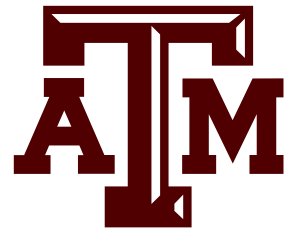


# CSCE 689: Special Topics in Trustworthy NLP

## Lecture 5: Attention, Transformers

Kuan-Hao Huang  
khhuang@tamu.edu



(Some slides adapted from Chris Manning, Karthik Narasimhan, Danqi Chen, and Vivian Chen)

# LaTeX Assignment

- LaTeX Assignment (1%)
- Due: Sep 11, 11:59pm

## CSCE 689: LaTeX Assignment

Your Name  
Your UID and email

### Overview

This assignment is designed to give you practice with LaTeX, which you are expected to use for your literature review, project proposal, and final report in this course.

### Instructions

For this assignment, you will create a PDF containing your answers using LaTeX. If this is your first time working with LaTeX, we recommend starting with this [short tutorial](#), which covers the basic features you will need for this course. Please use the Association for Computational Linguistics LaTeX template ([link](#)), a template widely used in major NLP conferences. We suggest using [Overleaf](#) as your online editor, since it automatically manages packages for you.

By default, the template is set to *review mode*. To switch to *final mode*, change:

Review Mode (Default)

`\usepackage[review]{acl}`

to:

Final Mode

`\usepackage[final]{acl}`

This allows you to display the author information. Be sure to include your name, UIN, and email.

The following sections contain questions on some commonly used LaTeX commands. There are a total of 100 points for this assignment. Please answer each question in a separate *section*, and submit the final PDF generated using LaTeX.

### 1 Including Equations [20pts]

Typeset the following expression using LaTeX:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \mathbf{w}_j} = -\frac{1}{m} \sum_{i=1}^m (y_i - \sigma(z_i)) \cdot \mathbf{x}_{i,j}$$

### 2 Including Images [20pts]

Select a picture of a cat and include it with a caption. The figure below is provided as an example.



Figure 1: This is a cute cat!

### 3 Including Tables [20pts]

Create a table that displays your name, UIN, and email. You can follow the example below as a template.

|       |                  |
|-------|------------------|
| Name  | Kuan-Hao Huang   |
| UIN   | 123456789        |
| Email | khhuang@tamu.edu |

Table 1: Example table.

### 4 Including Lists [20pts]


Create a list that displays your name, UIN, and email. You can follow the example below as a template.

- Name: Kuan-Hao Huang
- UIN: 123456789
- Email: khhuang@tamu.edu

### 5 Including Citations [20pts]

Use *BibTeX* to include the following paper: Paper 1 ([Vaswani et al., 2017](#)) and Paper 2 ([Devlin et al., 2019](#)). You can learn more about *BibTeX* [here](#).

# Course Schedule Change

|   |       |   |   |         |
|---|-------|---|---|---------|
|   | 10/15 | Project Highlight Presentations             |   |         |
| W9  | 10/20 | Multimodal Models                           | <a href="#">When and why vision-language models behave like bags-of-words, and what to do about it?</a> , ICLR 2023<br><a href="#">What's "up" with vision-language models? Investigating their struggle with spatial reasoning</a> , EMNLP 2023<br><a href="#">Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs</a> , CVPR 2024<br><a href="#">Why Is Spatial Reasoning Hard for VLMs? An Attention Mechanism Perspective on Focus Areas</a> , ICML 2025 | Student |
|   | 10/22 | In-Context Learning                         | <a href="#">Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?</a> , EMNLP 2022<br><a href="#">Not All Demonstration Examples are Equally Beneficial: Reweighting Demonstration Examples for In-Context Learning</a> , EMNLP-Findings 2023<br><a href="#">What Makes a Good Order of Examples in In-Context Learning</a> , ACL 2024<br><a href="#">Revisiting Demonstration Selection Strategies in In-Context Learning</a> , ACL 2024               | Student |
|  |       |   |   |         |
|   | 10/15 | Multimodal Models                           | <a href="#">When and why vision-language models behave like bags-of-words, and what to do about it?</a> , ICLR 2023<br><a href="#">What's "up" with vision-language models? Investigating their struggle with spatial reasoning</a> , EMNLP 2023<br><a href="#">Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs</a> , CVPR 2024<br><a href="#">Why Is Spatial Reasoning Hard for VLMs? An Attention Mechanism Perspective on Focus Areas</a> , ICML 2025 | Student |
| W9  | 10/20 | In-Context Learning                         | <a href="#">Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?</a> , EMNLP 2022<br><a href="#">Not All Demonstration Examples are Equally Beneficial: Reweighting Demonstration Examples for In-Context Learning</a> , EMNLP-Findings 2023<br><a href="#">What Makes a Good Order of Examples in In-Context Learning</a> , ACL 2024<br><a href="#">Revisiting Demonstration Selection Strategies in In-Context Learning</a> , ACL 2024               | Student |
|   | 10/22 | Project Highlight Presentations<br>(Remote) |   |         |

# Topic Sign-Up

- Sign-up: <https://tinyurl.com/2p9mr2wa>
  - Log in with TAMU account
  - Due: Sep 10 before lecture

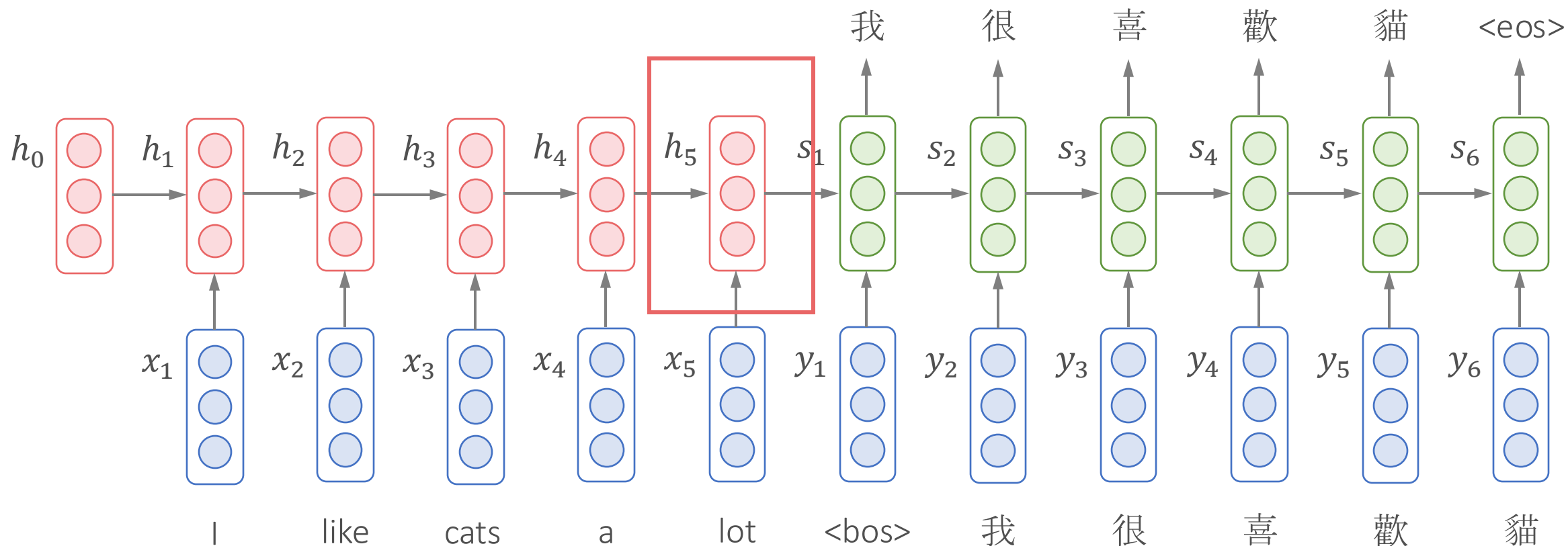
| Put Preference with Topic IDs |                                   |                      |              |              |              |              |              |              |              |              |              |
|-------------------------------|-----------------------------------|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Team                          | Member 1                          | Member 2 (optional)  | Preference 1 | Preference 2 | Preference 3 | Preference 4 | Preference 5 | Preference 6 | Preference 7 | Preference 8 | Preference 9 |
| Example                       | First_name Last_name              | First_name Last_name | 4            | 10           | 1            | 7            | 3            | 9            | 8            | 6            | 12           |
| 1                             | Kunal Jain                        |                      | 3            | 12           | 8            | 6            | 11           | 10           | 9            |              |              |
| 2                             | Muhan Gao                         |                      | 5            | 6            | 4            | 9            | 10           | 7            | 8            | 11           | 12           |
| 3                             | Serhii Honcharenko                |                      | 9            | 10           | 4            | 7            | 6            | 1            |              |              |              |
| 4                             | Oscar Chew                        |                      | 9            | 3            | 11           | 7            | 8            | 10           | 4            | 2            | 1            |
| 5                             | Junggeun Do                       |                      | 8            | 7            | 11           | 10           | 2            | 6            | 12           | 11           | 5            |
| 6                             | Jiongran Wang                     |                      | 3            | 9            | 10           | 6            | 4            | 2            | 7            | 5            | 1            |
| 7                             | Sicong Liang                      |                      | 10           | 9            | 3            | 13           | 7            |              |              |              |              |
| 8                             | Kowsalya Balamuralei Umamaheswari |                      | 3            | 10           | 12           | 8            |              |              |              |              |              |
| 9                             | Yi Wen                            |                      | 12           | 4            | 7            | 9            | 6            | 8            |              |              |              |
| 10                            | Quang Nguyen                      |                      | 6            | 3            | 4            | 12           | 2            | 1            | 8            | 9            | 7            |
| 11                            | Aaron Xu                          |                      | 4            | 8            | 7            | 3            | 9            | 6            | 11           | 2            | 1            |
| 12                            | Bhaskar Ruthvik Bikkina           |                      | 7            | 11           | 12           | 9            | 10           | 5            | 3            | 4            | 6            |
| 13                            |                                   |                      |              |              |              |              |              |              |              |              |              |
| 14                            |                                   |                      |              |              |              |              |              |              |              |              |              |
| 15                            |                                   |                      |              |              |              |              |              |              |              |              |              |
| 16                            |                                   |                      |              |              |              |              |              |              |              |              |              |
| 17                            |                                   |                      |              |              |              |              |              |              |              |              |              |

# Topic Study

- Topic Study (30%)
  - Literature Review (15%) [Due: 10/2]
    - Examples
      - [Survey of Prompting Methods](#)
      - [Survey of Mitigating Gender Bias](#)
      - [Survey of AI Alignment](#)
  - Topic Presentation (15%)
    - Email your slides to the instructor **at least 2 days** before your presentation

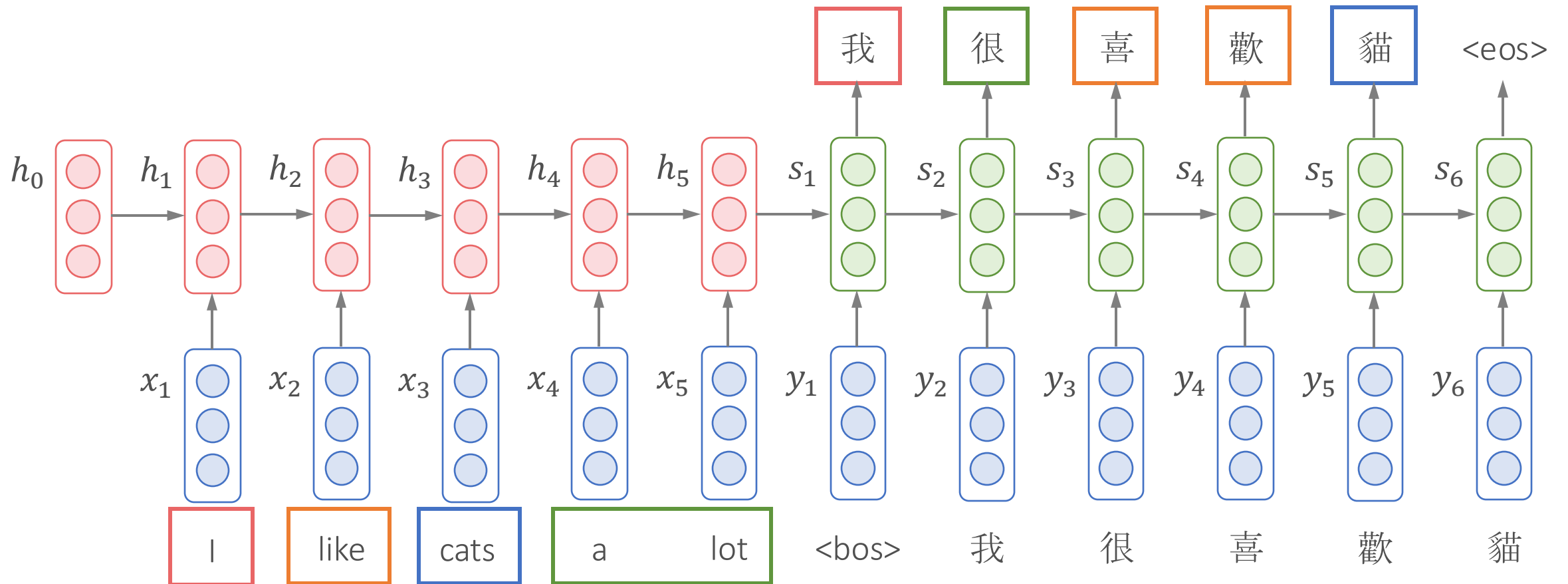
# Seq2Seq: Bottleneck

- A single vector needs to capture **all the information** about source sentence
- Longer sequences can still lead to **vanishing gradients**

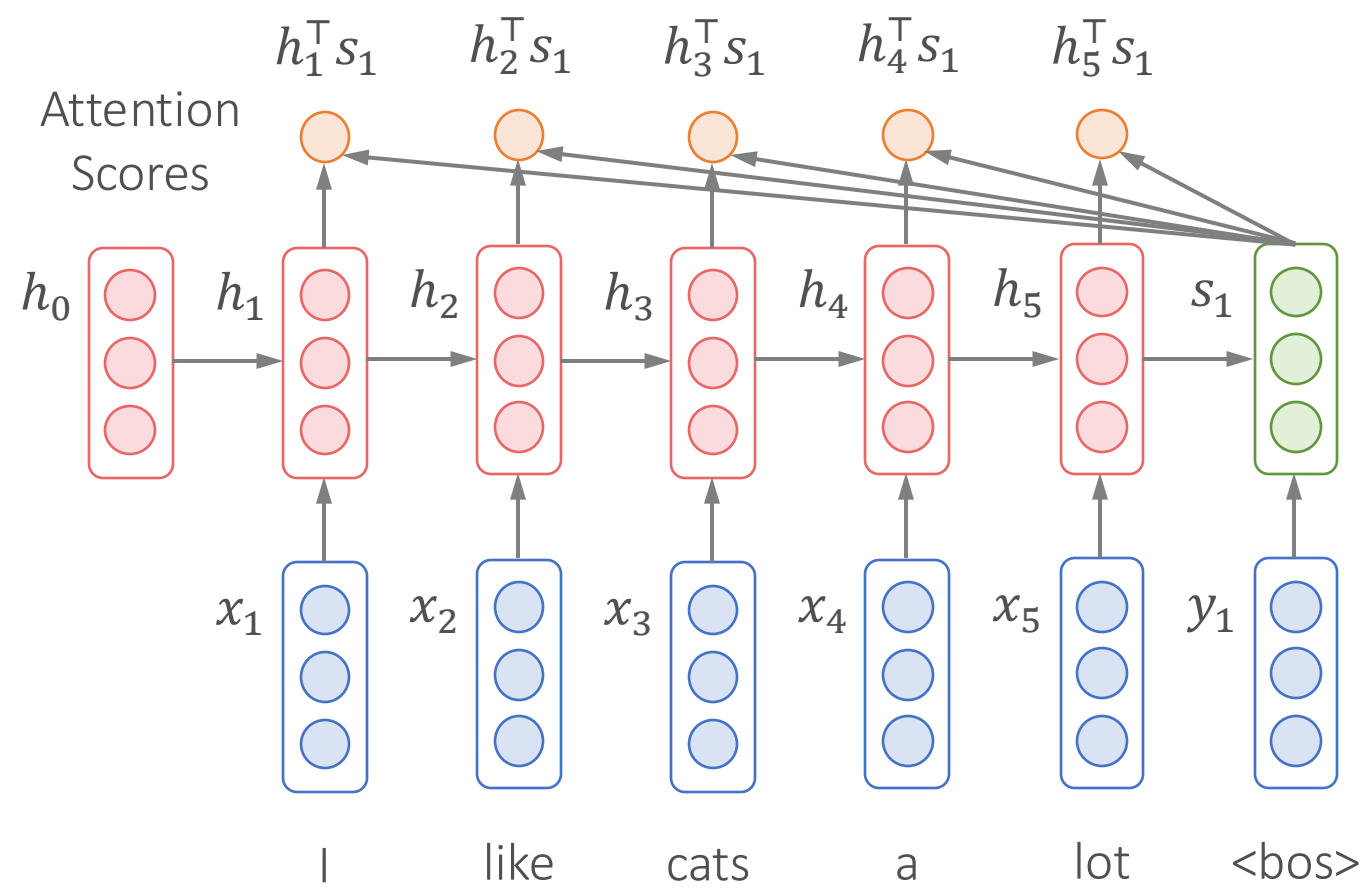


# Focus on A Particular Part When Decoding

- Each token classification requires different part of information from source sentence



