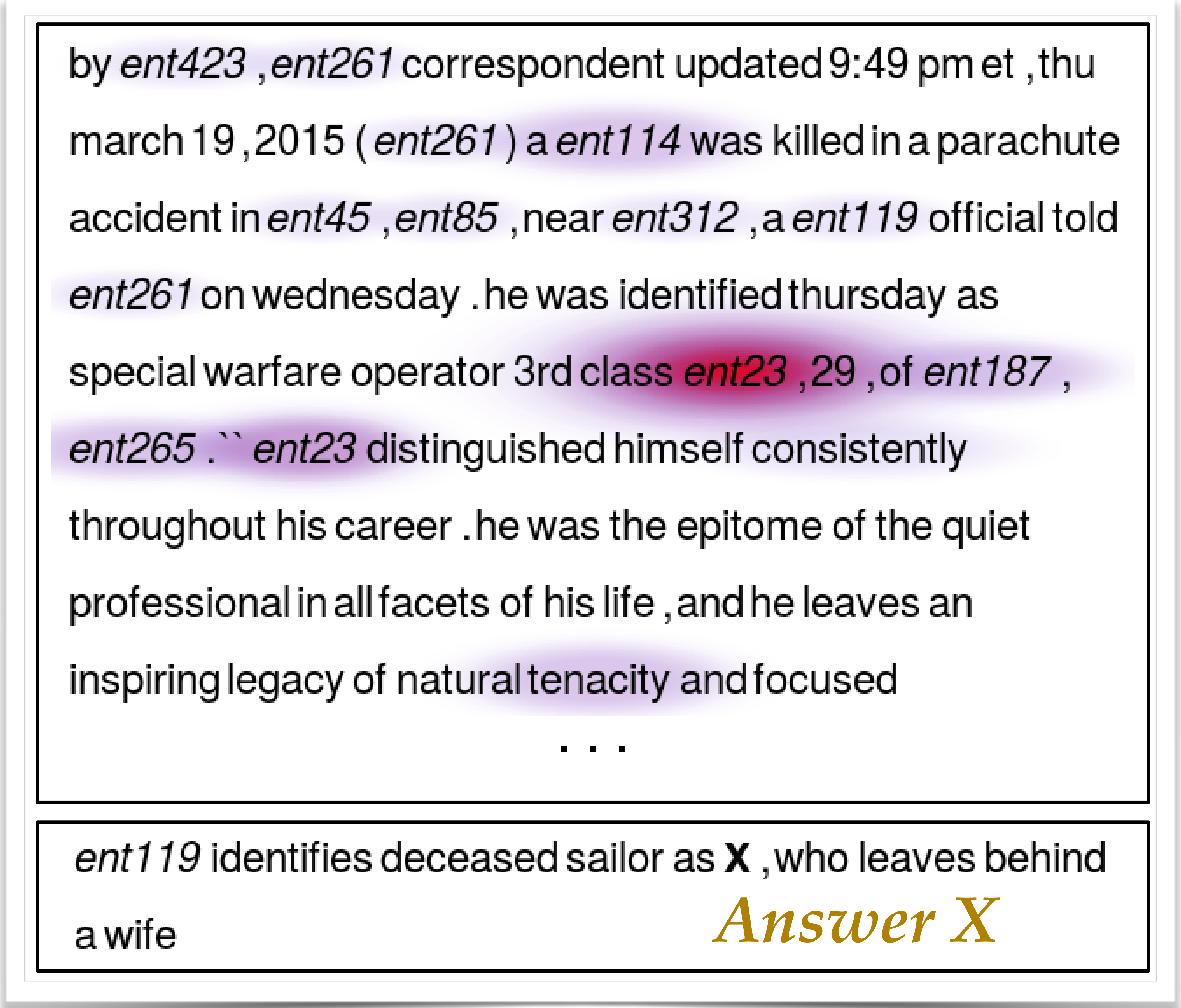
A dog is standing on a hardwood floor.

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

Query answering with attention over context

Document

Query

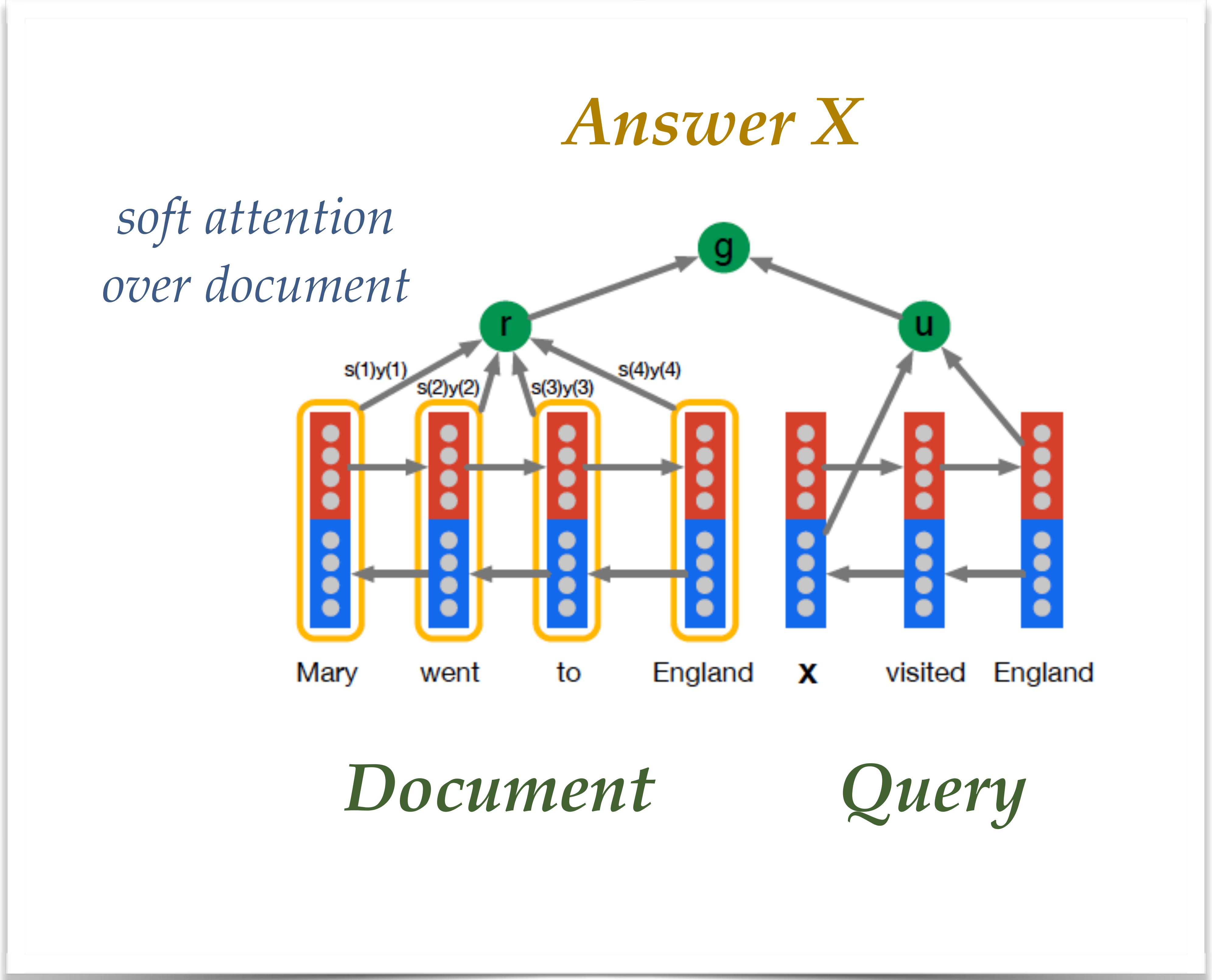


by *ent423* , *ent261* correspondent updated 9:49 pm et , thu march 19 , 2015 (*ent261*) a *ent114* was killed in a parachute accident in *ent45* , *ent85* , near *ent312* , a *ent119* official told *ent261* on wednesday . he was identified thursday as special warfare operator 3rd class *ent23* , 29 , of *ent187* , *ent265* . `` *ent23* distinguished himself consistently throughout his career . he was the epitome of the quiet professional in all facets of his life , and he leaves an inspiring legacy of natural tenacity and focused

. . .

