Attention is all you need ?

刘天禹

August 1st 2019

Timelines

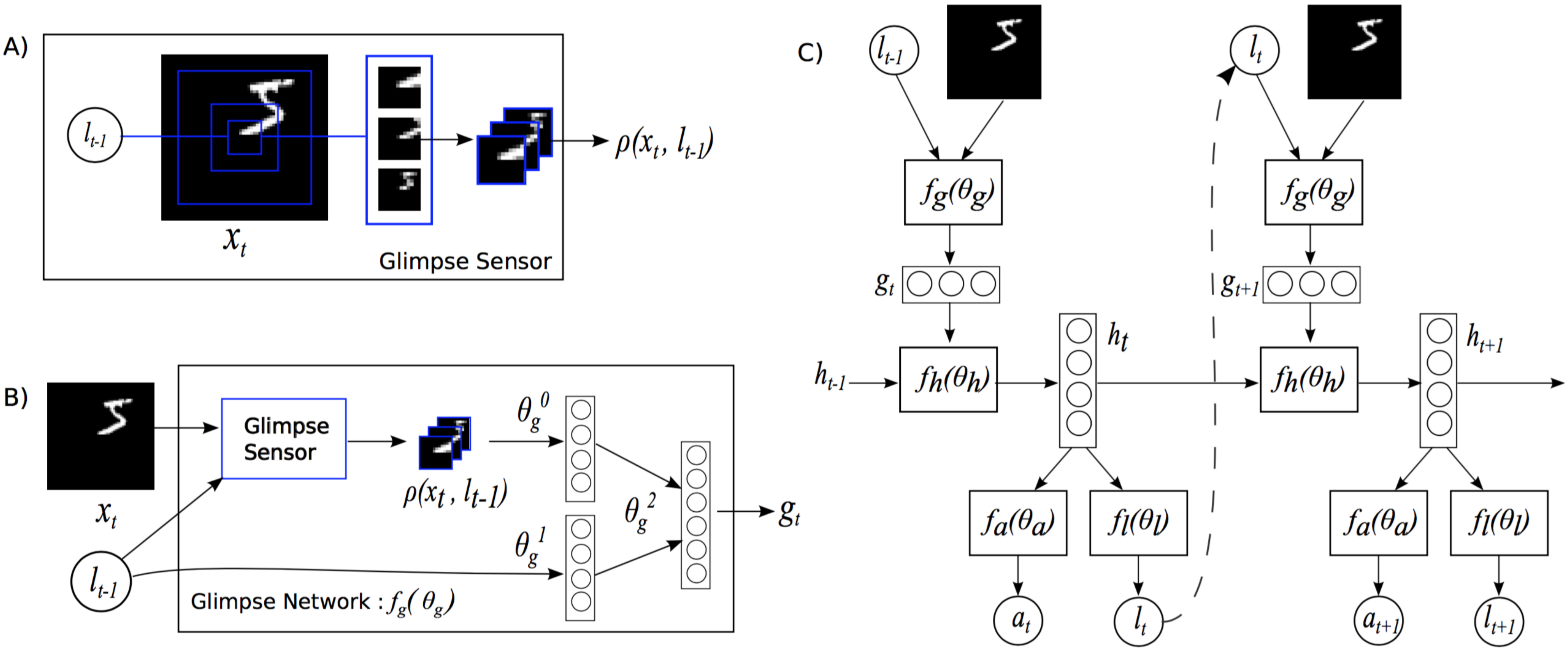
- 2014 Recurrent Models of Visual Attention
- 2017 Attention is All You Need
- 2018 Image Transformer
- 2018 Non-local Neural Networks
- 2019 Self-Attention GAN
- 2019 Multi-Channel Attention Selection GAN with Cascaded Semantic Guidance for Cross-View Image Translation

Recurrent Models of Visual Attention

Attention for Visual with Reinforcement Learning

Google DeepMind, Volodymyr Mnih & Alex Graves

NIPS 2014



$$a_t \sim p(\cdot | f_a(h_t; \theta_a))$$

$$l_t \sim p(\cdot | f_l(h_t; \theta_l)).$$

Reinforcement Learning

https://blog.csdn.net/yexiaogu1104/article/details/89455718

Attention is All You Need

"Attention is all you need"

Google Brain

December 2017

Sequence Learning

- Sequence Learning
  (Variable Length Sequence of Words of Pixels)
    - RNNs
    - LSTM&GRU

- Recurrent Convolution Structure

Stanford CS224N-L14 Winter 2019 https://www.youtube.com/watch?v=5vcj8kSwBCY&t=1619s

But…

- No Parallelization
- No Forward Thinking
- No explicit model of Long-term dependency
- **Computationally Wasteful**

How about Convolution？

- Trivial to parallelize

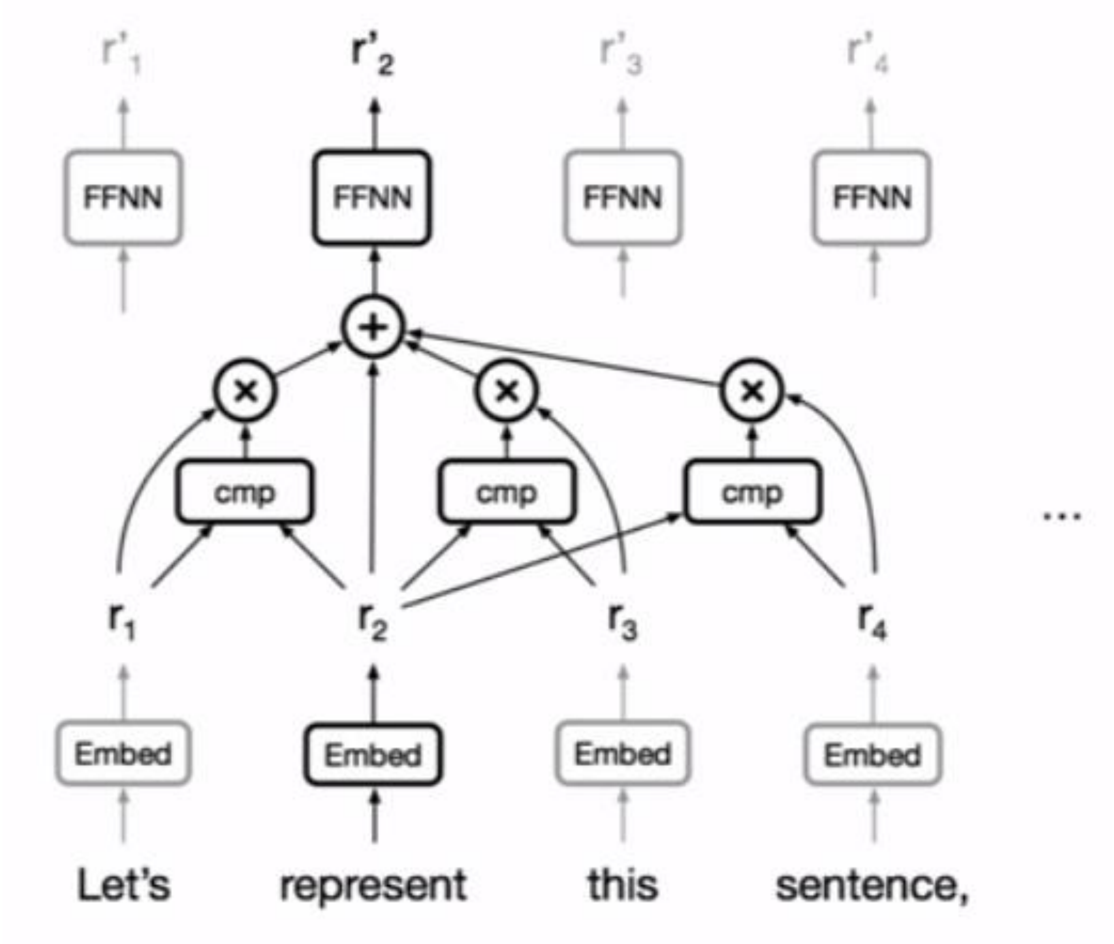
- Local dependencies

- **Long-distance require many layers**

Attention !