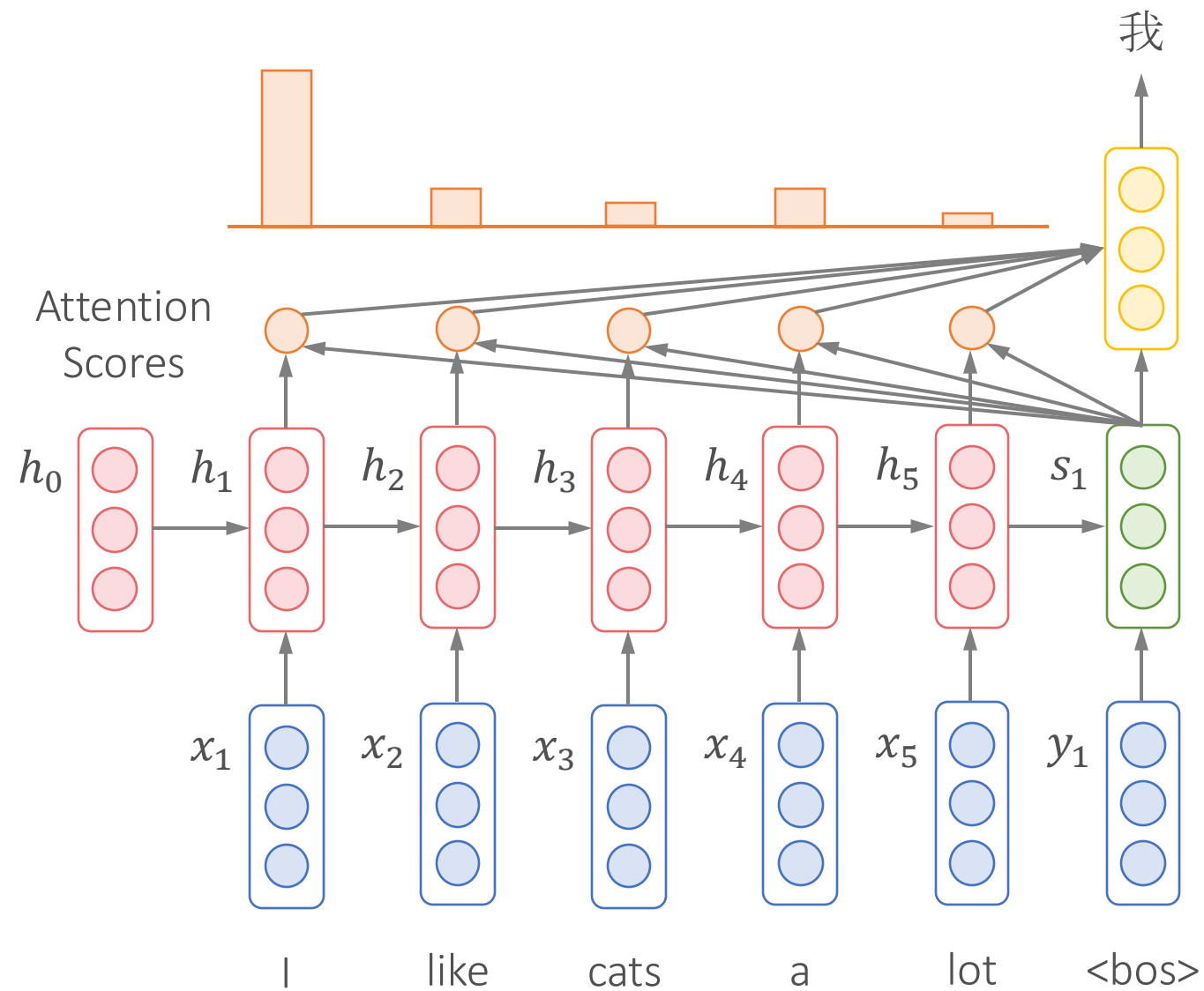
# RNN with Attention



Attention Scores  $\alpha_i = h_i^T s_1$



# RNN with Attention



Attention Scores

$$\alpha_i = h_i^\top s_1$$

Normalized  
Attention Scores

$$\hat{\alpha}_i = \text{softmax}(\alpha_i)$$

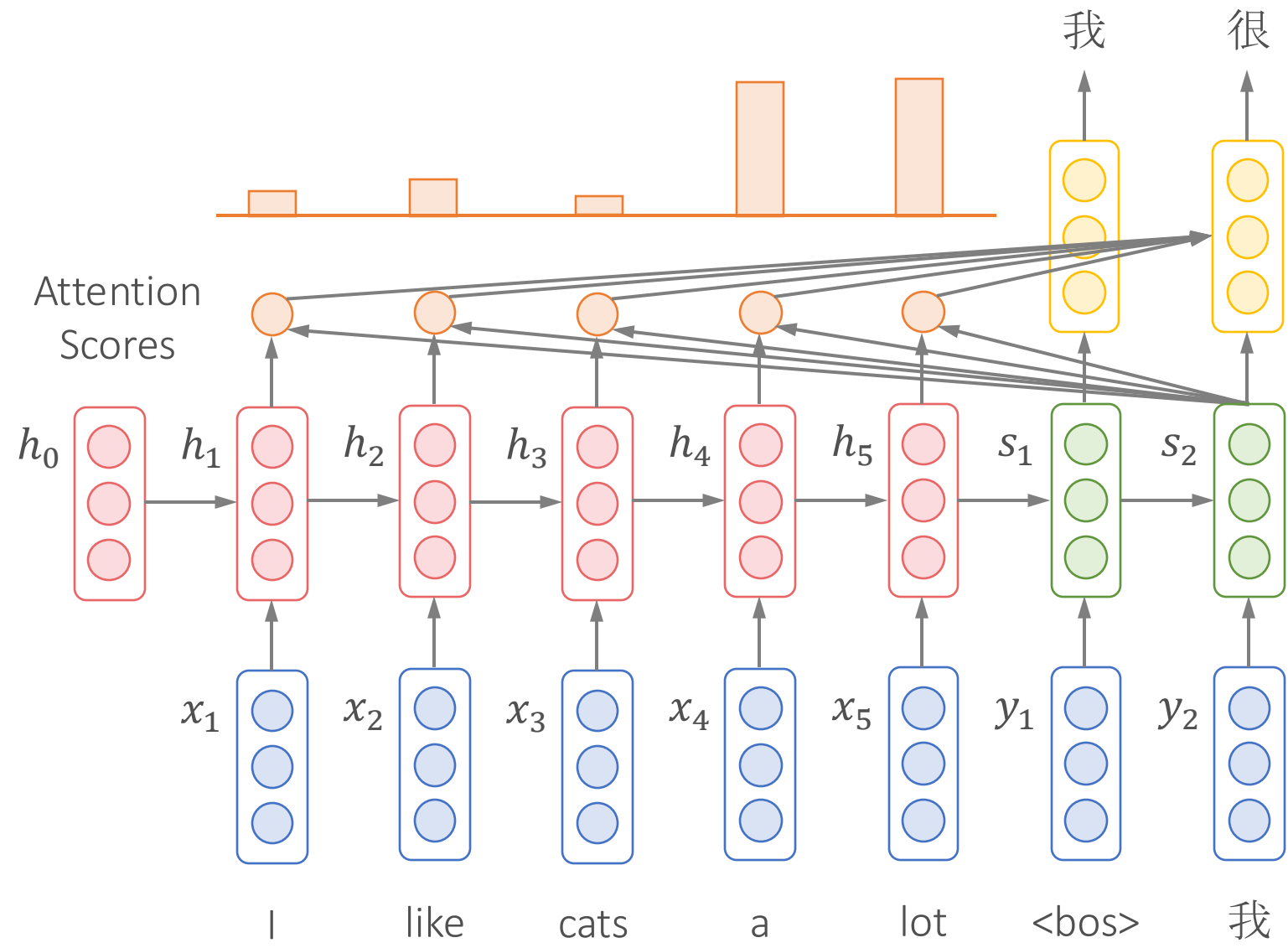
Weighted Sum

$$a = \sum_i \hat{\alpha}_i h_i$$

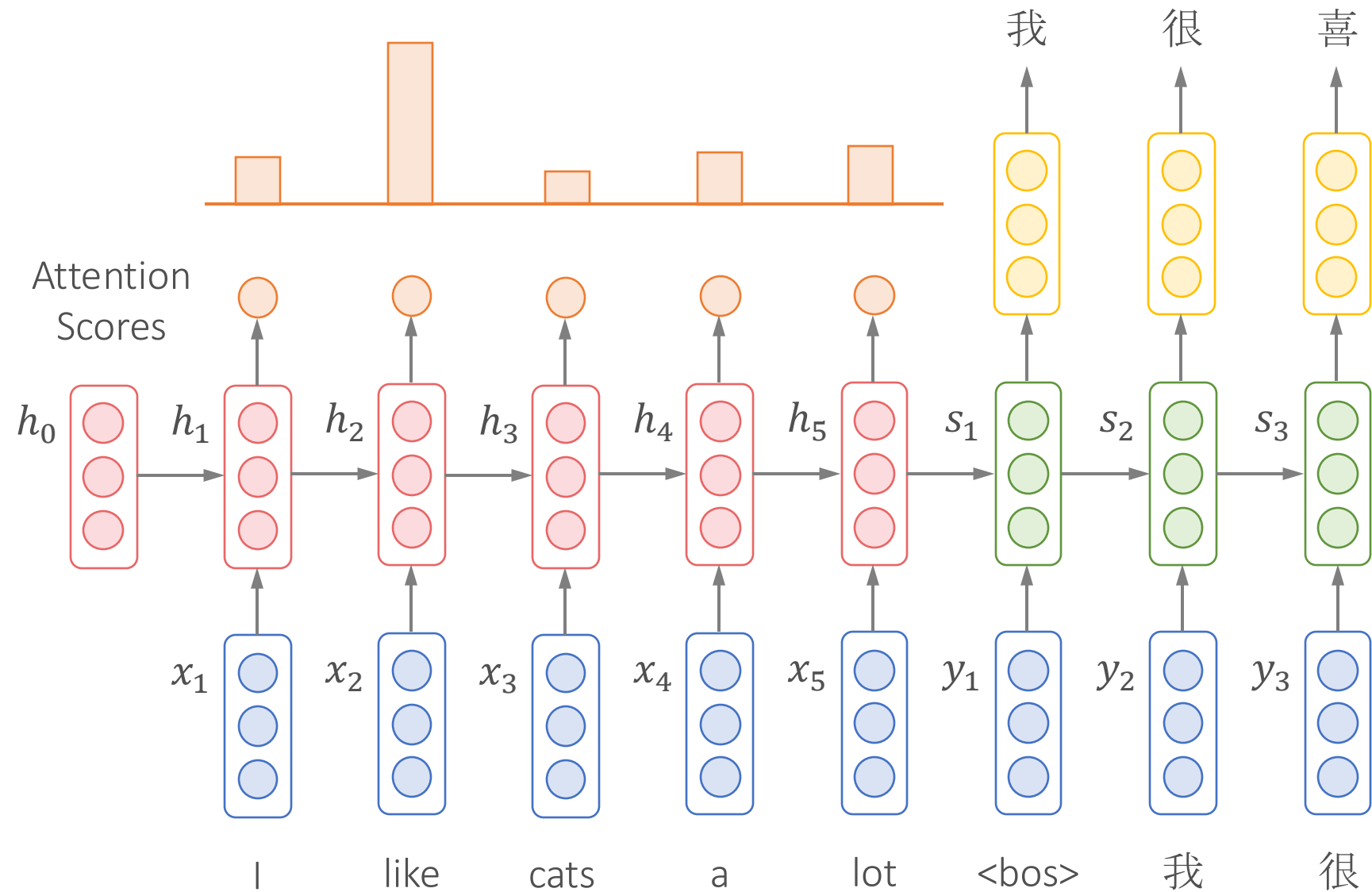
Attention  
Output

$$\tanh(\mathbf{W}[a; s_1])$$

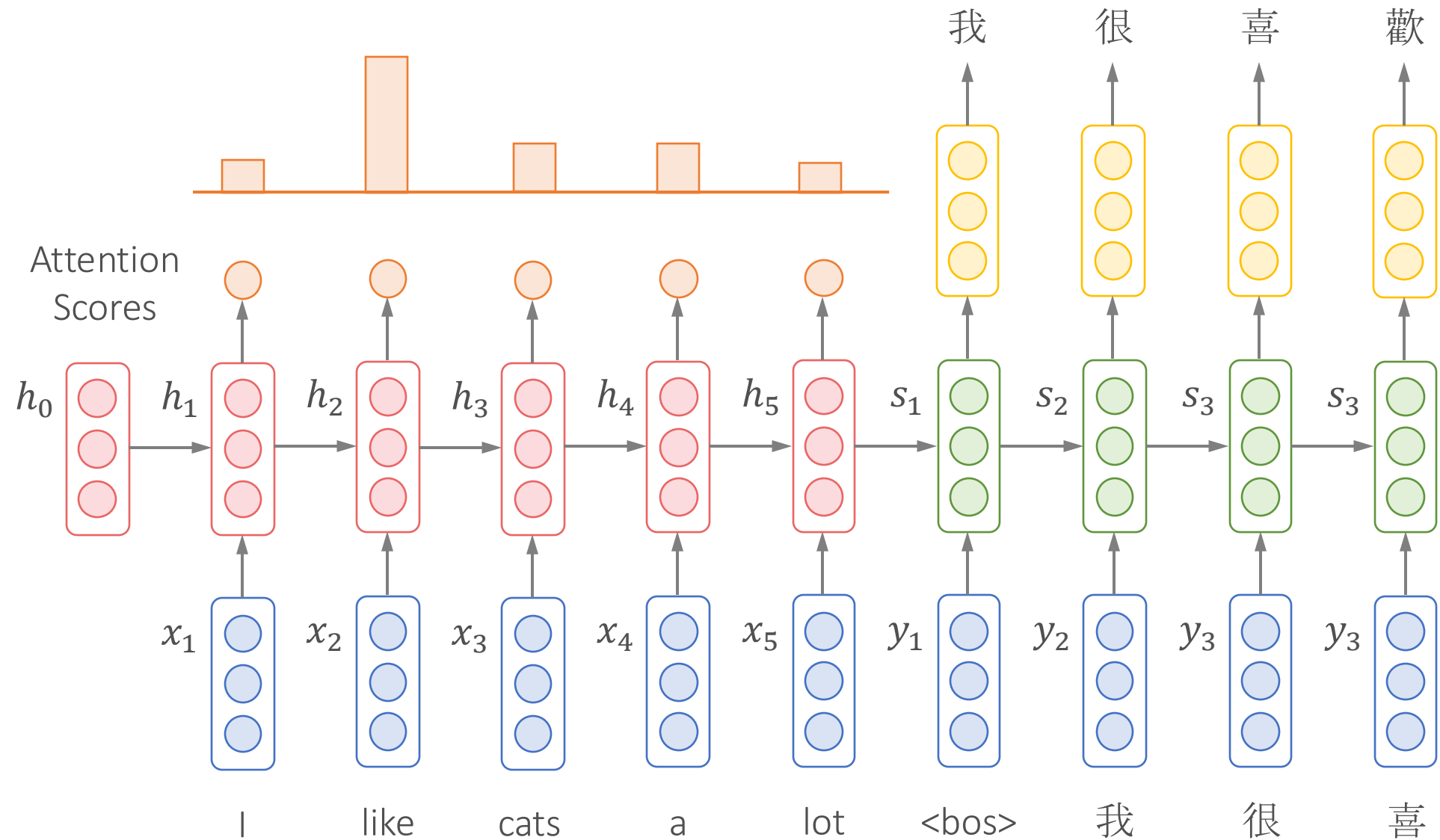
# RNN with Attention



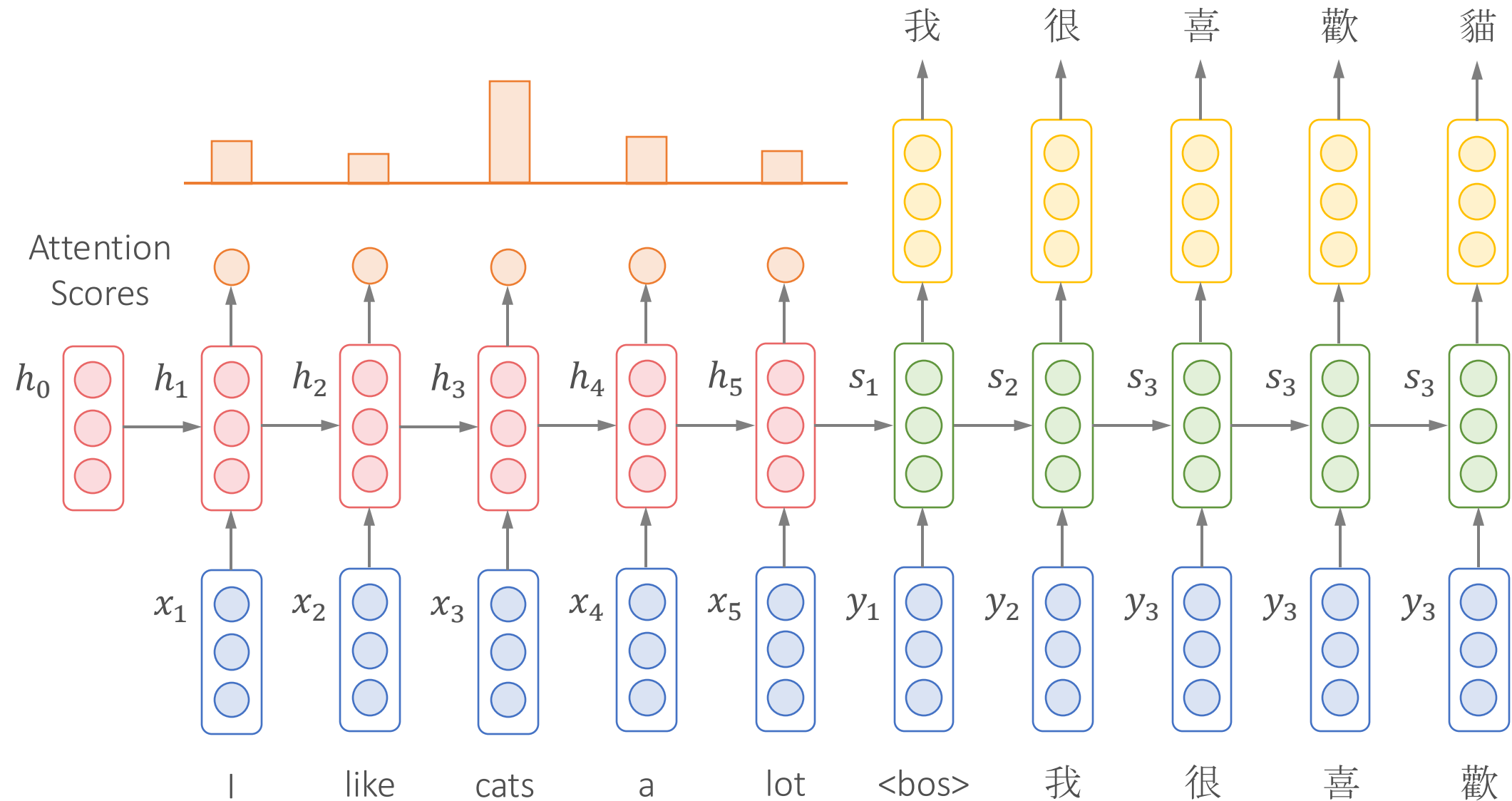
# RNN with Attention



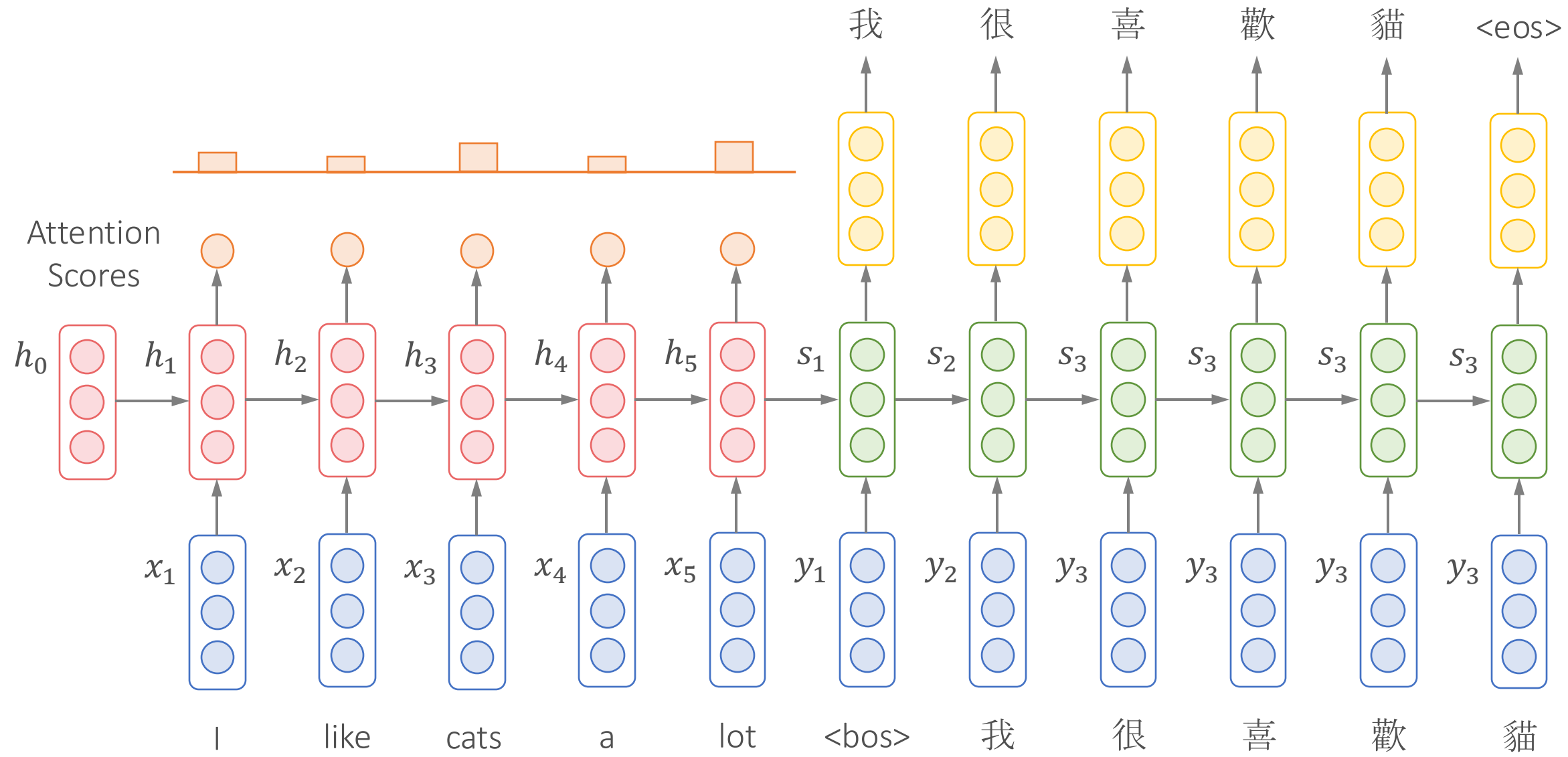
# RNN with Attention



# RNN with Attention



# RNN with Attention



# Different Types of Attention

Dot-Product Attention

$$h_i^\top s_j$$

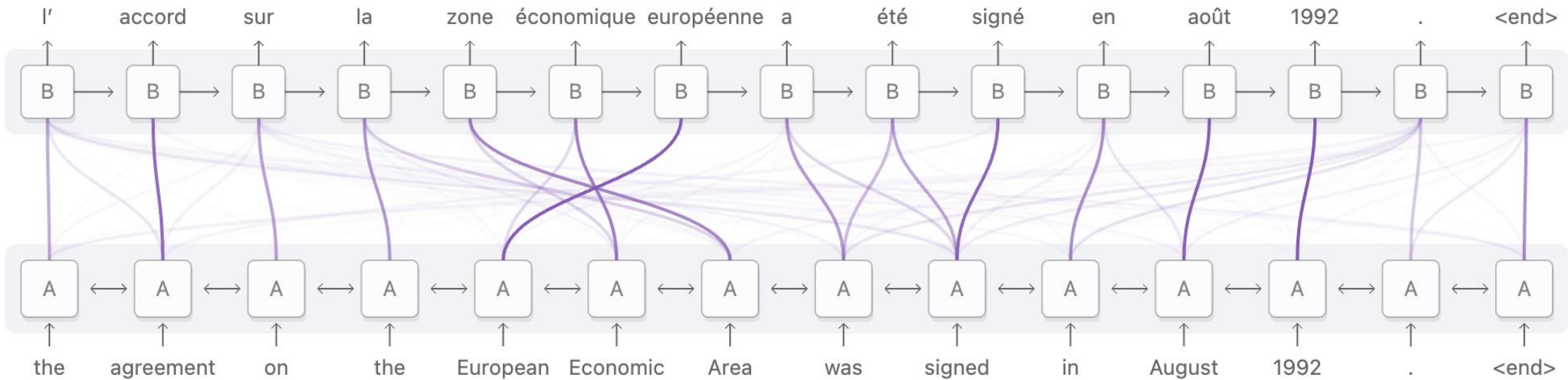
Multiplicative Attention

$$h_i^\top W s_j$$

Additive Attention

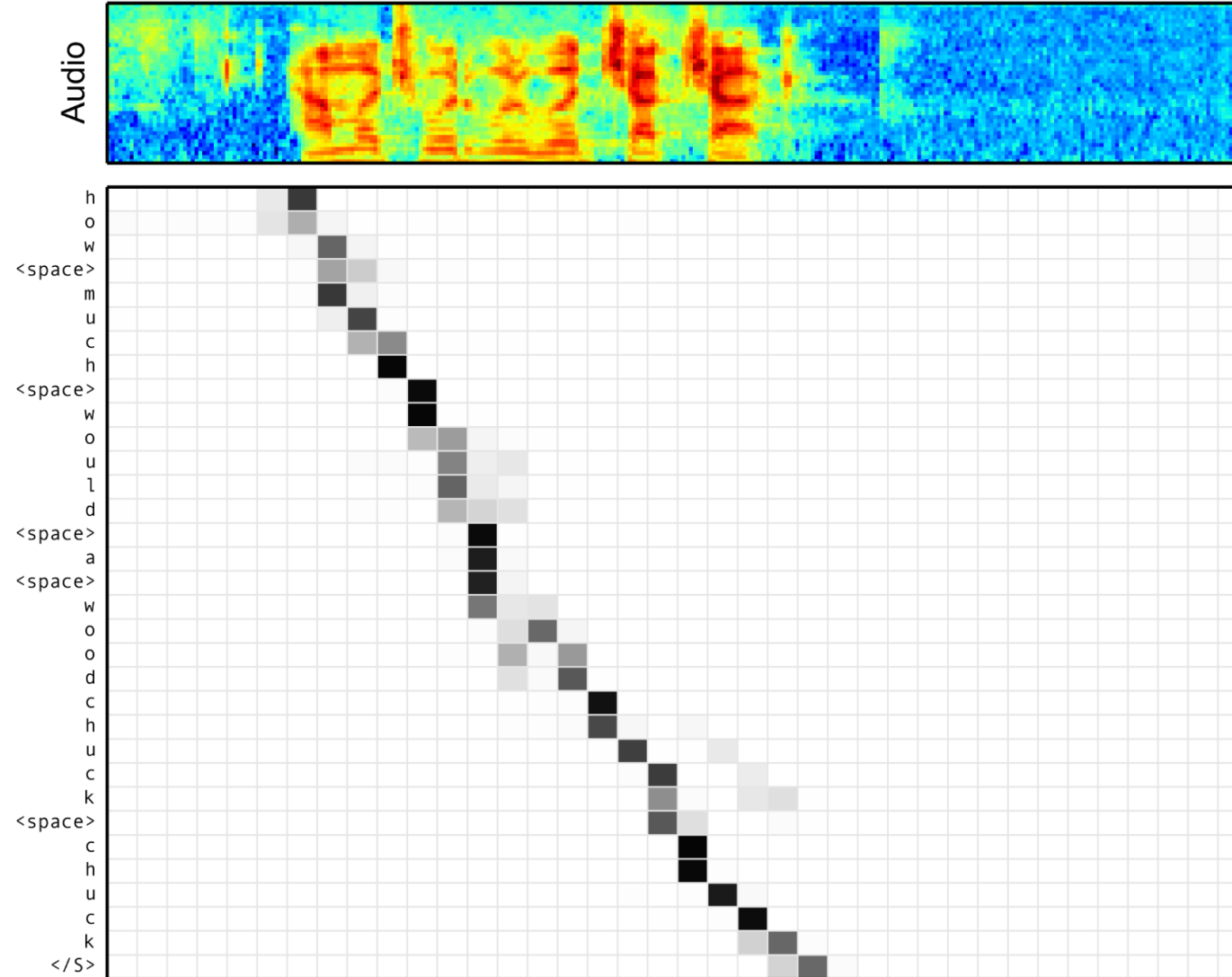
$$v^\top \tanh(W_1 h_i + W_2 s_j)$$

# Machine Translation with Attention

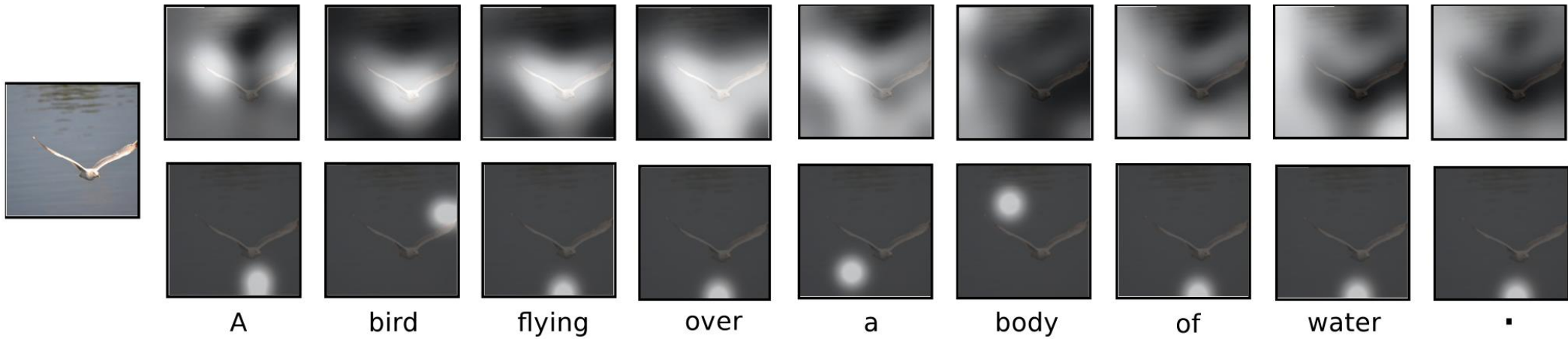




# Speech Recognition with Attention



# Image Captioning with Attention



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.



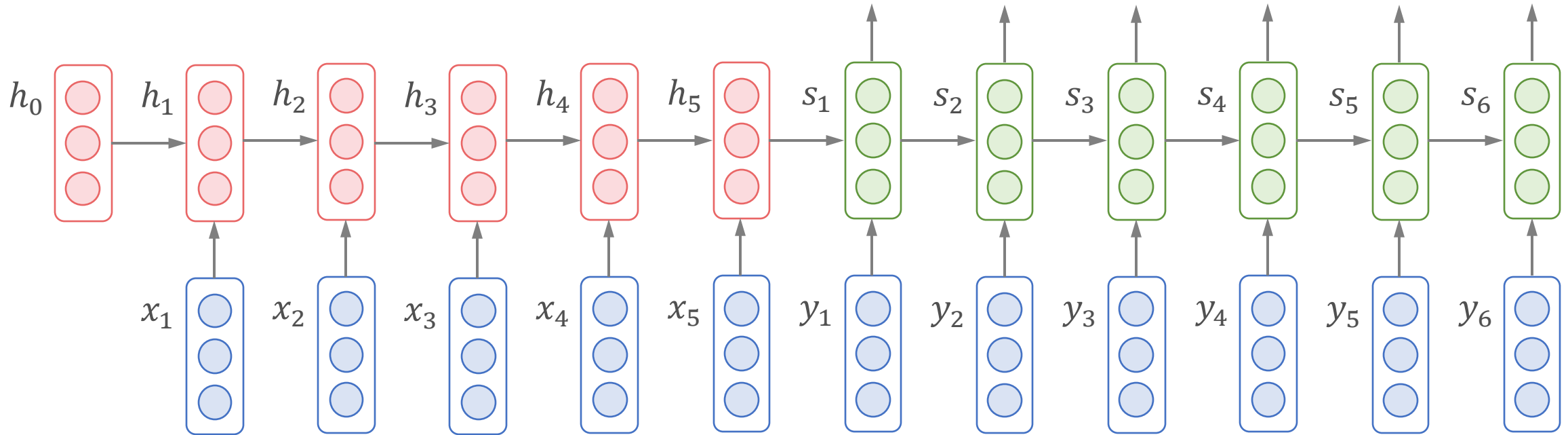
A group of people sitting on a boat in the water.



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

# Issues with RNN

- Longer sequences can lead to vanishing gradients → It is hard to capture long-distance information
- Lack parallelizability



# Transformers: Attention Is All You Need!

---

## Attention Is All You Need

---

**Ashish Vaswani\***  
Google Brain  
avaswani@google.com

**Noam Shazeer\***  
Google Brain  
noam@google.com

**Niki Parmar\***  
Google Research  
nikip@google.com

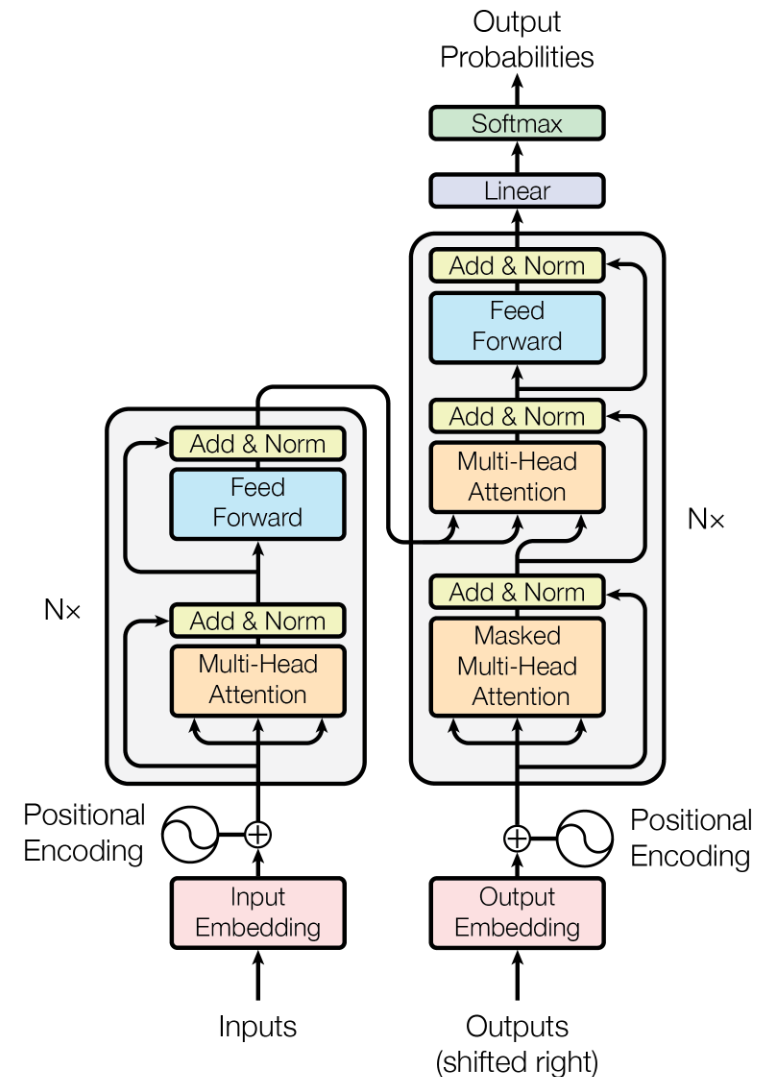
**Jakob Uszkoreit\***  
Google Research  
usz@google.com

**Llion Jones\***  
Google Research  
llion@google.com

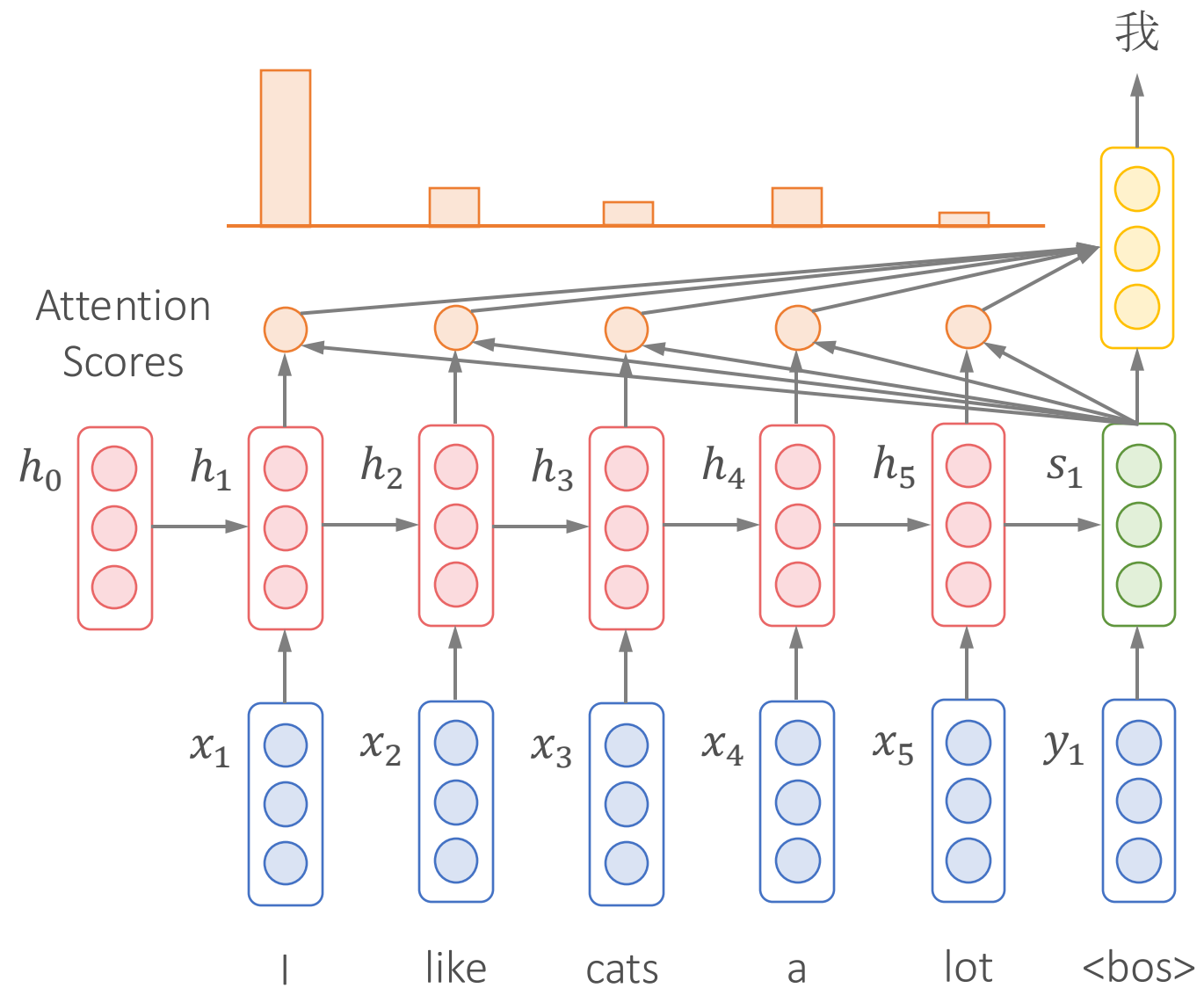
**Aidan N. Gomez\* †**  
University of Toronto  
aidan@cs.toronto.edu

