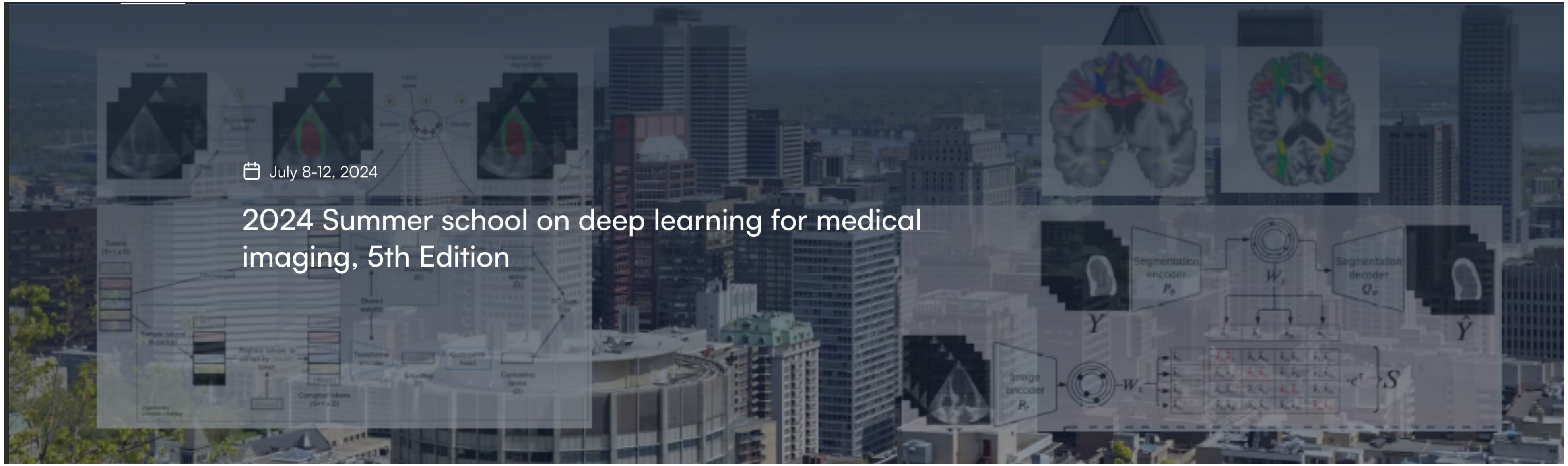
Advanced concepts in deep learning



THOME Nicolas – Prof. at SORBONNE University
ISIR Lab, MLIA TEAM

SORBONNE UNIVERSITÉ

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

MLIA Machine Learning & Deep Learning for Information Access

Sequential models

Up to now …

- Fully connected neural nets, parameters learning and training tricks
- Convolutional neural networks (ConvNets)
- Generative models

- **Sequential models:** dealing with “ordered sets”
- 2 main state-of-the-art deep architectures
  - Recurrent Neural Networks (RNNs)
  - Attention models & transfomers

Outline

I. Recurrent Neural Networks (RNNs)
   a) Vanilla RNNs
   b) RNN training
   c) RNN architectures
   d) Applications

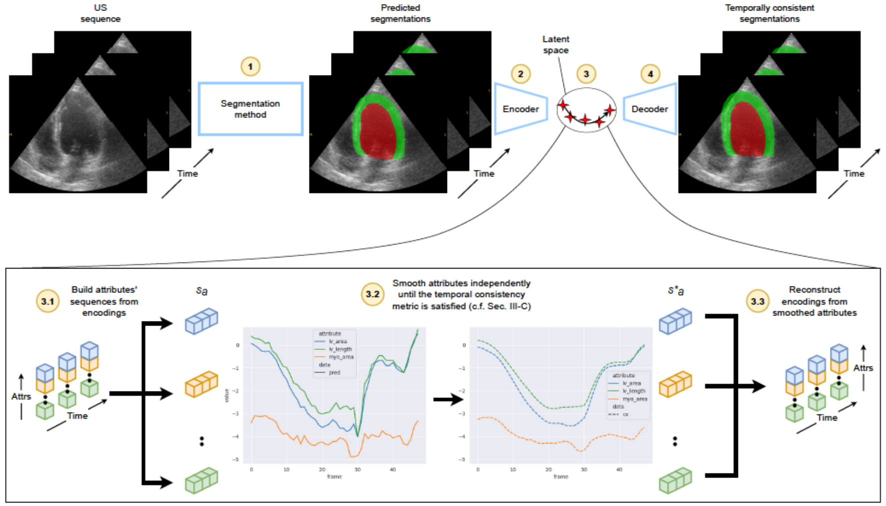
II. Attention models & transfomers

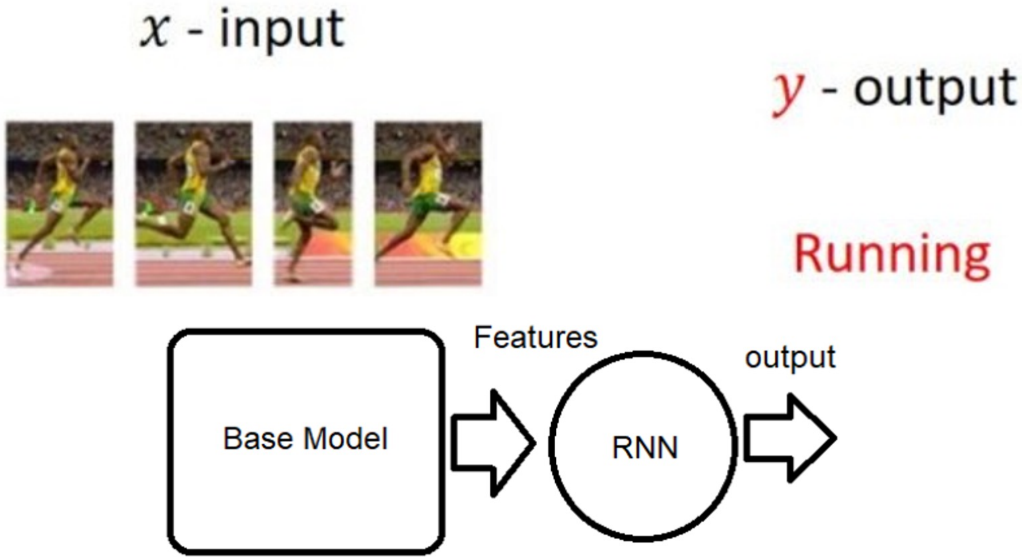
Recurrent Neural Networks (RNNs)

- Manipulating sequences of “ ordered tokens”, *i.e.,* atomic elements
  - Ex tokens: characters/word in NLP

"Hello I love you" ⇨ "Hello", "I", "love", "you"

  - Pixels/patches in images (how to define an order?)
  - Features a time t for time series, video: sequences of frames / feature embeddings with CNNs





Outline

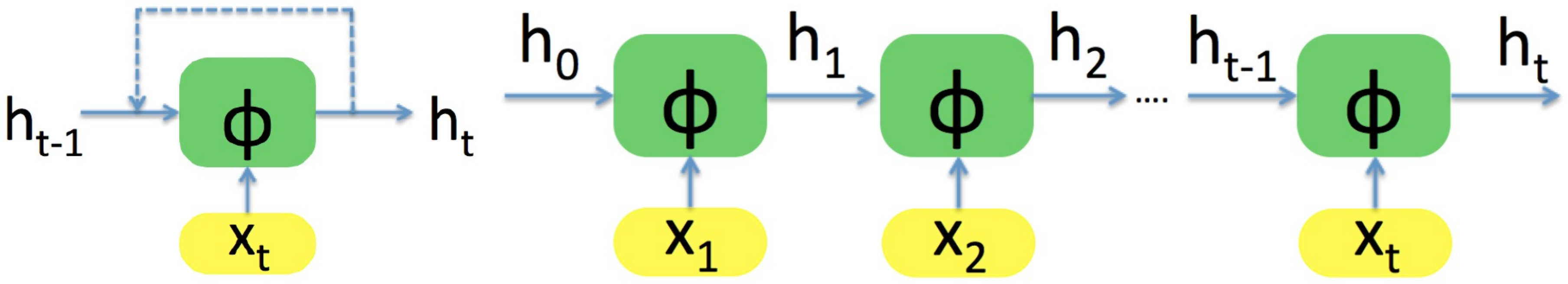
I. Recurrent Neural Networks (RNNs)
   a) Vanilla RNN model
   b) RNN training
   c) RNN architectures
   d) Applications

II. Attention models & transfomers

Recurrent Neural Networks (RNNs) [1]