ent119 identifies deceased sailor as **X** , who leaves behind a wife

Answer X



[Karl M Hermann et al. (2015) "Teaching Machines to Read and to Comprehend", NIPS]

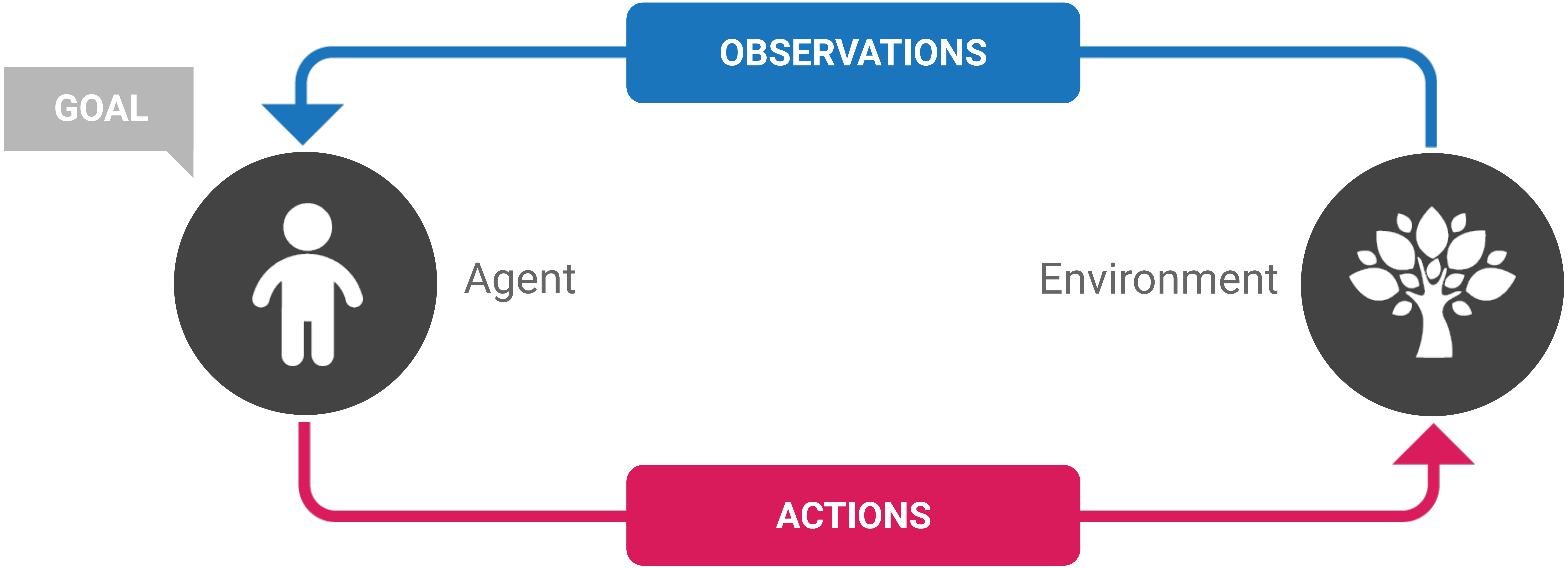
How? (what this talk will cover)

Fixed-memory language models
n-grams and Markov chains
Learning representations
Word embeddings
Maximum likelihood learning
Neural language models
Recurrent Neural Networks (RNNs)
Long Short-Term Memory RNNs
Attention and memory models
Control through Reinforcement Learning

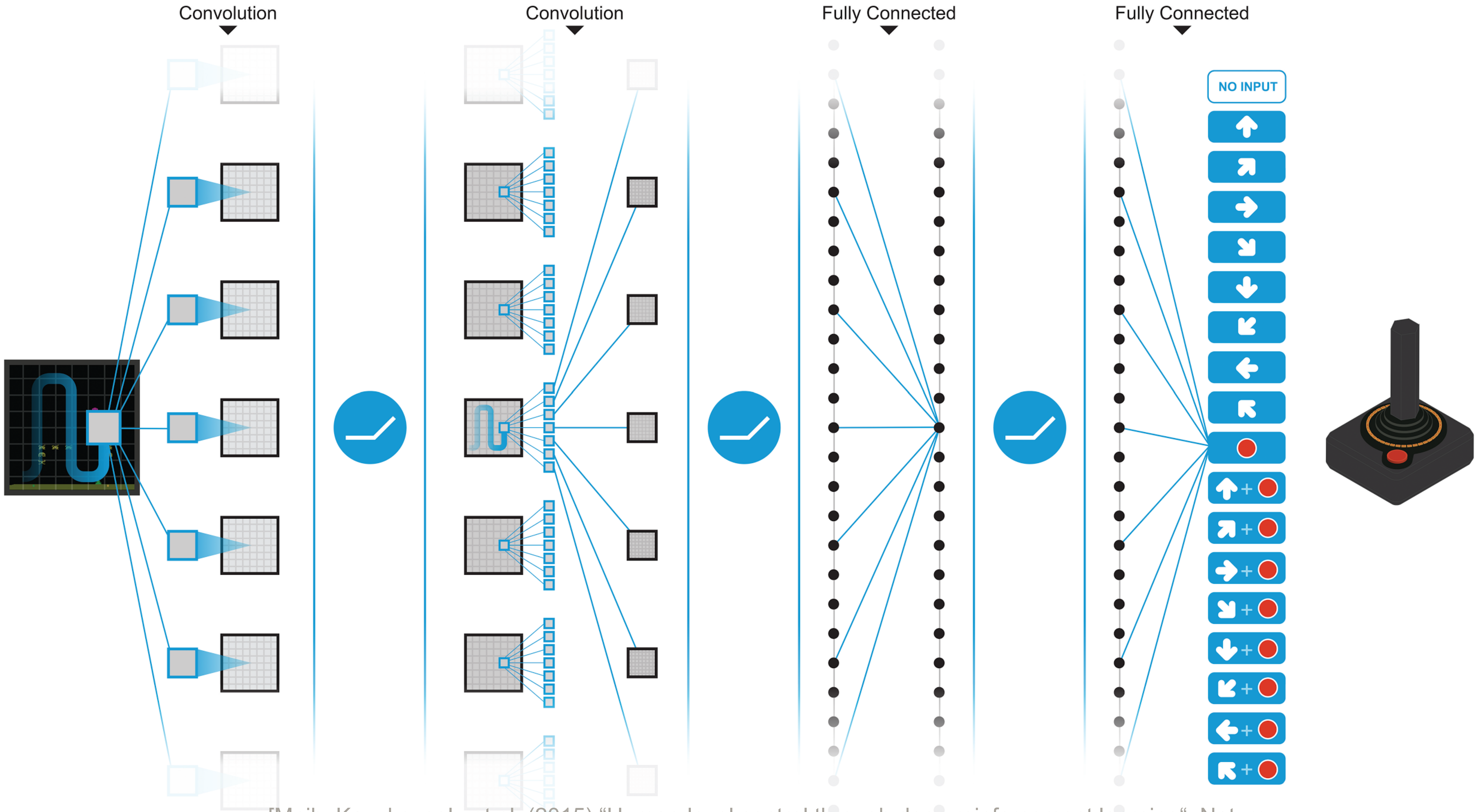
Language modeling
Sentence completion
Machine translation
Text generation
Speech recognition
Image captioning
Query answering
Playing 3D games
Learning to navigate

Reinforcement learning framework

[Mnih, Kavukcuoglu et al. (2015) “Human-level control through deep reinforcement learning“, Nature; Silver, Huang et al. (2016) “Mastering the game of Go with deep neural networks and tree search”, Nature]