**Łukasz Kaiser\***  
Google Brain  
lukaszkaiser@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com



# Look Back at RNN with Attention



Attention Scores

$$\alpha_i = h_i^\top s_1$$

Normalized  
Attention Scores

$$\hat{\alpha}_i = \text{softmax}(\alpha_i)$$

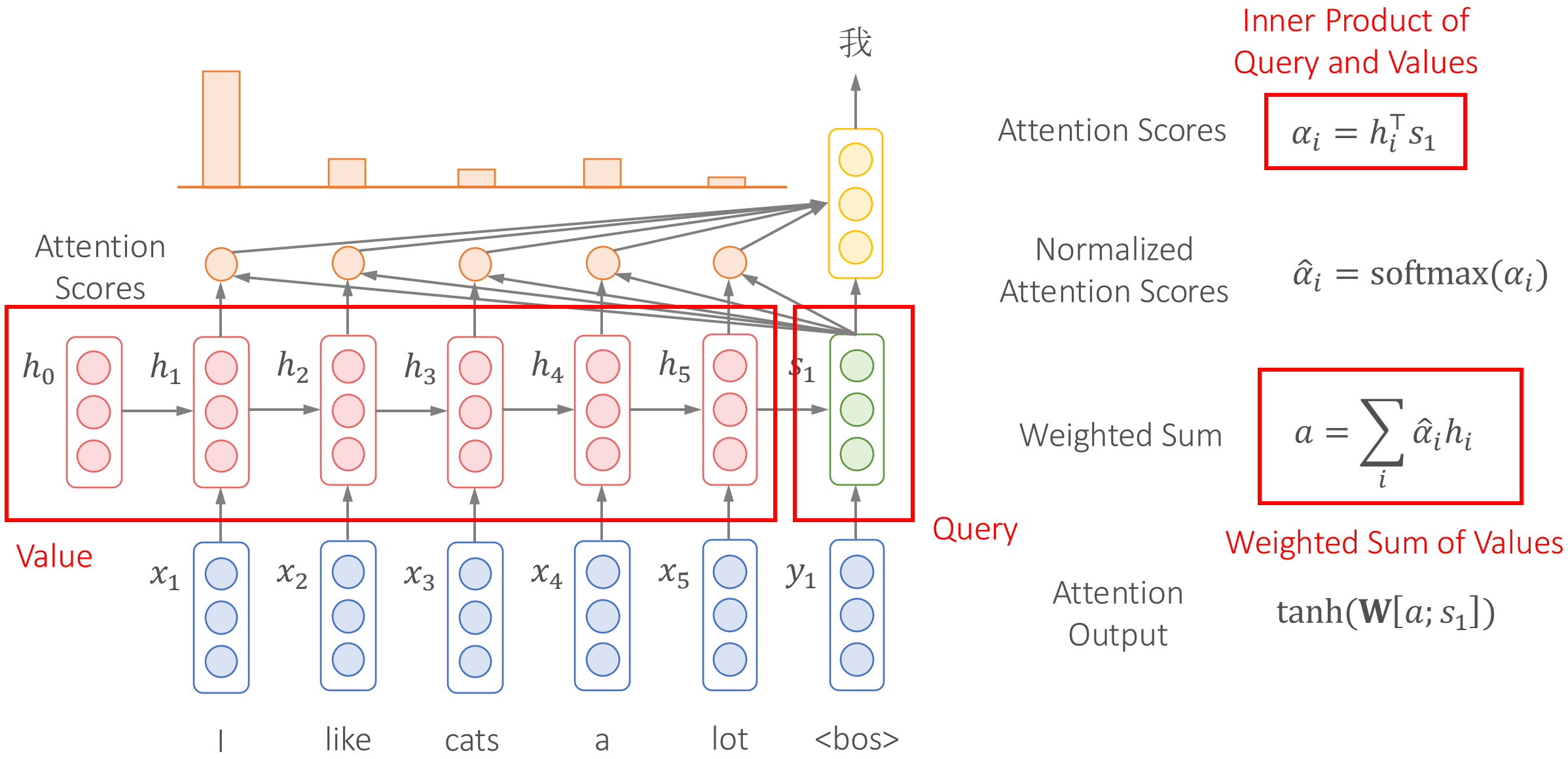
Weighted Sum

$$a = \sum_i \hat{\alpha}_i h_i$$

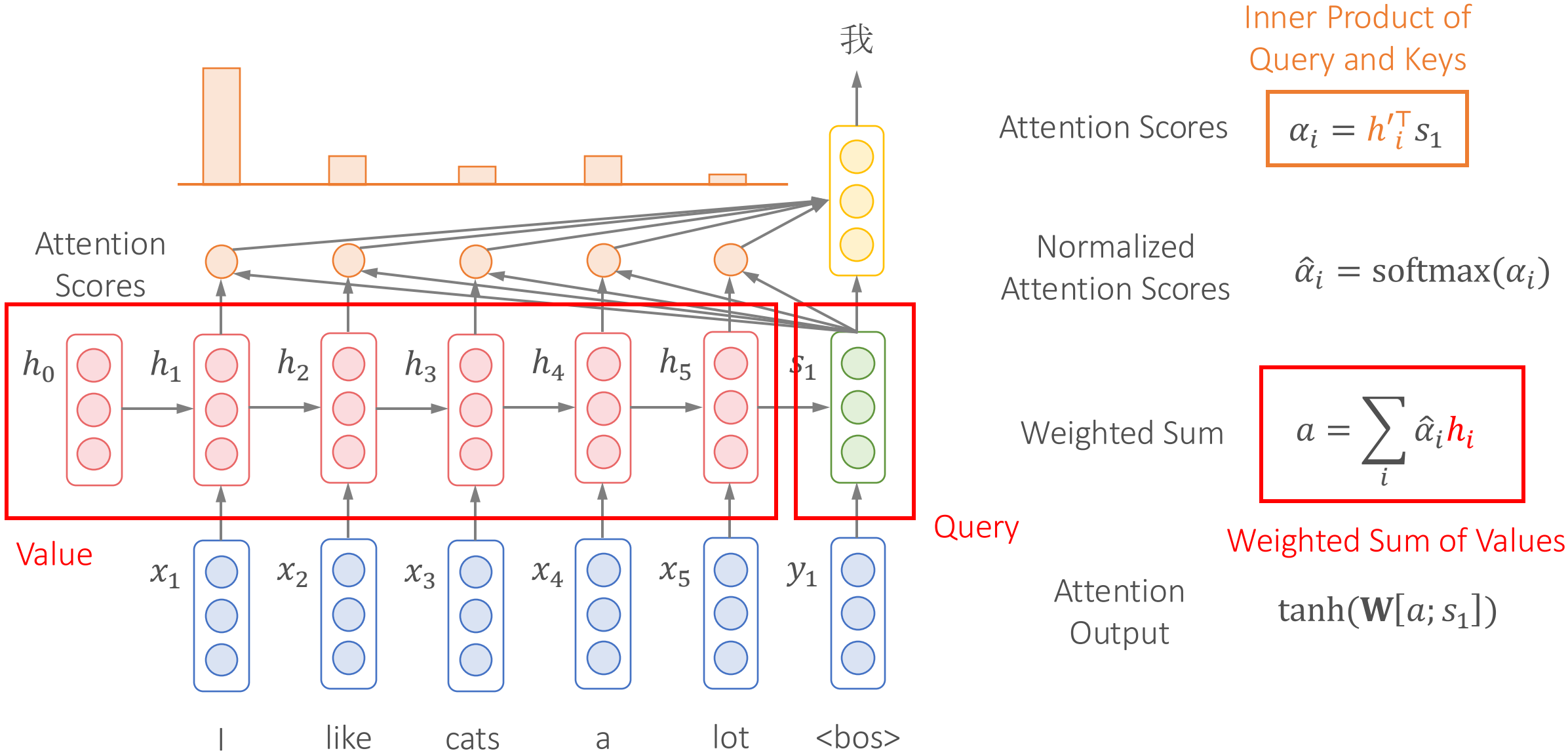
Attention  
Output

$$\tanh(\mathbf{W}[a; s_1])$$

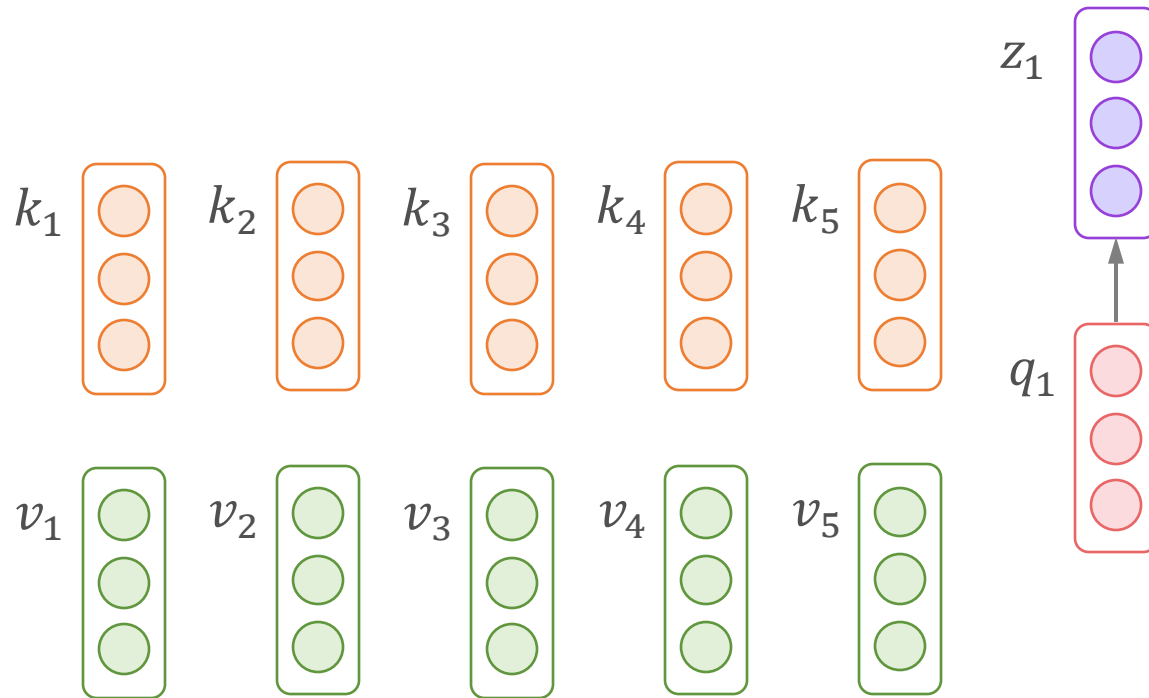
# Look Back at RNN with Attention



# Look Back at RNN with Attention – General Version



# Attention – General Version



Attention Scores

$$\alpha_i = k_i^\top q_1$$

Normalized  
Attention Scores

$$\hat{\alpha}_i = \text{softmax}(\alpha_i)$$

Weighted Sum

$$z_1 = \sum_i \hat{\alpha}_i v_i$$



# From Attention to Self-Attention

- Self-attention = attention from the sequence to itself
  - The queries, keys and values come from the same source
- Any word can be a **query**
- Any word can be a **key**
- Any word can be a **value**

# Self-Attention

Query

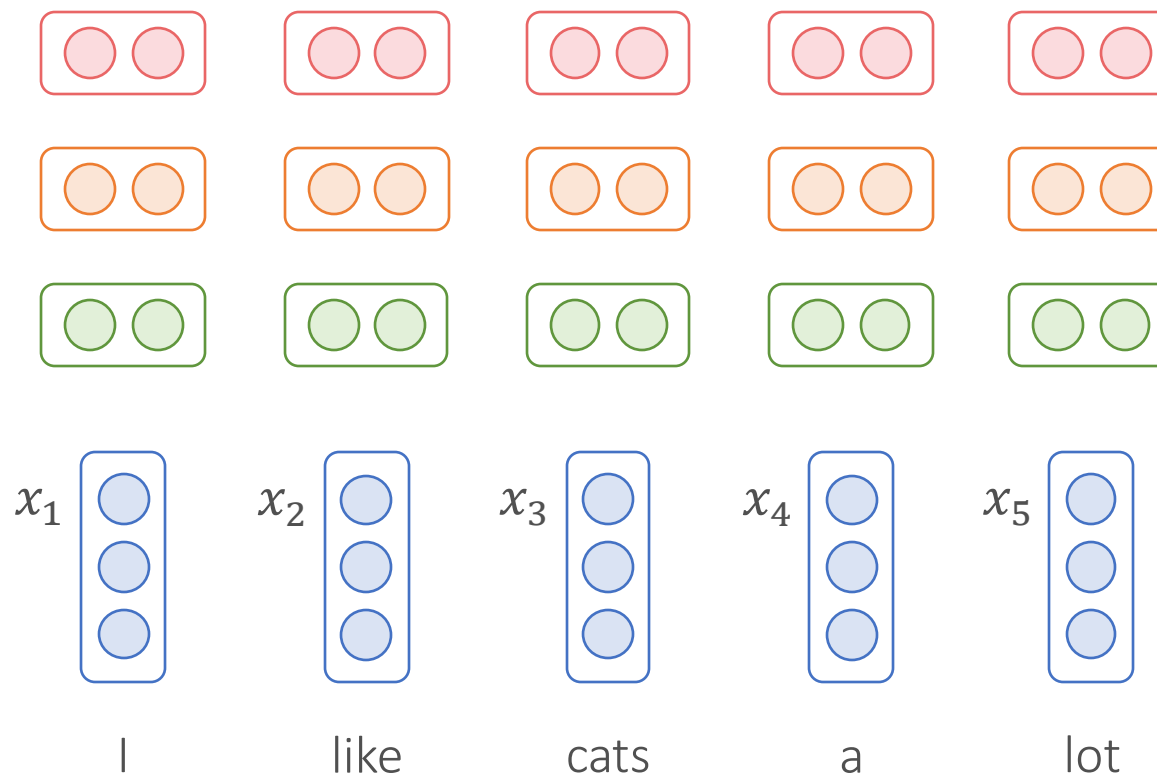
$$q_i = W^Q x_i$$

Key

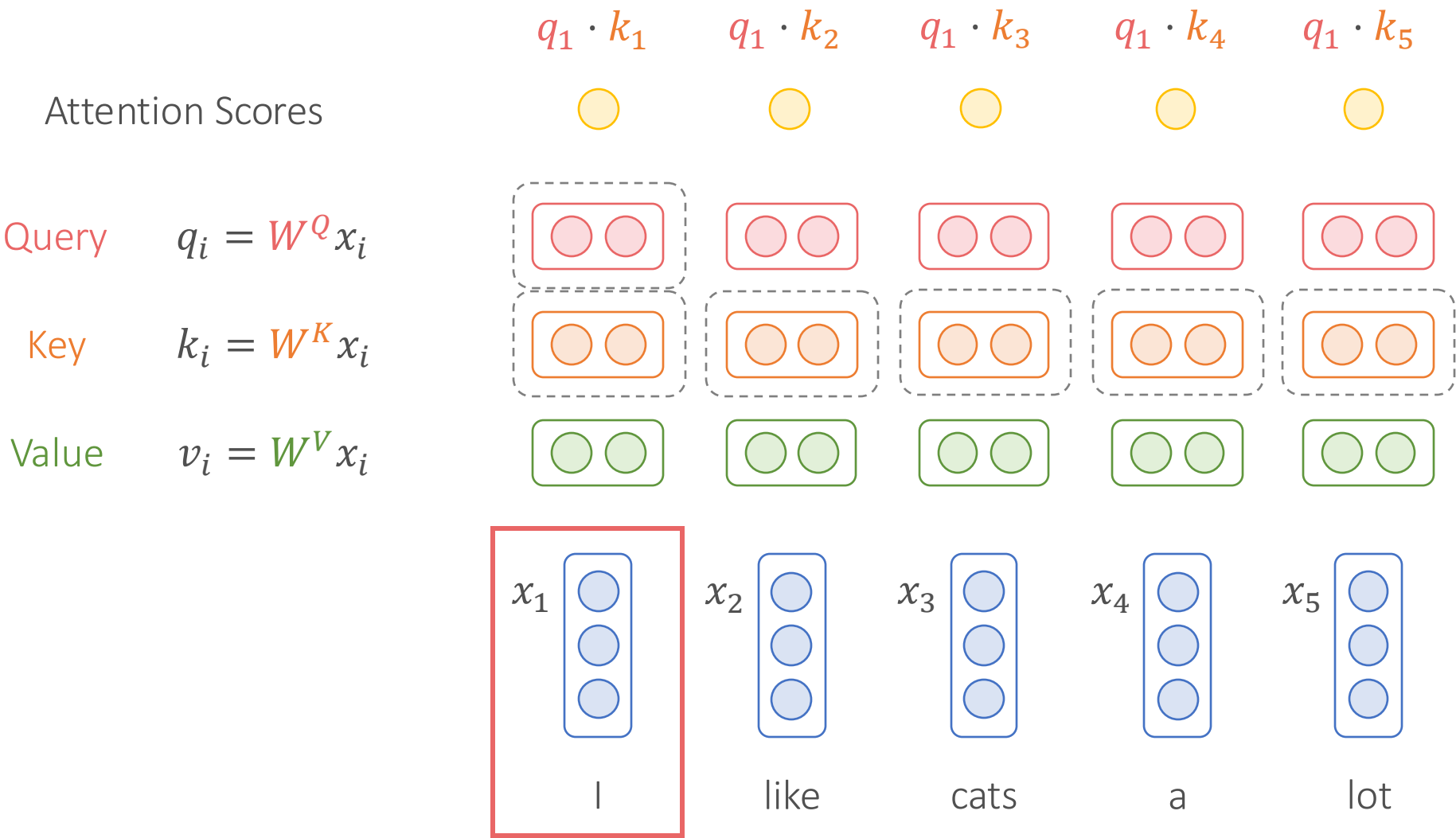
$$k_i = W^K x_i$$

Value

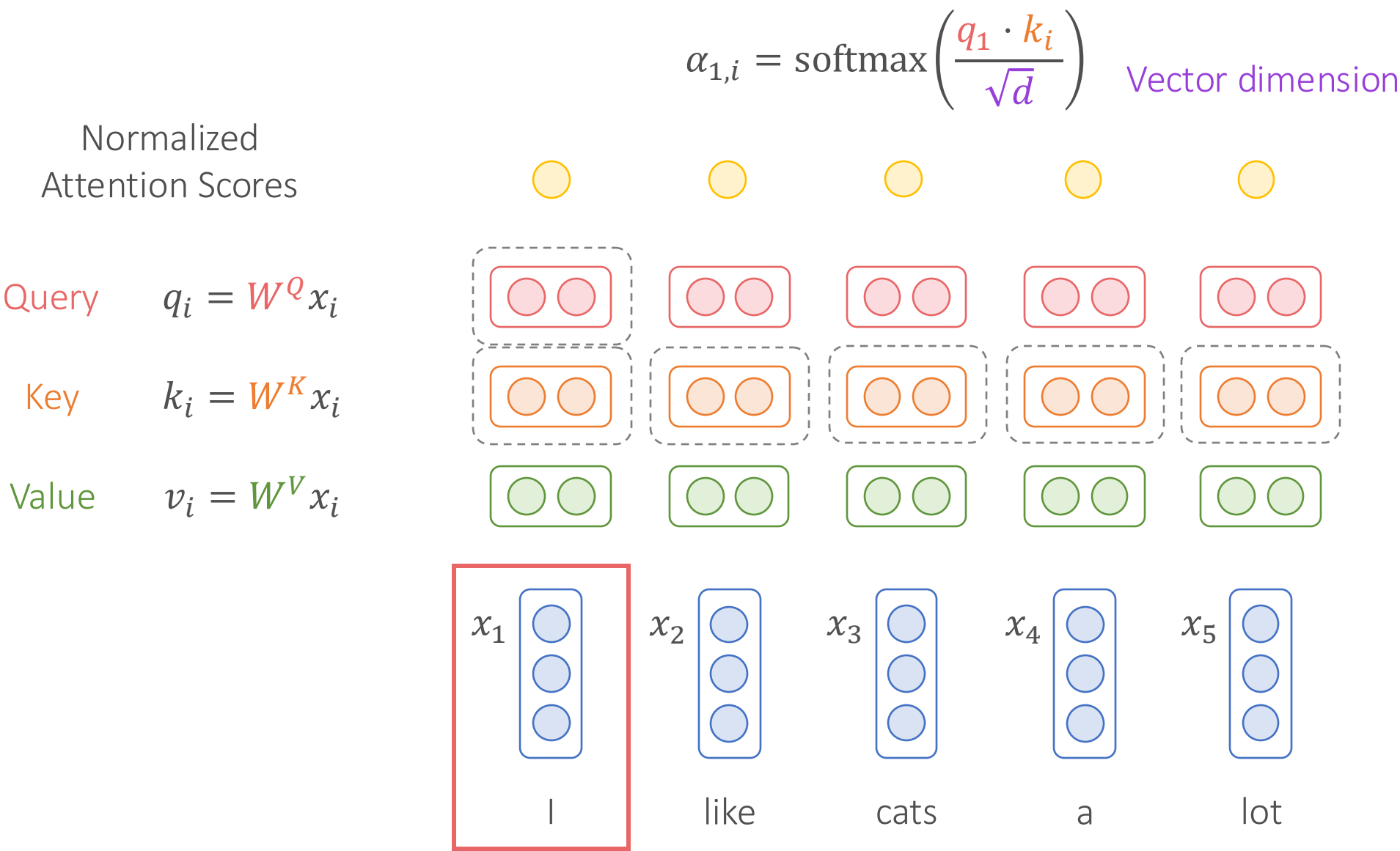
$$v_i = W^V x_i$$



# Self-Attention

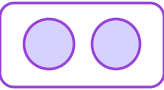


# Self-Attention



# Self-Attention

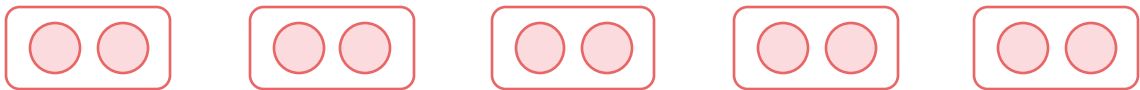
Weighted Sum


$$z_1 = \sum_i \alpha_{1,i} v_i$$

Normalized  
Attention Scores



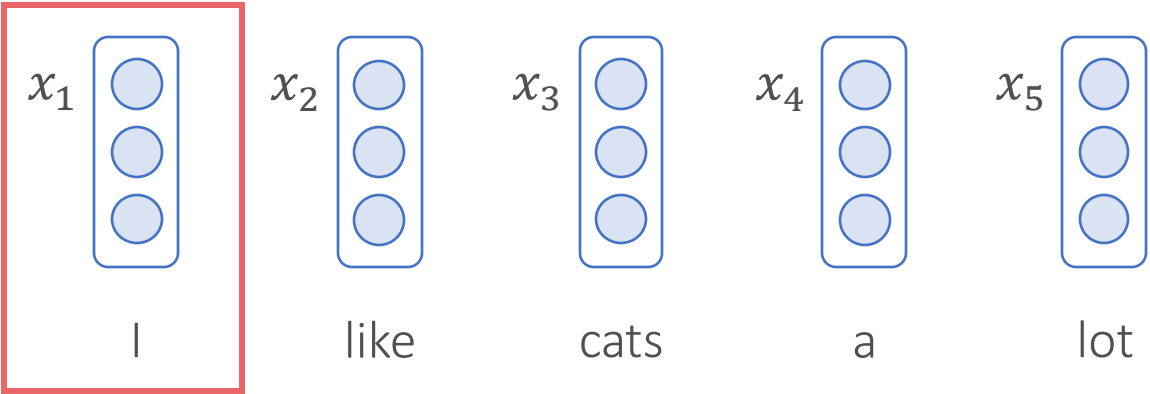
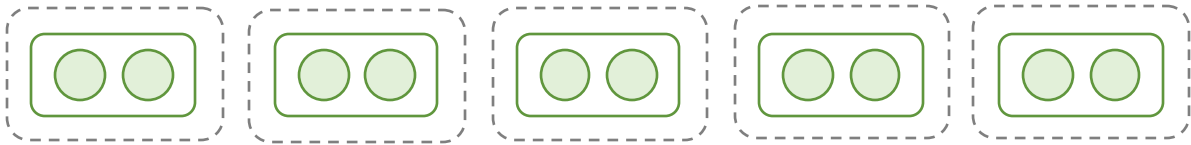
Query  $q_i = W^Q x_i$



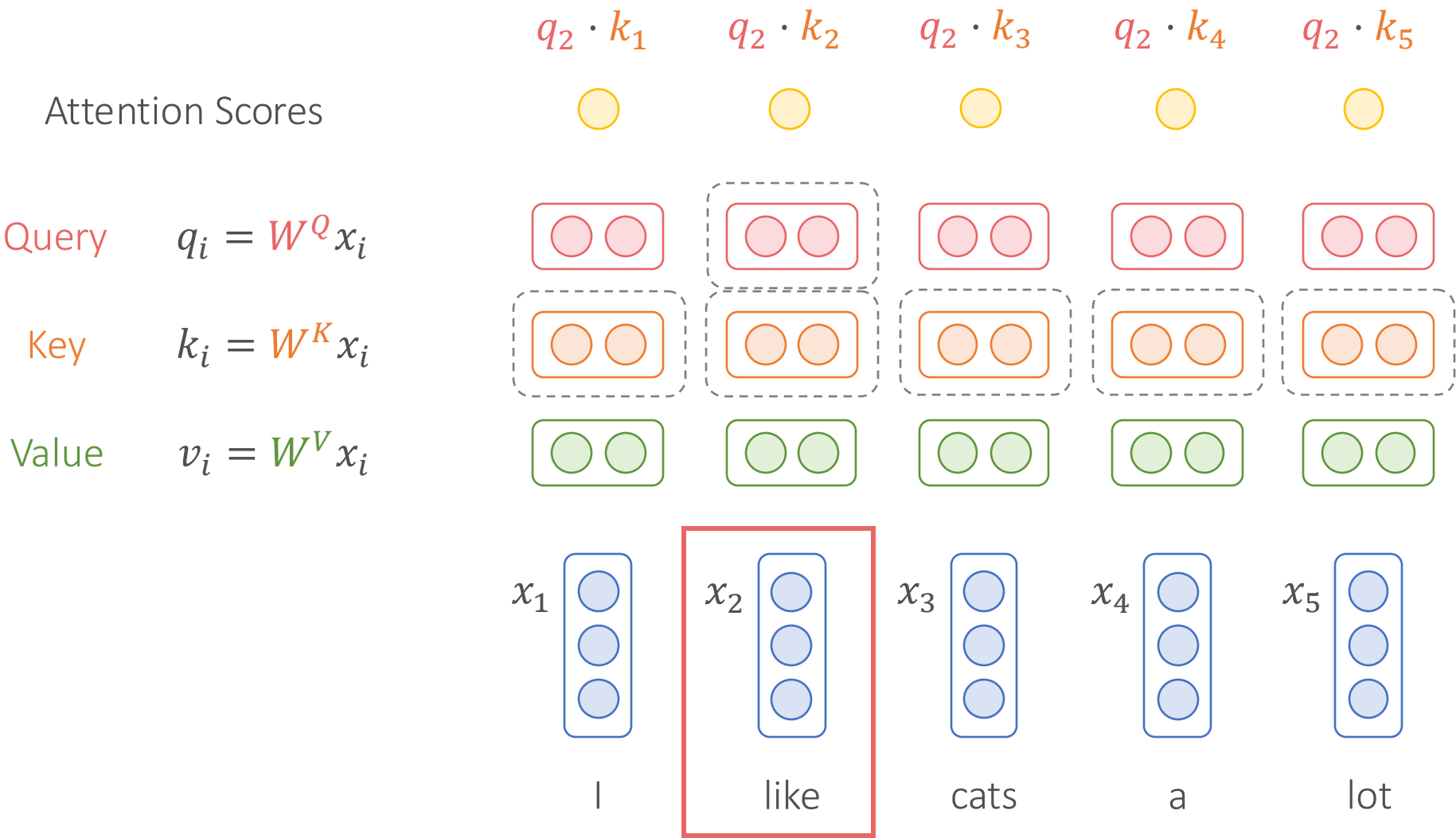
Key  $k_i = W^K x_i$



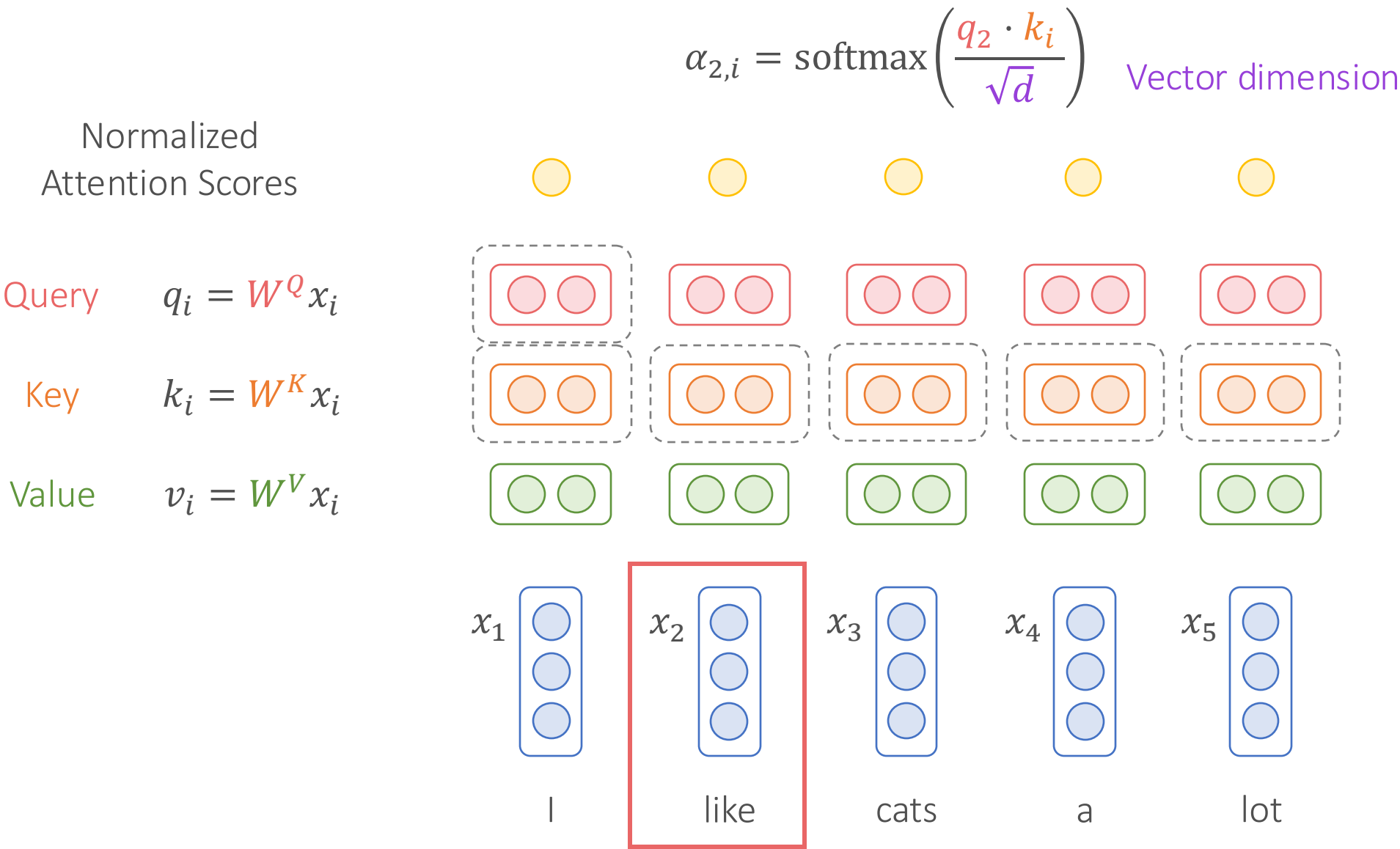
Value  $v_i = W^V x_i$



# Self-Attention



# Self-Attention



# Self-Attention

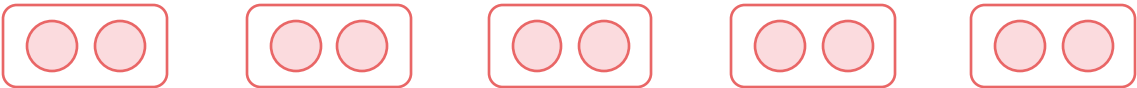
Weighted Sum

$$z_2 = \sum_i \alpha_{2,i} v_i$$

Normalized  
Attention Scores



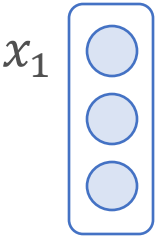
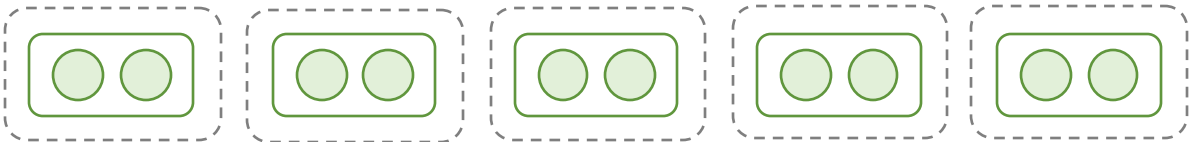
Query  $q_i = W^Q x_i$