- Input sequence $\{\mathrm{x}_t\}_{t\in\{1;T\}}$, $\mathrm{x}_t \in \mathbb{R}^d$
- Internal RNN state $\{\mathrm{h}_t\}_{t\in\{1;T\}}$, $\mathrm{h}_t \in \mathbb{R}^l$
- **RNN Cell:** $\mathrm{h}_t = \phi_t(\mathrm{x}_t, \mathrm{h}_{t-1})$
  - Loop, h_t depends on current x_t and previous state h_{t-1}
    - h_t: **memory of the network $\Leftrightarrow$ history up to time t**
  - In RNNs, function $\phi_t = \phi$ shared across time

Recurrent RNN view

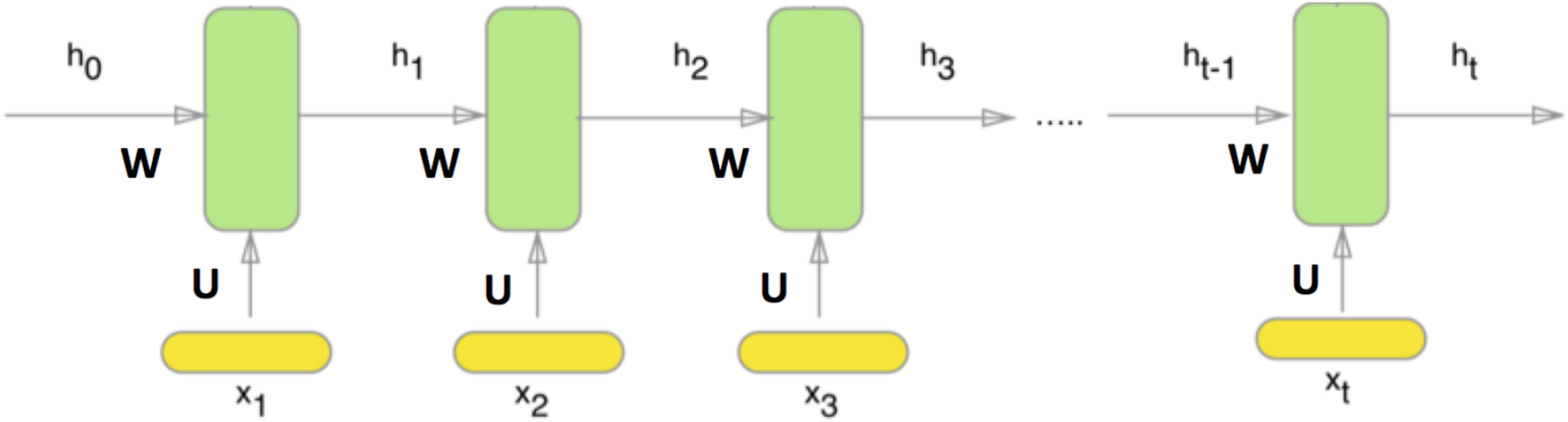


Unfolded RNN view

[1] Elman, J. L. (1990). Finding structure in time. COGNITIVE SCIENCE, 14(2):179–211.

Recurrent Neural Networks (RNNs)

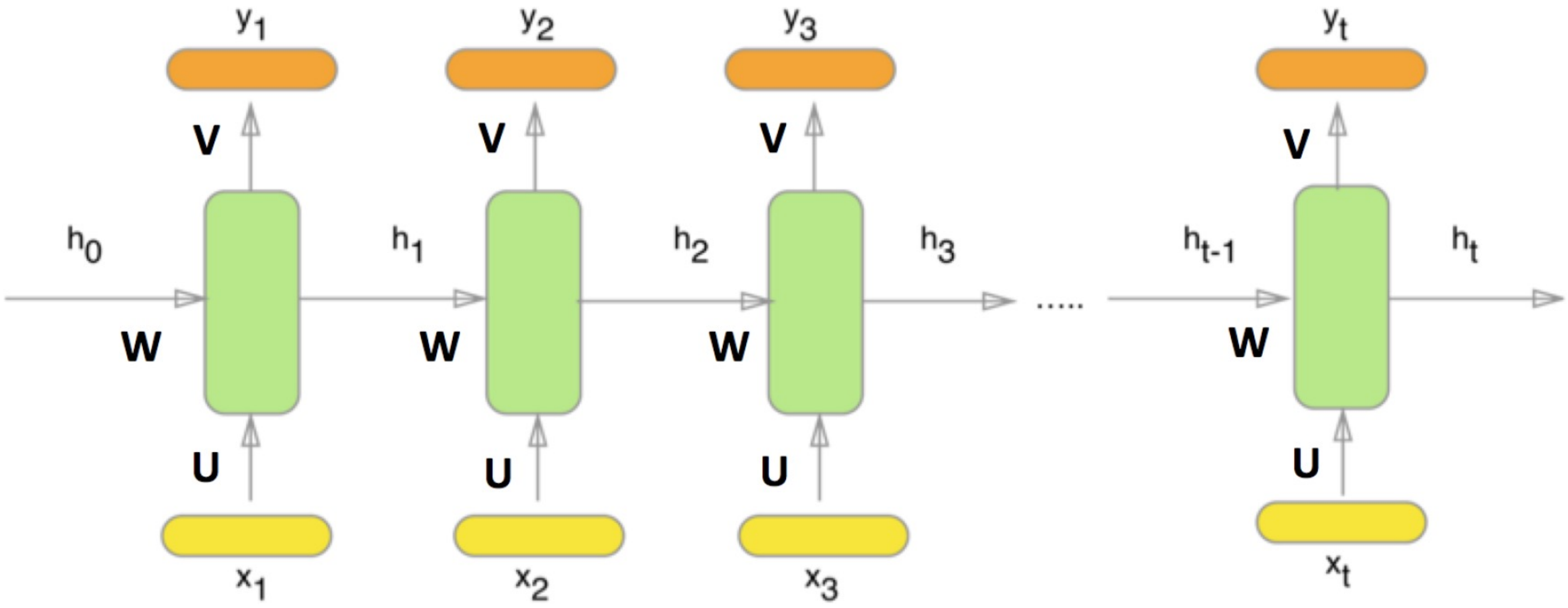
- **RNN Cell:** $h_t = \phi(x_t, h_{t-1})$
  - ϕ: linear projection of x_t and h_{t-1}, *i.e.* fully connected layers
  - $h_t = f(Ux_t + Wh_{t-1} + b_h)$
    - U matrix size $l \times d$, W matrix size $l \times l$ (b vector size l)
    - $f \leftarrow tanh$ non-linearity



Recurrent Neural Networks (RNNs)

- **<u>Depending on the task:</u>** prediction at each time step

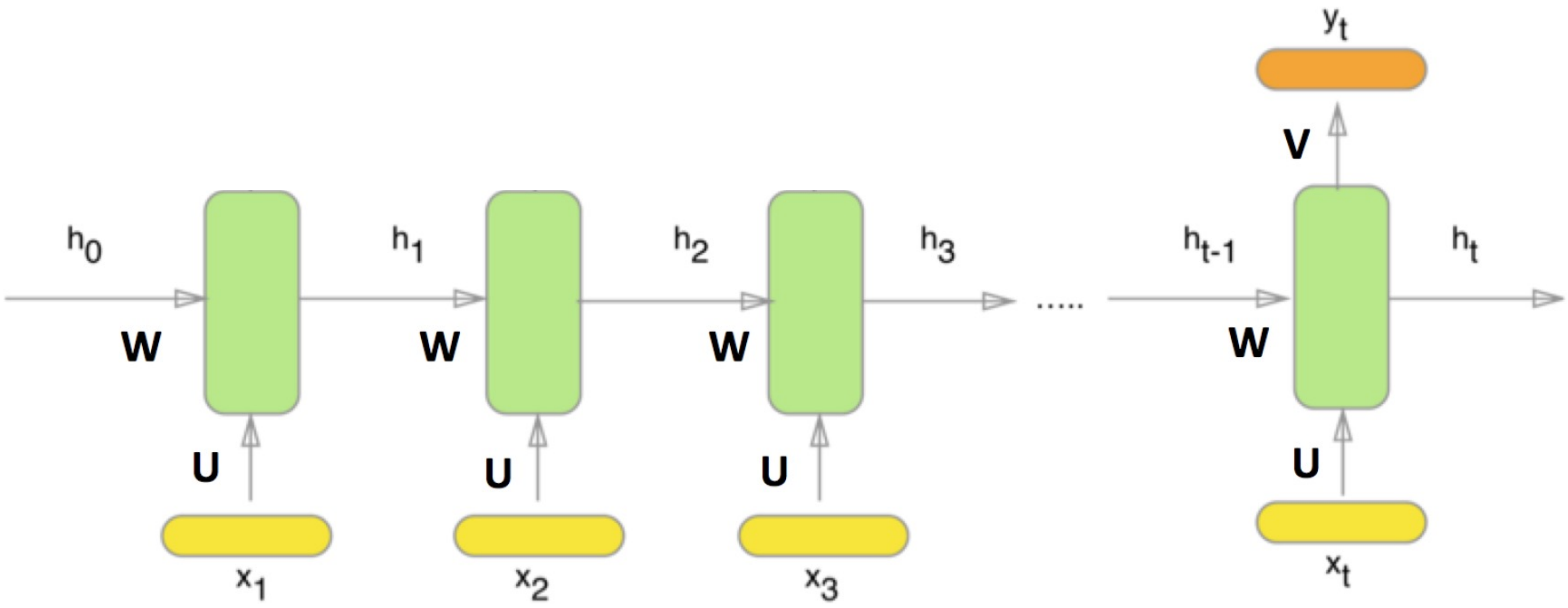
$$y_t = f'(Vh_t + b_y) \qquad \text{e.g., } f' \leftarrow \text{soft-max if } y_t \leftrightarrow \text{class probabilities}$$