- Crucial in Neural machine translation
- **How about representation ?**

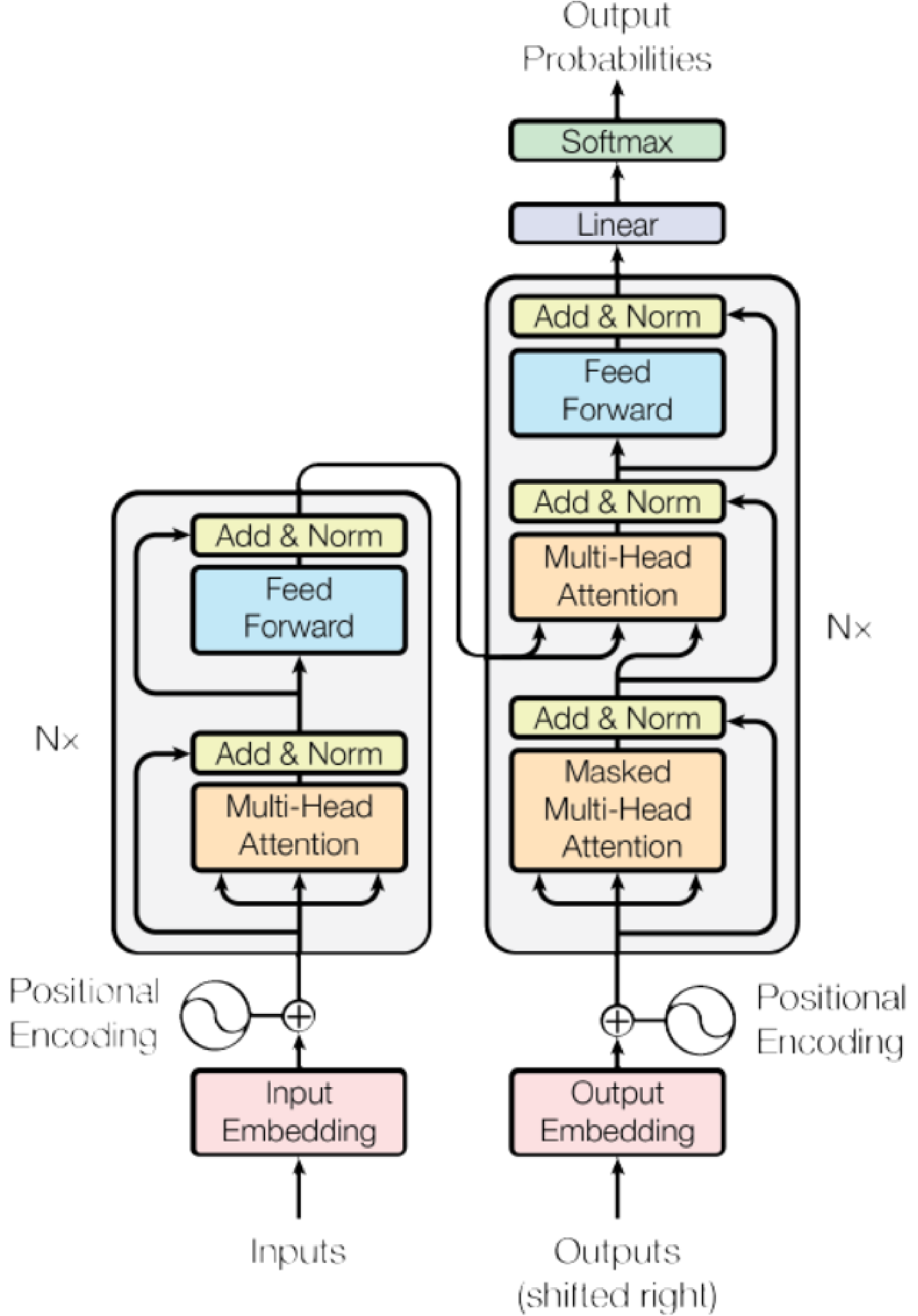
Self Attention



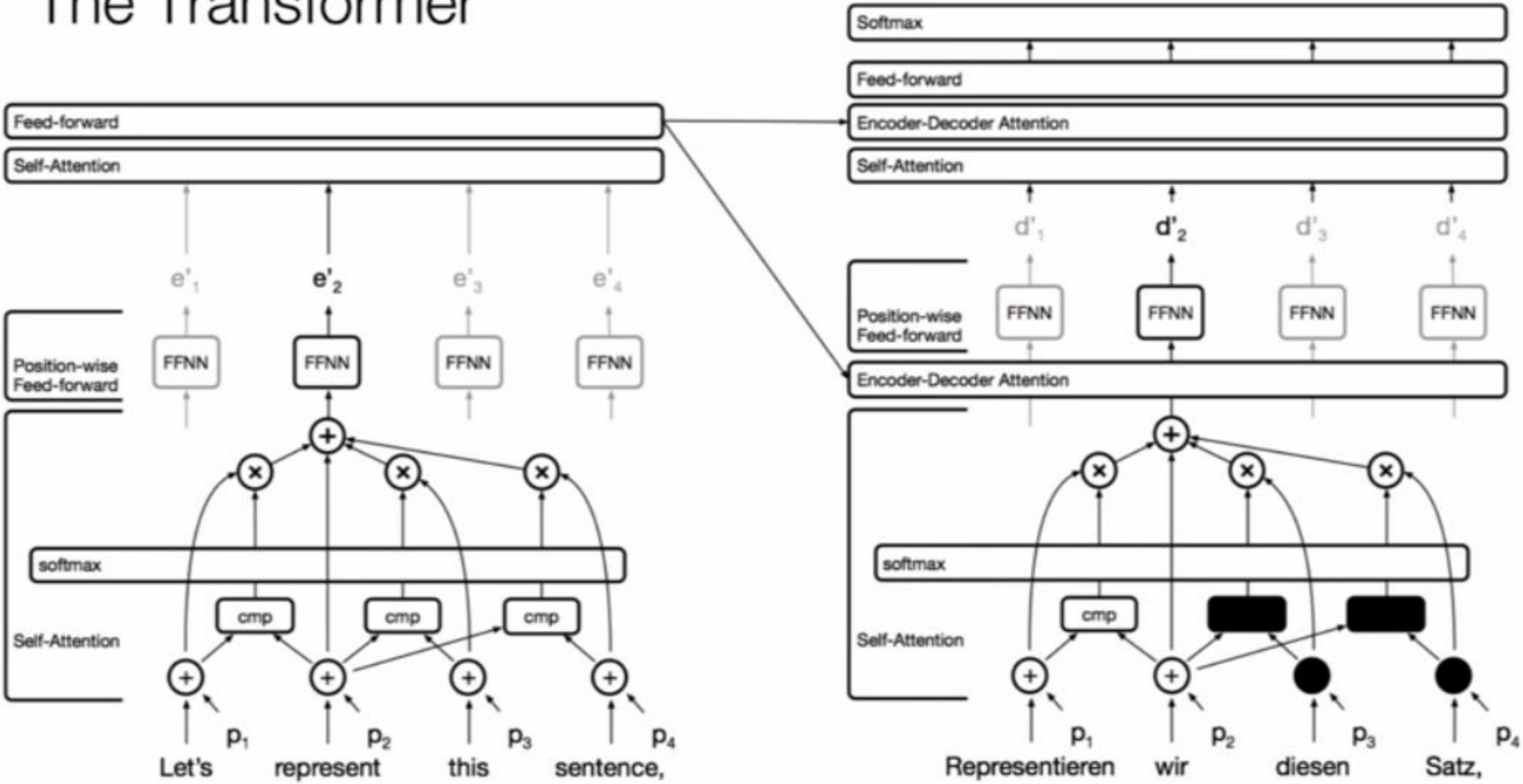
- Constant “Path length”
- Trivial to Parallelize
- Multiplicative Interaction