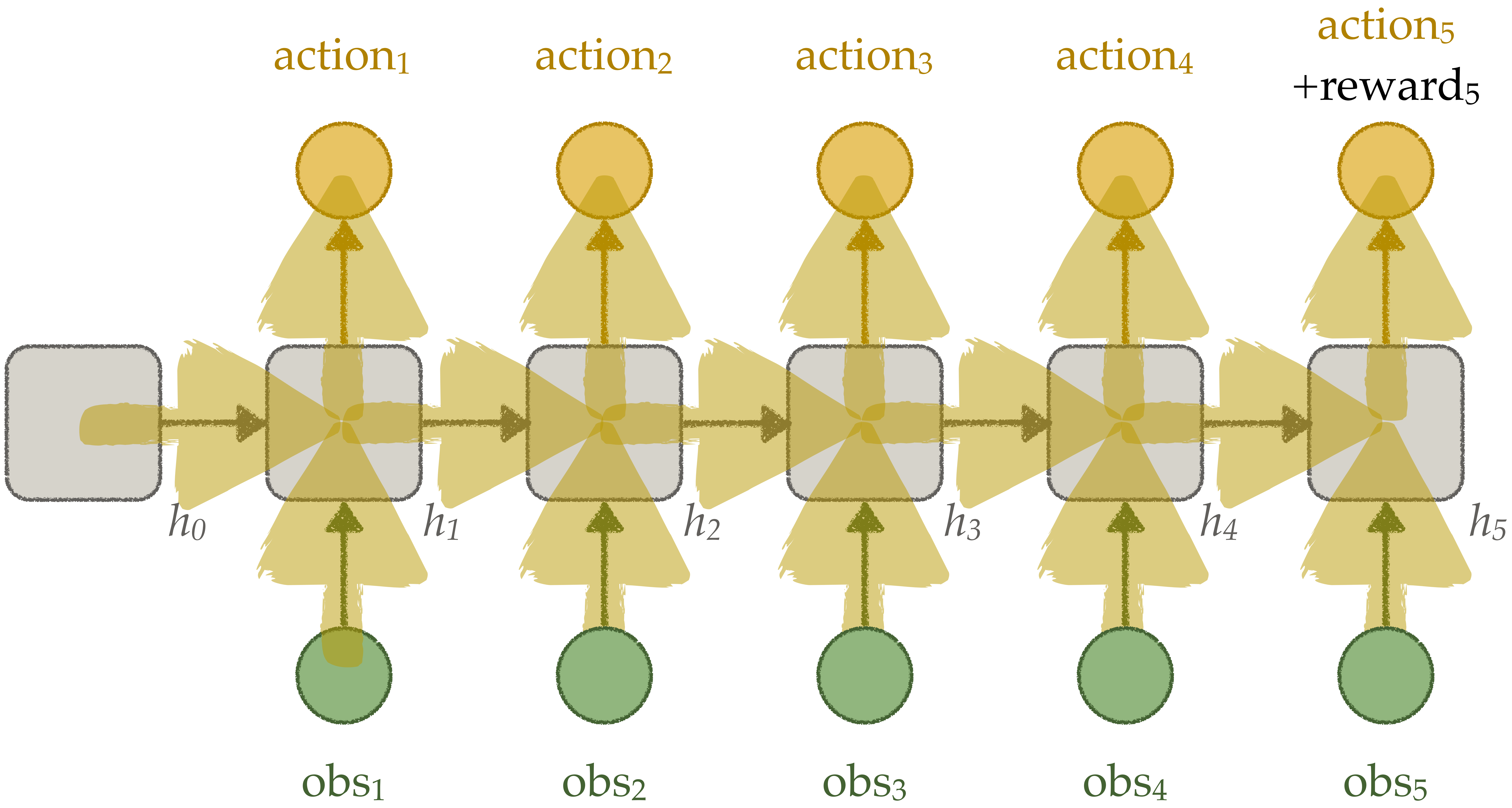


Reinforcement learning with plain convnets

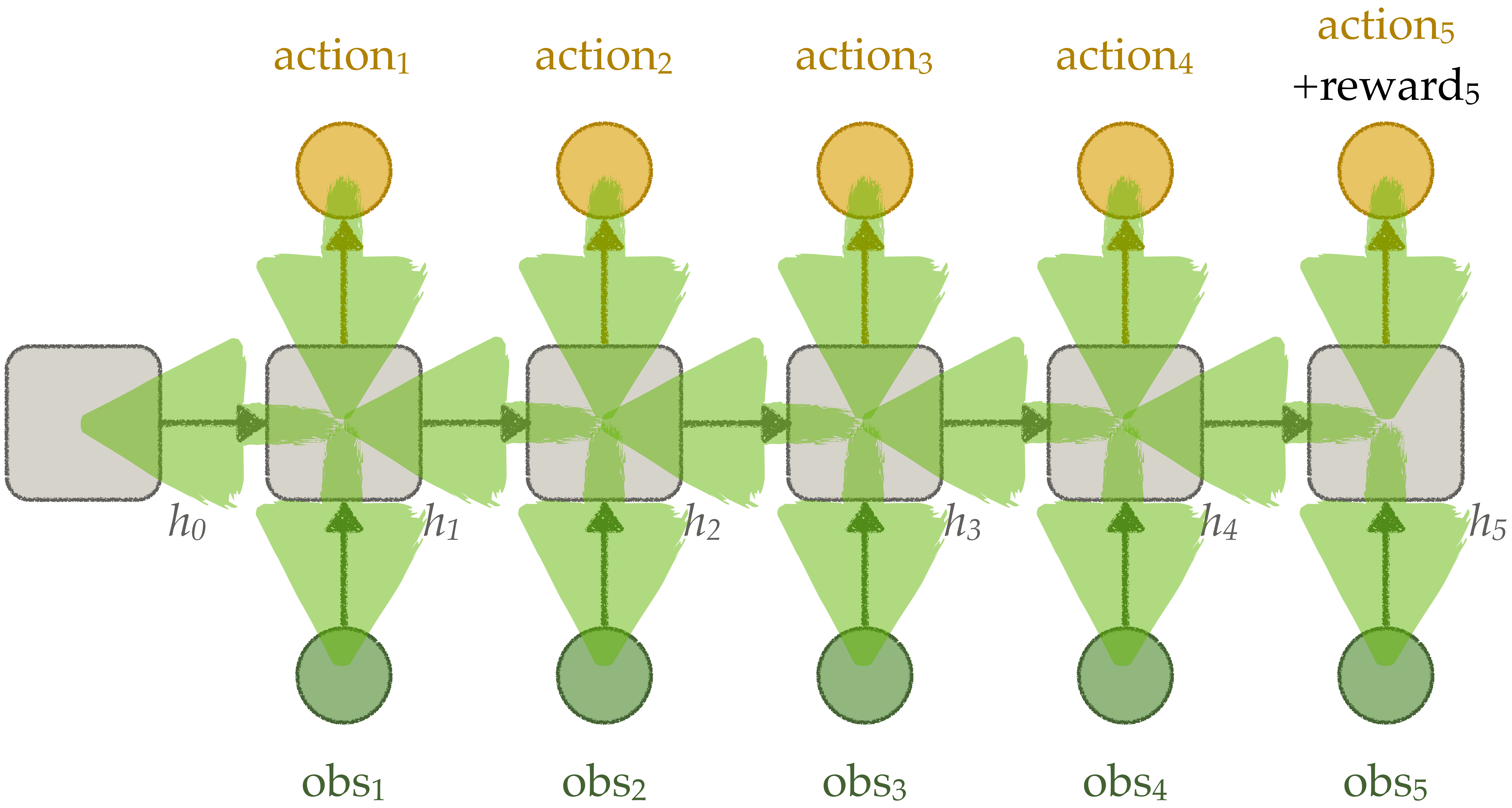


[Mnih, Kavukcuoglu et al. (2015) "Human-level control through deep reinforcement learning", Nature;
Silver, Huang et al. (2016) "Mastering the game of Go with deep neural networks and tree search", Nature]