Recurrent Neural Networks (RNNs)

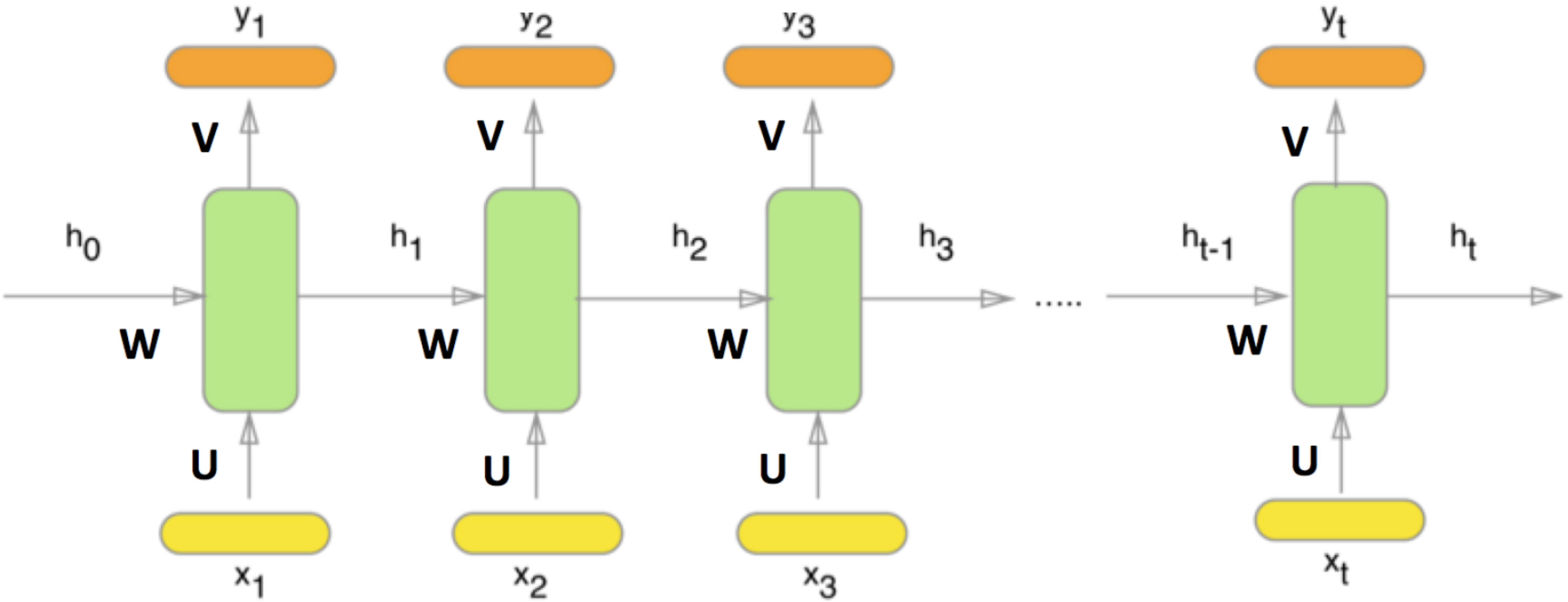
- **Depending on the task:** prediction at the last time step

$$y_t = f'(Vh_t + b_y) \qquad \text{e.g., } f' \leftarrow \text{soft-max if } y_t \leftrightarrow \text{class probabilities}$$



RNNs expressiveness

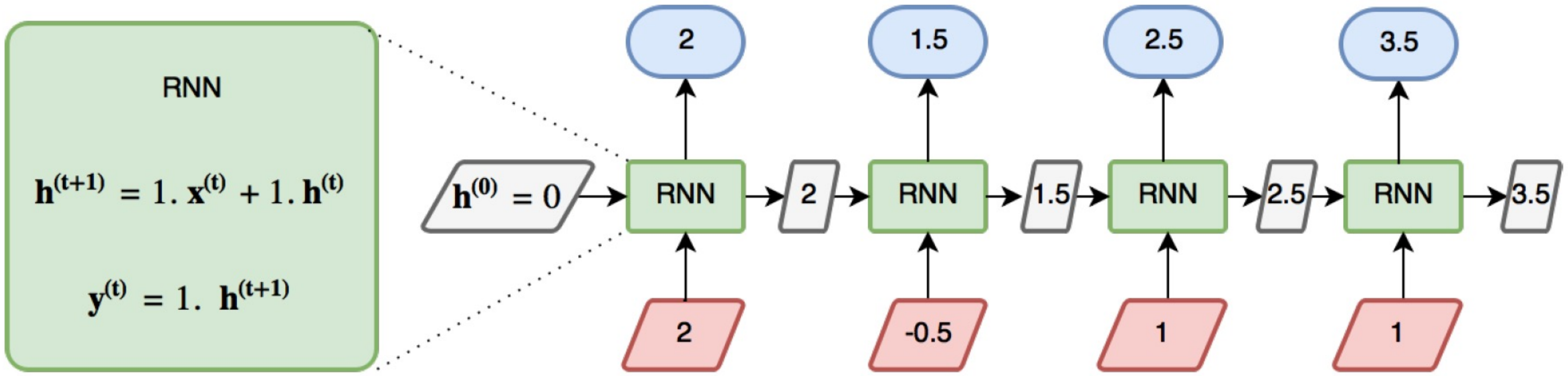
- **Recap:** Feed-forward neural networks are universal function approximators
- **Expressibility of the mapping between $\{x_t\}_{t\in\{1;T\}}$ and $\{y_t\}_{t\in\{1;T\}}$?**
  - RNNs are universal program approximators [2]
    - Can approximate any any computable function, *i.e.* Turing machine
  - RNNs can approximate any measurable sequence to sequence mapping



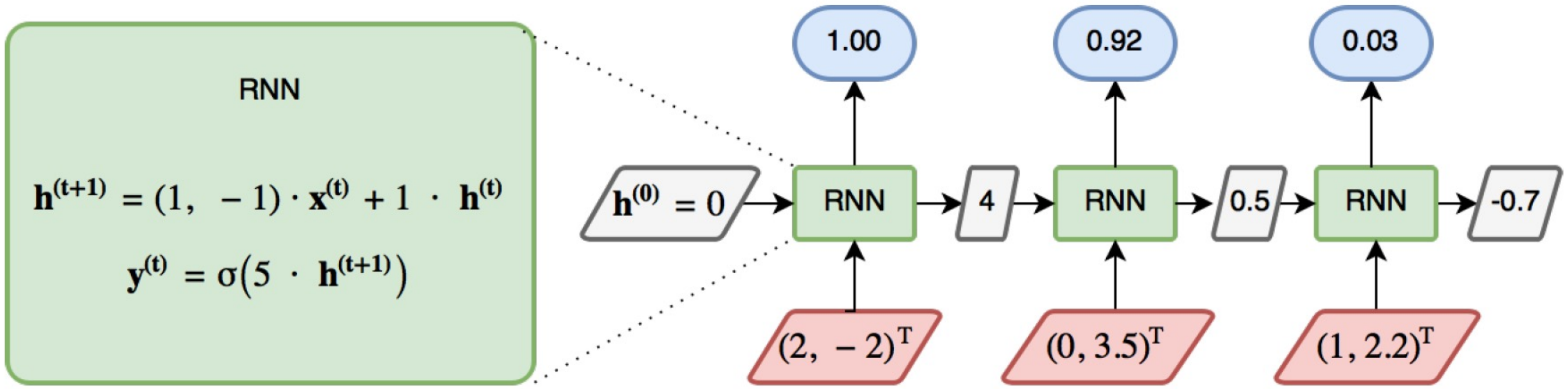
[2] H. Siegelmann, and E. Sontag. On the computational power of neural nets. J. Comput. Syst. Sci., 1995.