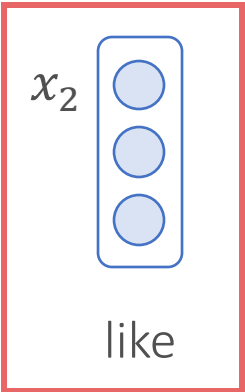
Key  $k_i = W^K x_i$



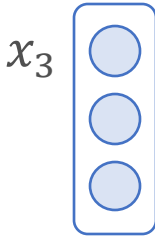
Value  $v_i = W^V x_i$



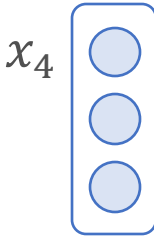
I



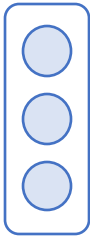
like



cats



a



lot



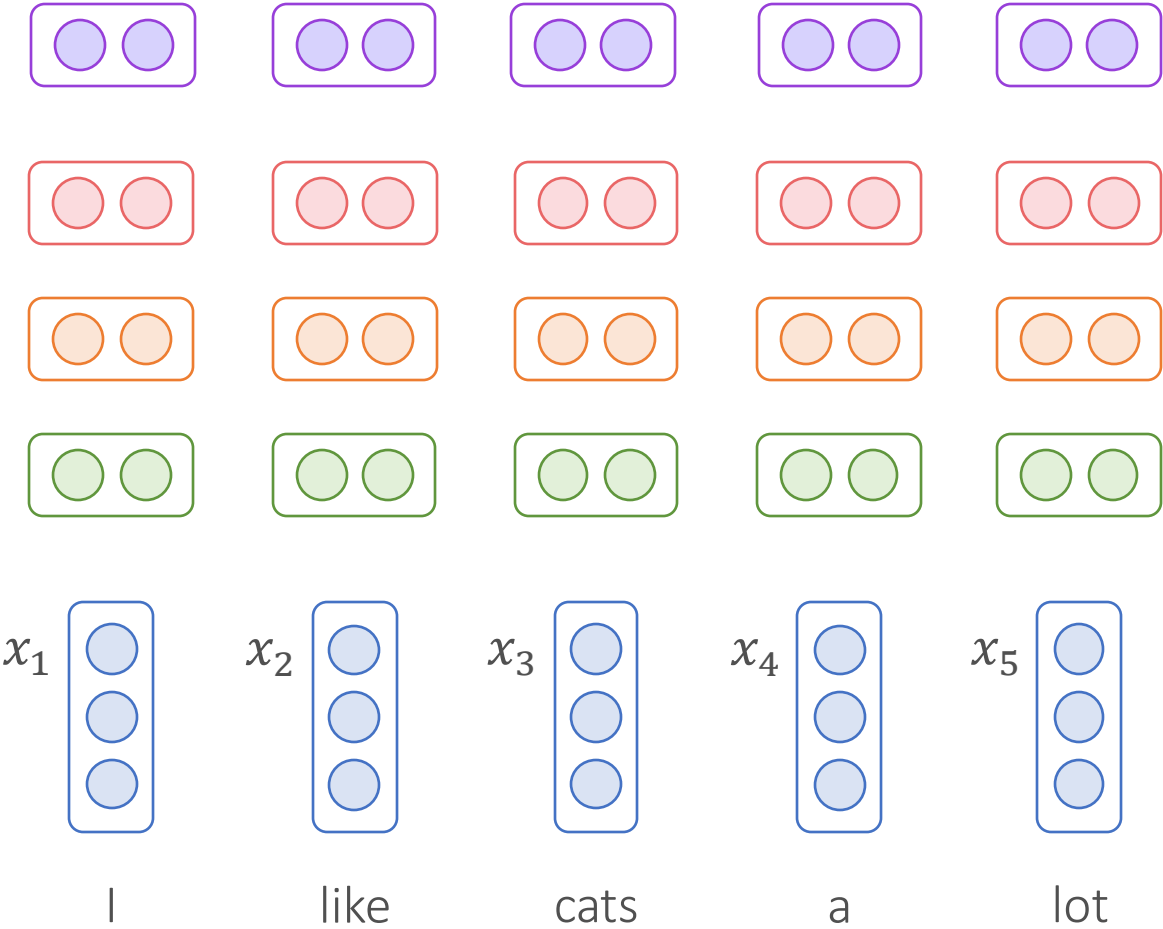
# Self-Attention

Self-Attention Output

Query  $q_i = W^Q x_i$

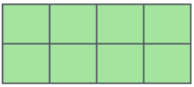
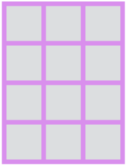
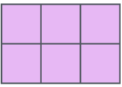
Key  $k_i = W^K x_i$

Value  $v_i = W^V x_i$



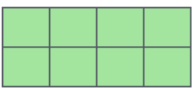
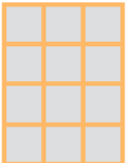
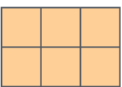
# Self-Attention – Matrix Form

$\mathbf{X}$   
 Word 1  
 Word 2


 $\times$ 

 $=$ 


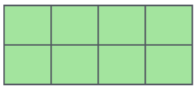
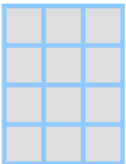
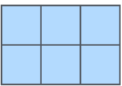
$\mathbf{Q}$

$\mathbf{X}$   
 Word 1  
 Word 2


 $\times$ 

 $=$ 


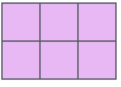
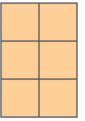
$\mathbf{K}$

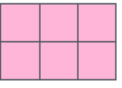
$\mathbf{X}$   
 Word 1  
 Word 2


 $\times$ 

 $=$ 


$\mathbf{V}$

$$\text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$

$\mathbf{Q}$ 

 $\times$ 

 $\sqrt{d_k}$

$\mathbf{Z}$   


$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$

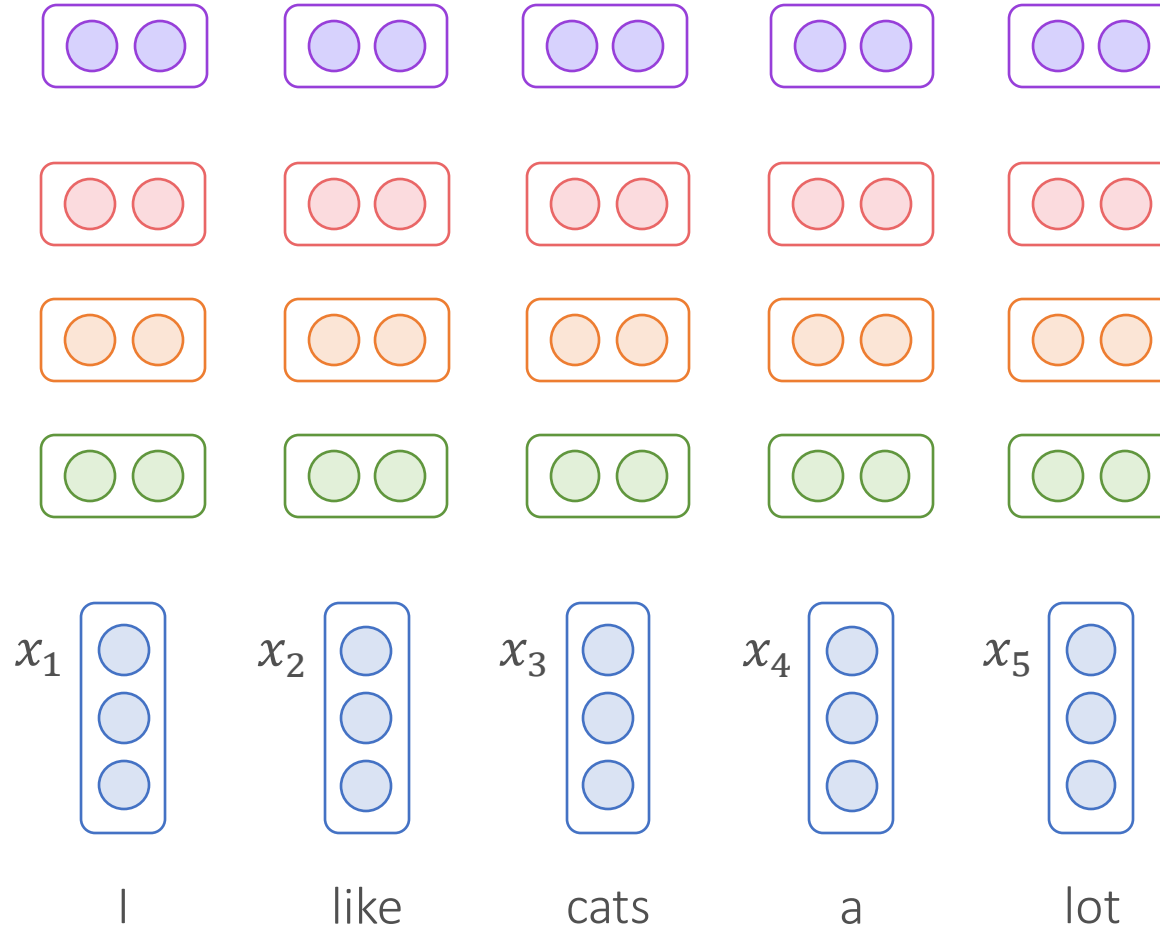
# Single-Head Attention

Self-Attention Output

Query  $q_i = W^Q x_i$

Key  $k_i = W^K x_i$

Value  $v_i = W^V x_i$



# Multi-Head Attention

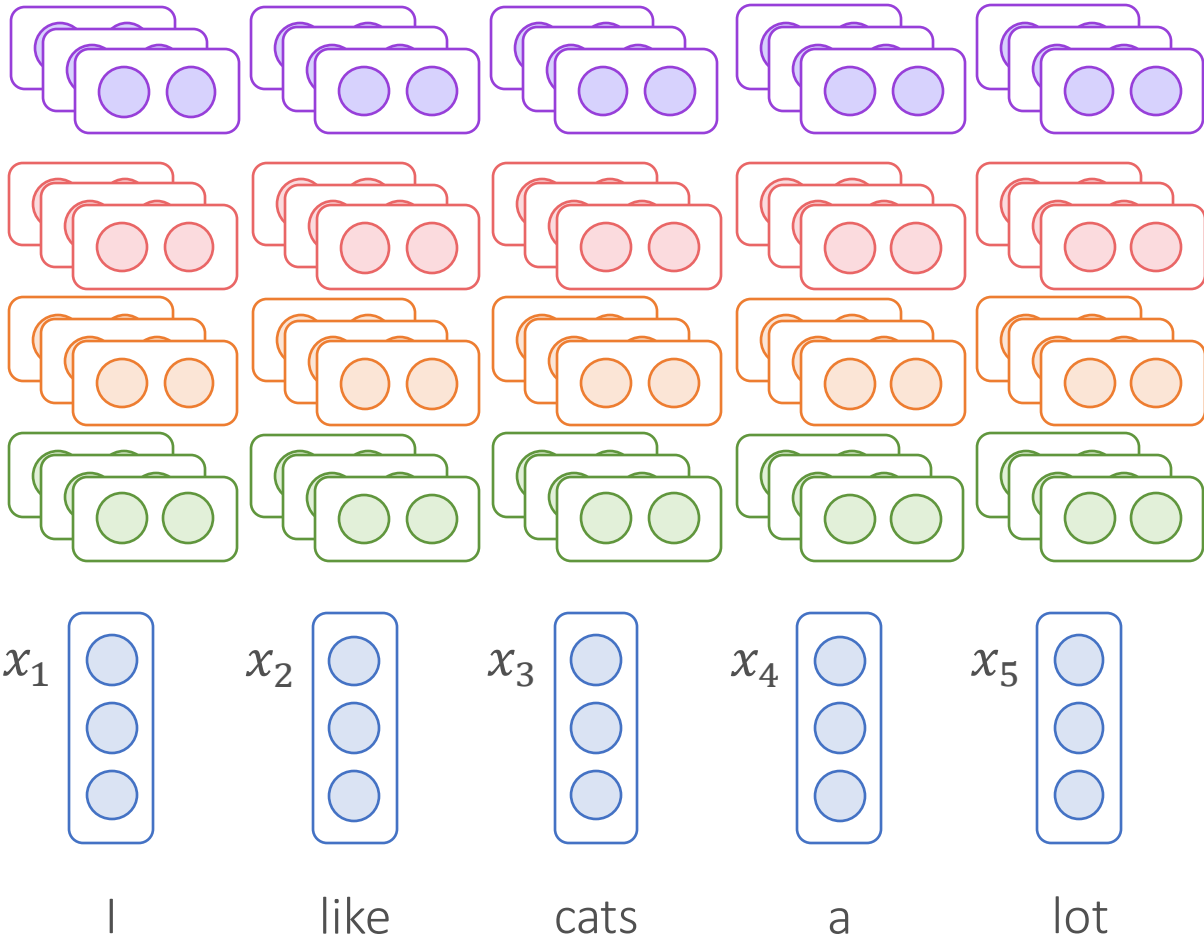
Each attention head focuses on different parts of understanding!

Multi-Attention Output

Query  $q_i = W_j^Q x_i$

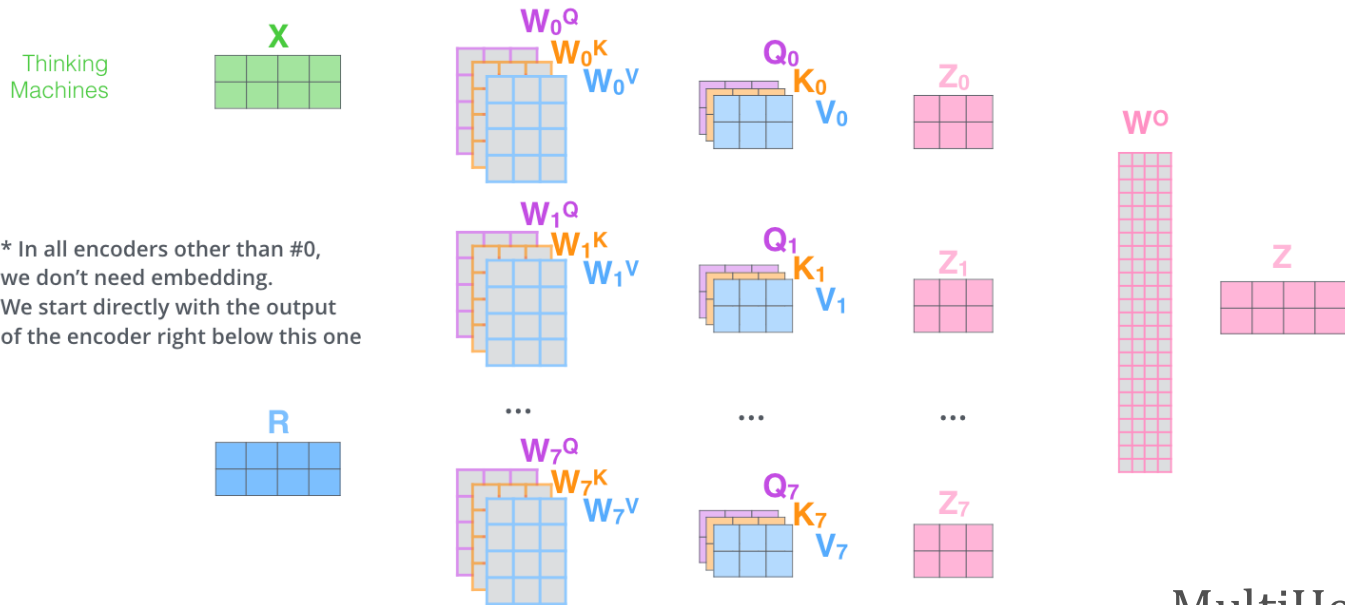
Key  $k_i = W_j^K x_i$

Value  $v_i = W_j^V x_i$



# Multi-Head Attention – Matrix Form

- 1) This is our input sentence\*
- 2) We embed each word\*
- 3) Split into 8 heads. We multiply  $X$  or  $R$  with weight matrices
- 4) Calculate attention using the resulting  $Q/K/V$  matrices
- 5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^O$  to produce the output of the layer

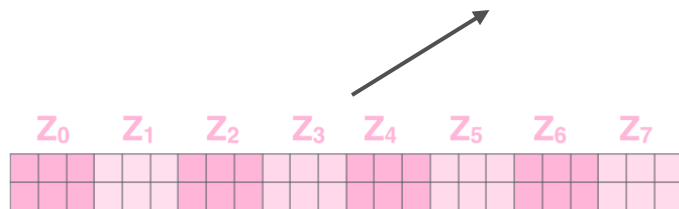


\* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

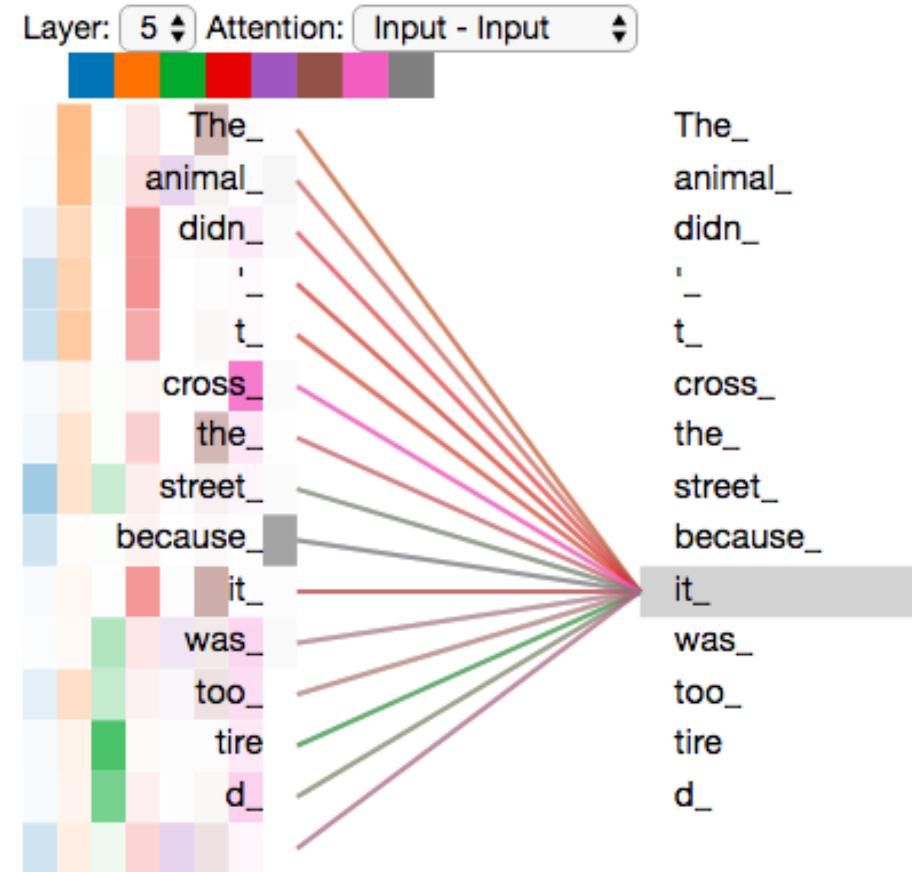
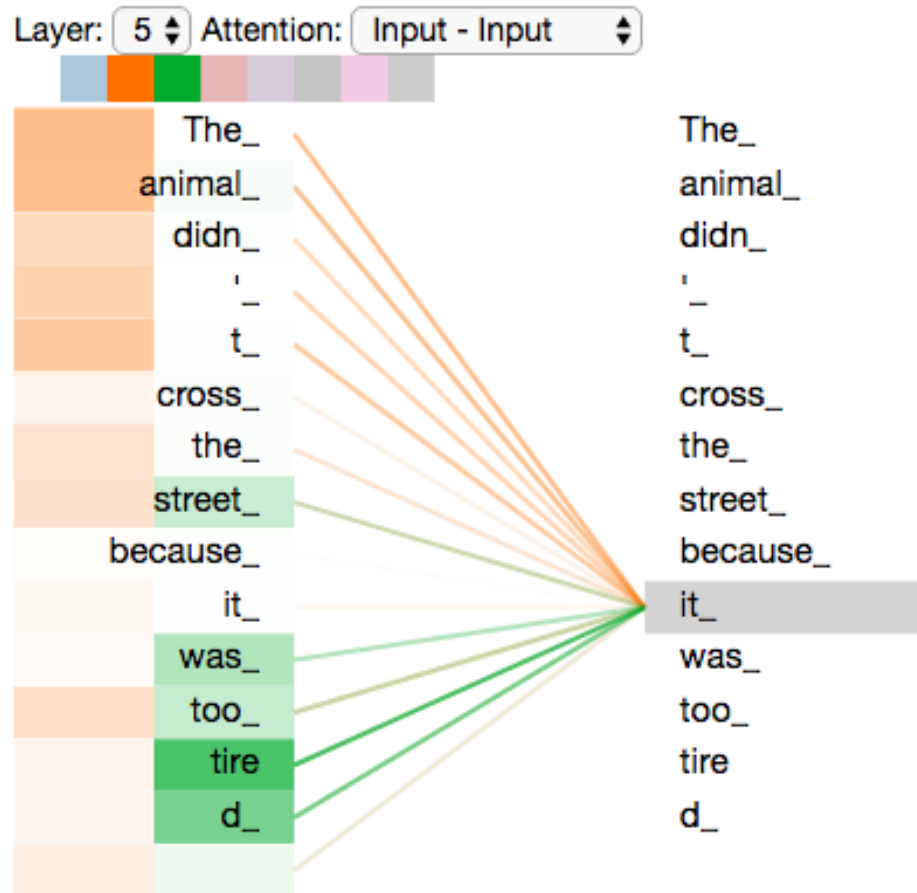
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

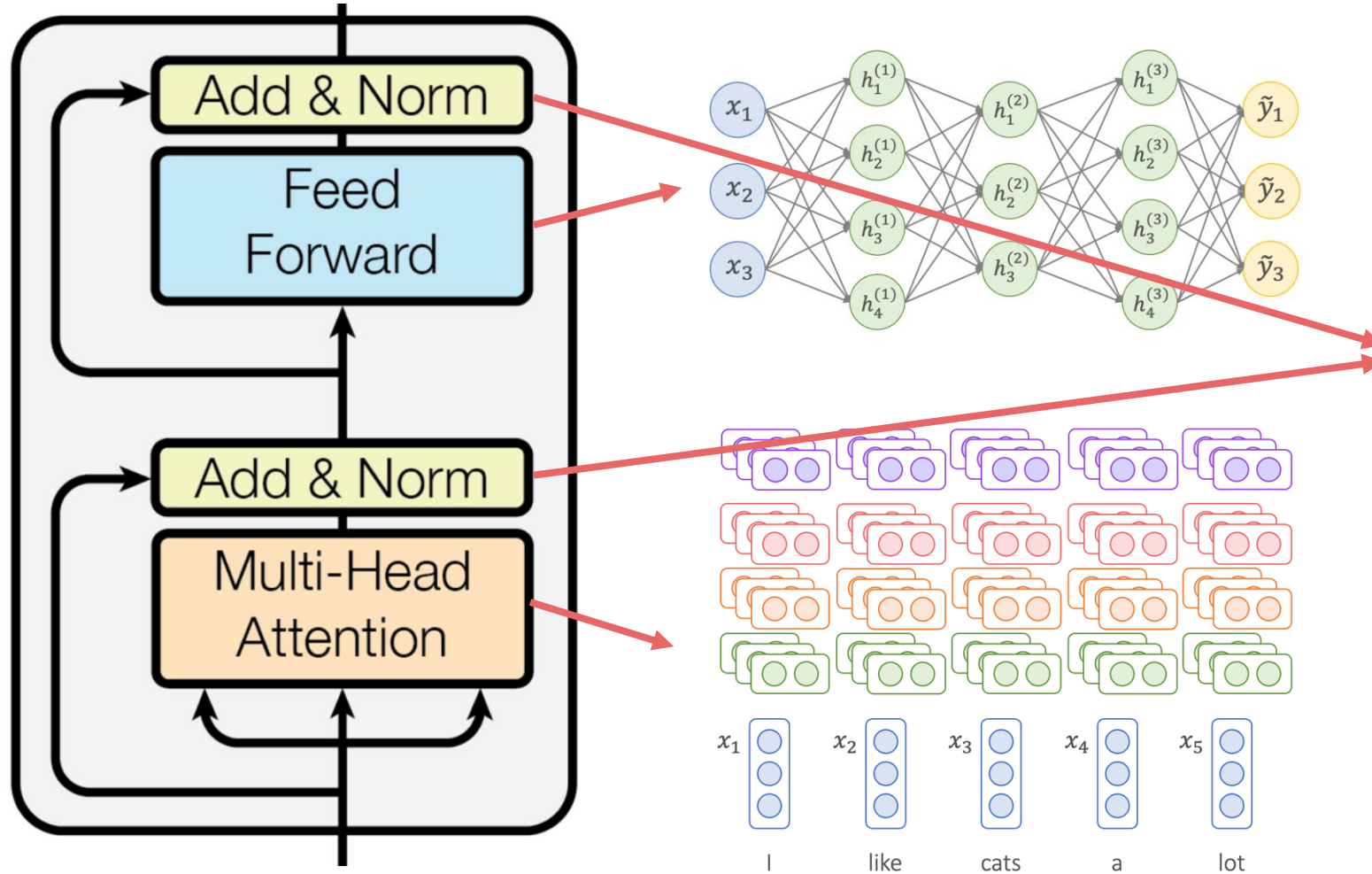
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$



# What Does Multi-Head Attention Learn?



# Transformer Layer



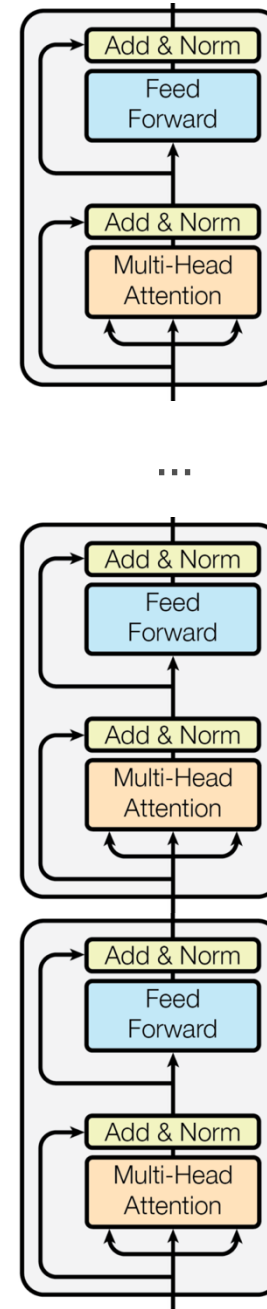
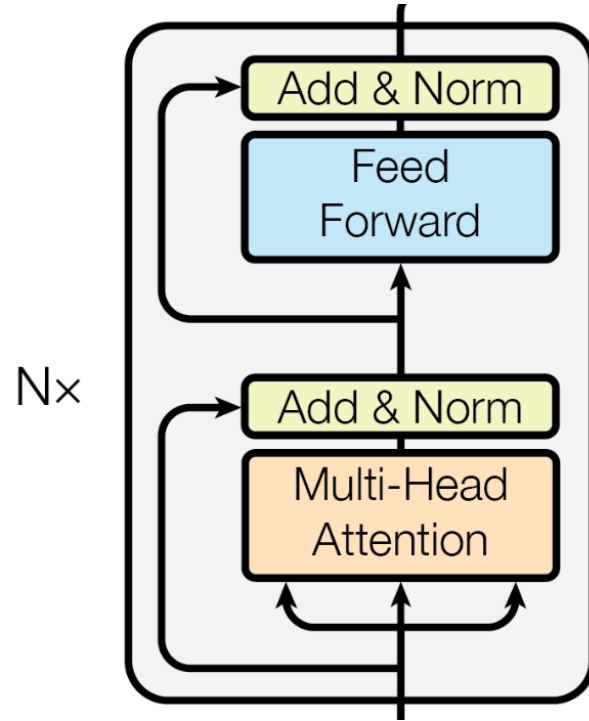
$\text{LayerNorm}(x + \text{Sublayer}(x))$

$$y = \frac{x - \mathbf{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} * \gamma + \beta$$

Residual connection (He et al., 2016)

Layer normalization (Ba et al., 2016)

# Transformer Encoder





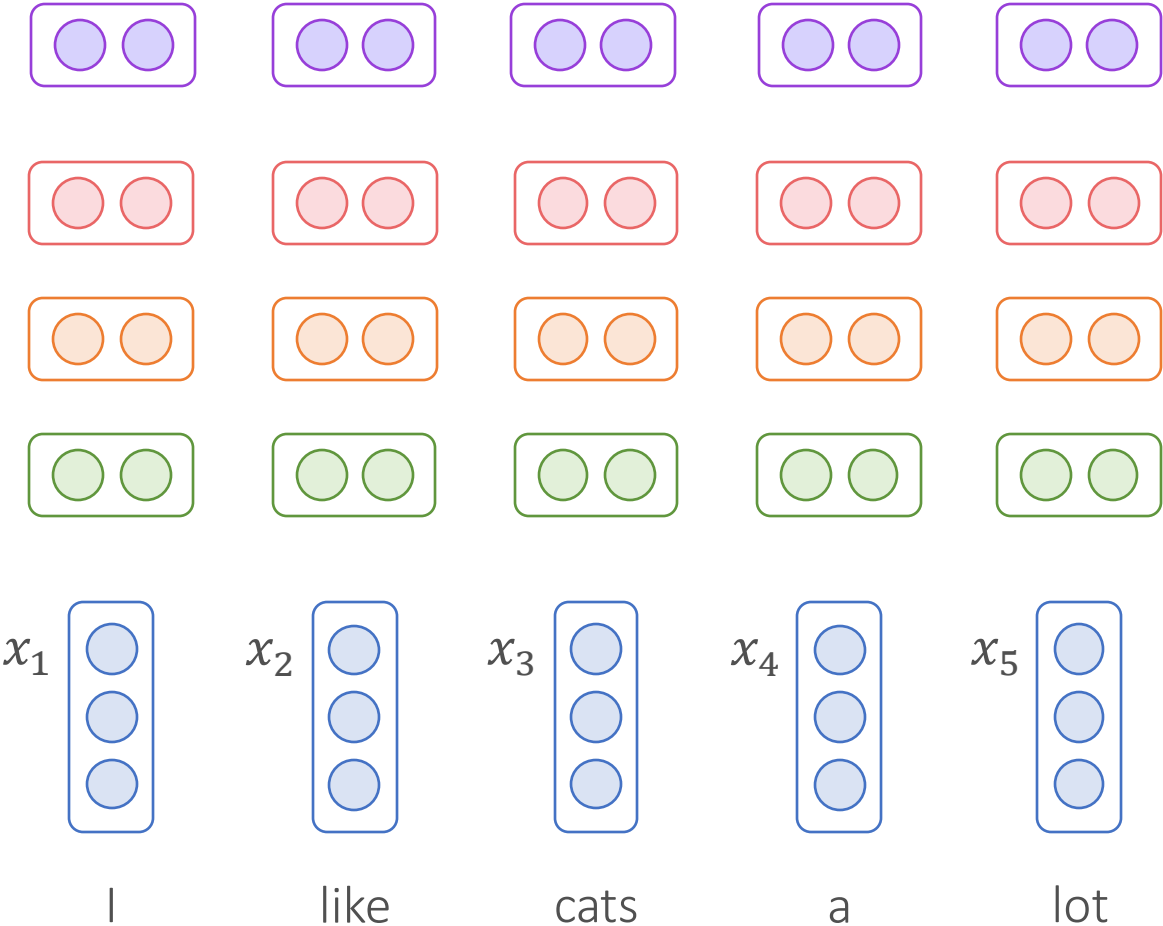
# How About Word Order?