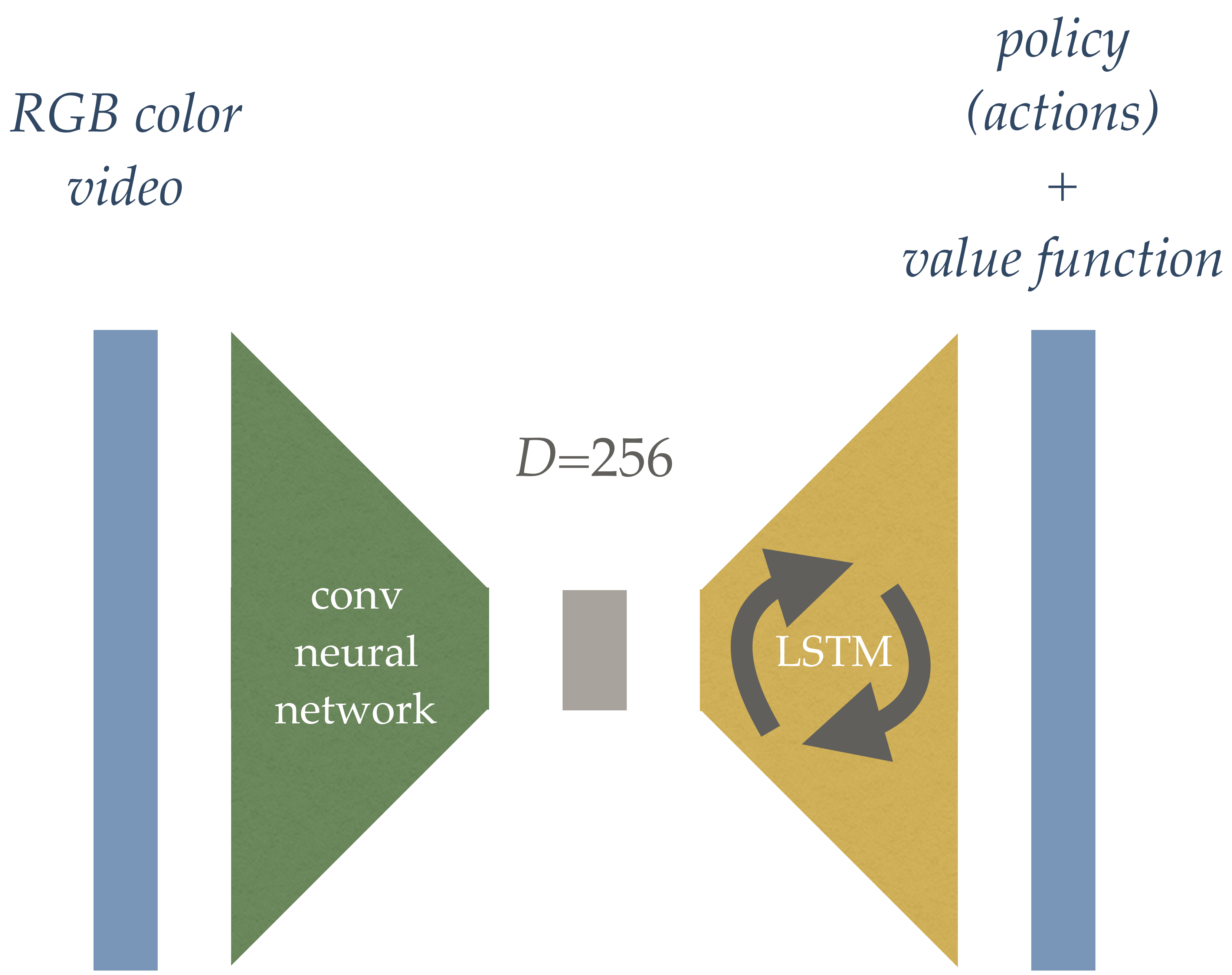
Reinforcement learning with RNNs



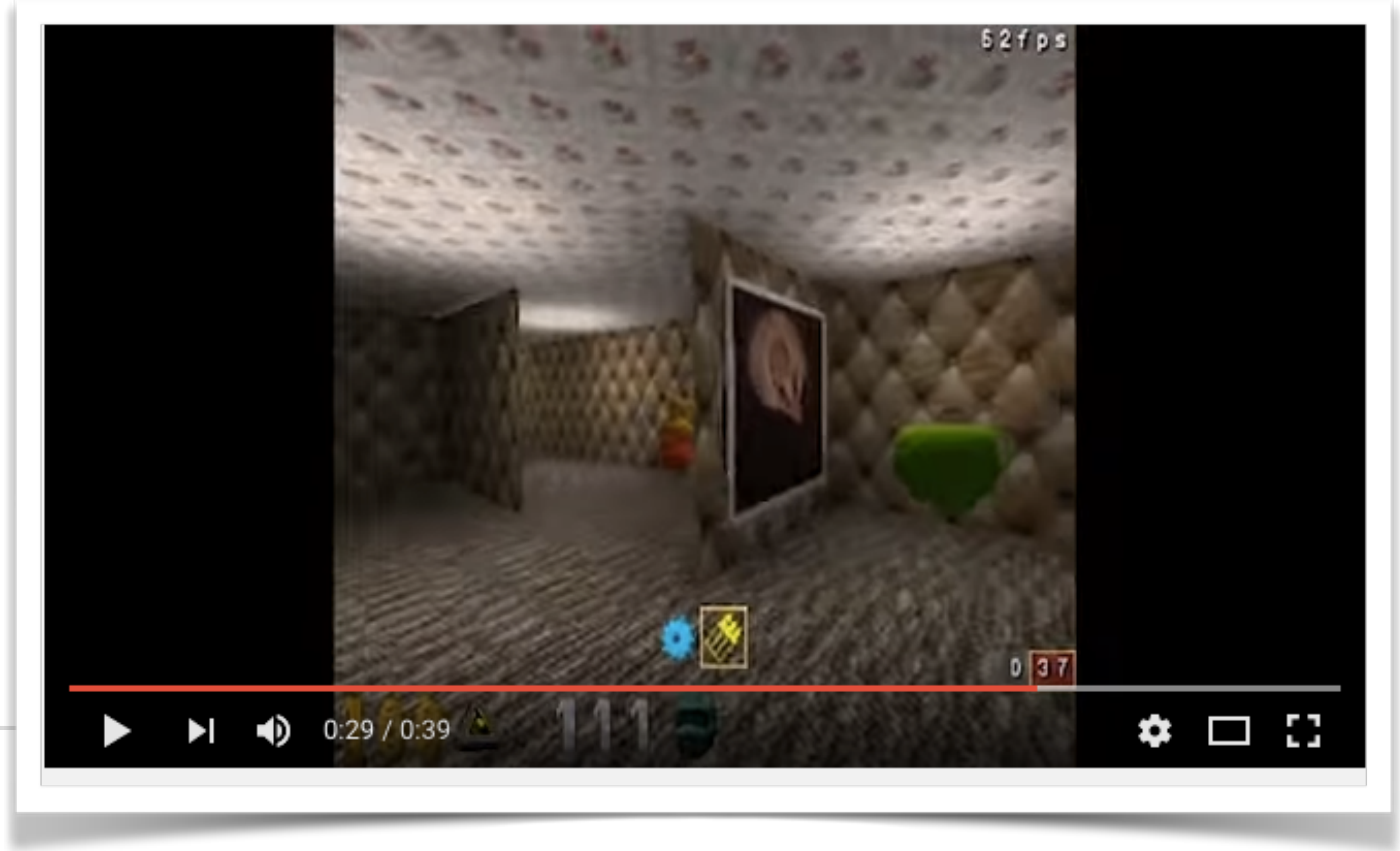
Reinforcement learning with RNNs



Reinforcement learning of 3D game controllers

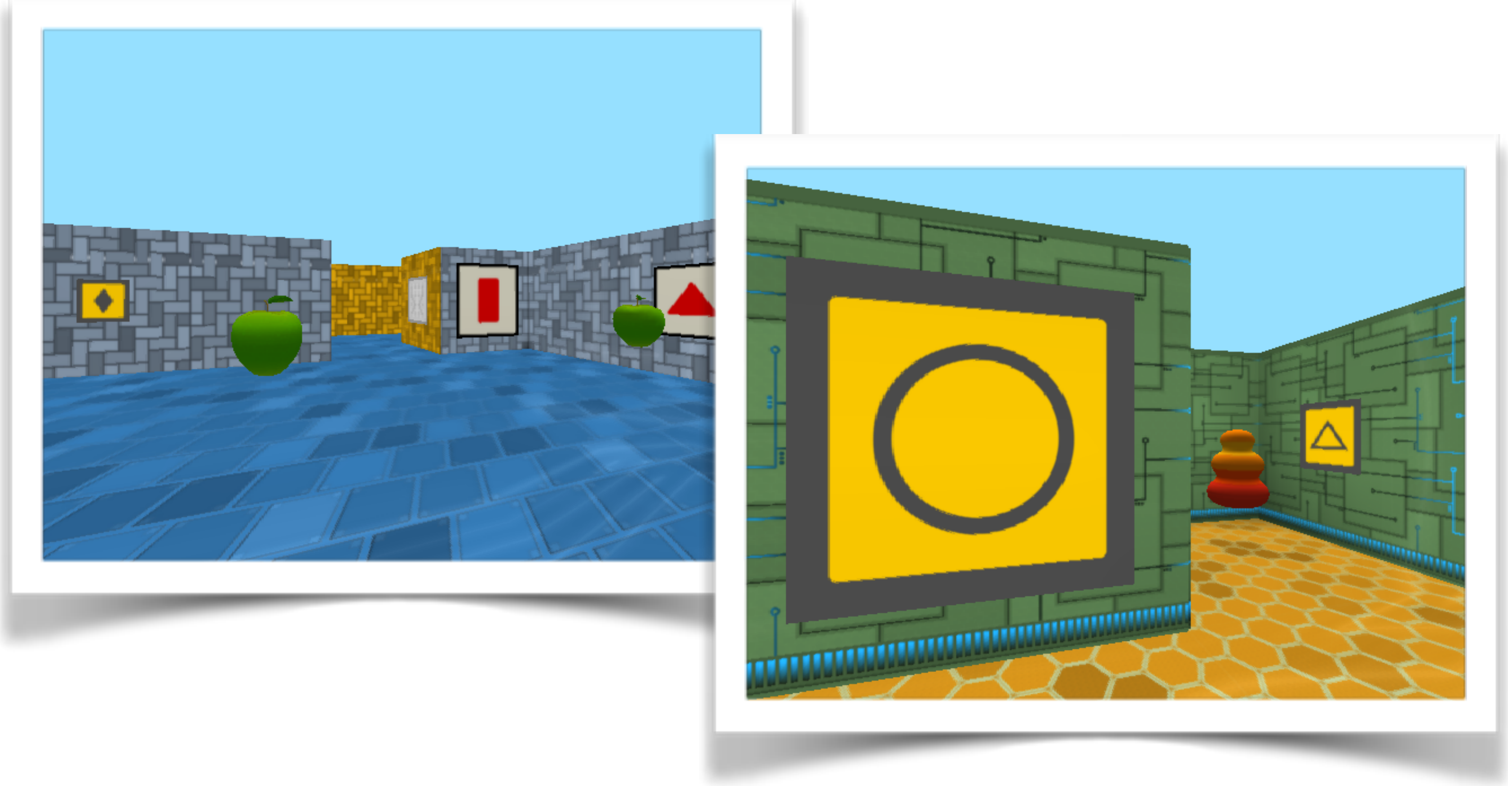


convolutional network + LSTM
trained on 120M frames of video game emulator