RNNs expressiveness: examples

- Computing sum



- Determining if sum of 1st dimension values > than sum of 2nd dimension



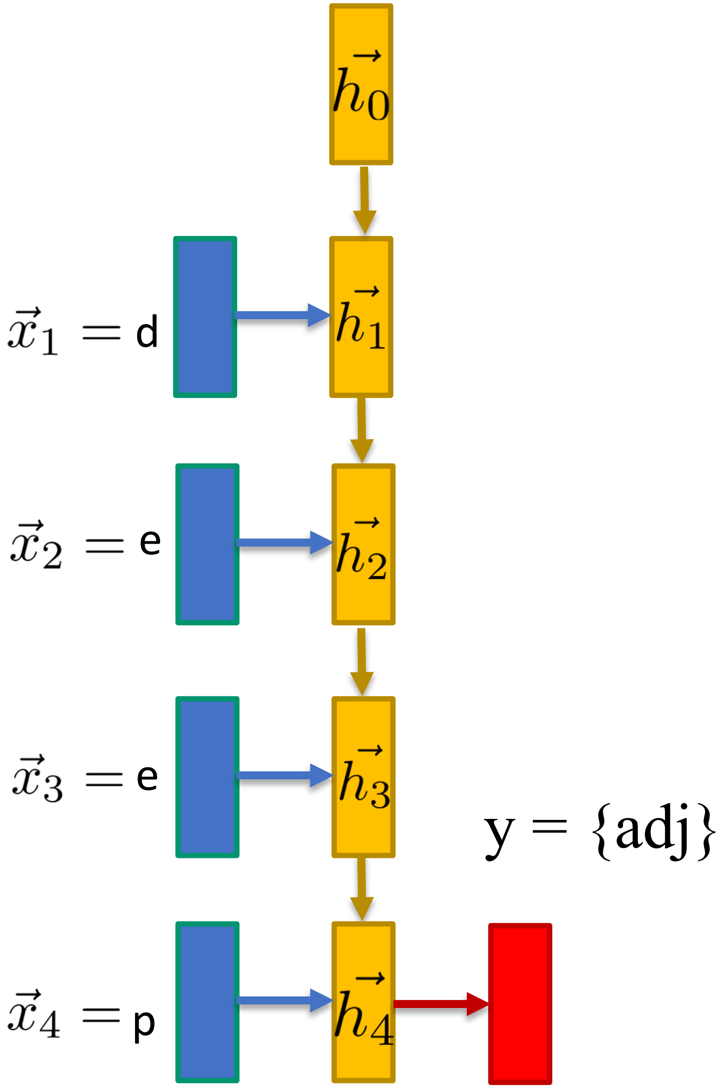
Toy ex: grammatical analysis, 'deep' => adj

- Input x_t: character, one-hot encoding

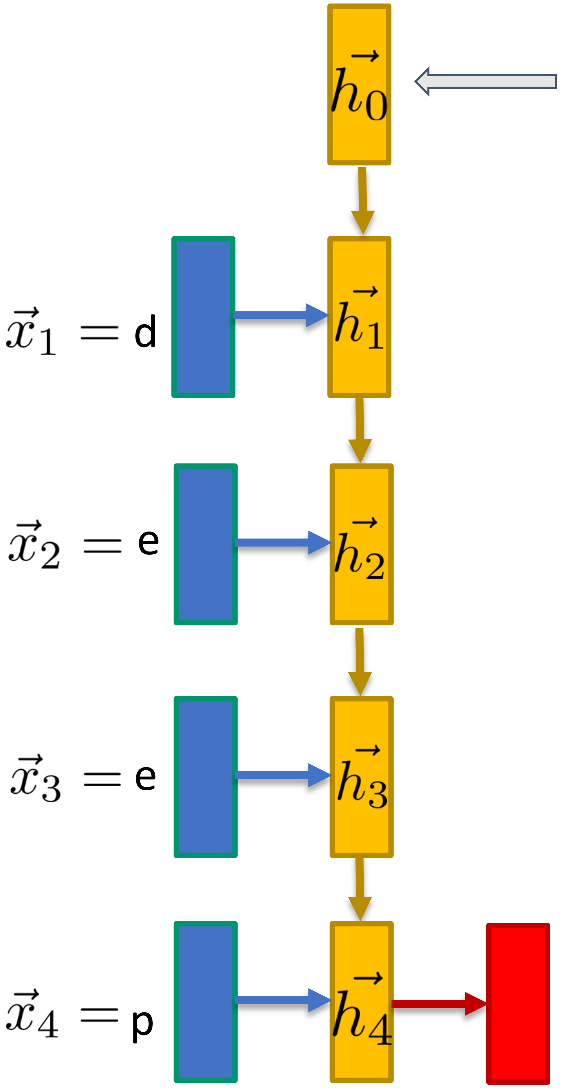
$$\left.\begin{aligned} 'a' &= [1,0,0,\ldots,0] \\ 'b' &= [0,1,0,\ldots,0] \\ 'c' &= [0,0,1,\ldots,0] \end{aligned}\right\} \in R^{V}$$

- Output y_t: grammatical class

$$\left.\begin{aligned} \text{'verb'} &= [1,0,0,\ldots,0] \\ \text{'noun'} &= [0,1,0,\ldots,0] \\ \text{'adj'} &= [0,0,1,\ldots,0] \end{aligned}\right\} \in R^{M}$$