Self-Attention Output

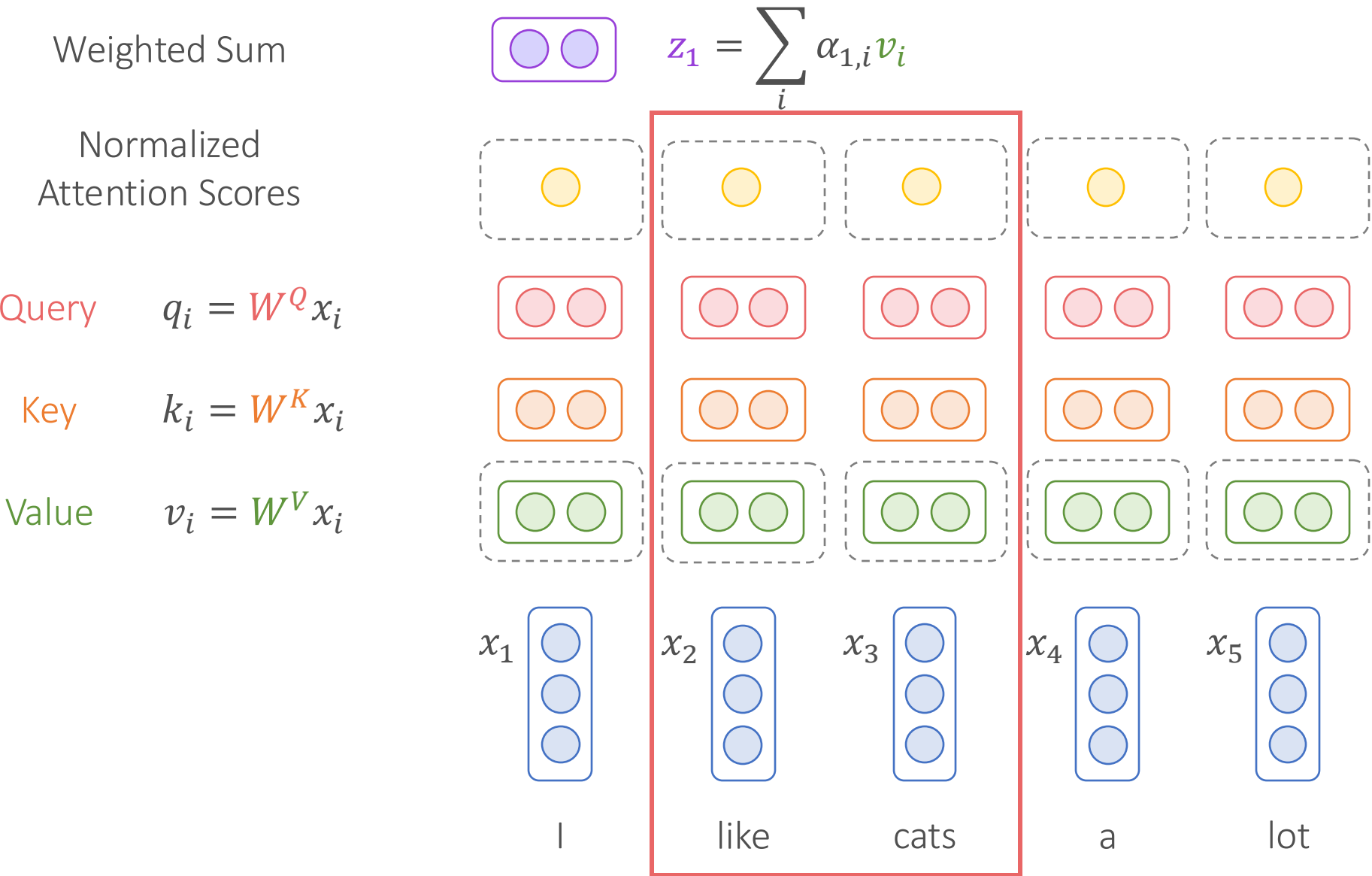
Query  $q_i = W^Q x_i$

Key  $k_i = W^K x_i$

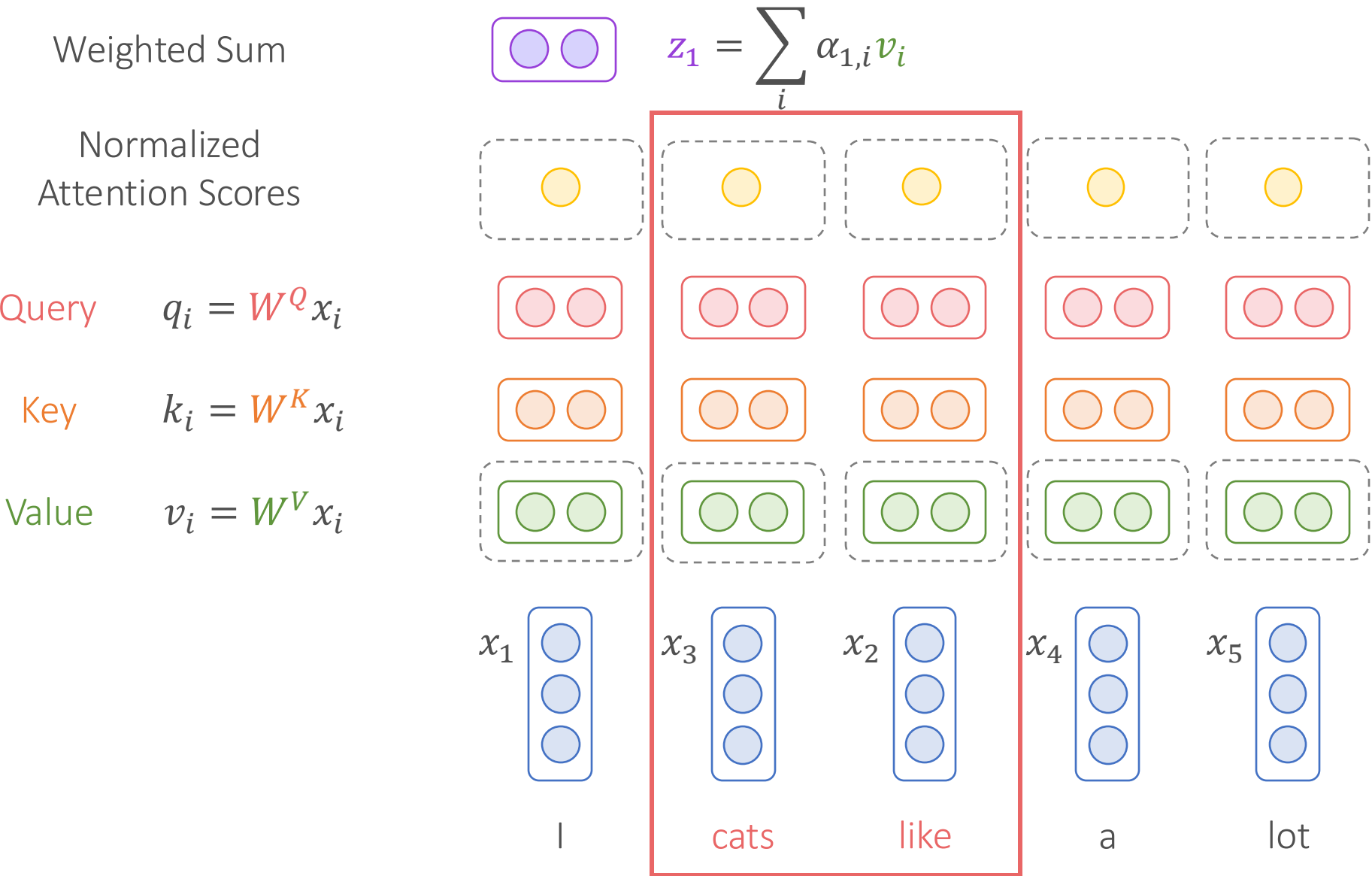
Value  $v_i = W^V x_i$



# How About Word Order?

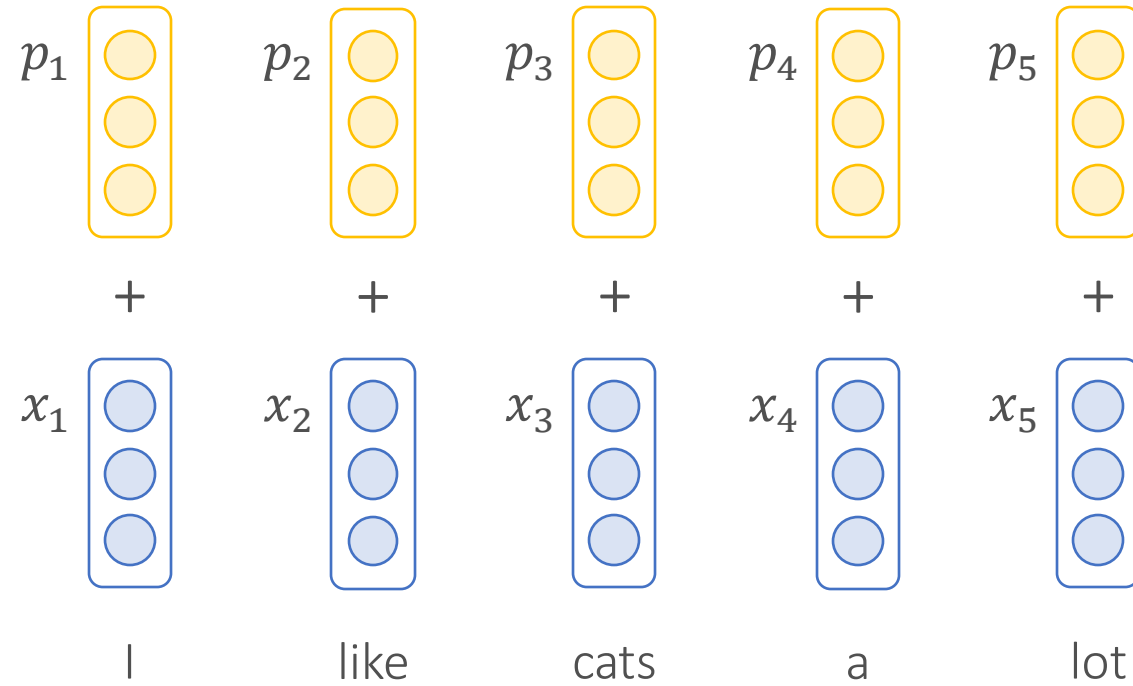


# How About Word Order?



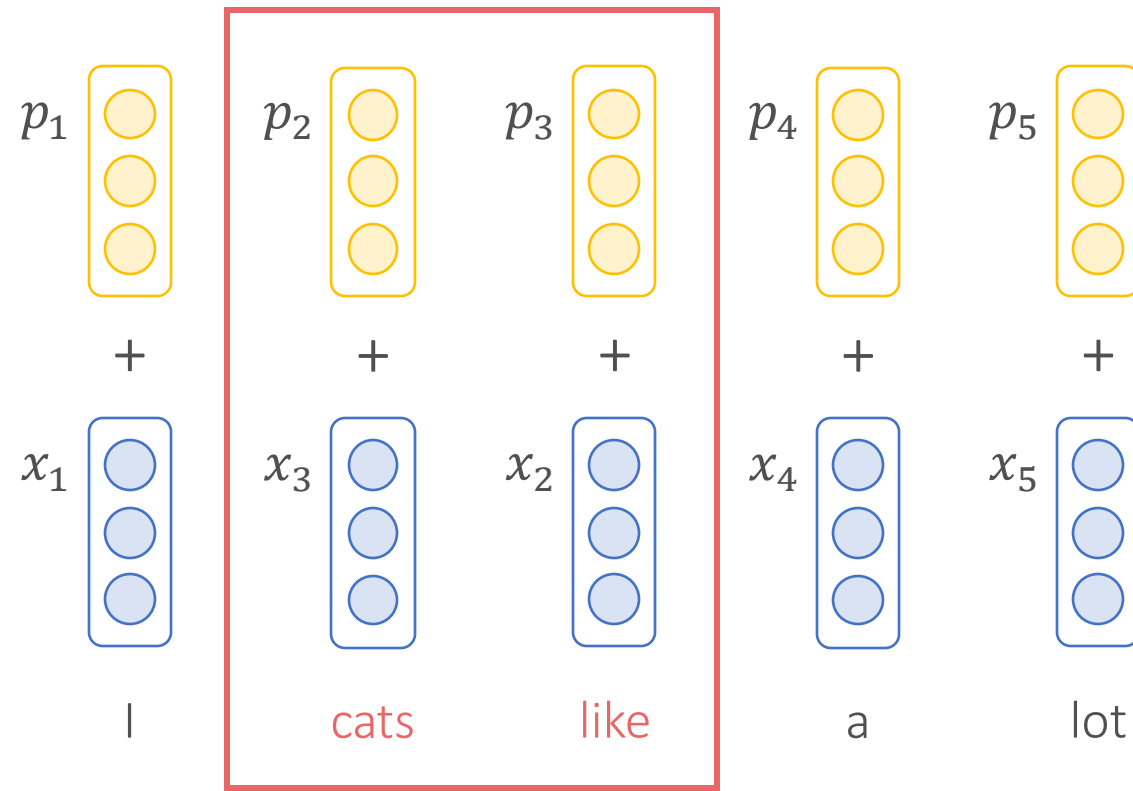
# Solution: Positional Encoding

$$x_i \leftarrow x_i + PE_i$$



# Solution: Positional Encoding

$$x_i \leftarrow x_i + PE_i$$



# Solution: Positional Encoding

- Unique encoding for each position
- Closer positions should have more similar encodings
- Distance between neighboring positions should be the same

# Sinusoidal Positional Encoding

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

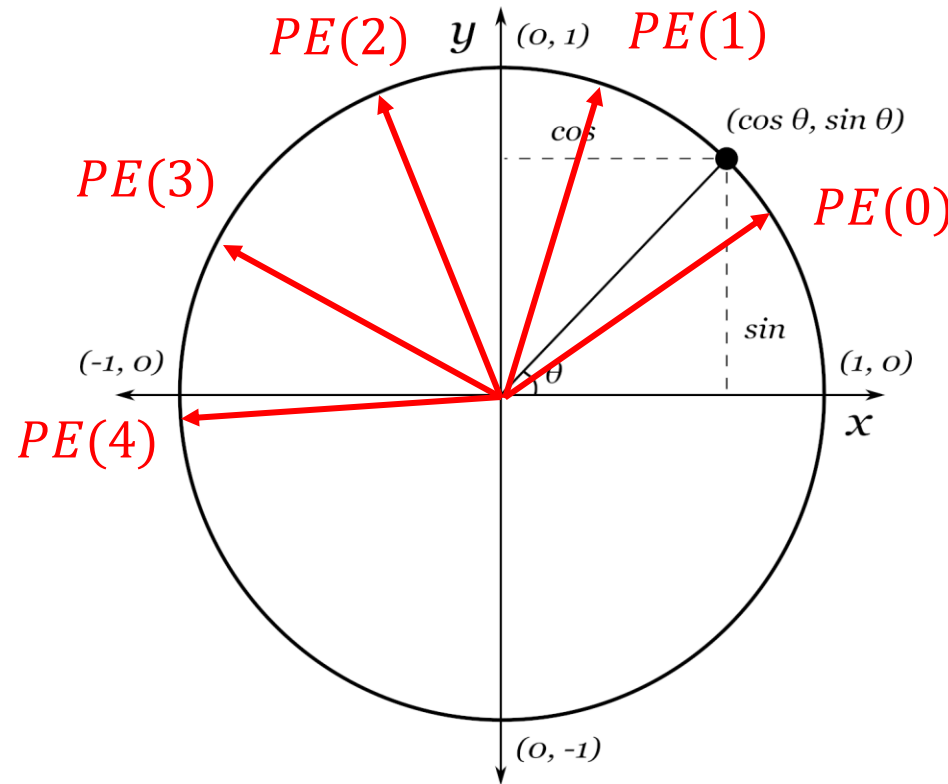
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

Why this?

# Sinusoidal Positional Encoding: Intuition

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

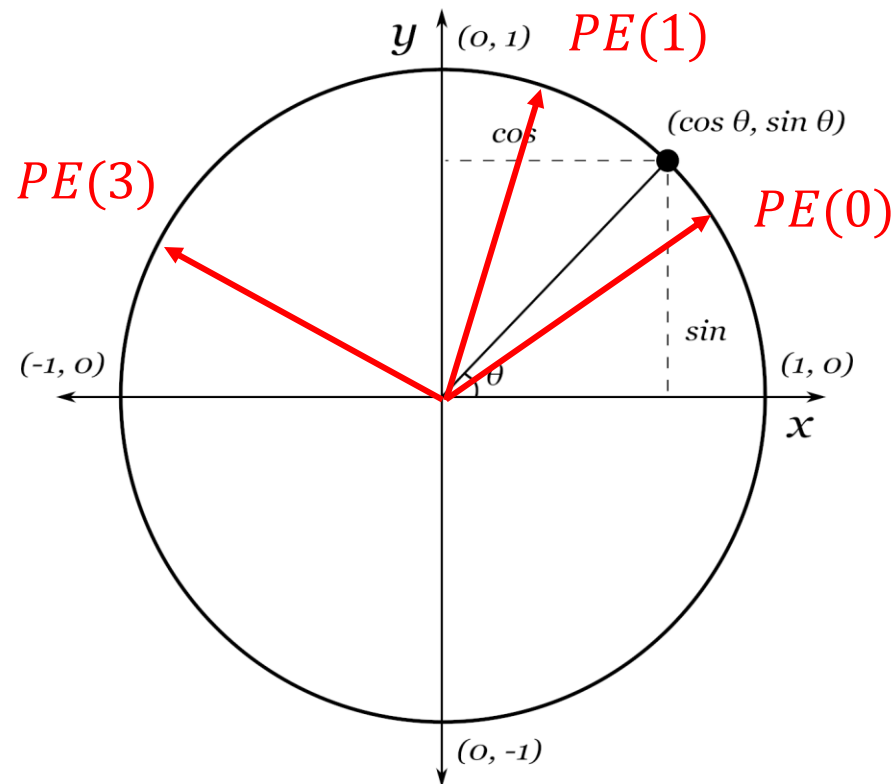




# Sinusoidal Positional Encoding: Intuition

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$



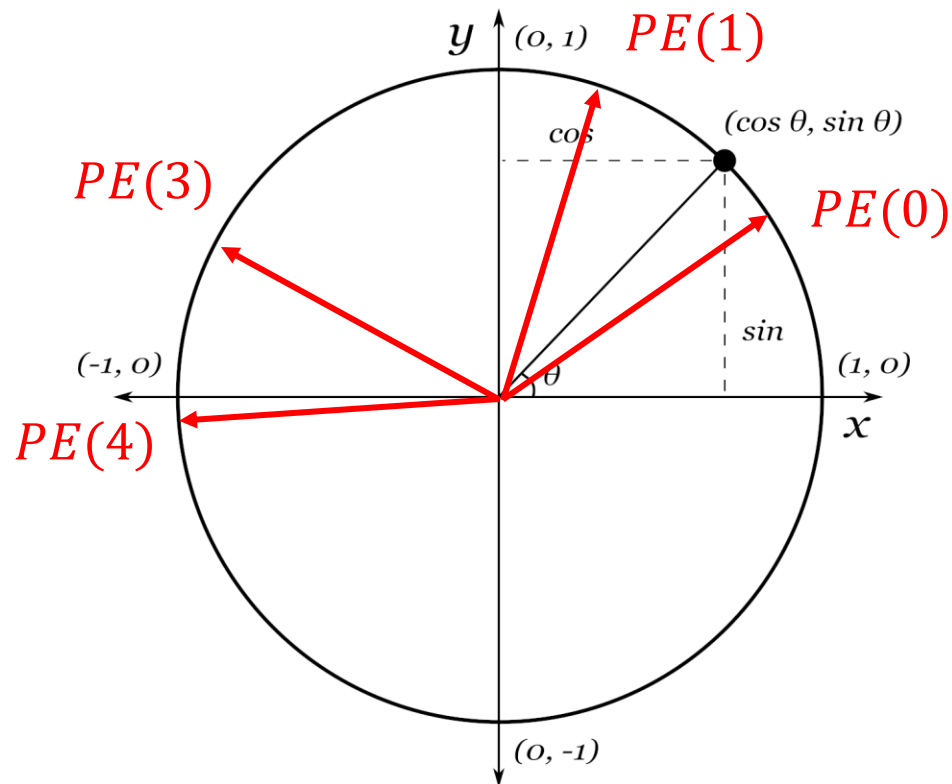
$$\text{Cosine}(PE(0), PE(1)) > \text{Cosine}(PE(0), PE(3))$$

Closer positions should have more similar encodings

# Sinusoidal Positional Encoding: Intuition

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$



$$\text{Cosine}(PE(0), PE(1)) = \text{Cosine}(PE(3), PE(4))$$

Distance between neighboring positions should be the same

# Sinusoidal Positional Encoding: Intuition

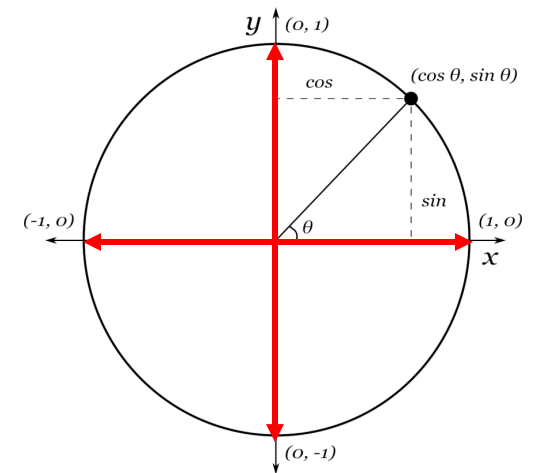
- How to expand to high-dimension?
- Let's consider **binary** positional encoding first
- How to use 4 bits to represent position 0~15?

|     |   |   |   |   |      |   |   |   |   |
|-----|---|---|---|---|------|---|---|---|---|
| 0 : | 0 | 0 | 0 | 0 | 8 :  | 1 | 0 | 0 | 0 |
| 1 : | 0 | 0 | 0 | 1 | 9 :  | 1 | 0 | 0 | 1 |
| 2 : | 0 | 0 | 1 | 0 | 10 : | 1 | 0 | 1 | 0 |
| 3 : | 0 | 0 | 1 | 1 | 11 : | 1 | 0 | 1 | 1 |
| 4 : | 0 | 1 | 0 | 0 | 12 : | 1 | 1 | 0 | 0 |
| 5 : | 0 | 1 | 0 | 1 | 13 : | 1 | 1 | 0 | 1 |
| 6 : | 0 | 1 | 1 | 0 | 14 : | 1 | 1 | 1 | 0 |
| 7 : | 0 | 1 | 1 | 1 | 15 : | 1 | 1 | 1 | 1 |

# Sinusoidal Positional Encoding: Intuition

- How to expand to high-dimension?
- Let's consider **binary** positional encoding first
- How to use 4 bits to represent position 0~15?

|     |   |   |   |   |      |   |   |   |   |
|-----|---|---|---|---|------|---|---|---|---|
| 0 : | 0 | 0 | 0 | 0 | 8 :  | 1 | 0 | 0 | 0 |
| 1 : | 0 | 0 | 0 | 1 | 9 :  | 1 | 0 | 0 | 1 |
| 2 : | 0 | 0 | 1 | 0 | 10 : | 1 | 0 | 1 | 0 |
| 3 : | 0 | 0 | 1 | 1 | 11 : | 1 | 0 | 1 | 1 |
| 4 : | 0 | 1 | 0 | 0 | 12 : | 1 | 1 | 0 | 0 |
| 5 : | 0 | 1 | 0 | 1 | 13 : | 1 | 1 | 0 | 1 |
| 6 : | 0 | 1 | 1 | 0 | 14 : | 1 | 1 | 1 | 0 |
| 7 : | 0 | 1 | 1 | 1 | 15 : | 1 | 1 | 1 | 1 |



# Sinusoidal Positional Encoding: Intuition

- How to expand to high-dimension?
- Let's consider **binary** positional encoding first
- How to use 4 bits to represent position 0~15?

|     |   |   |   |   |      |   |   |   |   |
|-----|---|---|---|---|------|---|---|---|---|
| 0 : | 0 | 0 | 0 | 0 | 8 :  | 1 | 0 | 0 | 0 |
| 1 : | 0 | 0 | 0 | 1 | 9 :  | 1 | 0 | 0 | 1 |
| 2 : | 0 | 0 | 1 | 0 | 10 : | 1 | 0 | 1 | 0 |
| 3 : | 0 | 0 | 1 | 1 | 11 : | 1 | 0 | 1 | 1 |
| 4 : | 0 | 1 | 0 | 0 | 12 : | 1 | 1 | 0 | 0 |
| 5 : | 0 | 1 | 0 | 1 | 13 : | 1 | 1 | 0 | 1 |
| 6 : | 0 | 1 | 1 | 0 | 14 : | 1 | 1 | 1 | 0 |
| 7 : | 0 | 1 | 1 | 1 | 15 : | 1 | 1 | 1 | 1 |

# Sinusoidal Positional Encoding: Intuition

- How to expand to high-dimension?
- Let's consider **binary** positional encoding first
- How to use 4 bits to represent position 0~15?

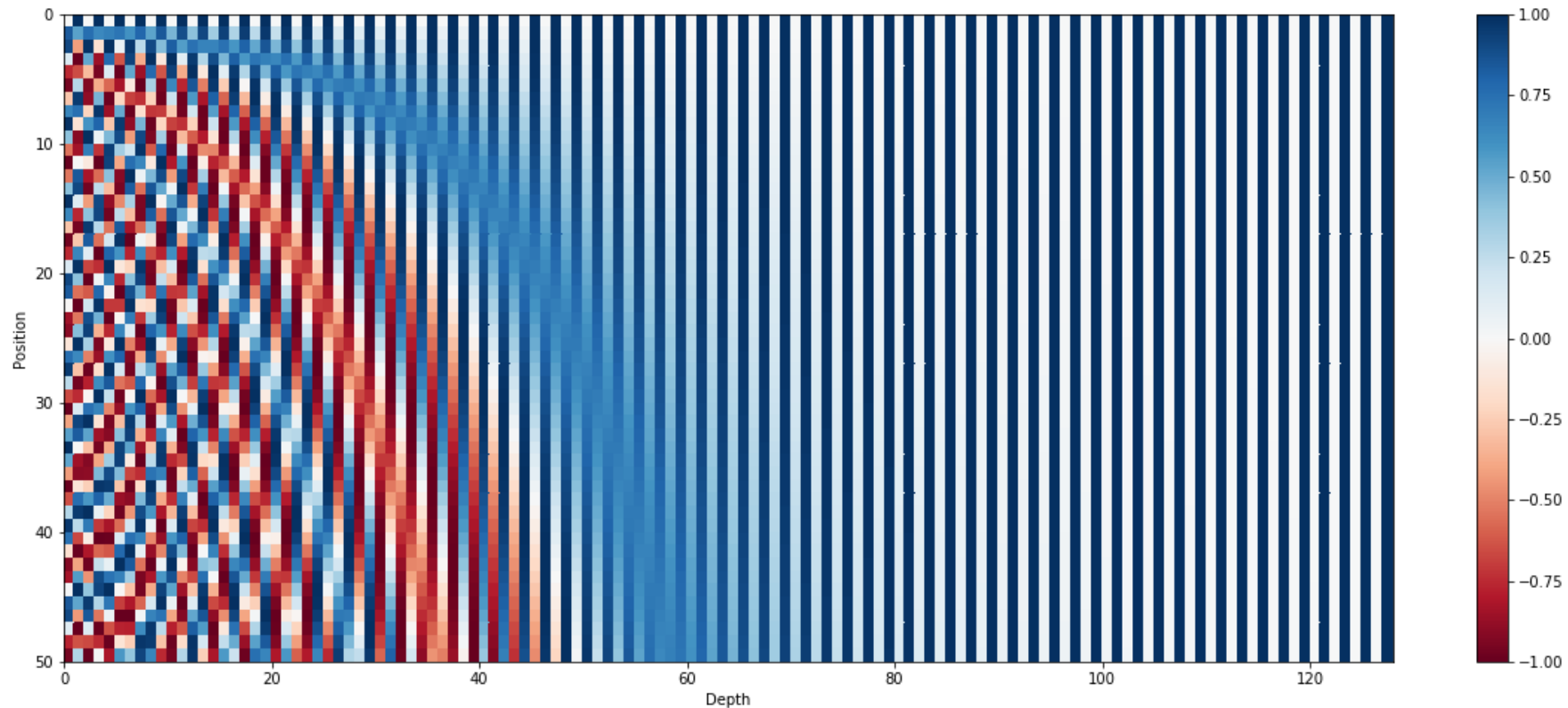
|     |   |   |   |   |      |   |   |   |   |
|-----|---|---|---|---|------|---|---|---|---|
| 0 : | 0 | 0 | 0 | 0 | 8 :  | 1 | 0 | 0 | 0 |
| 1 : | 0 | 0 | 0 | 1 | 9 :  | 1 | 0 | 0 | 1 |
| 2 : | 0 | 0 | 1 | 0 | 10 : | 1 | 0 | 1 | 0 |
| 3 : | 0 | 0 | 1 | 1 | 11 : | 1 | 0 | 1 | 1 |
| 4 : | 0 | 1 | 0 | 0 | 12 : | 1 | 1 | 0 | 0 |
| 5 : | 0 | 1 | 0 | 1 | 13 : | 1 | 1 | 0 | 1 |
| 6 : | 0 | 1 | 1 | 0 | 14 : | 1 | 1 | 1 | 0 |
| 7 : | 0 | 1 | 1 | 1 | 15 : | 1 | 1 | 1 | 1 |