using Reinforcement Learning
Asynchronous Advantage Actor-Critic



[Mnih, Badia et al. (2016)
"Asynchronous methods for deep reinforcement learning", ICML]

Learning to navigate in complex environments



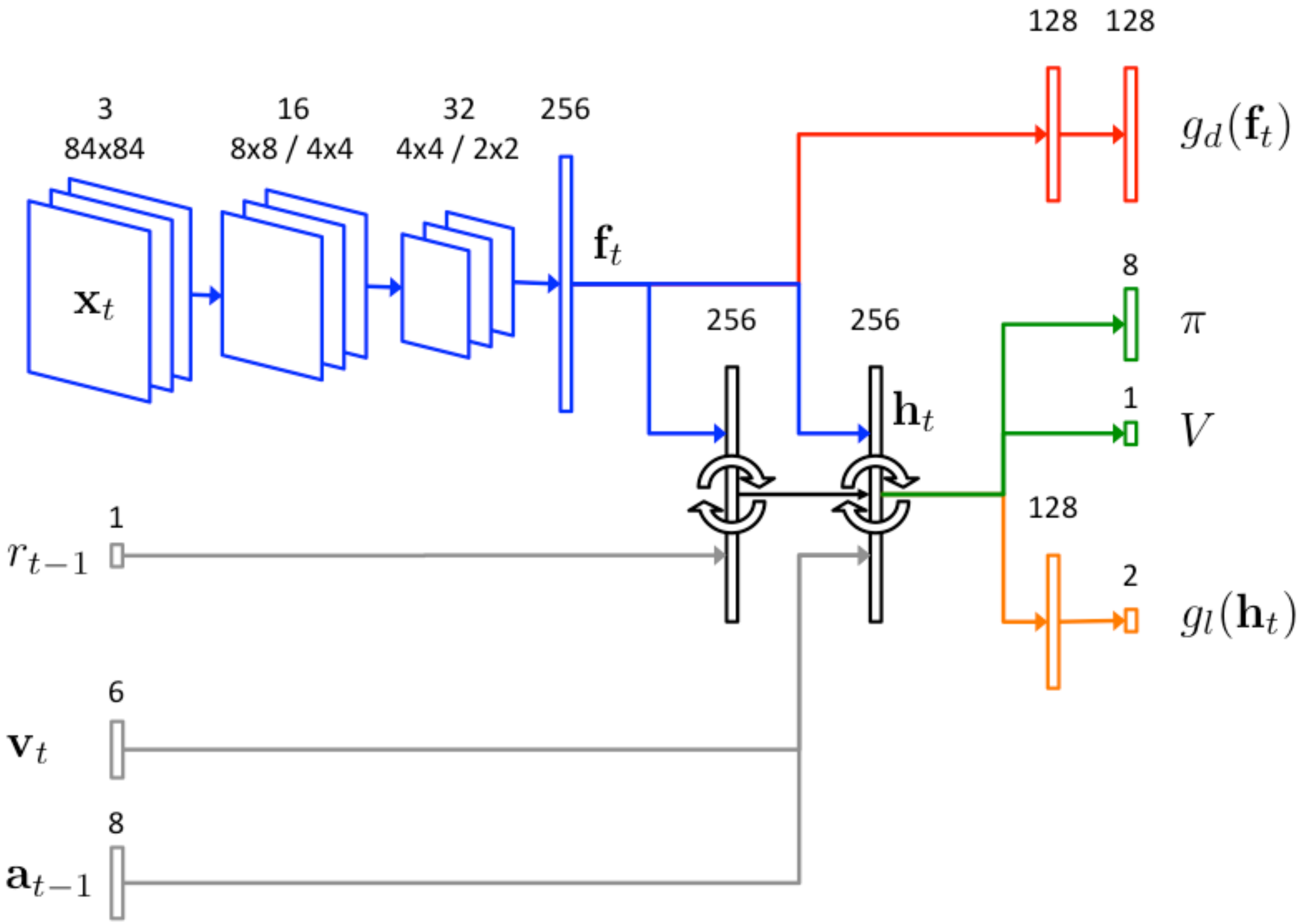
Multi-task learning
on multiple input modalities