Ex: grammatical analysis, ‘deep’ => adj



“0” vector or learned

Shared weights

$$\vec{h}_i = f_a(W^{xh}\vec{x}_i + W^{hh}\vec{h}_{i-1})$$

$$y = \text{softmax}(W^{hy}\vec{h}_4)$$

$y = \{\text{adj}\}$ → Loss

e.g., cross-entropy

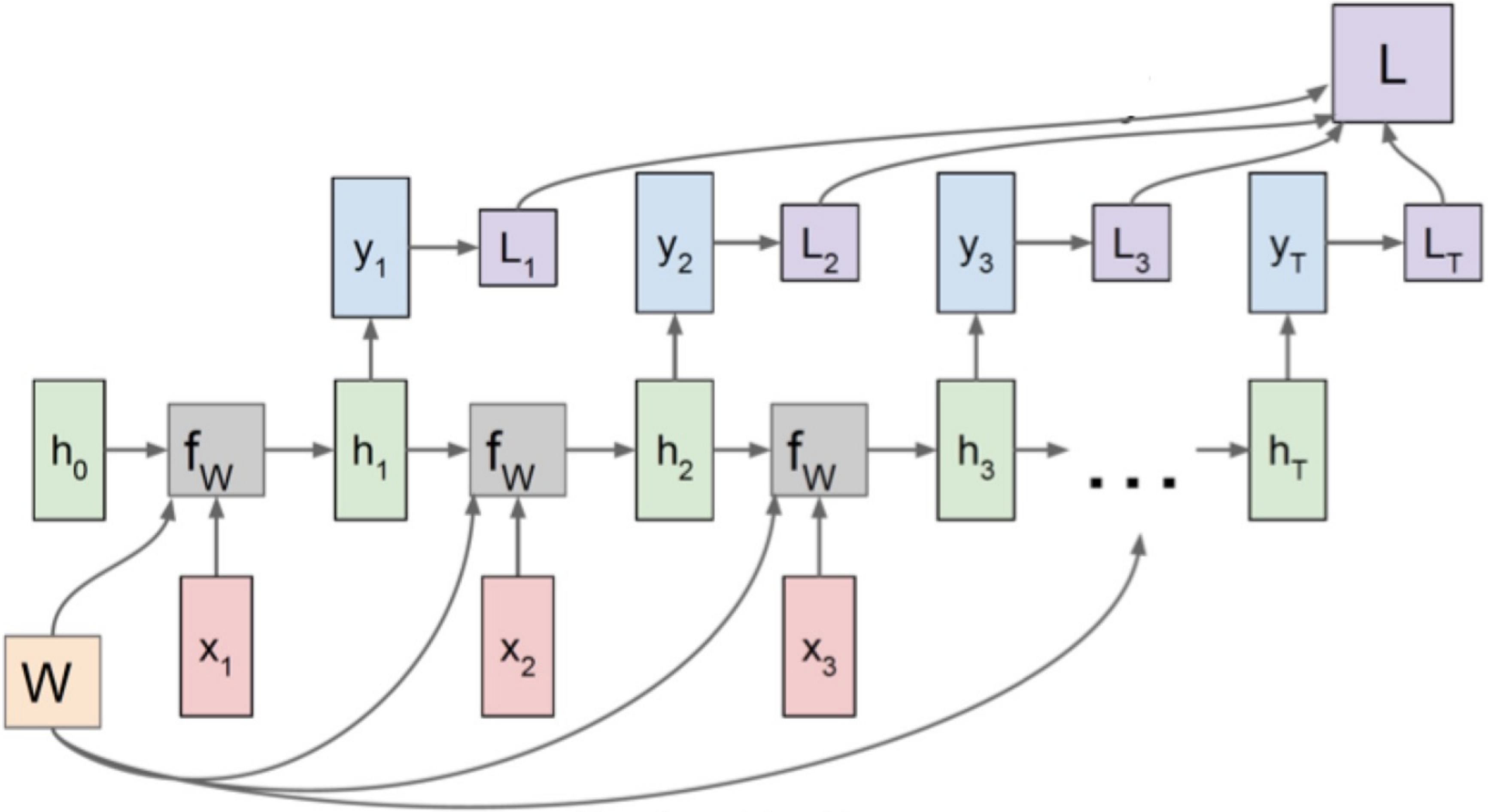
Outline

I. Recurrent Neural Networks (RNNs)
   a) Vanilla RNNs
   b) RNN training
   c) RNN architectures
   d) Applications

II. Attention models & transfomers

RNN training: formulation

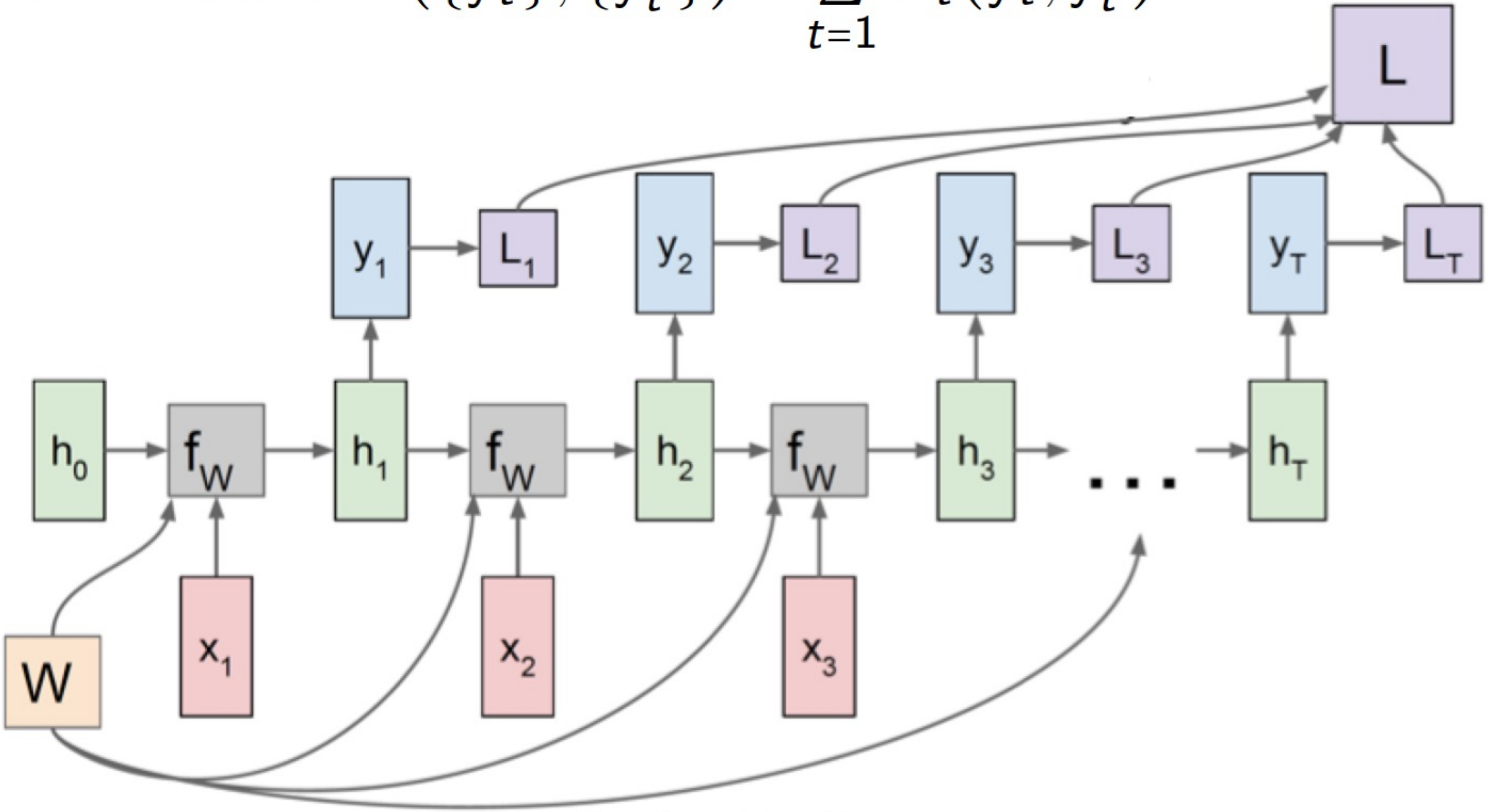
- Comparing output prediction $\{y_t\}_{t\in\{1;T\}}$ with supervision $\{y_t^*\}$
  - Task-dependent, *e.g.* only $\{y_T^*\}$ in many-to-one



Credit: Fei-Fei

RNN training: formulation

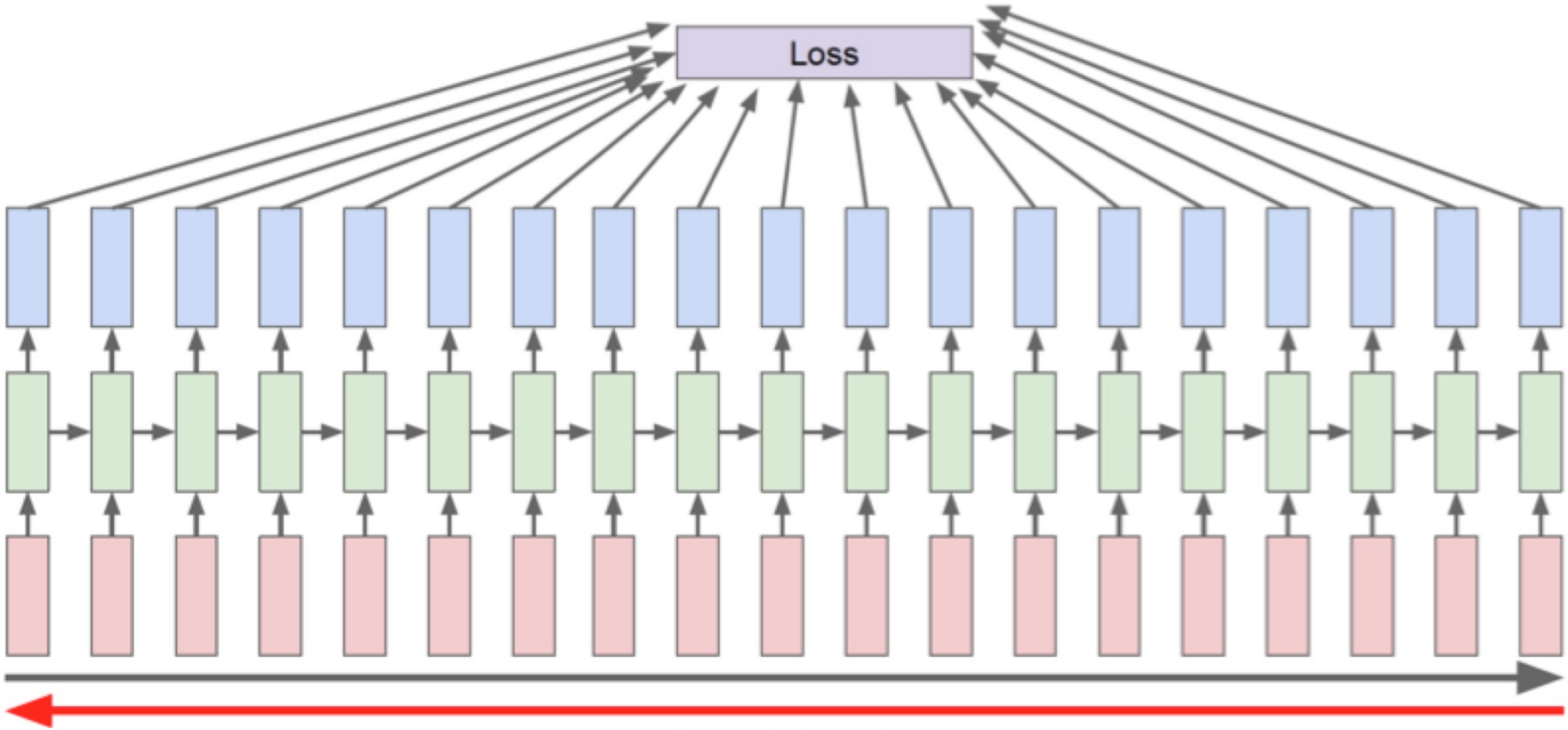
- Loss function at time t: $\mathcal{L}_t(\mathrm{y}_t, \mathrm{y}_t^*)$, *e.g.* cross-entropy (classification)
- Total loss function $\mathcal{L}(\{\mathrm{y}_t\}, \{\mathrm{y}_t^*\}) = \sum_{t=1}^{T} \mathcal{L}_t(\mathrm{y}_t, \mathrm{y}_t^*)$