The Transformer

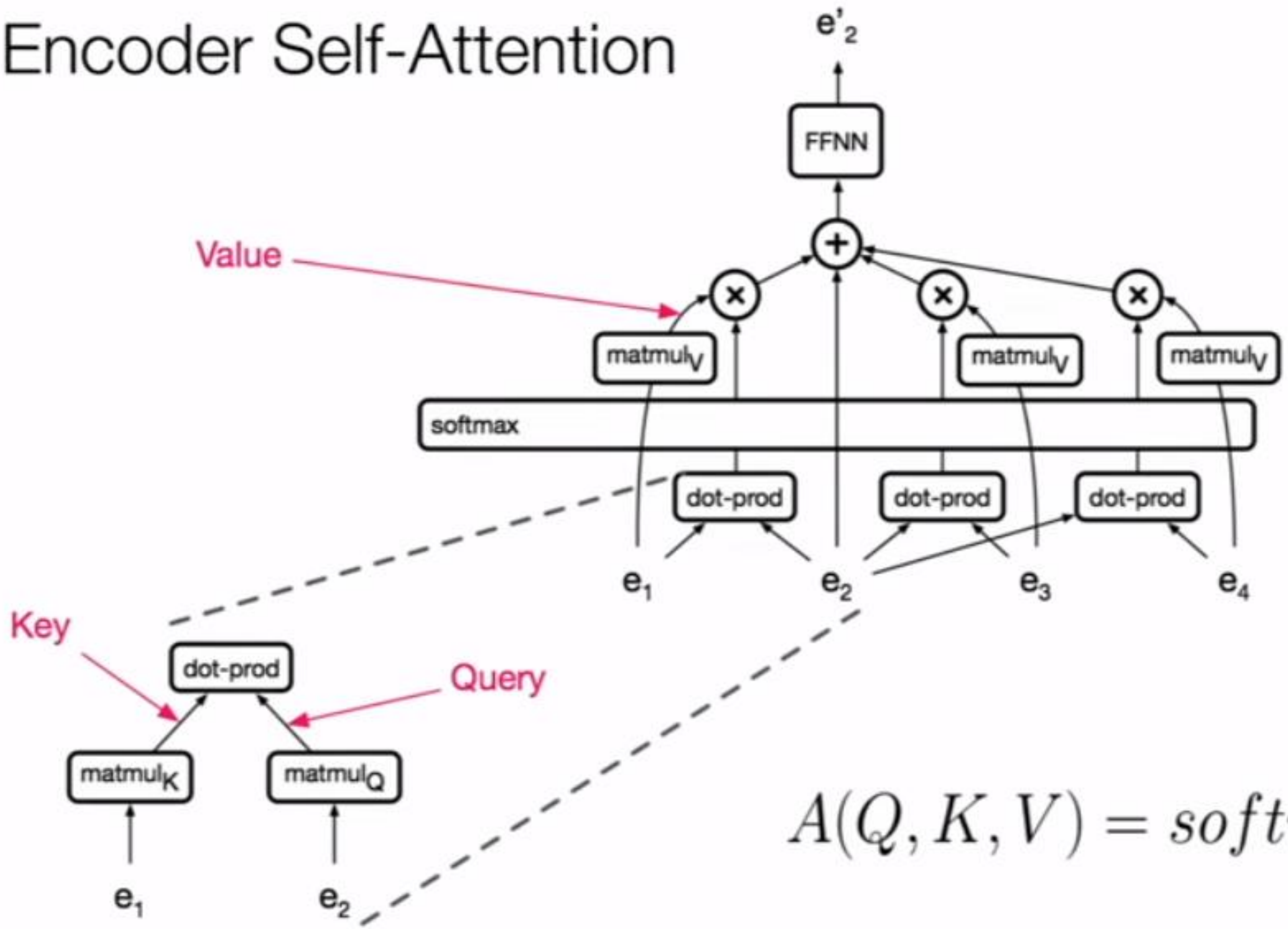
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



The Transformer



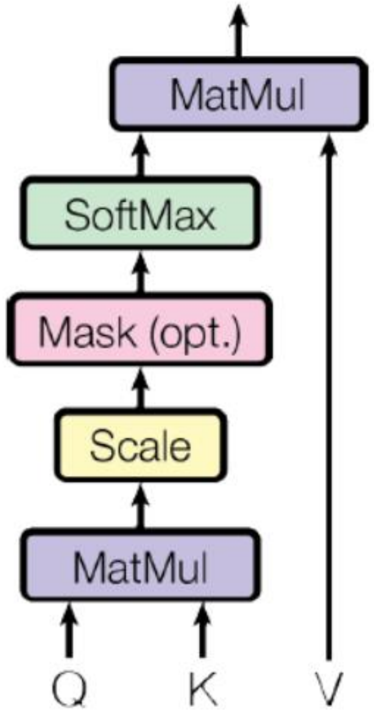
Encoder Self-Attention



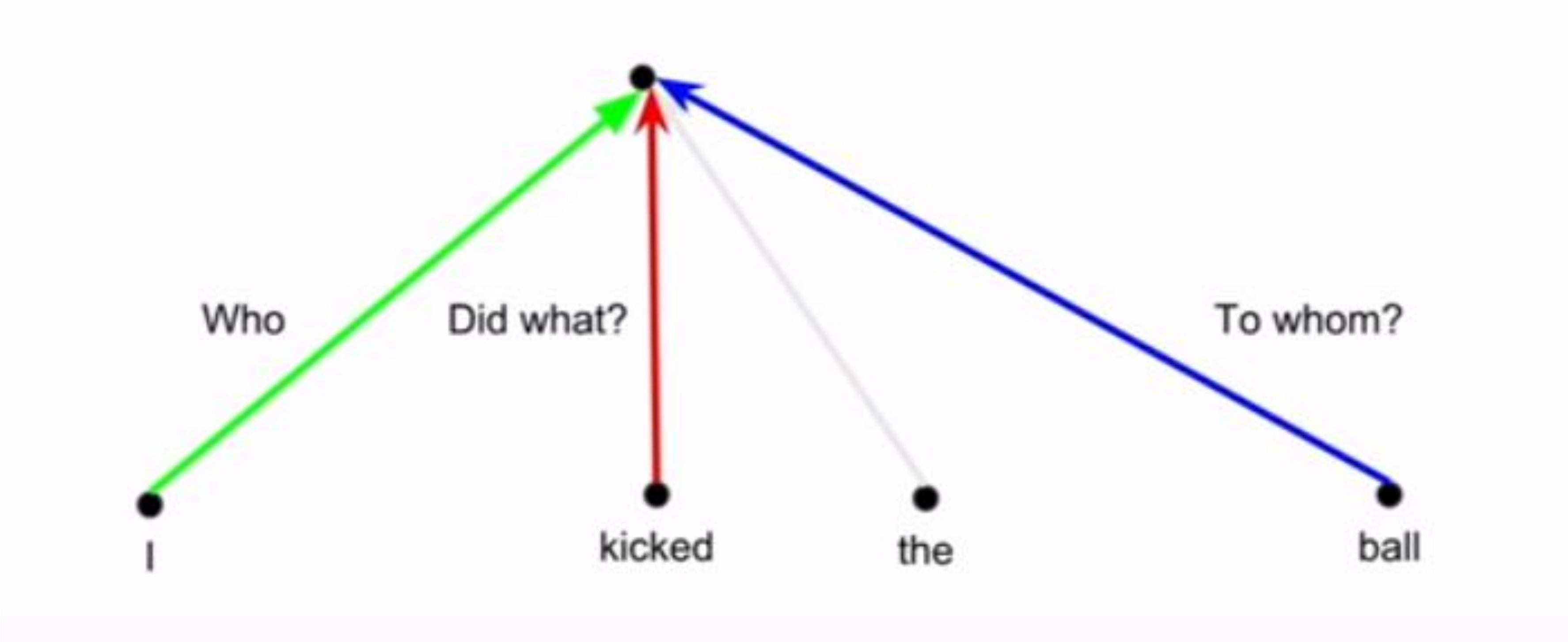
$$A(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Attention Head