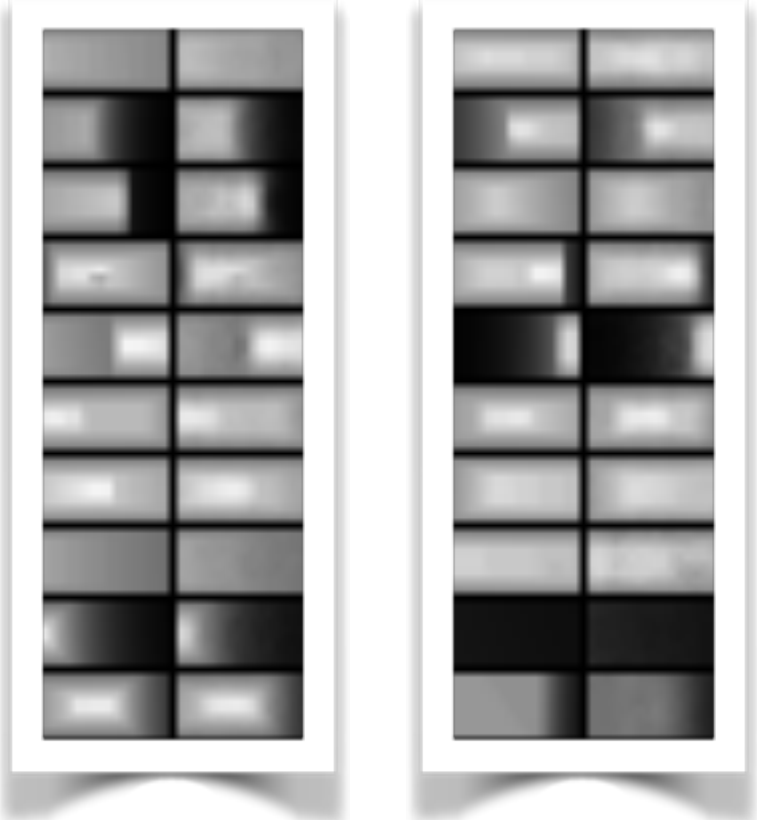
Visual inputs

Previous reward

Agent-relative velocity

Previous action

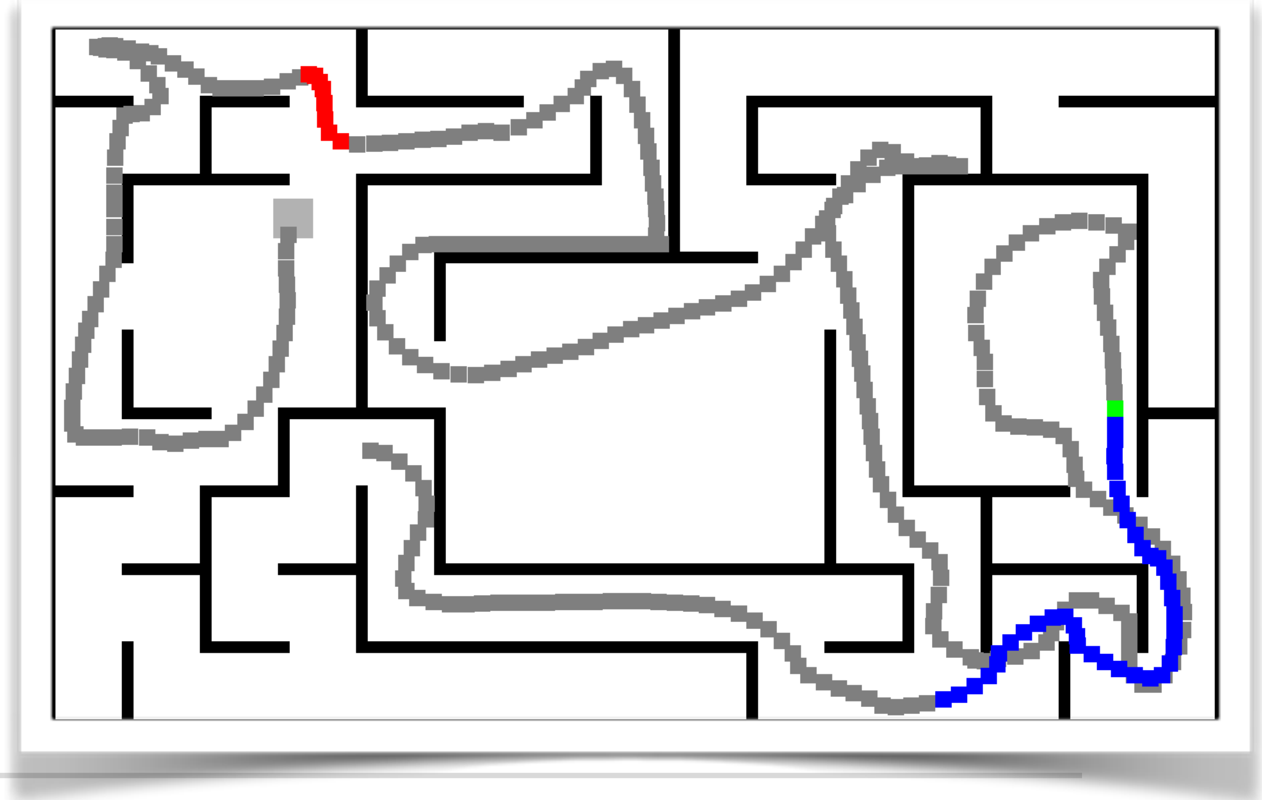




Depth prediction

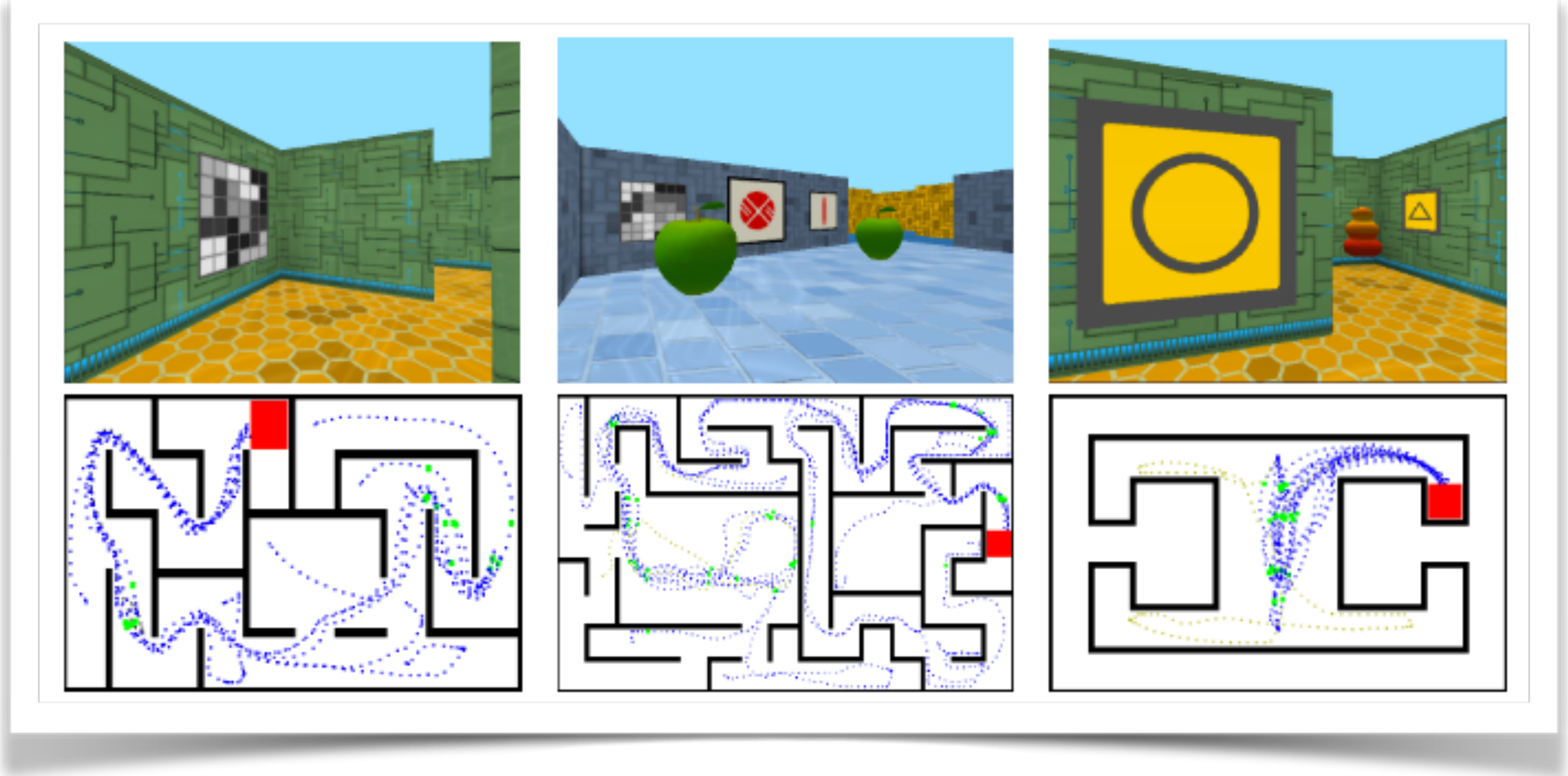
Reinforcement learning

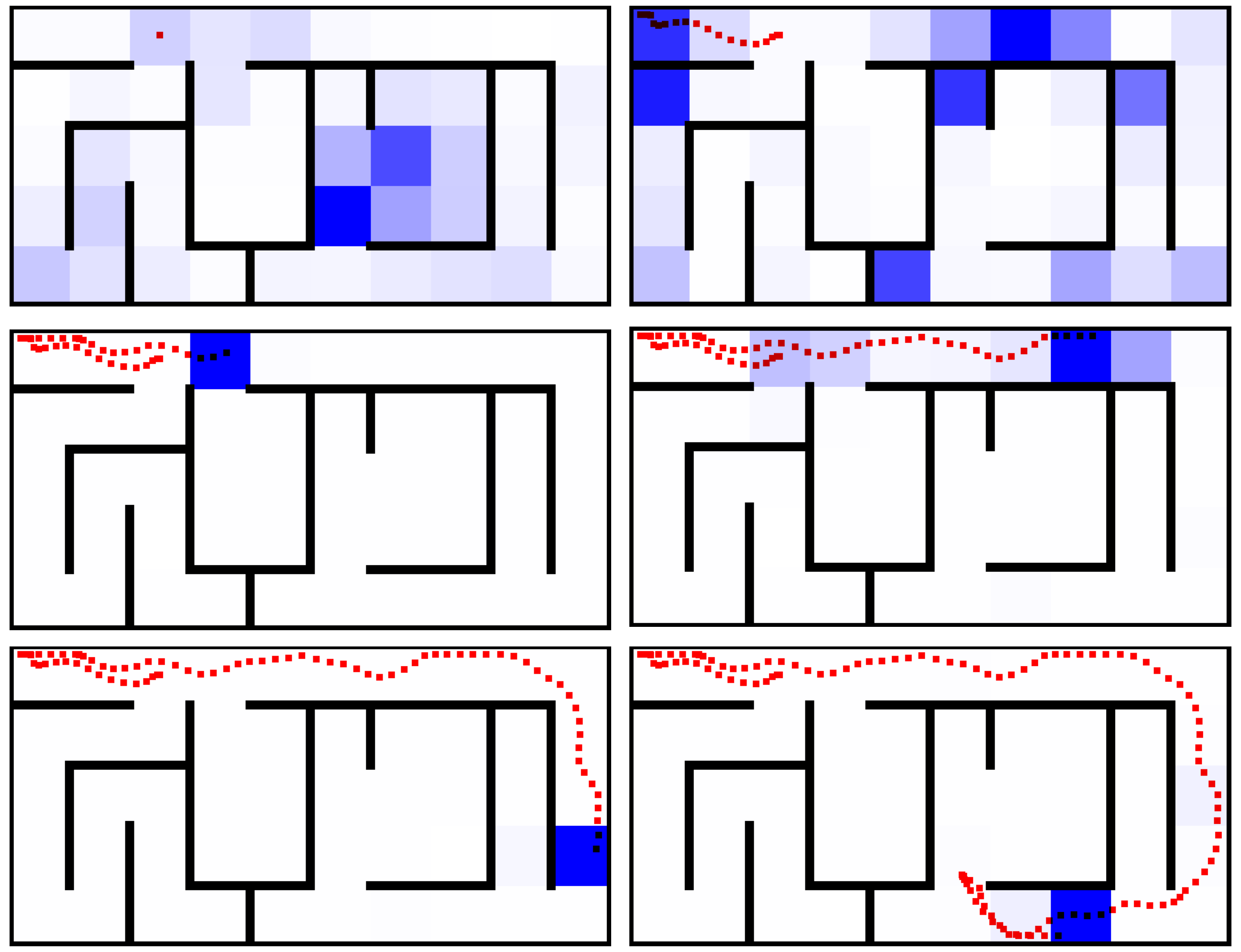