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

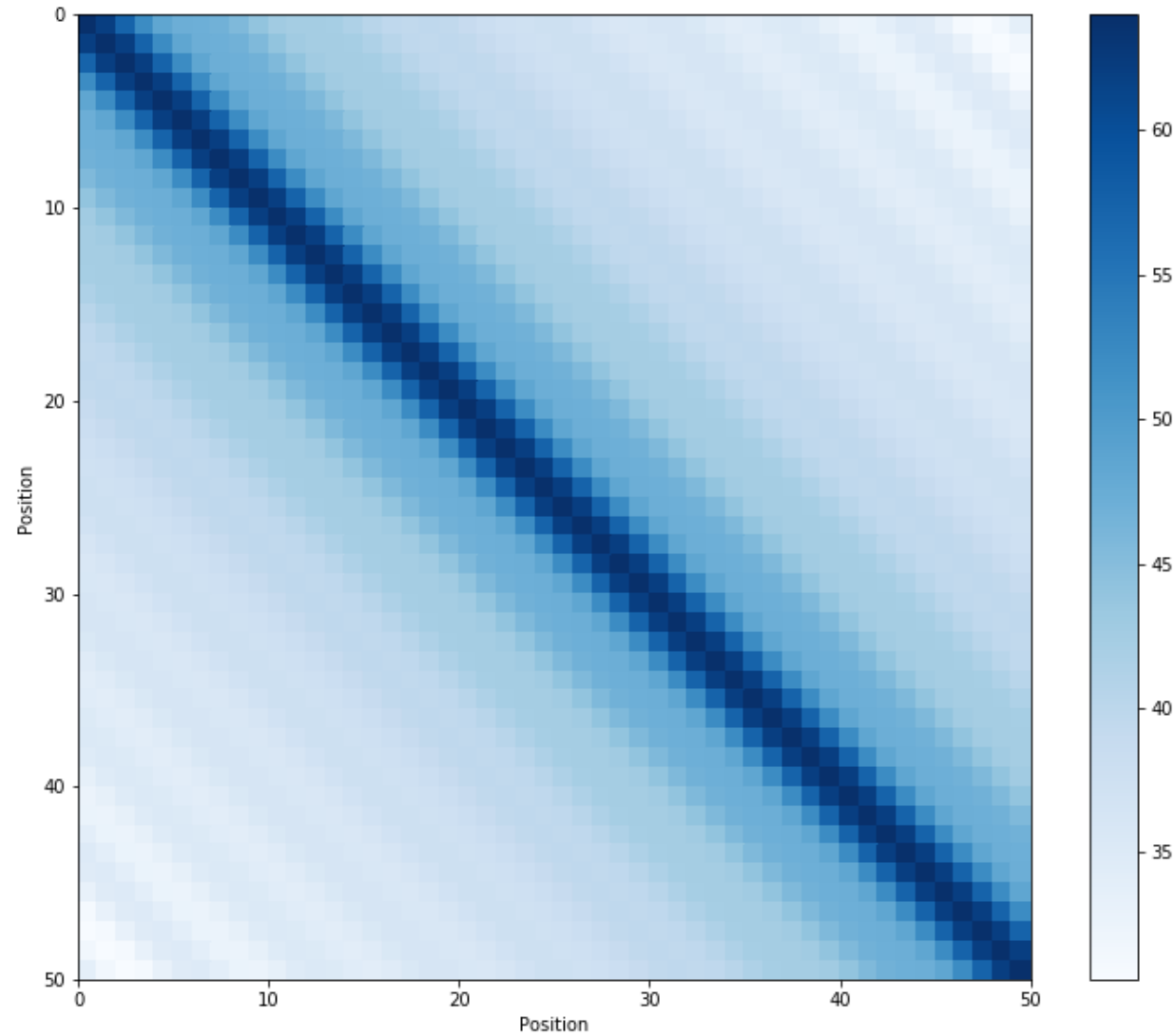
Soft version of alternating bits

# Sinusoidal Positional Encoding

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

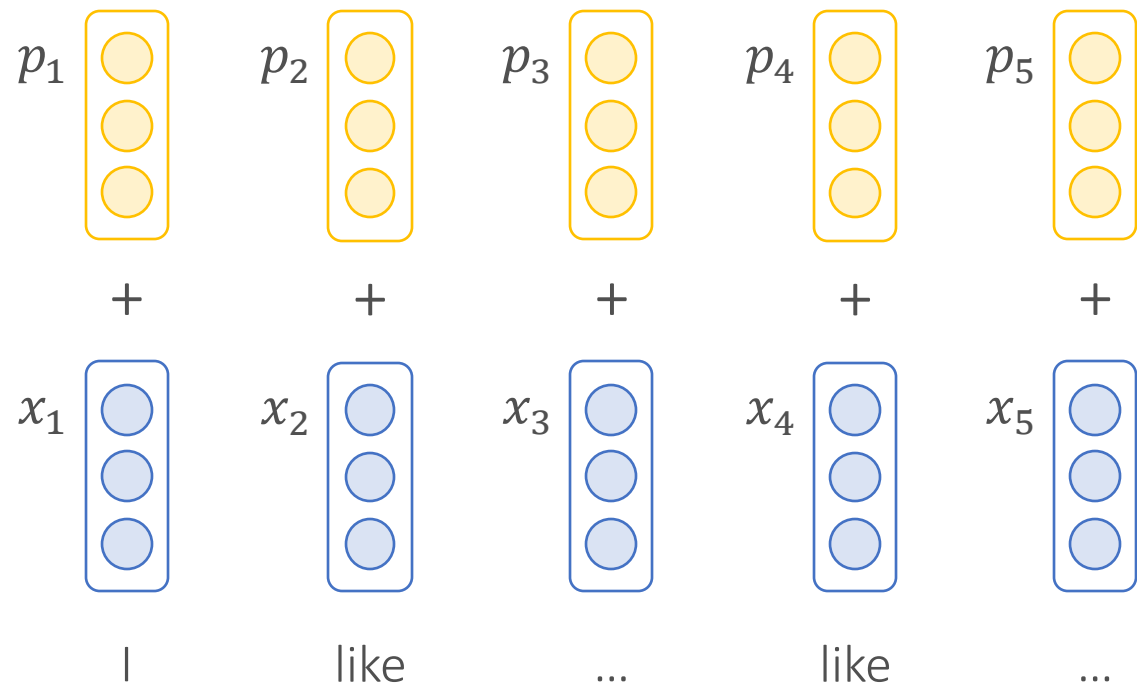


# Sinusoidal Positional Encoding





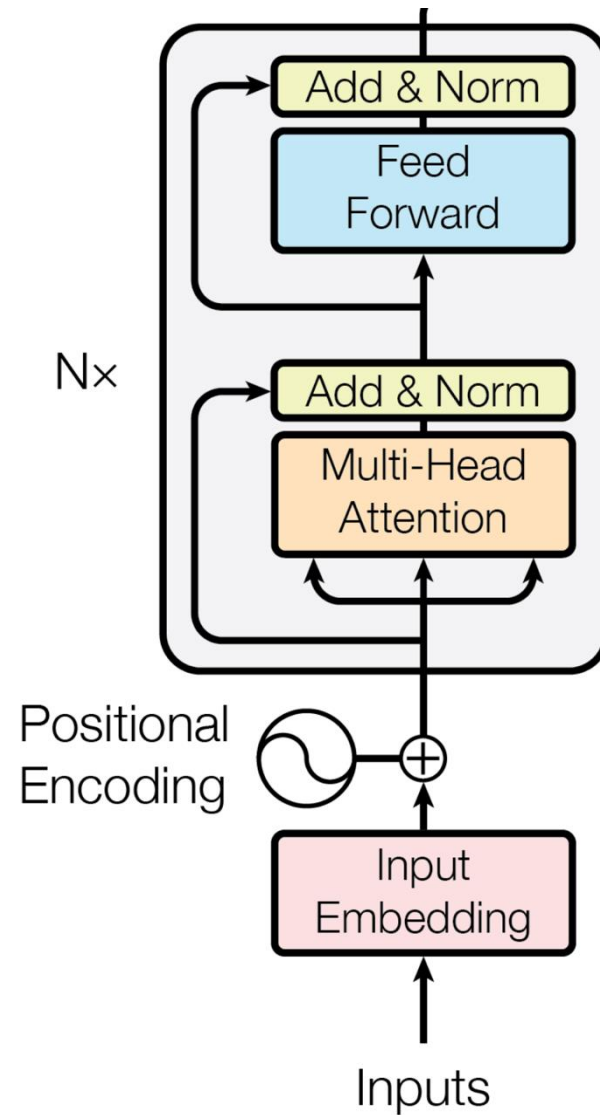
# Positional Encoding



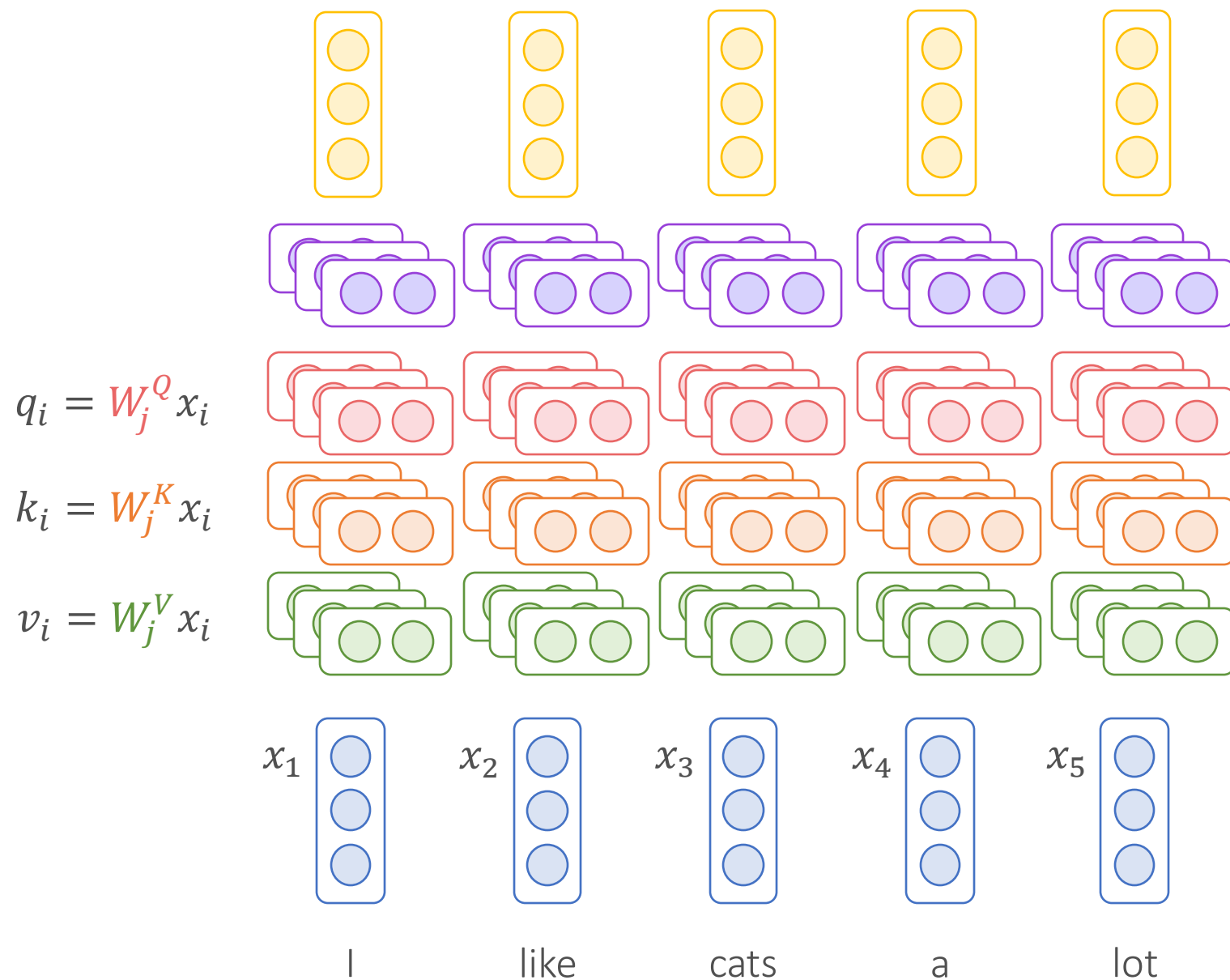
$$\begin{aligned} (E(I) + PE(1)) (E(like) + PE(2)) &= E(I)E(like) + E(I)PE(2) + PE(1)E(like) + PE(1)PE(2) \\ (E(I) + PE(1)) (E(like) + PE(4)) &= E(I)E(like) + E(I)PE(4) + PE(1)E(like) + PE(1)PE(4) \end{aligned}$$

In expectation, they are the same                      Position difference

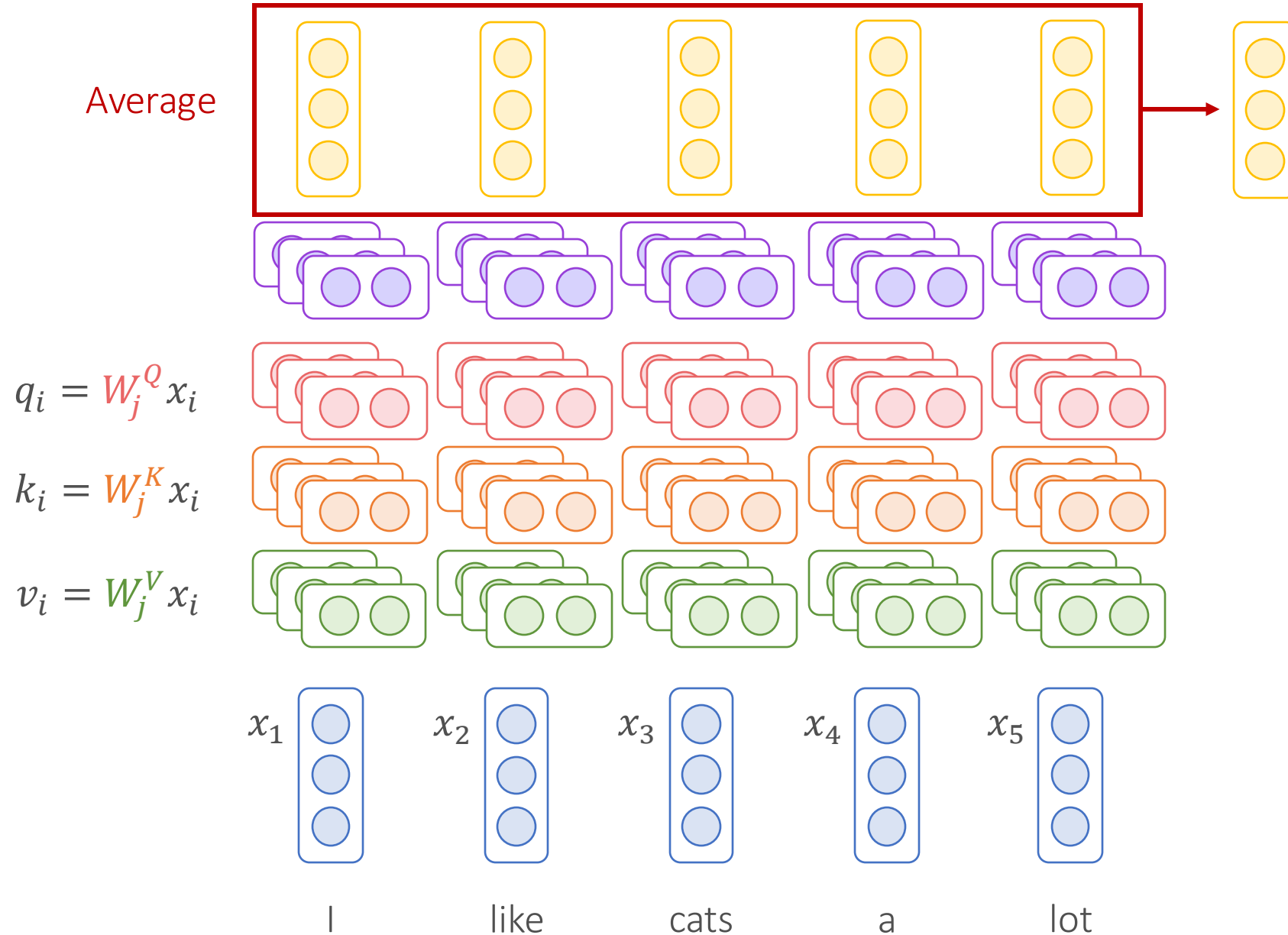
# Transformer Encoder with Positional Encoding



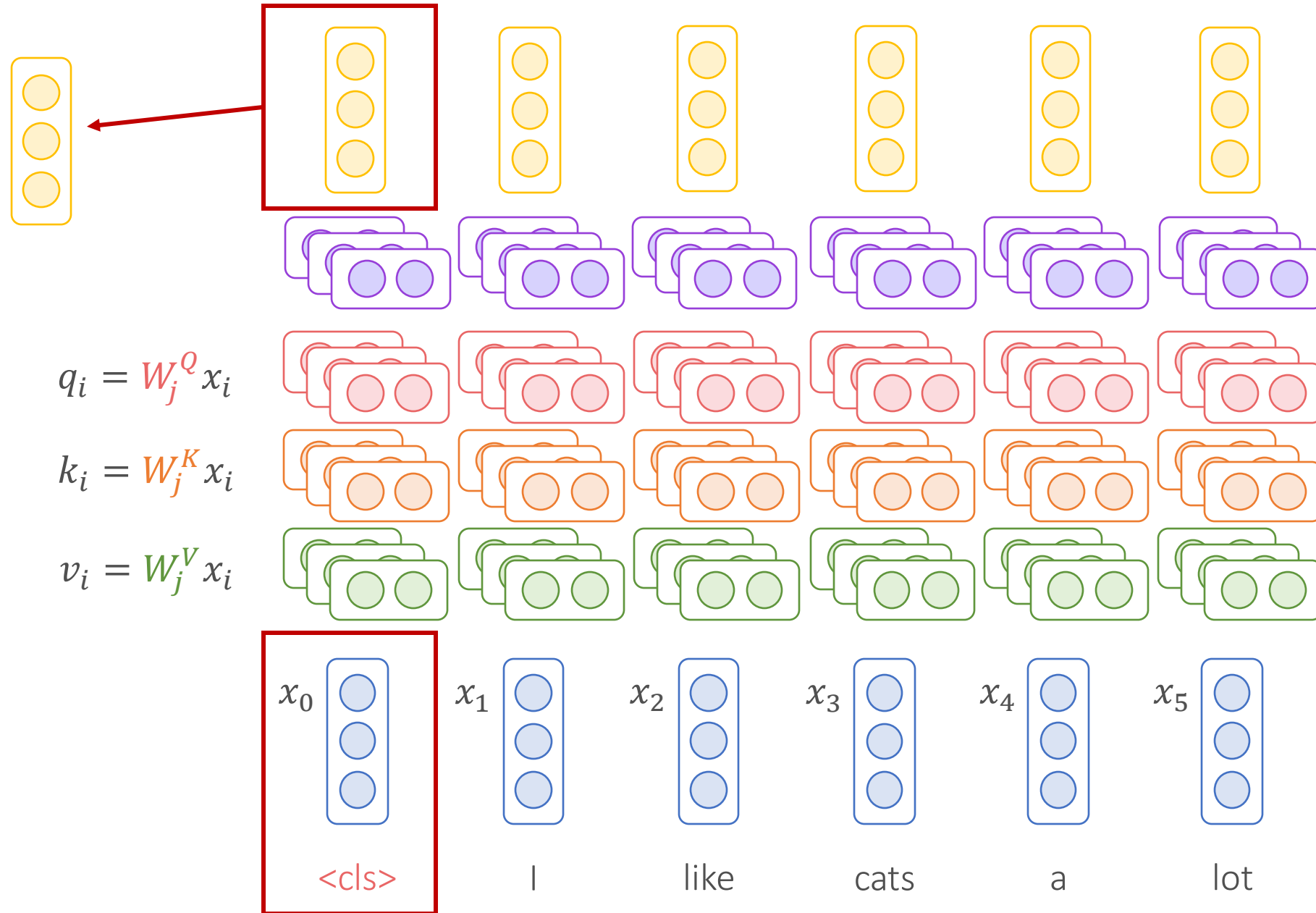
# Transformer as Token-Level Encoder



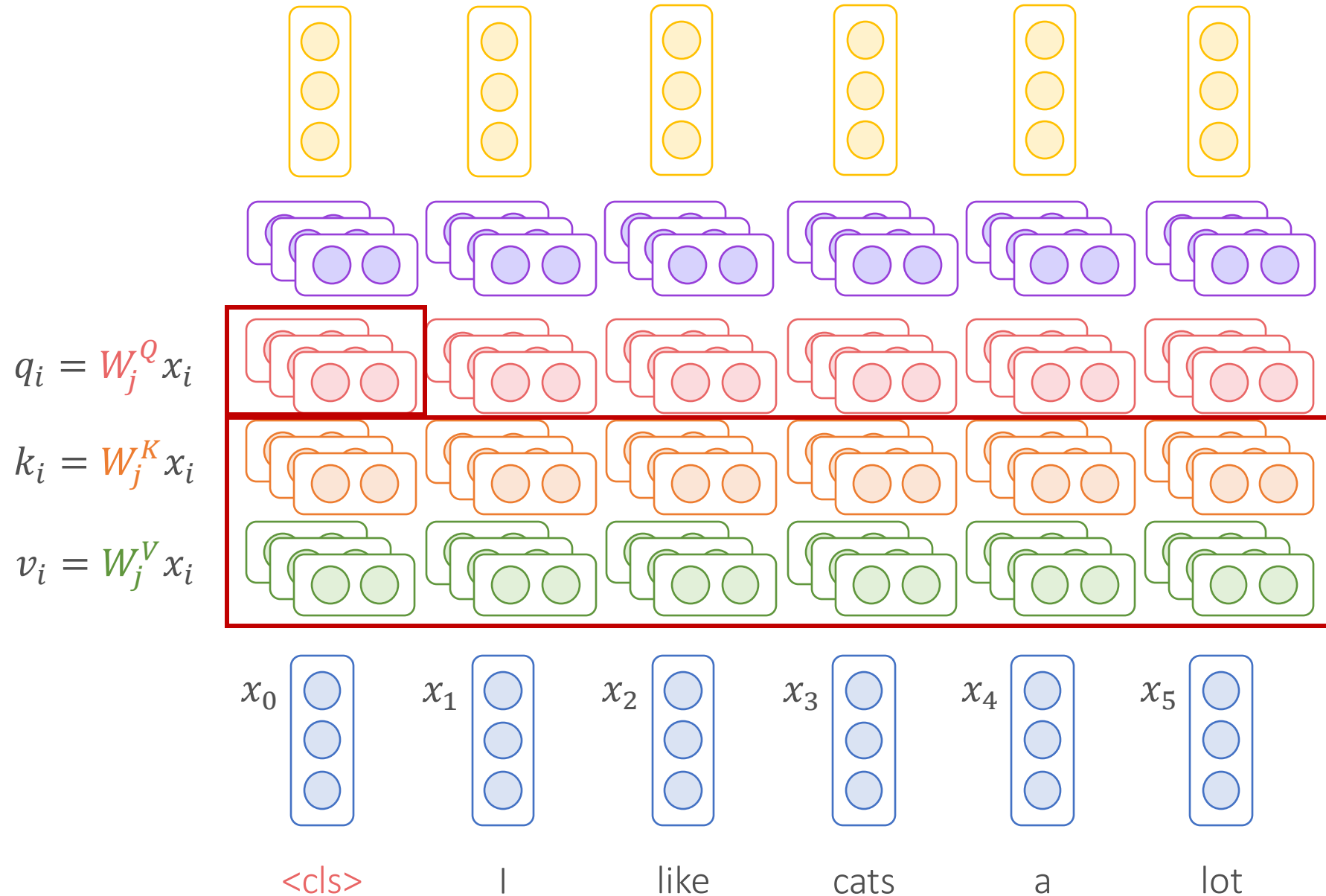
# Transformer as Sentence-Level Encoder



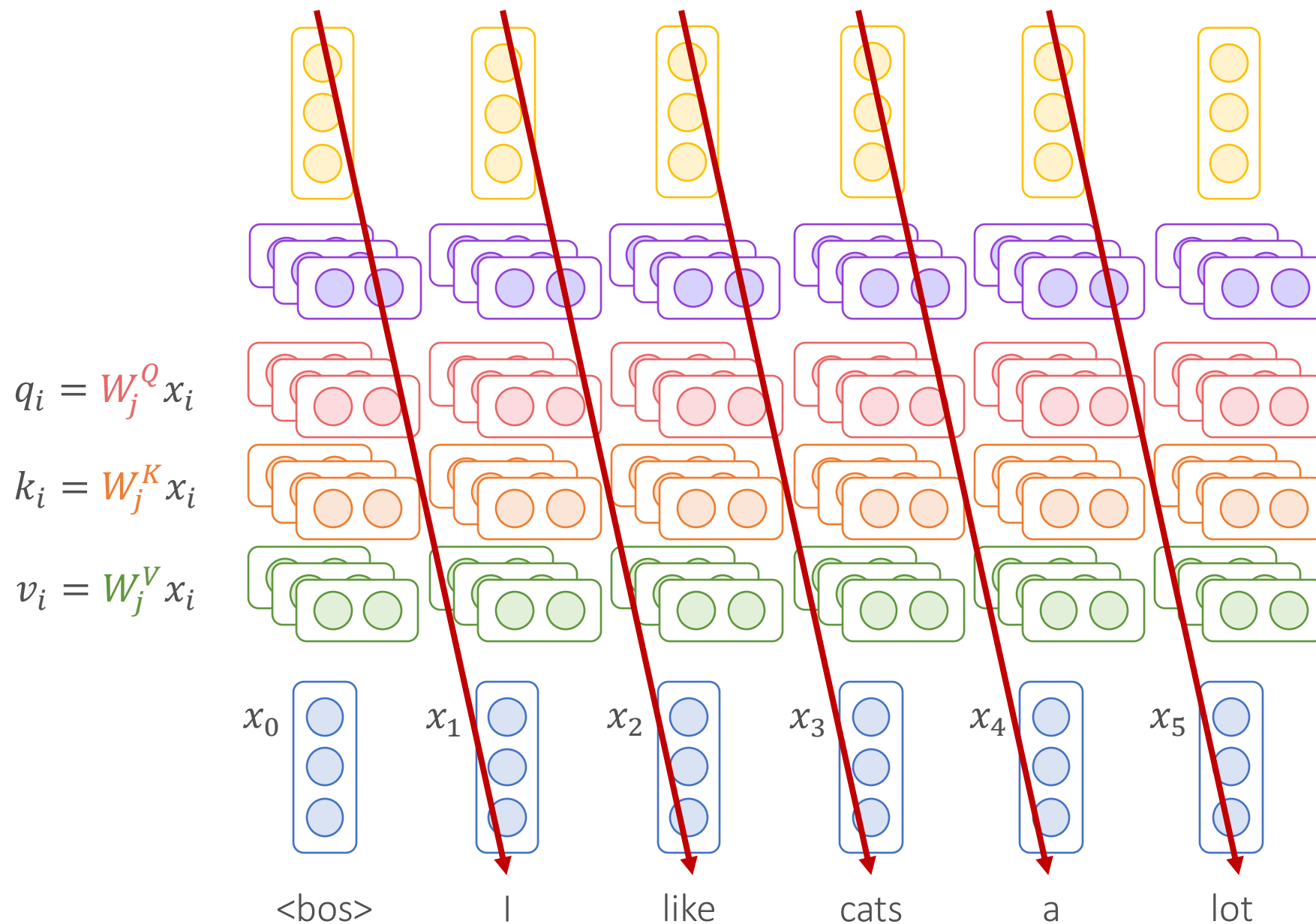
# Transformer as Sentence-Level Encoder



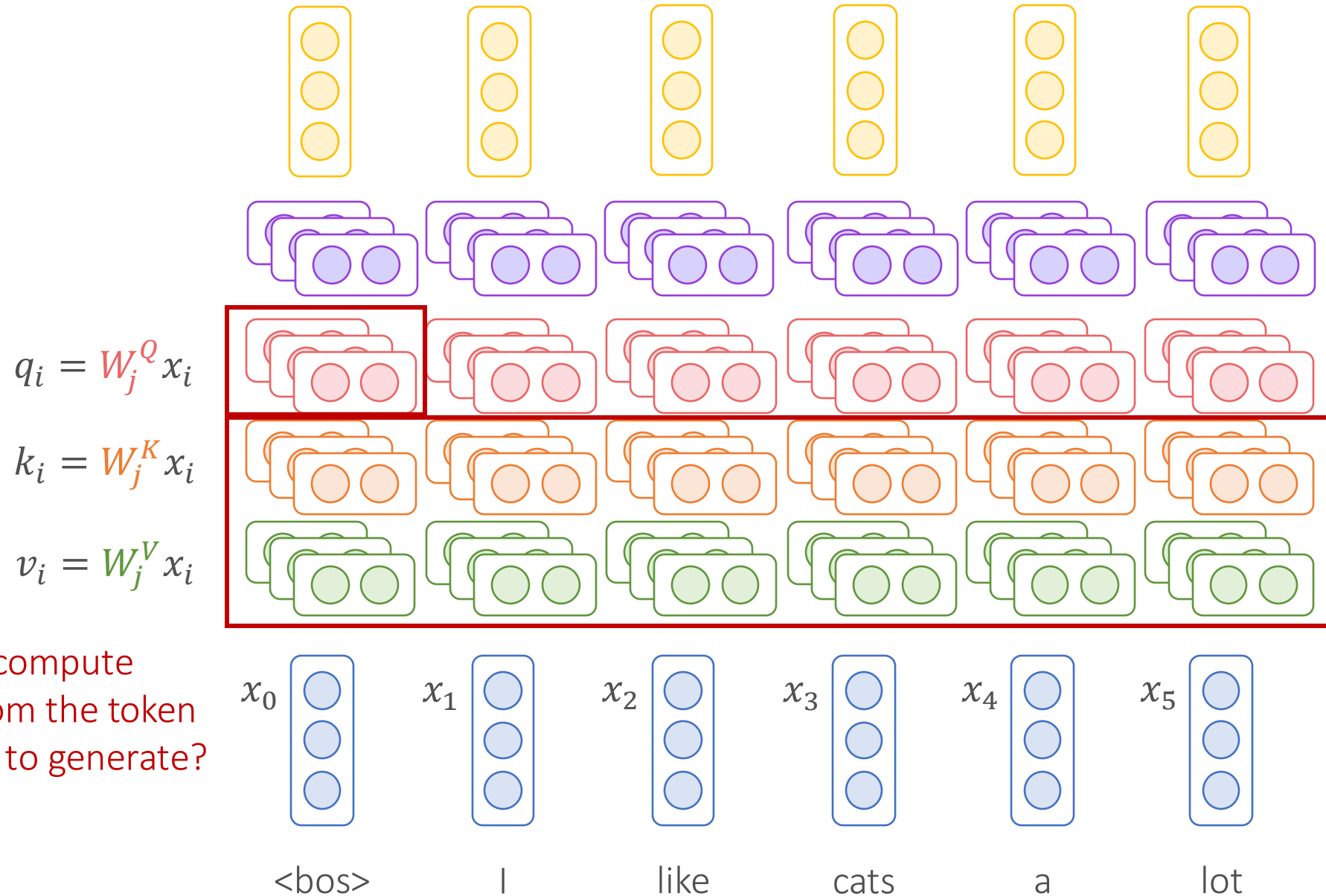
# Transformer as Sentence-Level Encoder



# Transformer as Decoder?



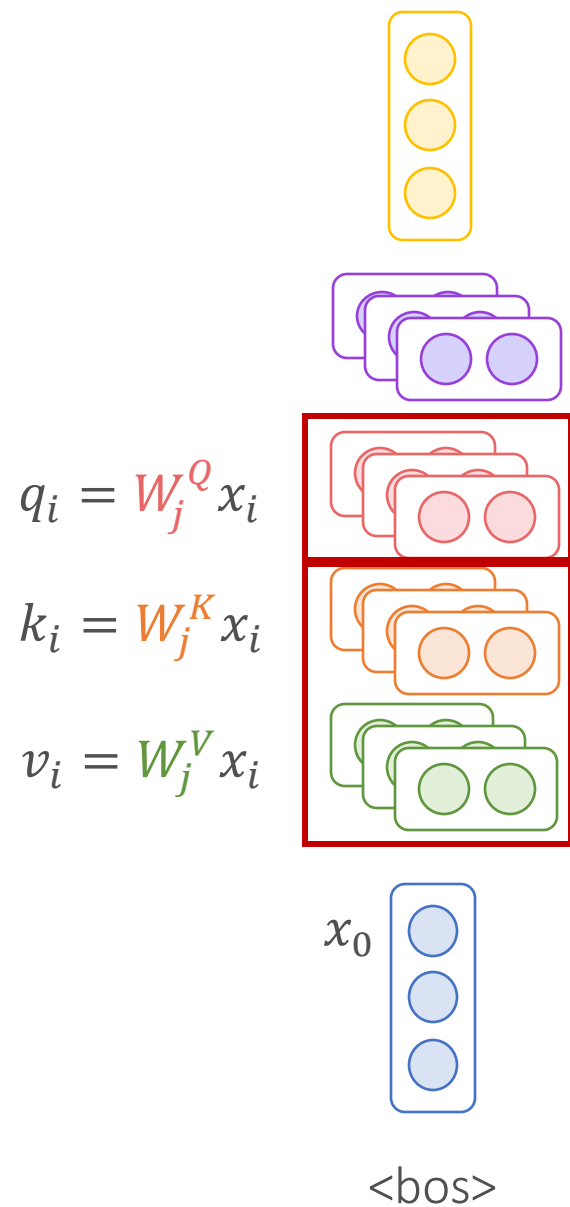
# Transformer as Decoder?