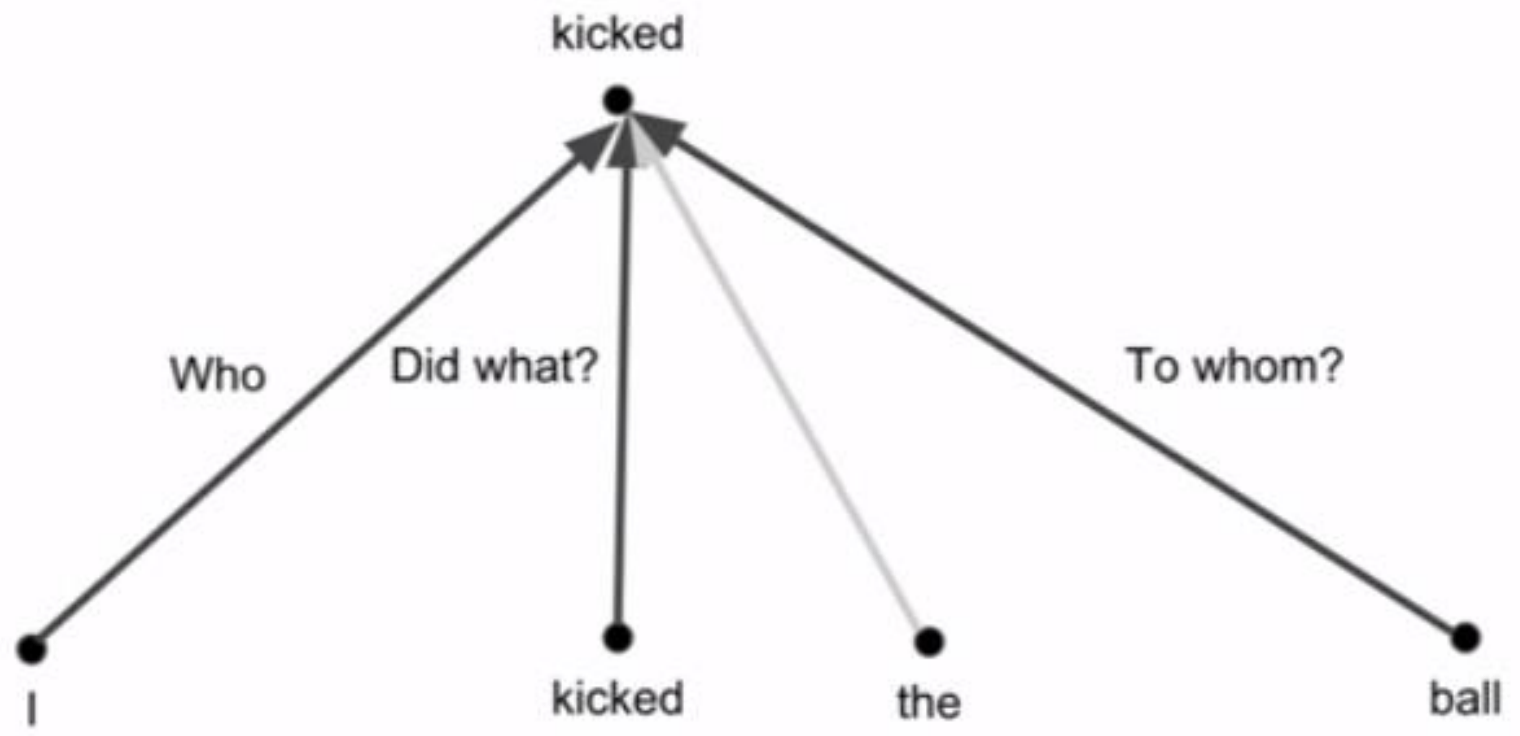
Scaled Dot-Product Attention



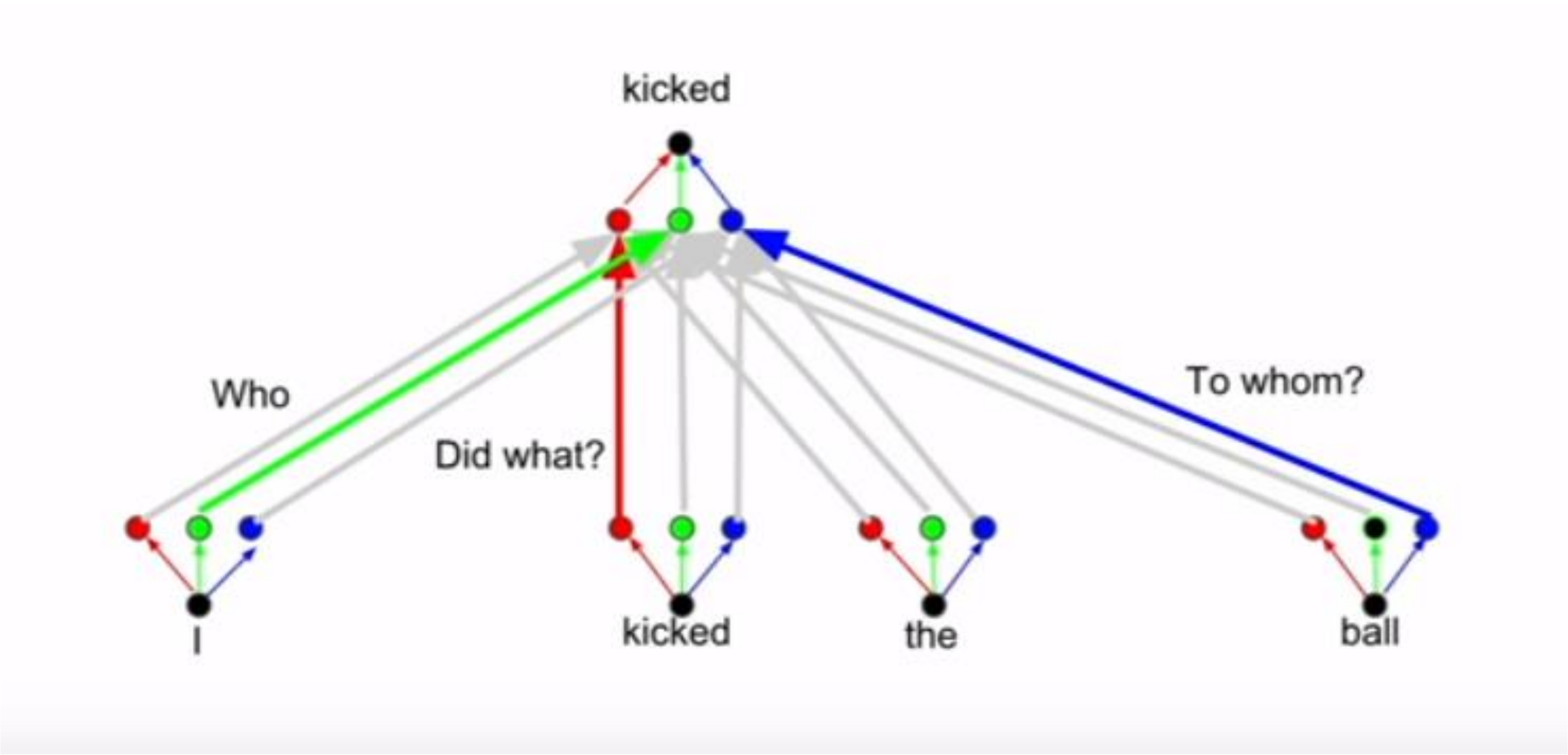
For Convolution