Credit: Fei-Fei

Back-Propagation Through Time (BPTT)

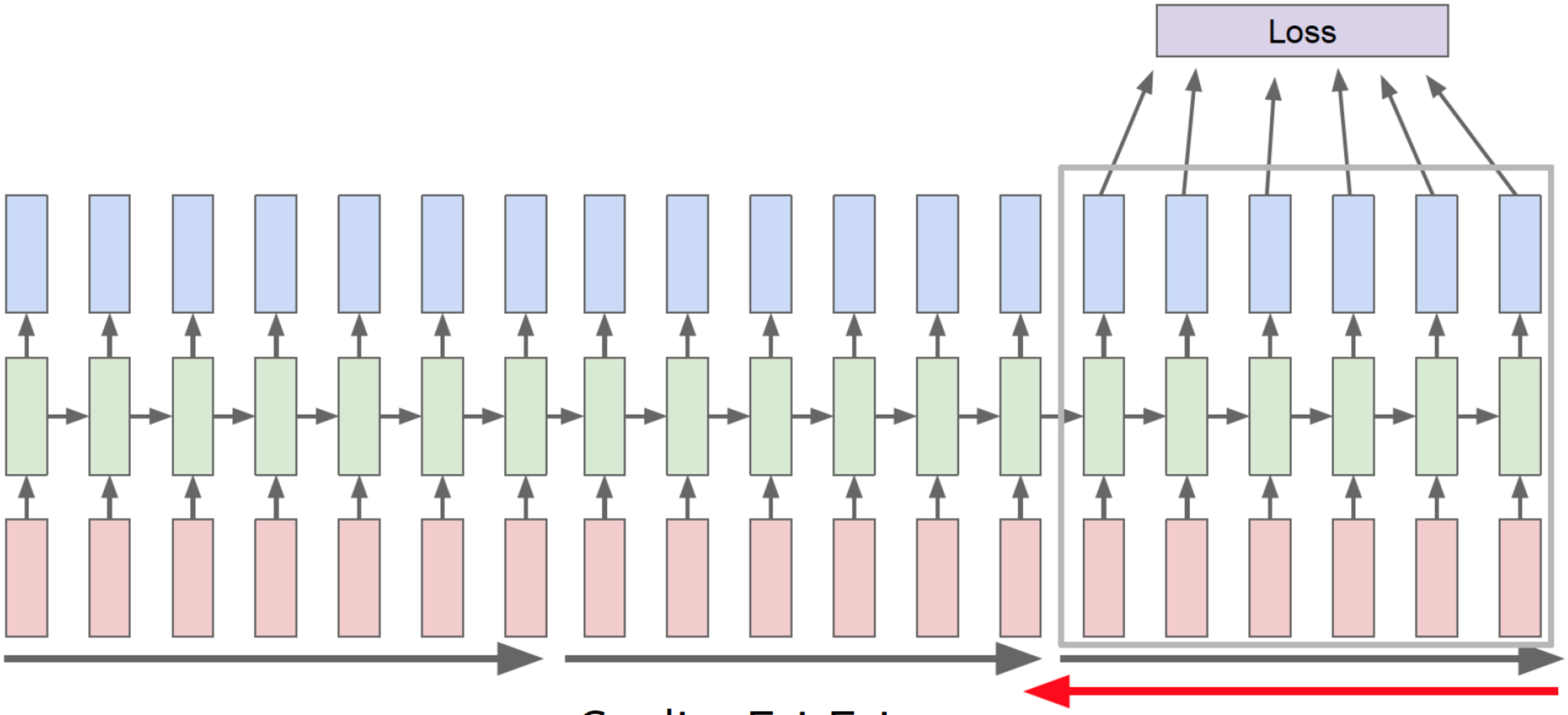
- Similar to standard back-prop, with time (sequence) ~ depth



Credit: Fei-Fei

Truncated BPTT

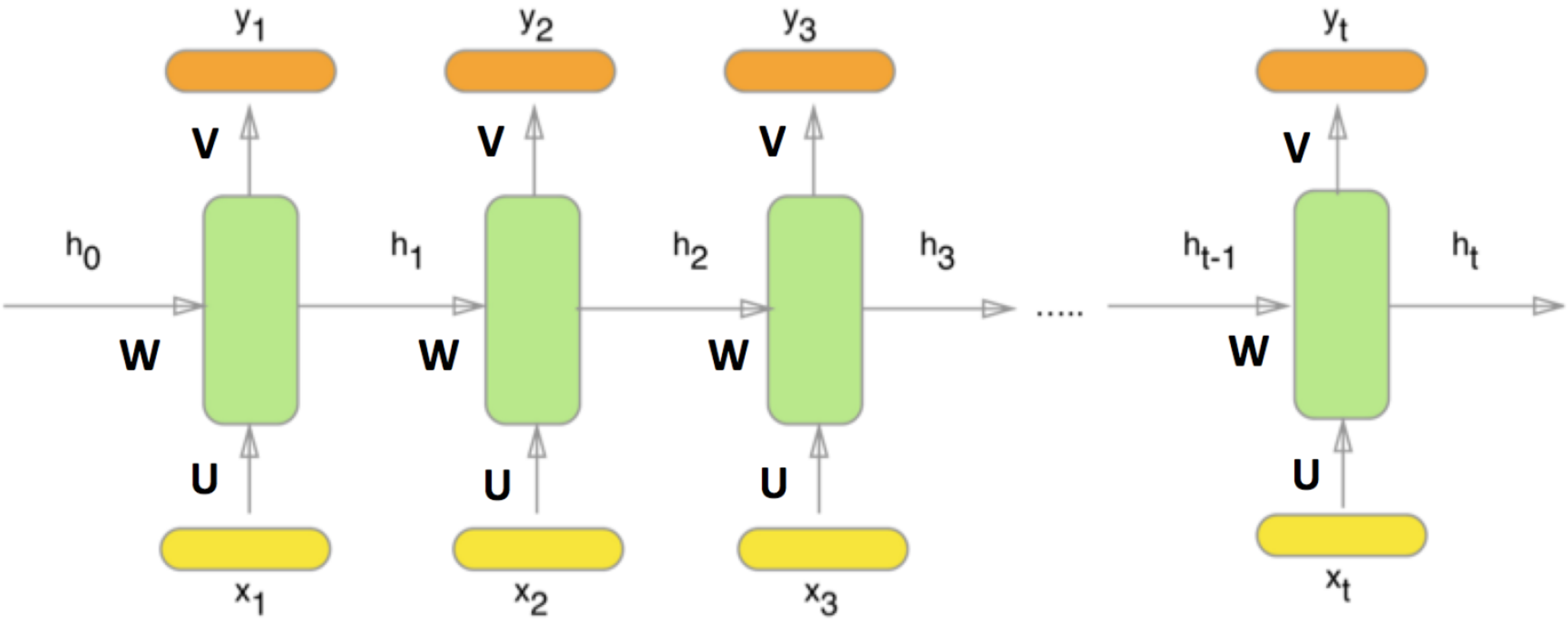
- Issue for large T => storage, + vanishing gradients issues (next)
- Truncated BPTT => BPTT on local temporal windows