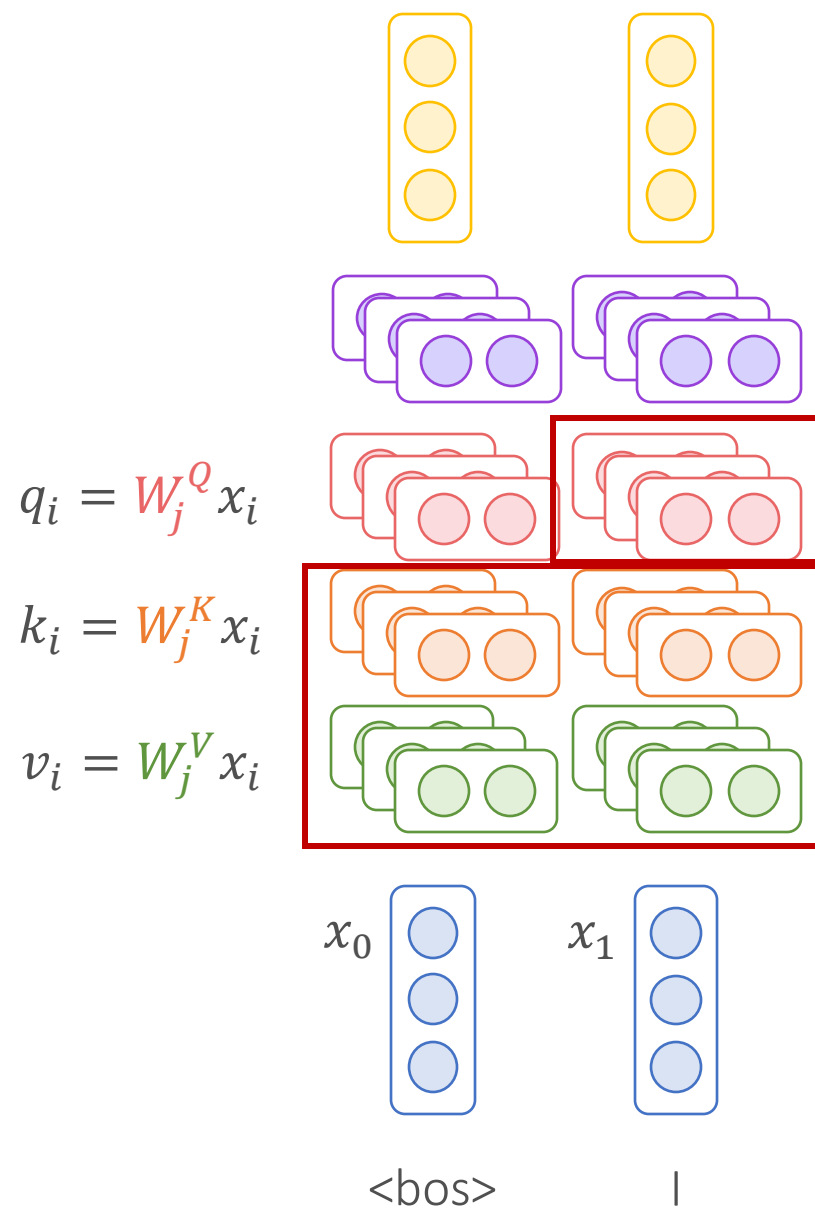
How to compute  
attention from the token  
we are going to generate?