For Attention



Multi-head Attention



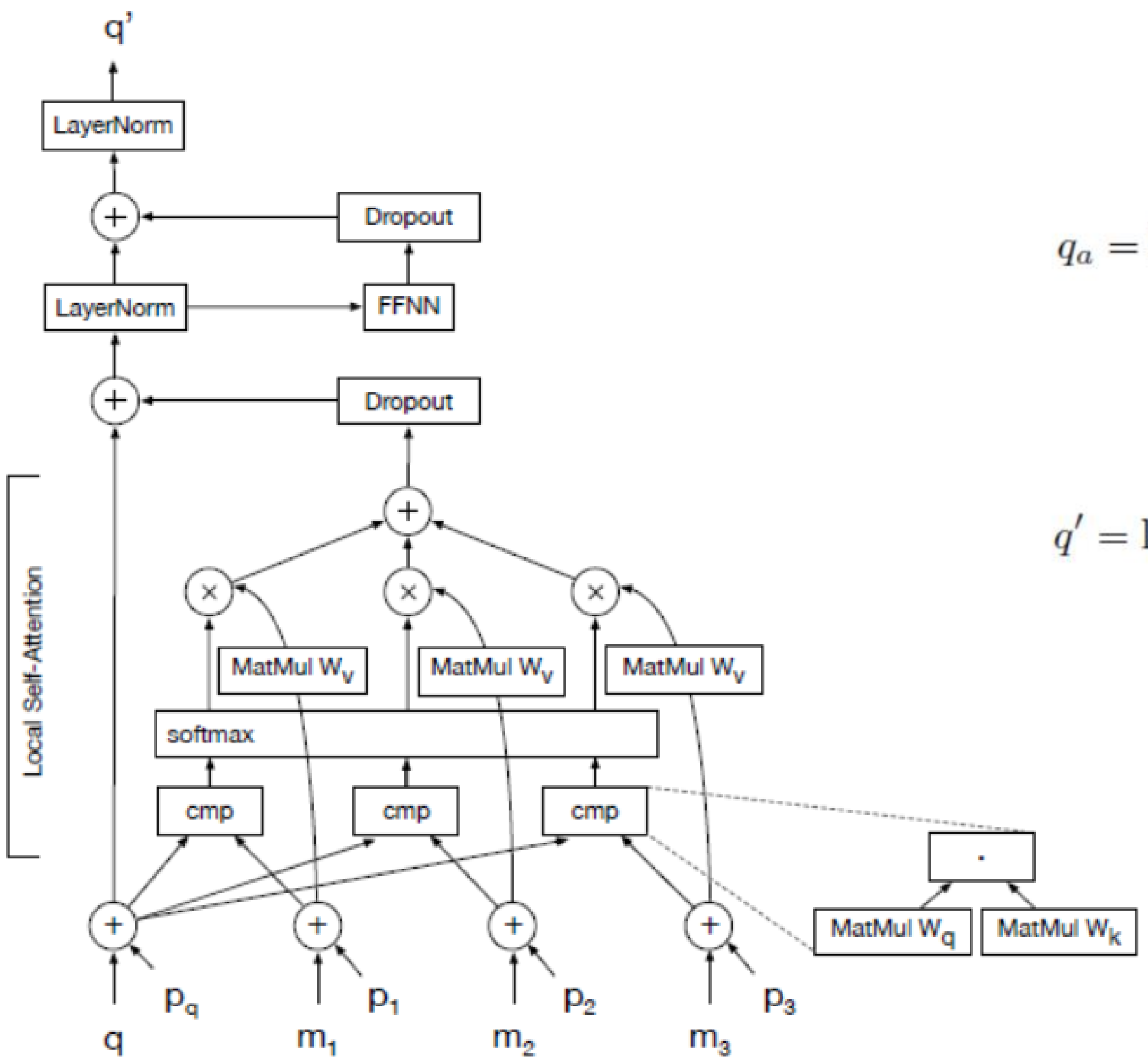
	FLOPs
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$