Credit: Fei-Fei

BPTT: Gradient Computation

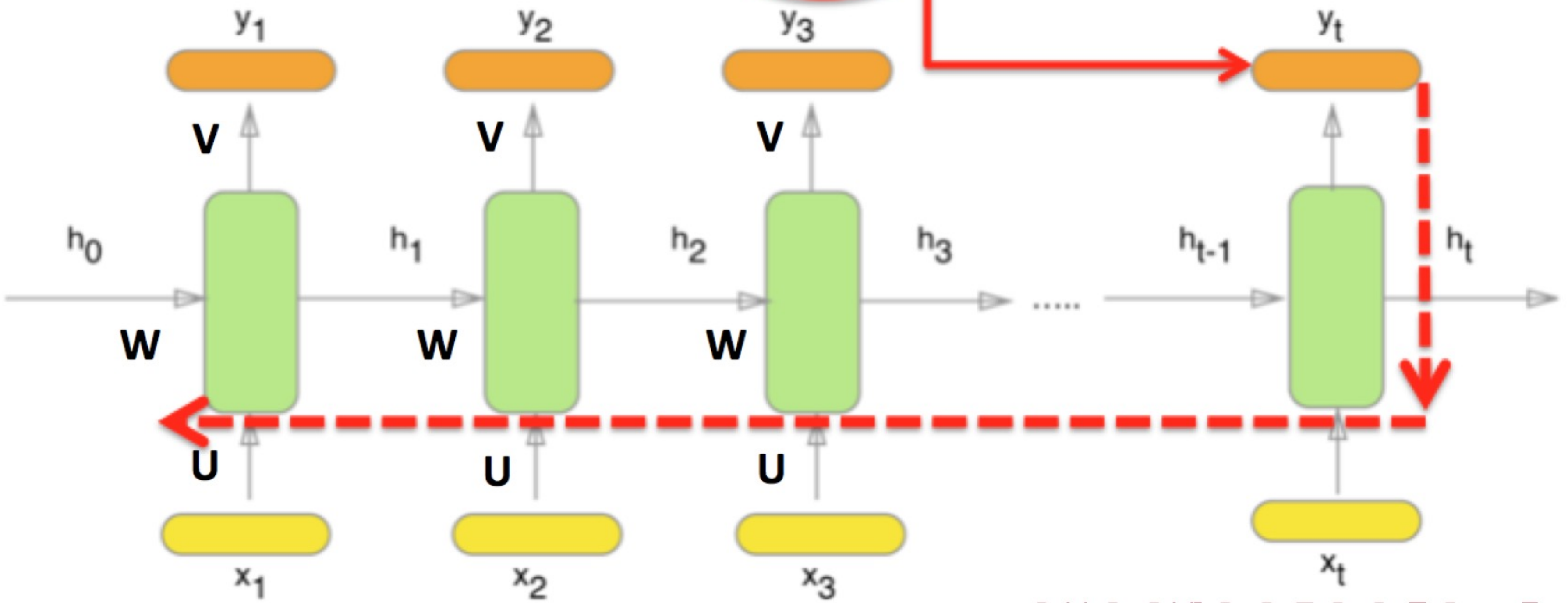
- **BPTT:** computing gradient $\frac{\partial \mathcal{L}_t}{\partial \mathrm{W}}$, $\frac{\partial \mathcal{L}_t}{\partial \mathrm{U}}$, $\frac{\partial \mathcal{L}_t}{\partial \mathrm{V}}$ (+biases)
- **Unfolded RNN:** same spirit as back-prop with fully connected networks (chain rule)
  - **BUT:** shared parameters W, U, V across time



BPTT: Gradient Computation

- **Shared parameters** W, U, V **across time**
  $\Rightarrow$ **gradients depend on the whole past history**
- Ex: for W: $\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$

$$\frac{\partial \mathcal{L}_t(\mathbf{y}_t, \mathbf{y}_t^*)}{\partial \mathbf{W}} = \sum_{k=1}^{t} \frac{\partial \mathcal{L}_t}{\partial \mathbf{y}_t} \frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$



BPTT: Gradient Computation

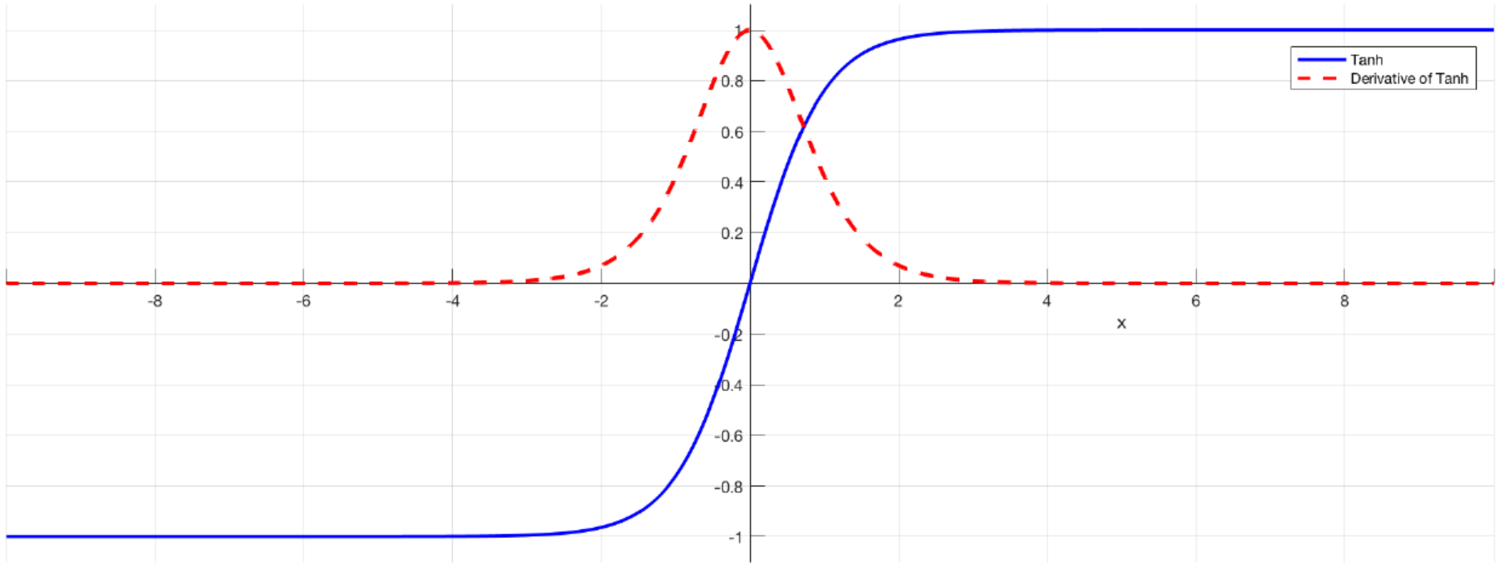
- $\frac{\partial \mathcal{L}_t}{\partial \mathrm{W}} = \sum_{k=1}^{t} \frac{\partial \mathcal{L}_t}{\partial \mathrm{y}_t} \frac{\partial \mathrm{y}_t}{\partial \mathrm{h}_t} \boxed{\frac{\partial \mathrm{h}_t}{\partial \mathrm{h}_k}} \frac{\partial \mathrm{h}_k}{\partial \mathrm{W}}$
- Chain rule (again): $\frac{\partial \mathrm{h}_t}{\partial \mathrm{h}_k} = \prod_{j=k+1}^{t} \frac{\partial \mathrm{h}_j}{\partial \mathrm{h}_{j-1}}$
- $\mathrm{h}_t = f(\mathrm{U}\mathrm{x}_t + \mathrm{W}\mathrm{h}_{t-1} + \mathrm{b}_h)$, *e.g.* f tanh
- Jacobian matrix $\frac{\partial \mathrm{h}_j}{\partial \mathrm{h}_{j-1}} = \mathrm{W}^T \mathit{diag}[f'(\mathrm{h}_{j-1})]$

$$\Rightarrow \text{Analyzing } \boxed{\left\|\frac{\partial \mathrm{h}_t}{\partial \mathrm{h}_k}\right\| = \left\| \prod_{j=k+1}^{t} \mathrm{W}^T \mathit{diag}[f'(\mathrm{h}_{j-1})]\right\|}$$

BPTT: Exploding and Vanishing Gradients

$$||\frac{\partial \mathrm{h}_t}{\partial \mathrm{h}_k}|| = || \prod_{j=k+1}^{t} \mathrm{W}^T diag[f'(\mathrm{h}_{j-1})]|| \leq (\beta_w \beta_h)^{t-k}$$

- β_h for activation (tanh=1, sigmoid=0.25), β_w for W (largest eigenvalue)
  - $\beta_h \cdot \beta_w > 1 \Rightarrow$ exploding gradients
  - $\beta_h \cdot \beta_w < 1 \Rightarrow$ vanishing gradients
- True for any deep networks, exacerbated for RNNs



Exploding and Vanishing Gradients: challenges & solutions

Main challenge: modeling long-range dependencies

Solution for exploding

- Regularization, e.g., $||W||_2$
- Simple common strategy: gradient clipping

$\Rightarrow$ Exploding gradients relatively easy to detect and fix

Solution for vanishing

- Use Truncated BPTT, but smaller range dependencies
- Using ReLU activation instead of tanh/sigmoid
- Specific architectures/models, e.g. GRU/LSTM

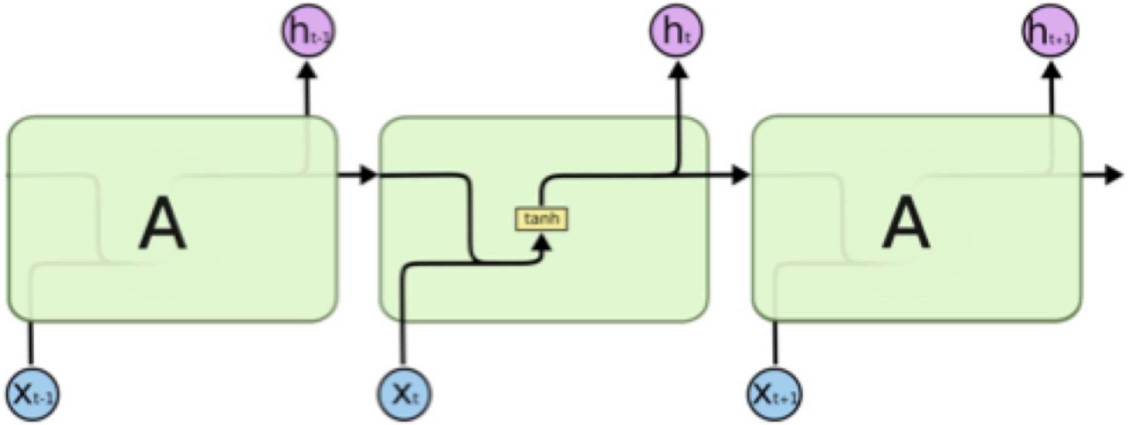
Outline

I. Recurrent Neural Networks (RNNs)
   a) Vanilla RNNs
   b) RNN training
   c) Specific RNN architectures
   d) Applications