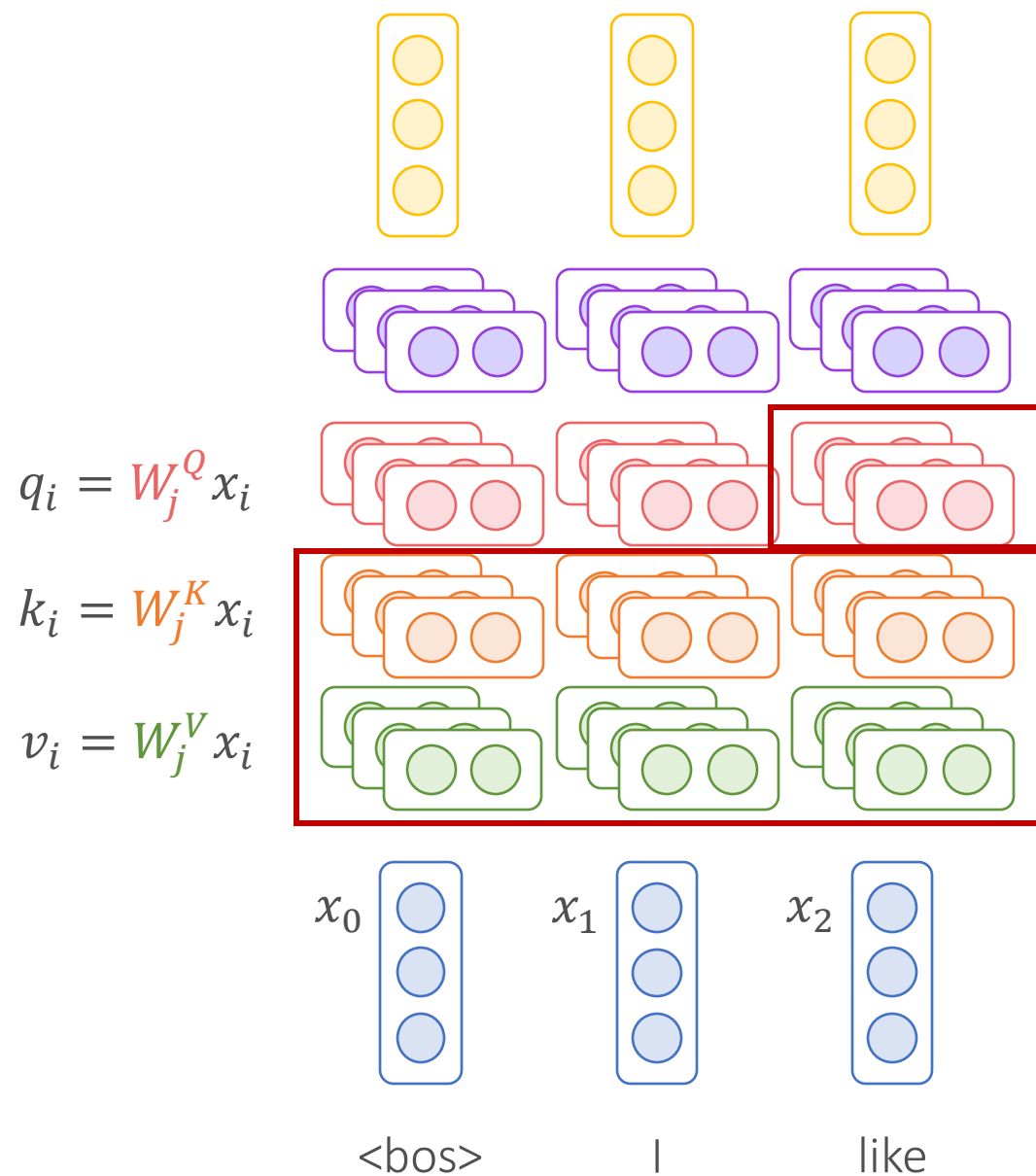
# Transformer Decoder



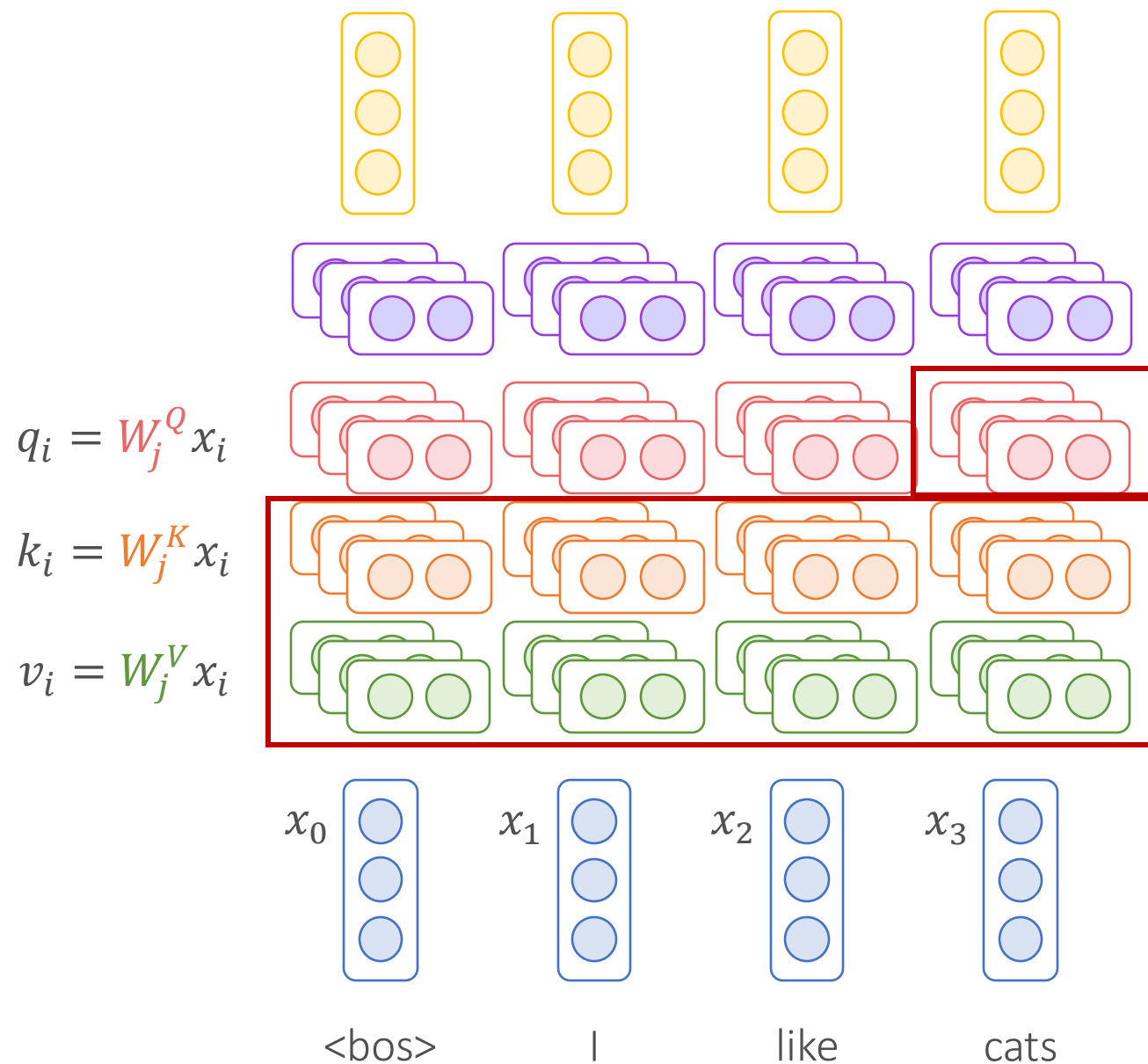
# Transformer Decoder



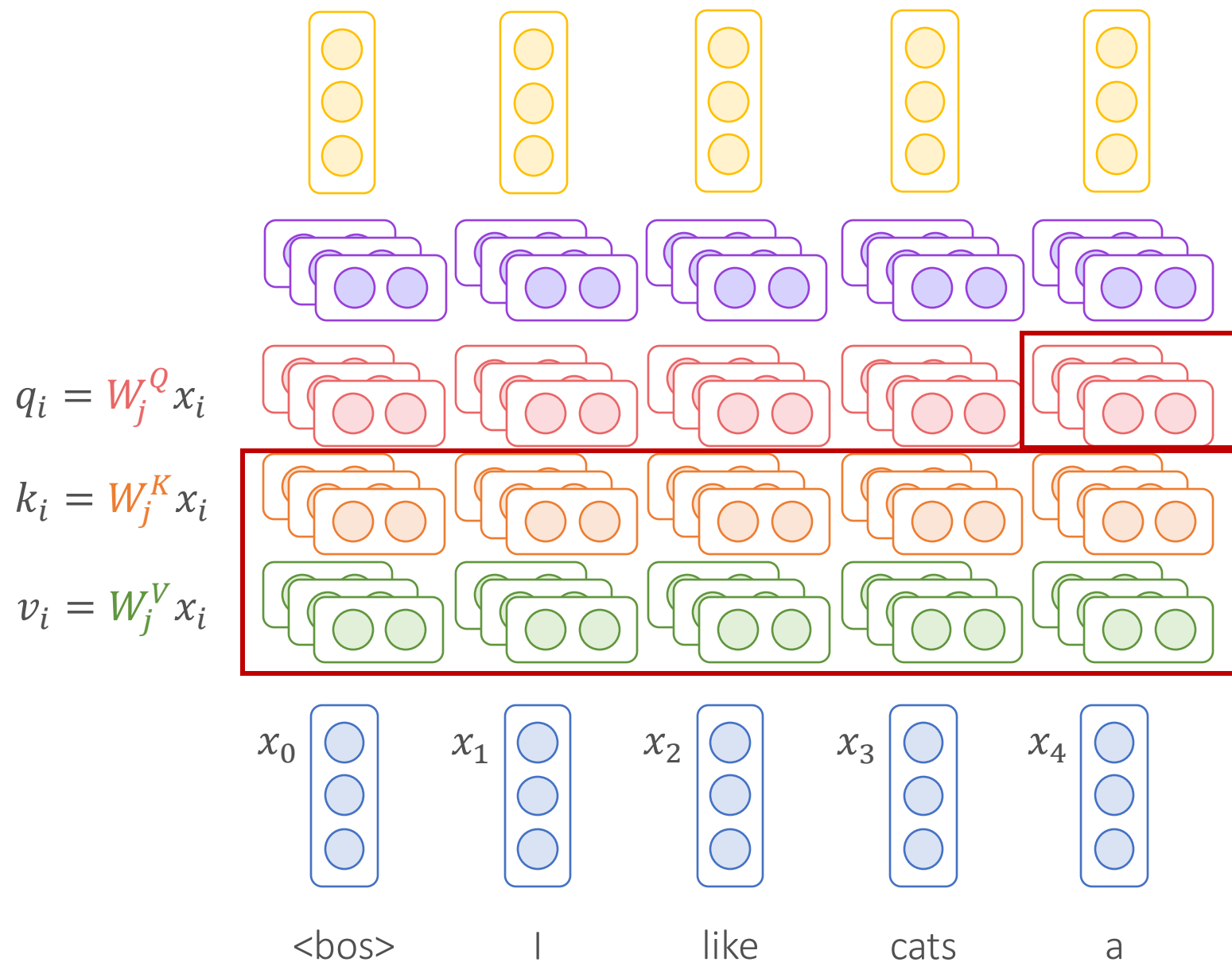
# Transformer Decoder



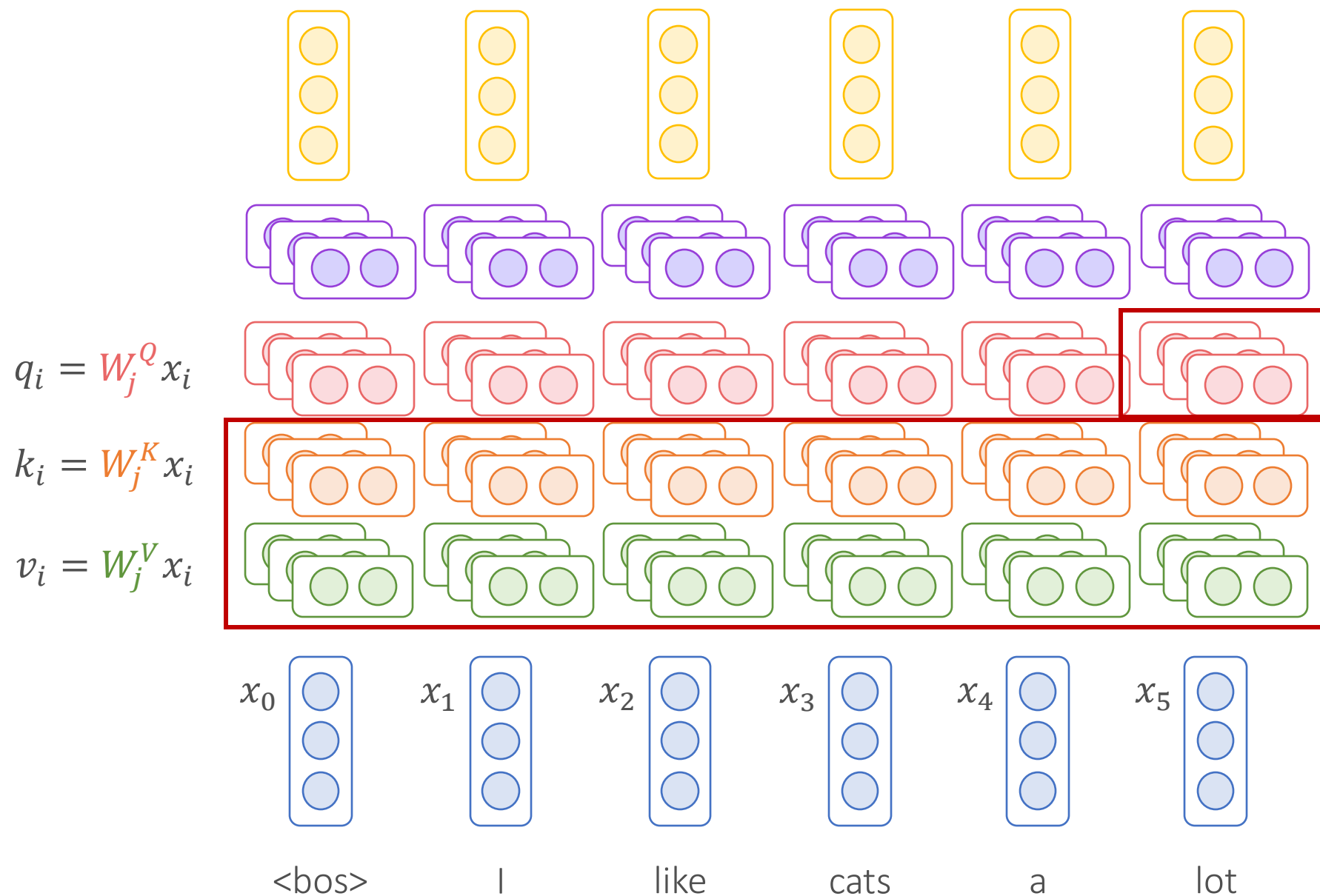
# Transformer Decoder



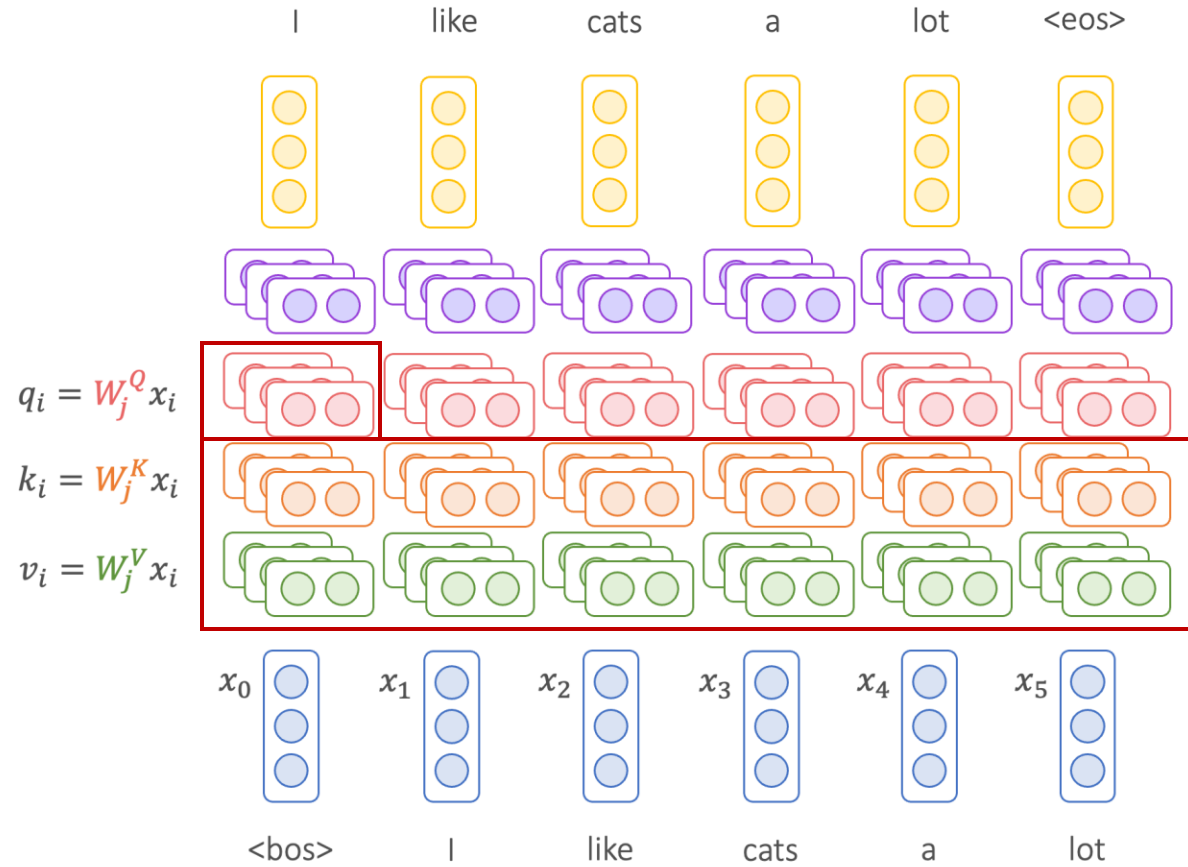
# Transformer Decoder



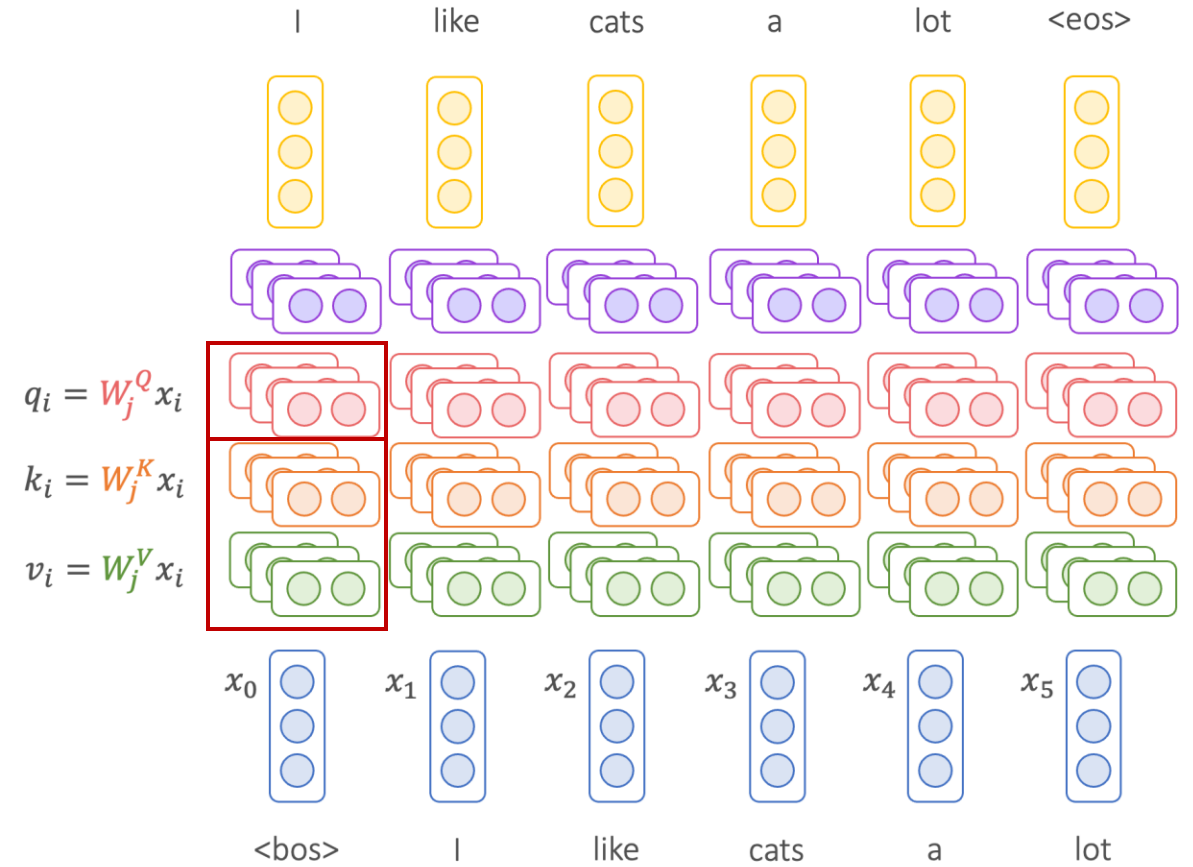
# Transformer Decoder



# Transformer Encoder vs. Transformer Decoder

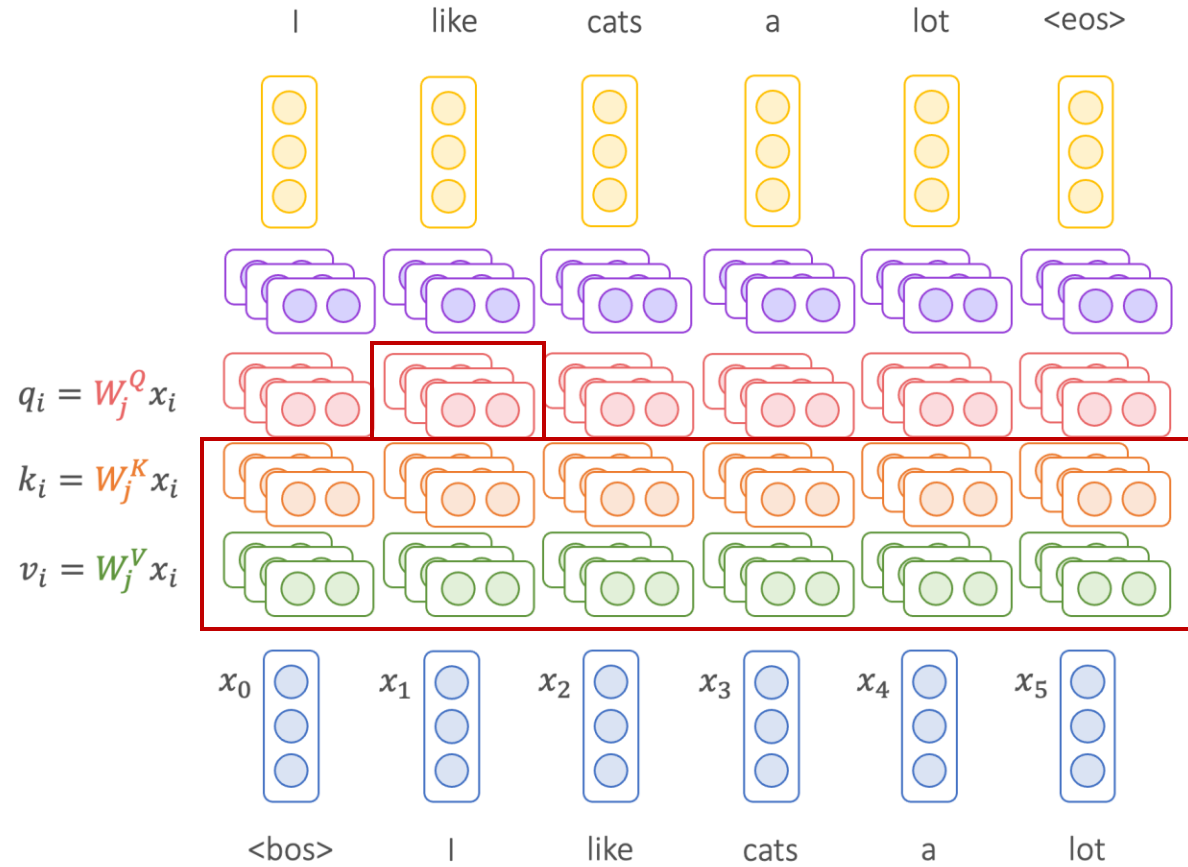


Transformer Encoder

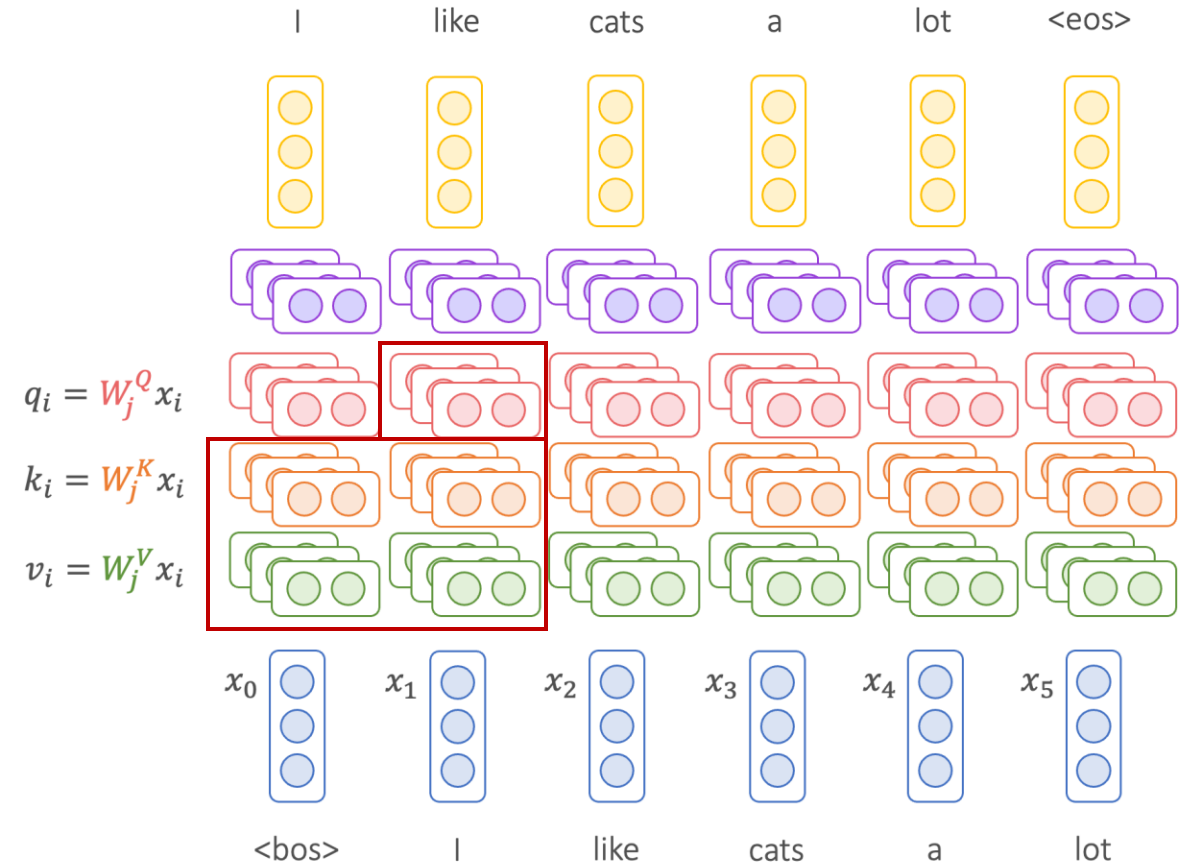


Transformer Decoder

# Transformer Encoder vs. Transformer Decoder



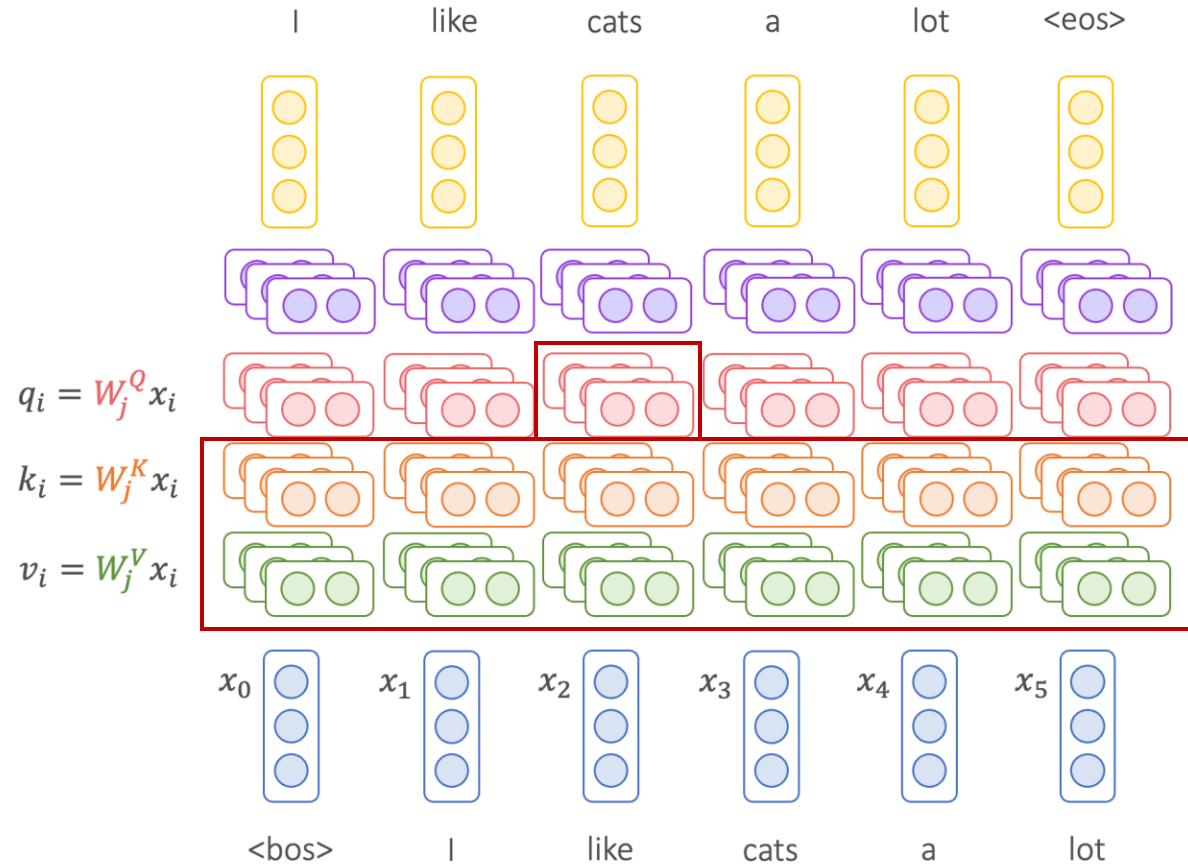
Transformer Encoder



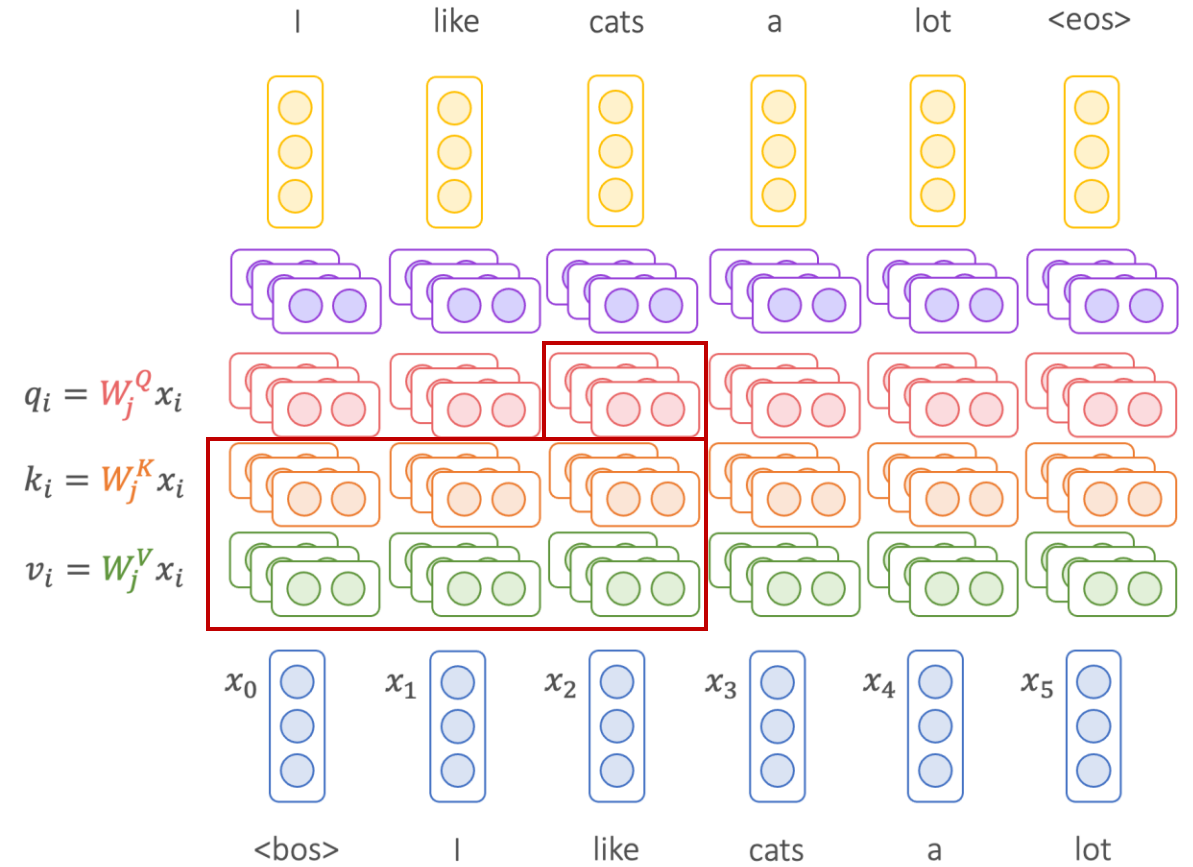
Transformer Decoder



# Transformer Encoder vs. Transformer Decoder



Transformer Encoder

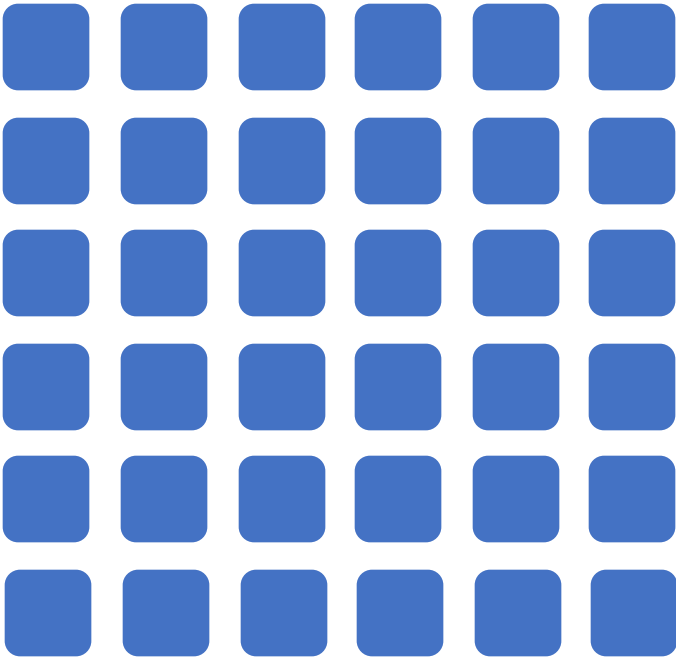
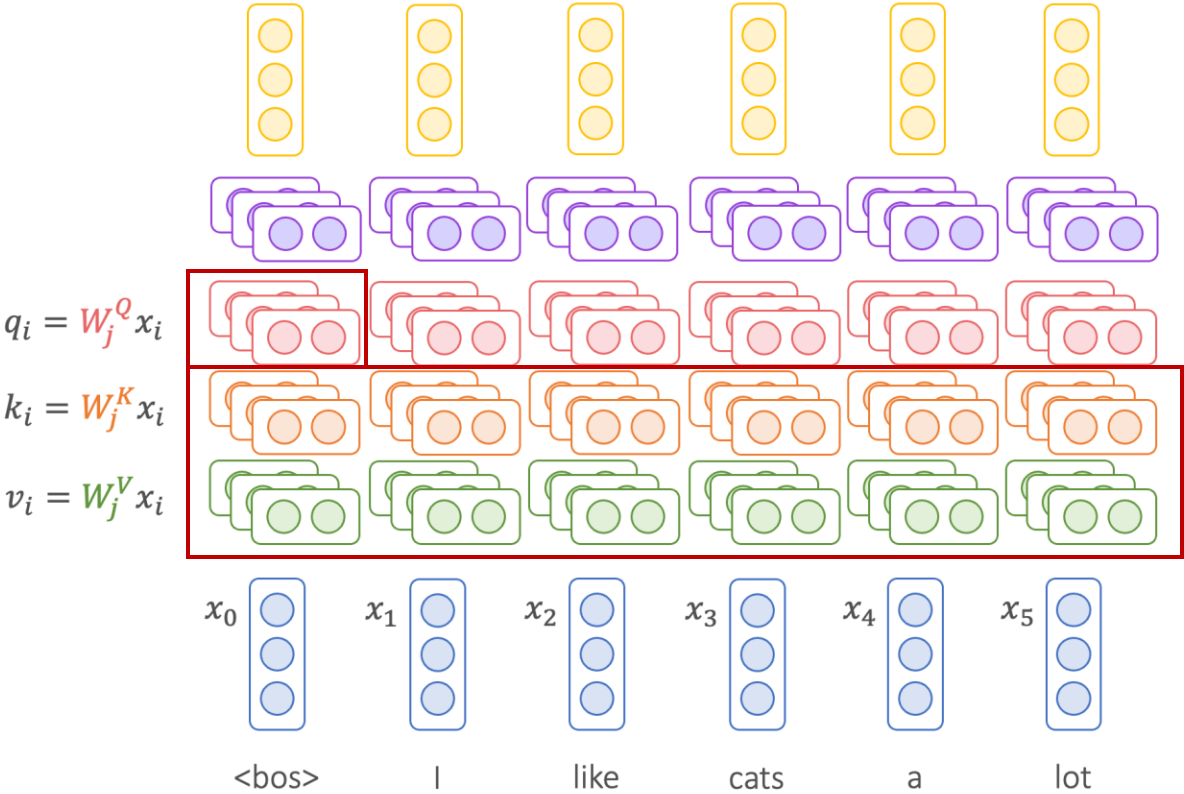


Transformer Decoder

# Transformer Encoder vs. Transformer Decoder

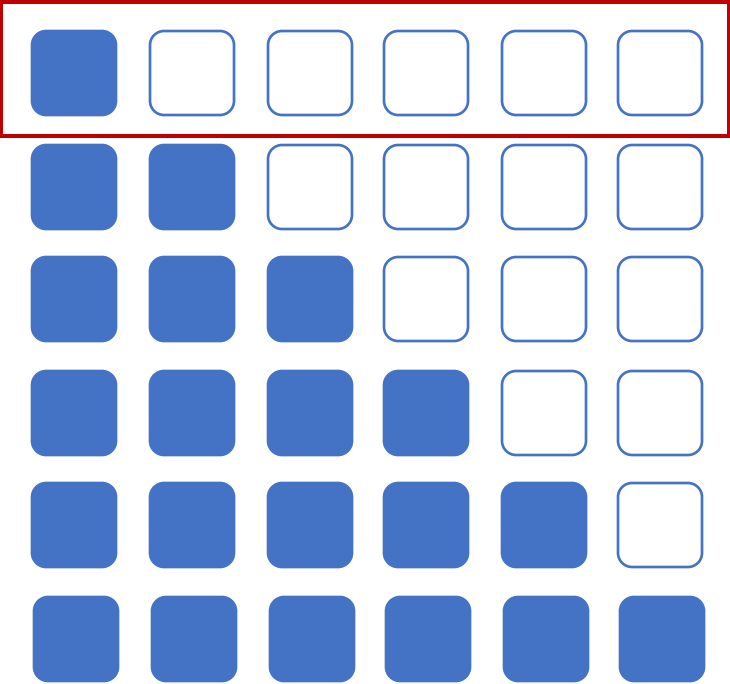
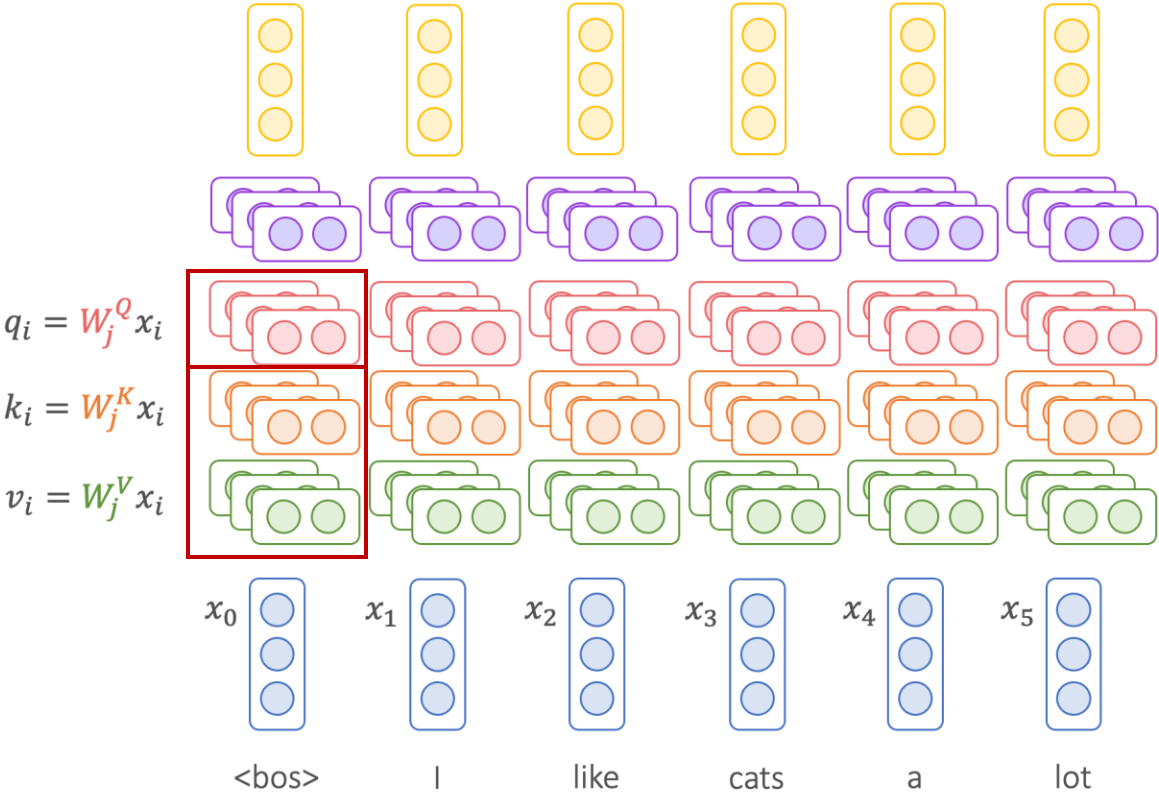
- When computing attention for one word
  - Encoder: can see the words **before and after** this word
  - Decoder: can see the words **only before** this word

# Masked Attention for Transformer Encoder



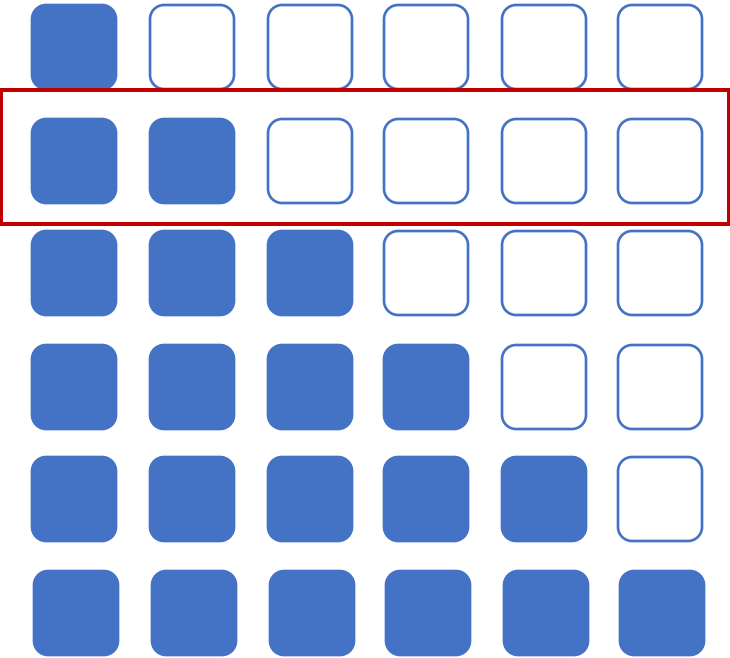
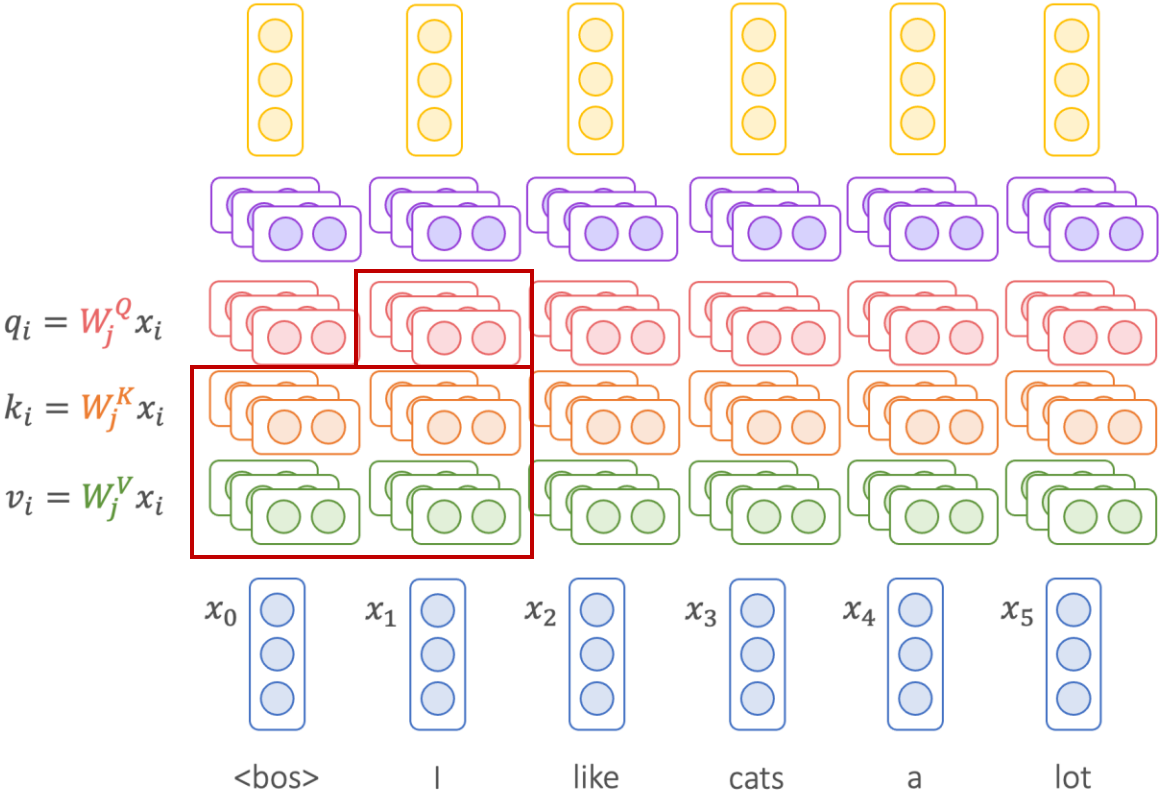
No Masking

# Masked Attention for Transformer Decoder



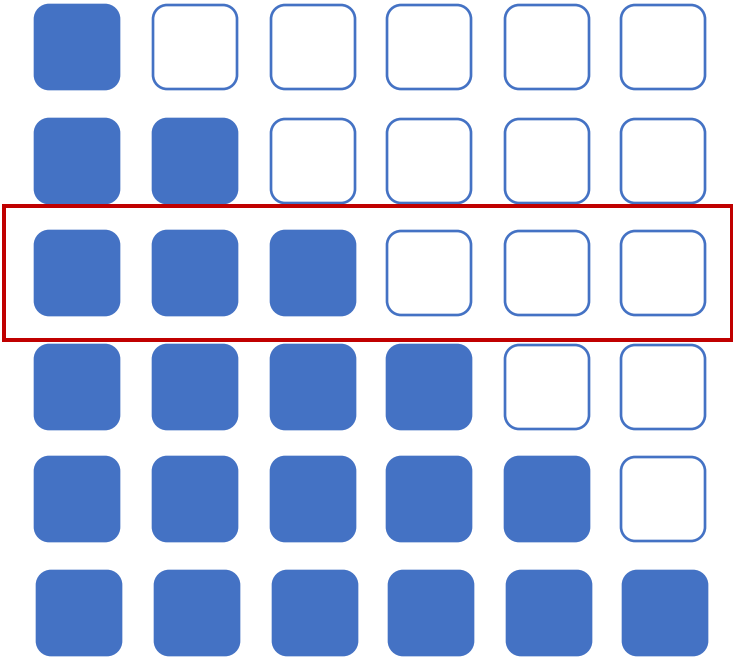
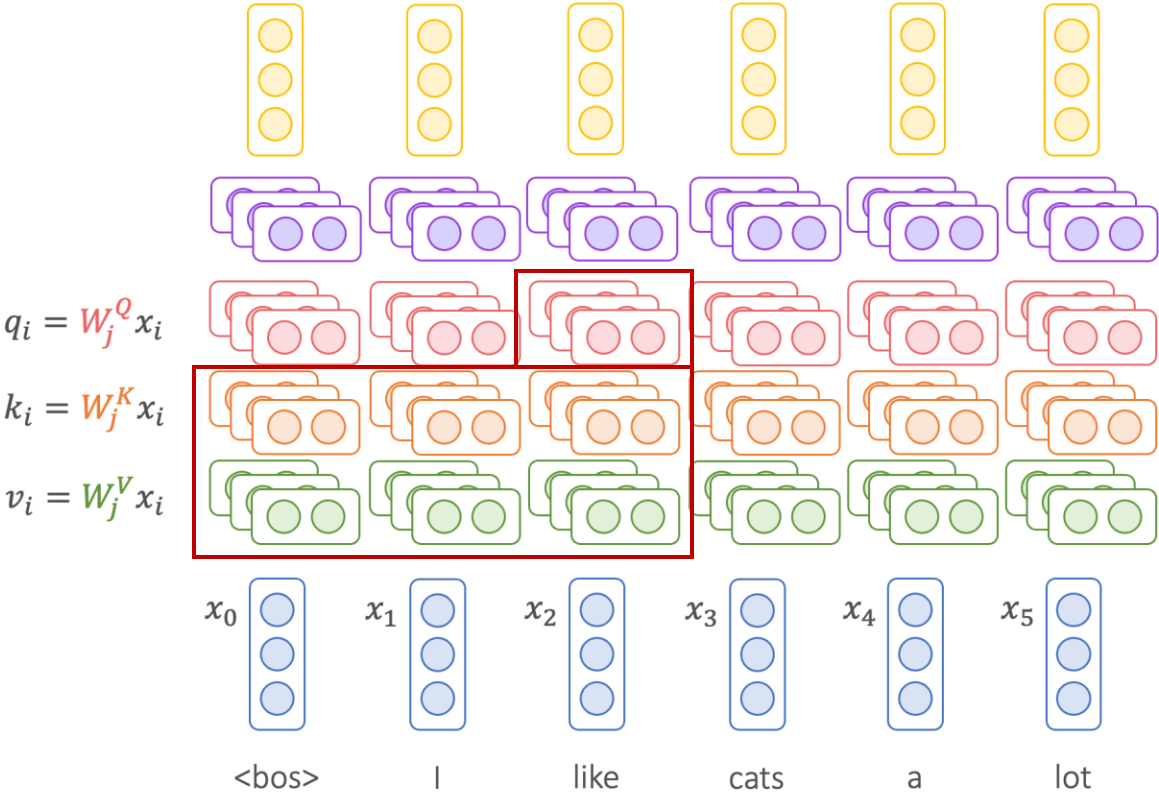
Causal Masking

# Masked Attention for Transformer Decoder



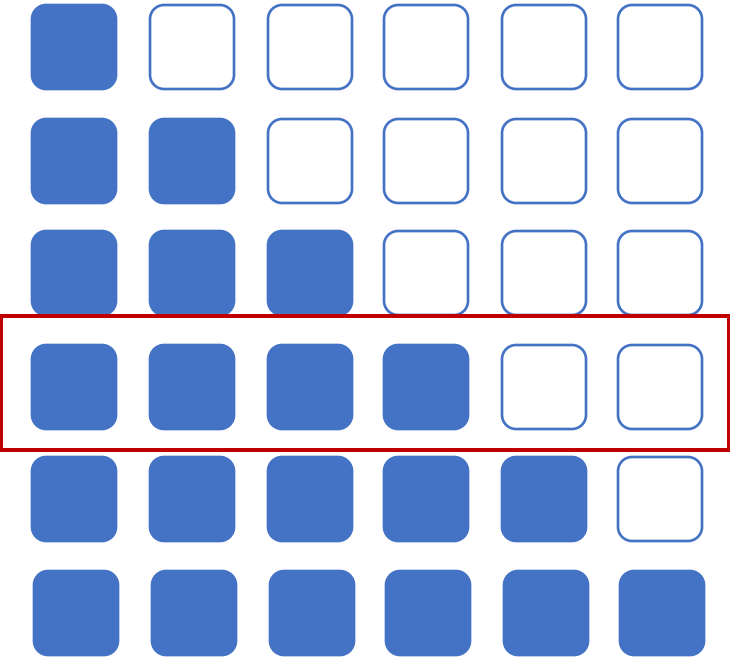
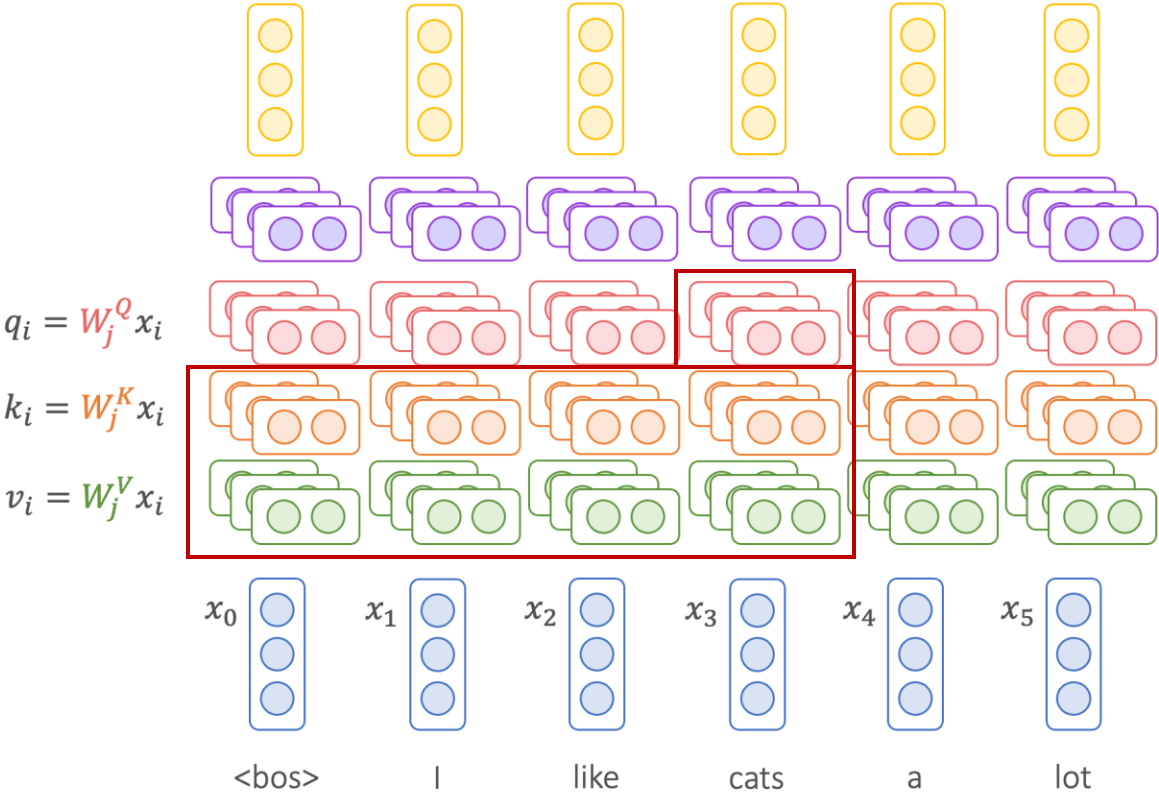
Causal Masking

# Masked Attention for Transformer Decoder



Causal Masking

# Masked Attention for Transformer Decoder



Causal Masking

# Masked Attention: Implementation