Image Transformer

Generate Image with Self-attention（Transformer）

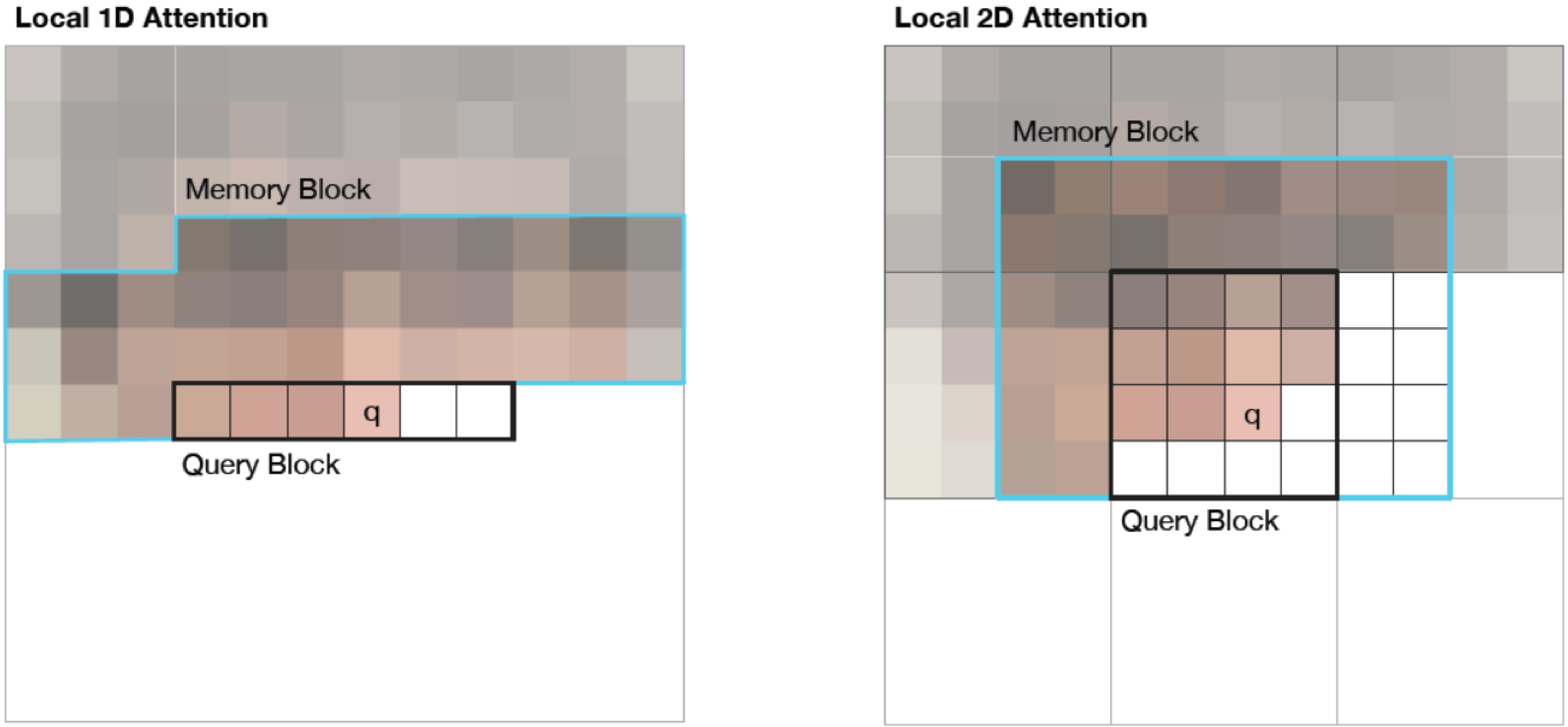
Google Brain

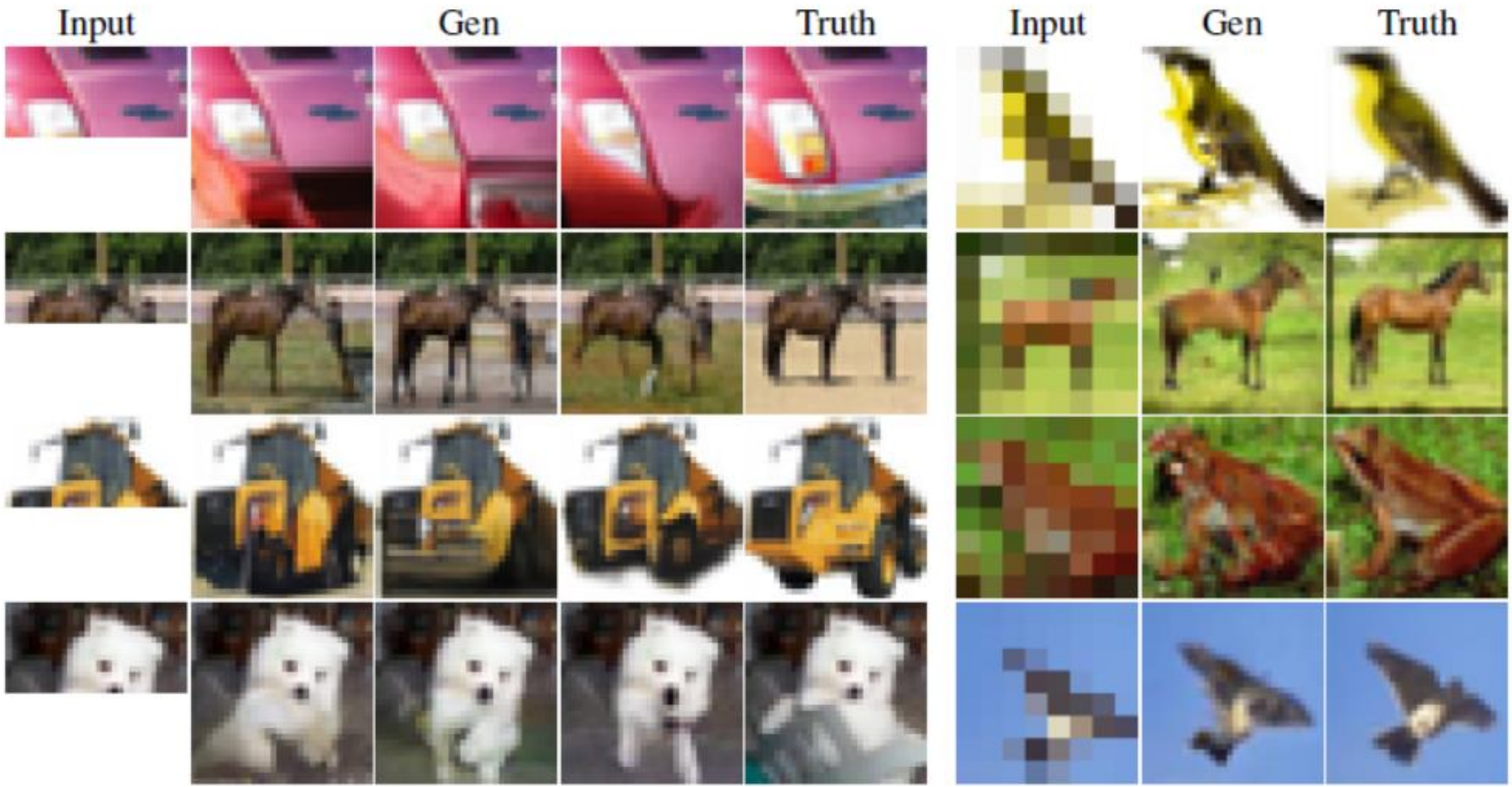
June 2018



$$q_a = \text{layernorm}(q + \text{dropout}(\text{softmax}\left(\frac{W_q q (M W_k)^T}{\sqrt{d}}\right) M W_v)) \quad (1)$$

$$q' = \text{layernorm}(q_a + \text{dropout}(W_1 \text{ReLu}(W_2 q_a))) \quad (2)$$


Local 1D Attention
Memory Block
q
Query Block
Local 2D Attention
Memory Block
q
Query Block