Loop closure prediction



[Mirowski, Pascanu, Viola, Soyer, Ballard, Banino, Denil, Goroshin, Sifre, Kavukcuoglu, Kumaran and Hadsell (2016) "Learning to navigate in complex environments", *under review*]

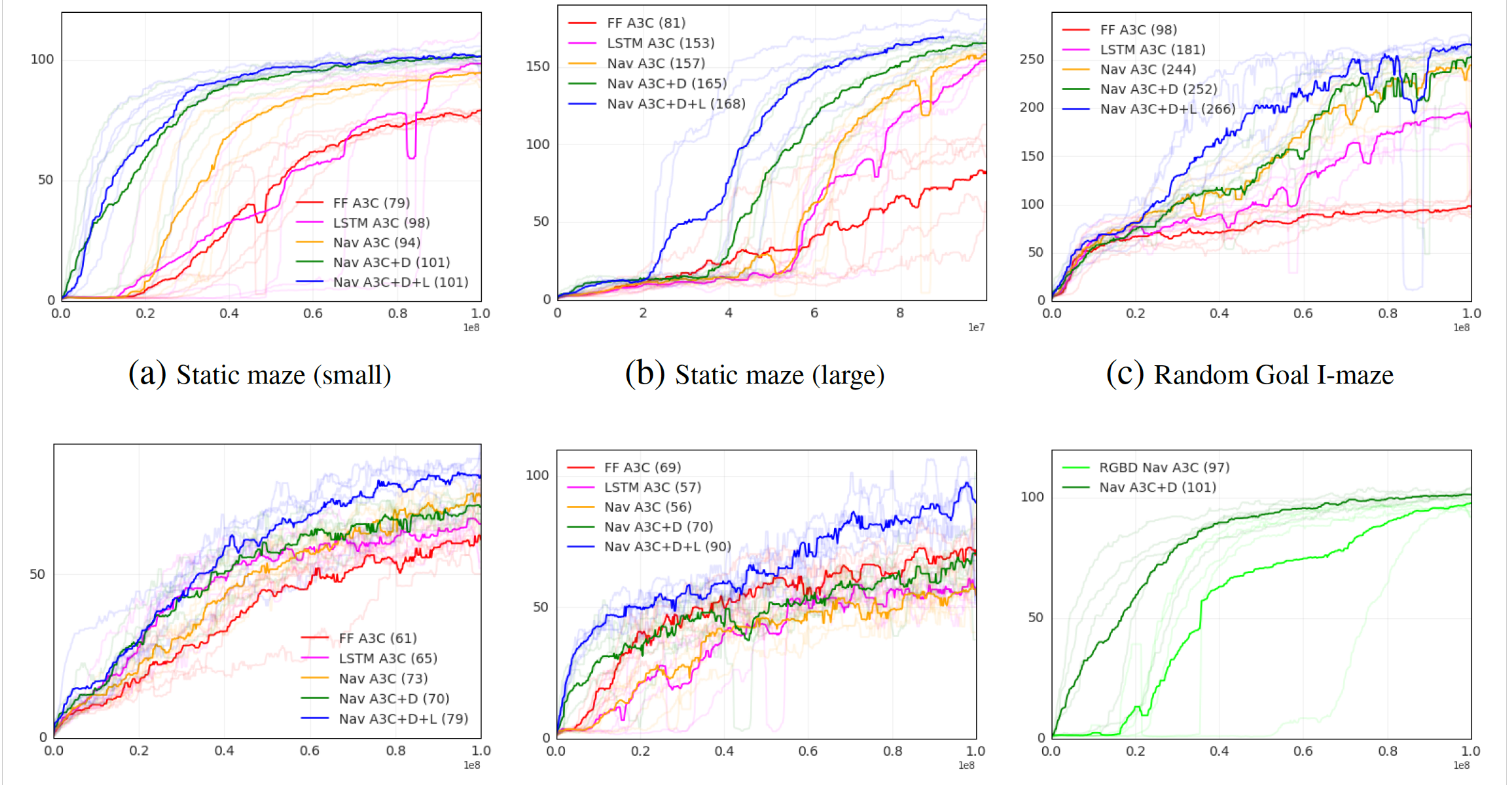
Learning to navigate in complex environments