Input
Gen
Truth
Input
Gen
Truth

Non-local Neural Networks

Non-local Means X Self-Attention

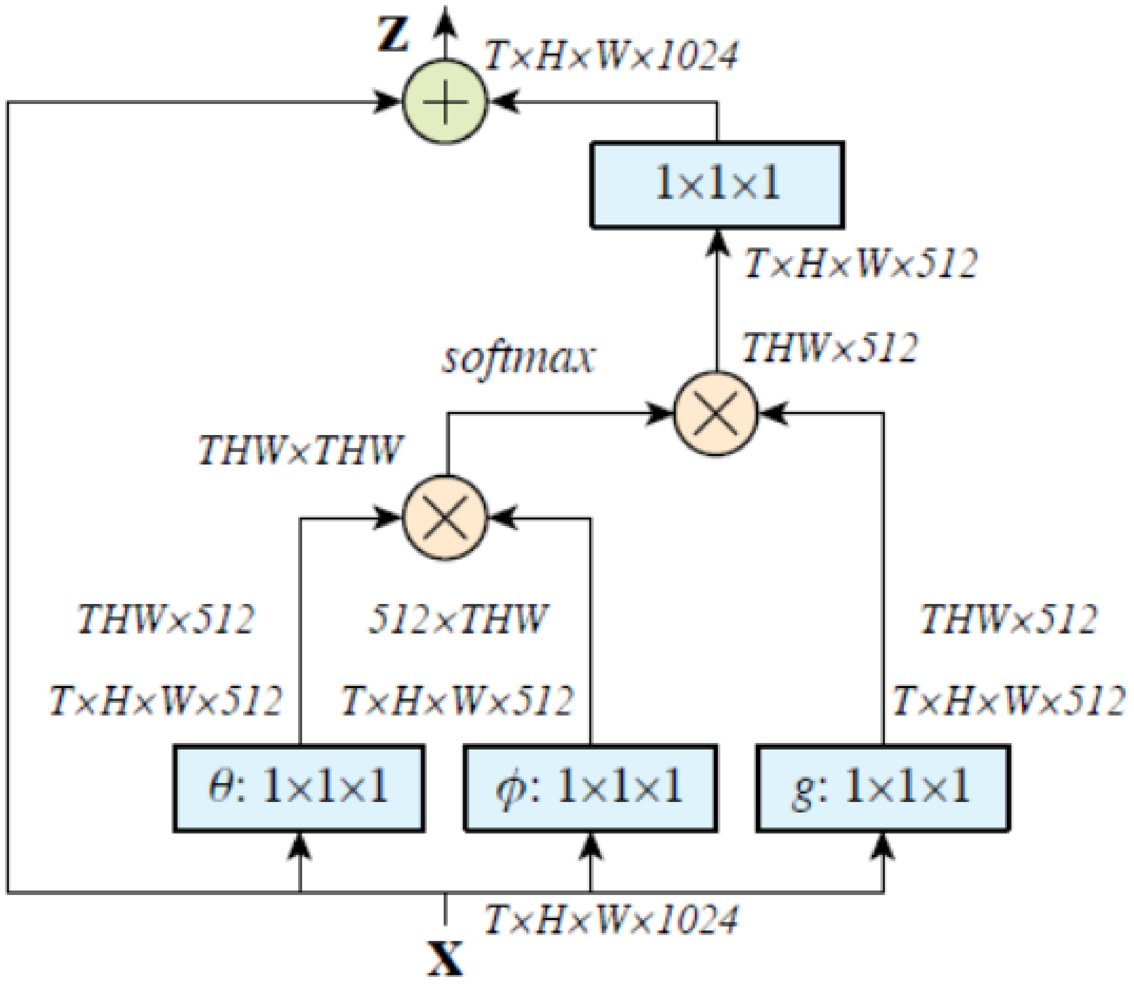
Kaiming He

CVPR 2018

Non-local Means

$$y_i = \frac{1}{C(x)} \sum_j \boxed{f(x_i, x_j)} g(x_j)$$

- Gaussian $f(x_i, x_j) = \exp(x_i^T \cdot x_j)$
- Embedded Gaussian $f(x_i, x_j) = \exp(\theta(x_i^T) \cdot \phi(x_j))$
- Dot Product $f(x_i, x_j) = \theta(x_i^T) \cdot \phi(x_j)$
- Concatenation $f(x_i, x_j) = ReLU(w_f^T [\theta(x_i) \cdot \phi(x_j)])$



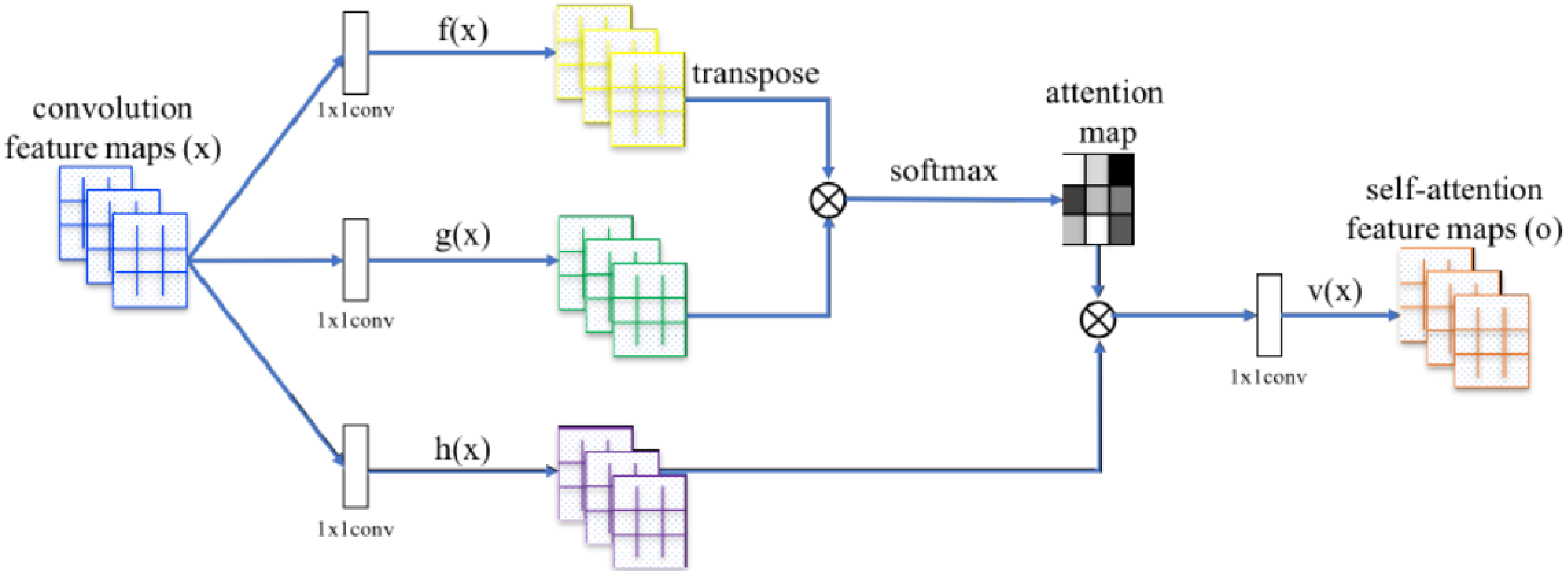
Z
T×H×W×1024
1×1×1
T×H×W×512
THW×512
softmax
THW×THW
THW×512
512×THW
THW×512
T×H×W×512
T×H×W×512
T×H×W×512
θ: 1×1×1
ϕ: 1×1×1
g: 1×1×1
T×H×W×1024
X

Self-Attention GAN

“FINALLY, we have dogs with four legs”

Ian Goodfellow

May 2018



convolution
feature maps (x)
1x1conv
1x1conv
1x1conv
f(x)
g(x)
h(x)
transpose
softmax
attention
map
1x1conv
v(x)
self-attention
feature maps (o)

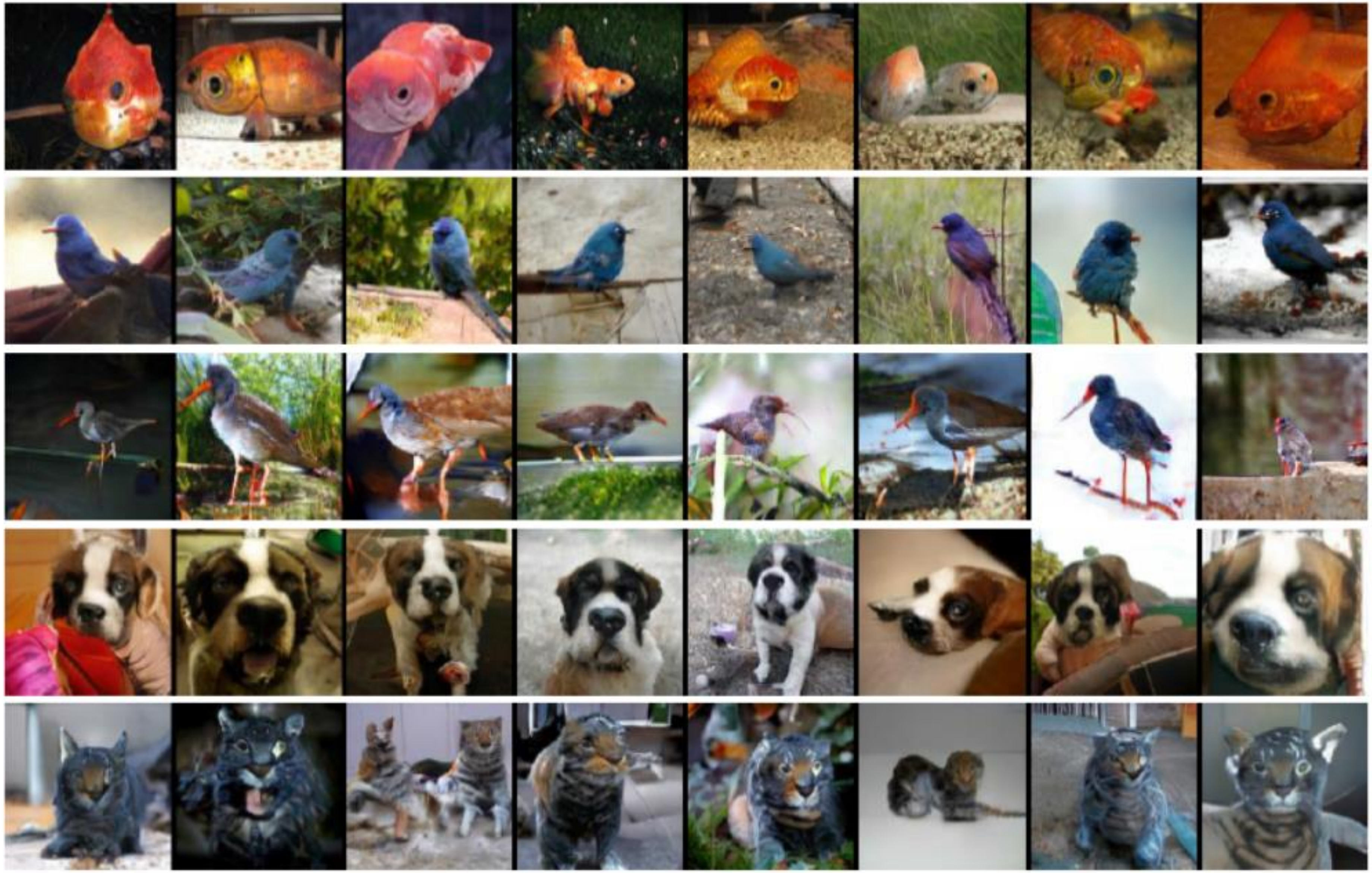
goldfish
(**44.4**, 58.1)

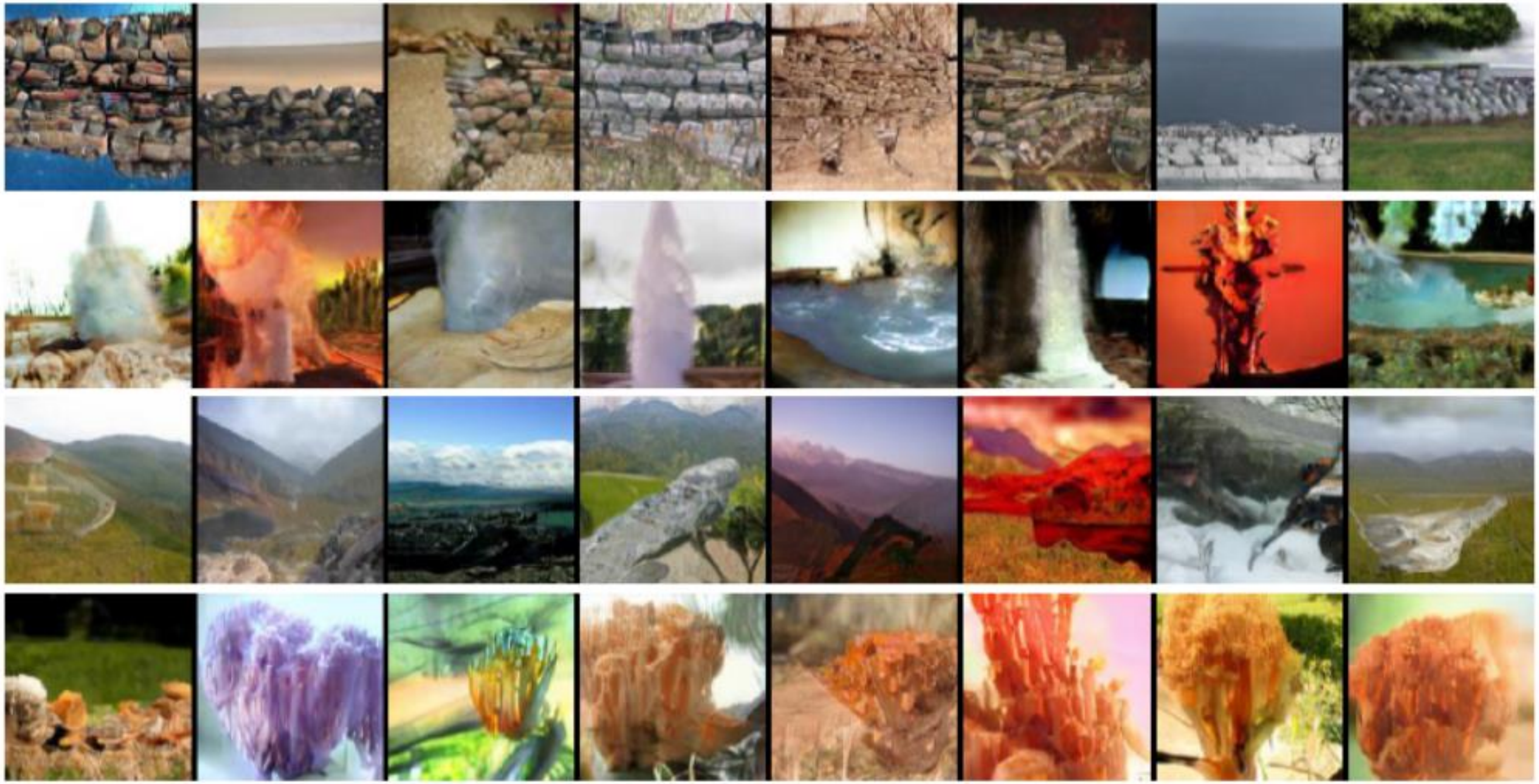
indigo
bunting
(**53.0**, 66.8)

redshank
(**48.9**, 60.1)

saint
bernard
(**35.7**, 55.3)

tiger
cat
(**88.1**, 90.2)





stone wall (57.5, **49.3**)

geyser (21.6, **19.5**)

valley (39.7, **26.0**)

coral fungus (38.0, **37.2**)

Model	Inception Score	Intra FID	FID
AC-GAN (Odena et al., 2017)	28.5	260.0	/
SNGAN-projection (Miyato & Koyama, 2018)	36.8	92.4	27.62*
SAGAN	**52.52**	**83.7**	**18.65**

Multi-Channel Attention Selection GAN with Cascaded Semantic Guidance for Cross-View Image Translation

Attention of the future

CVPR 2019（oral accepted）