Position decoding

Higher rewards, faster than **plain convnet** or **convnet+LSTM**



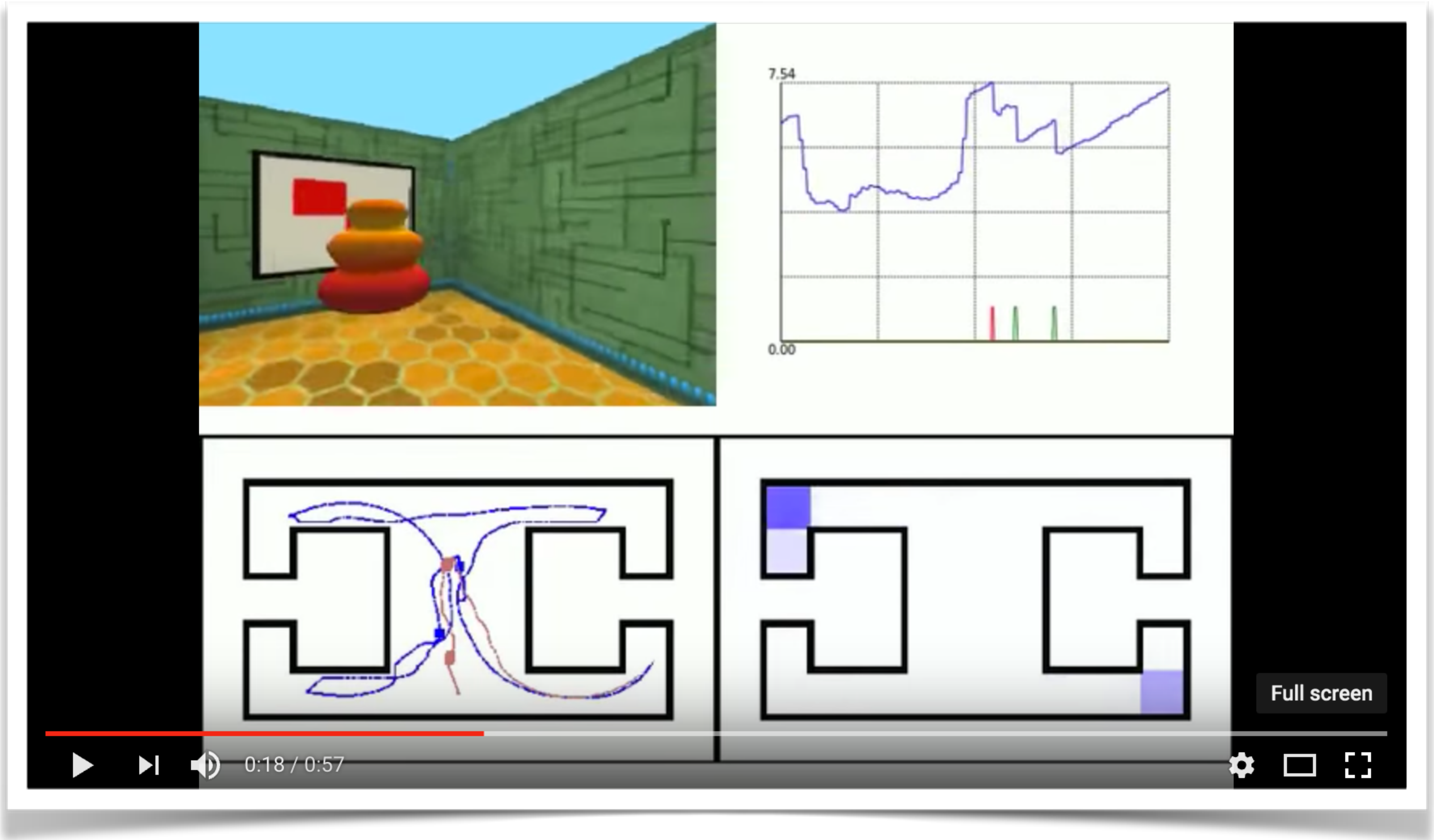
(a) Static maze (small) (b) Static maze (large) (c) Random Goal I-maze

(d) Random Goal maze (small) (e) Random Goal maze (large) (f) Static maze: depth input vs target

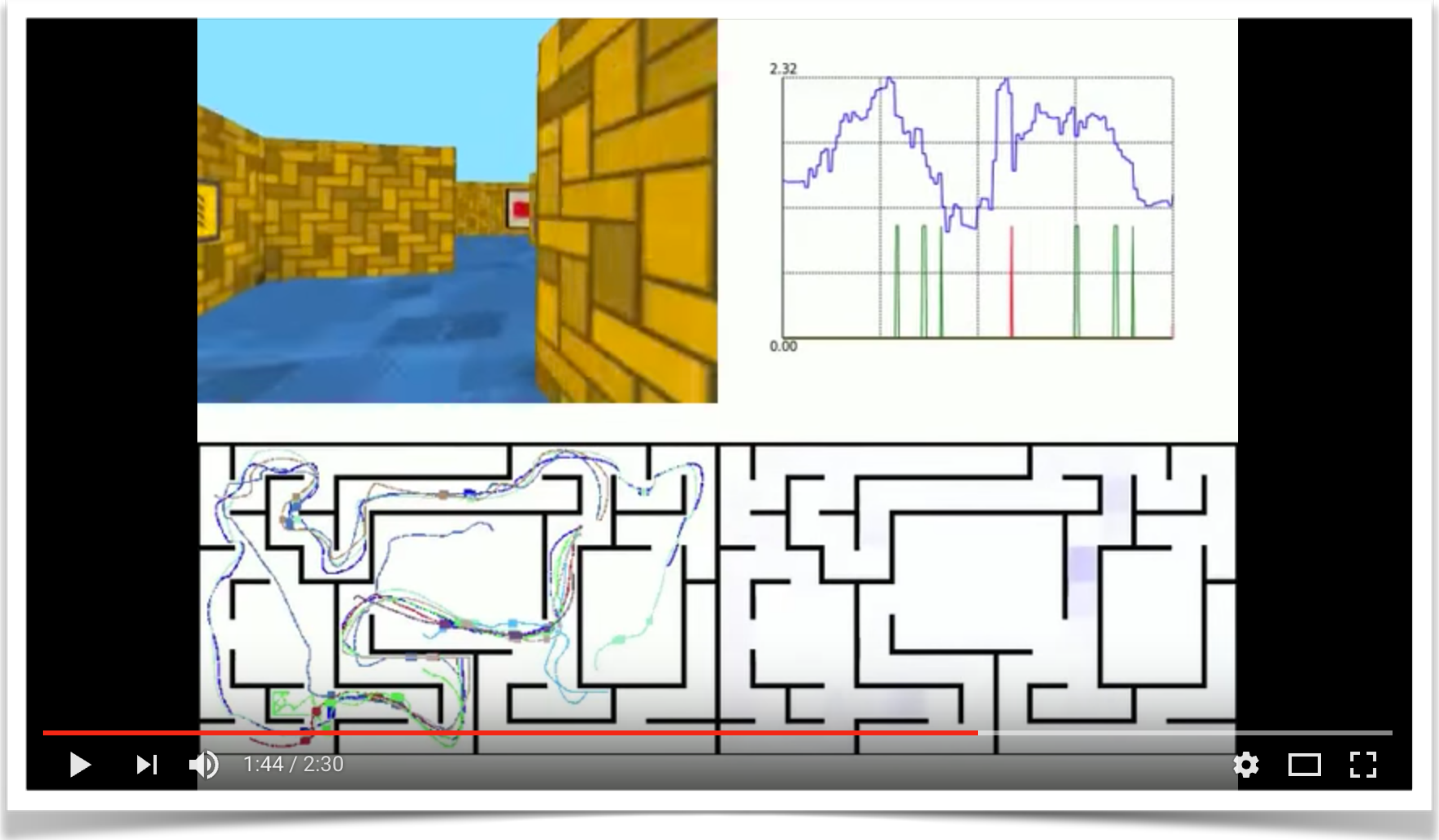
[Mirowski, Pascanu, Viola, Soyer, Ballard, Banino, Denil, Goroshin, Sifre, Kavukcuoglu, Kumaran and Hadsell (2016) "Learning to navigate in complex environments", *under review*]

Learning to navigate in complex environments

Small **I-maze** (goal in **random branch**): remembers which branch contains goal



Large random maze with **random goal**



[Mirowski, Pascanu, Viola, Soyer, Ballard, Banino, Denil, Goroshin, Sifre, Kavukcuoglu, Kumaran and Hadsell (2016) “Learning to navigate in complex environments”, *under review*]

Thank you!

piotr.mirowski@computer.org

These slides will also be posted on:
piotrmirowski.wordpress.com

www.deepmind.com/**research**/publications/

www.deepmind.com/**careers**/

Take-aways

Lecture notes in Deep Learning (Nando de Freitas, Oxford):
“Recurrent nets and LSTM”
https://www.youtube.com/watch?v=56TYLaQN4N8
“Generating sequences with RNNs”
https://www.youtube.com/watch?v=-yX1SYeDHbg

LSTM code for Lua+**Torch7**:
https://github.com/karpathy/char-rnn/
https://github.com/jcjohnson/torch-rnn

LSTM code for Python+**TensorFlow**:
https://www.tensorflow.org/versions/r0.8/tutorials/recurrent/index.html