II. Attention models & transfomers

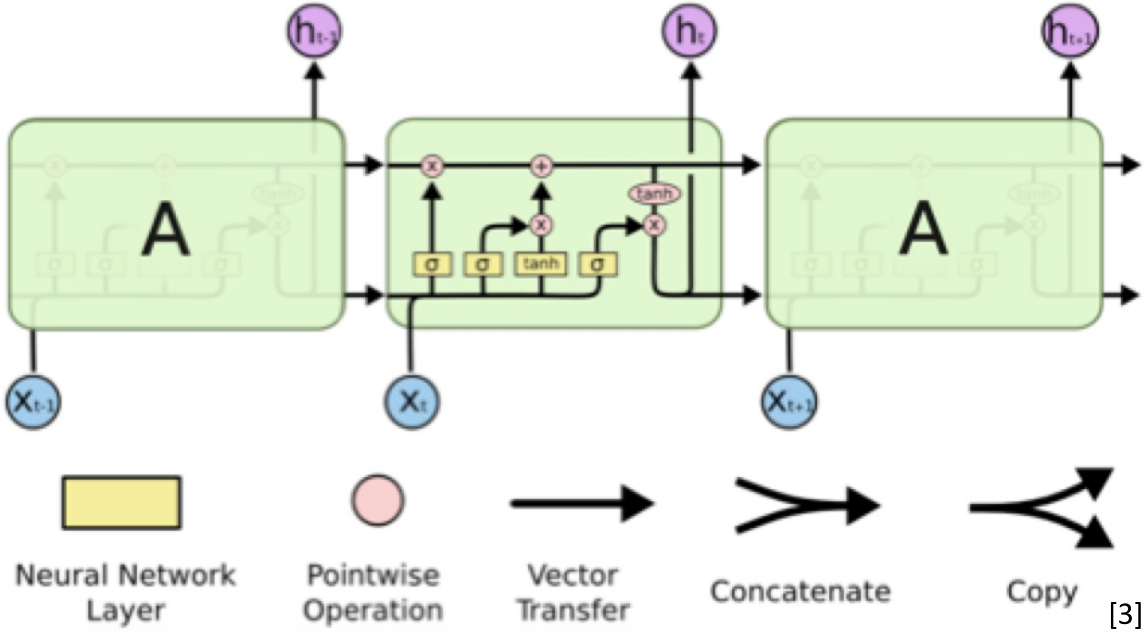
Long Short-Term Memory (LSTM) [3]

- Recap: Vanilla RNN cell



$$h_t = \tanh\left(W\begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix}\right)$$

- Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997]



$$\begin{pmatrix}i\\ f\\ o\\ g\end{pmatrix} = \begin{pmatrix}\sigma\\ \sigma\\ \sigma\\ \tanh\end{pmatrix} W\begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

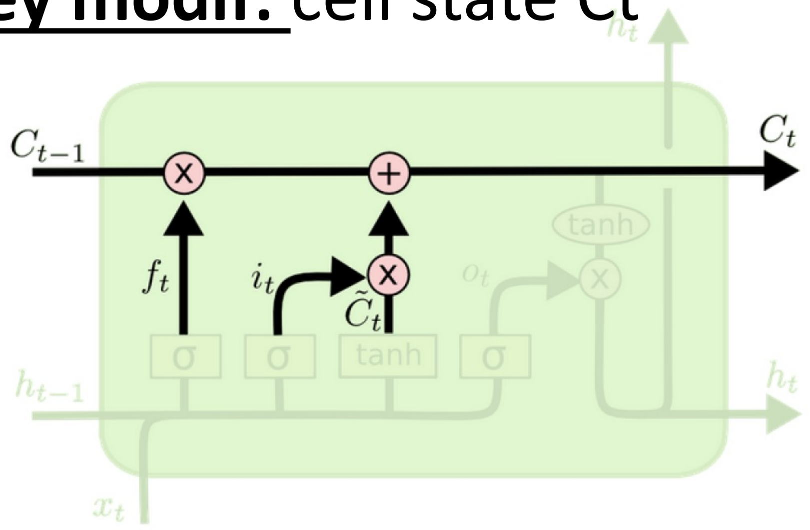
$$h_t = o \odot \tanh(c_t)$$

Credit: C. Olah

[3] S. Hochreiter, J. Schmidhuber.. Long short-term memory. Neural Comput, 1997

Long Short-Term Memory (LSTM)

Key modif: cell state Ct



$$\boxed{C_t = f_t * C_{t-1} + i_t * \tilde{C}_t}$$

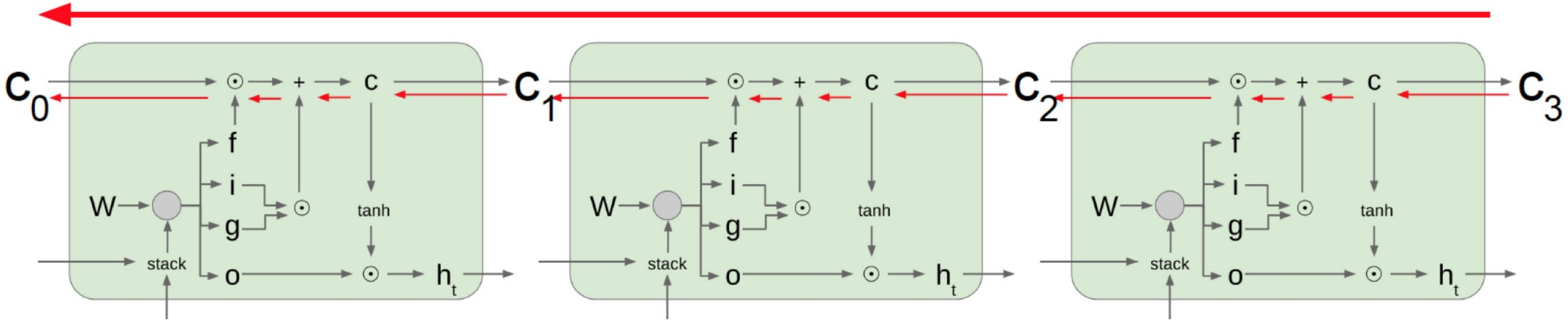
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$ Forget

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$ Input

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

- Only elementwise multiplication and addition, no matrix multiply by W

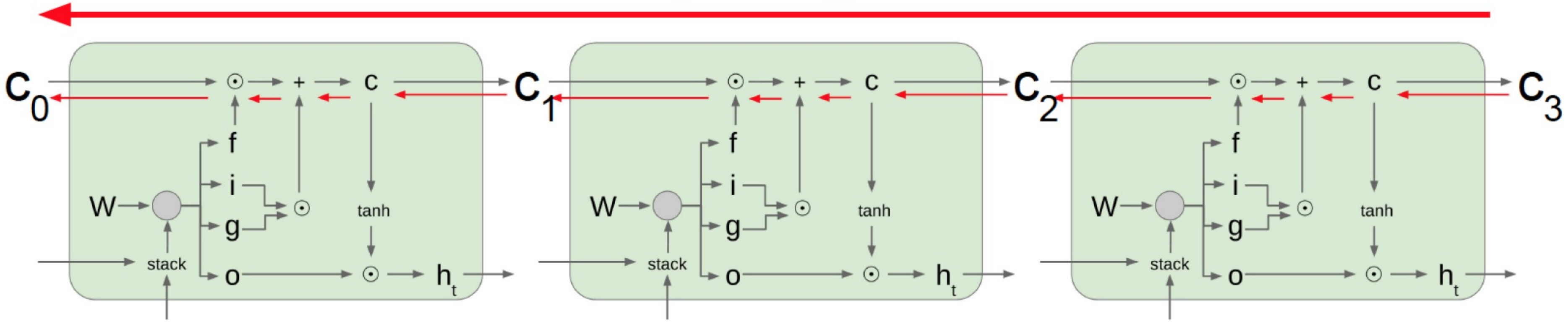
Key property: Uninterrupted gradient flow!



Long Short-Term Memory (LSTM)

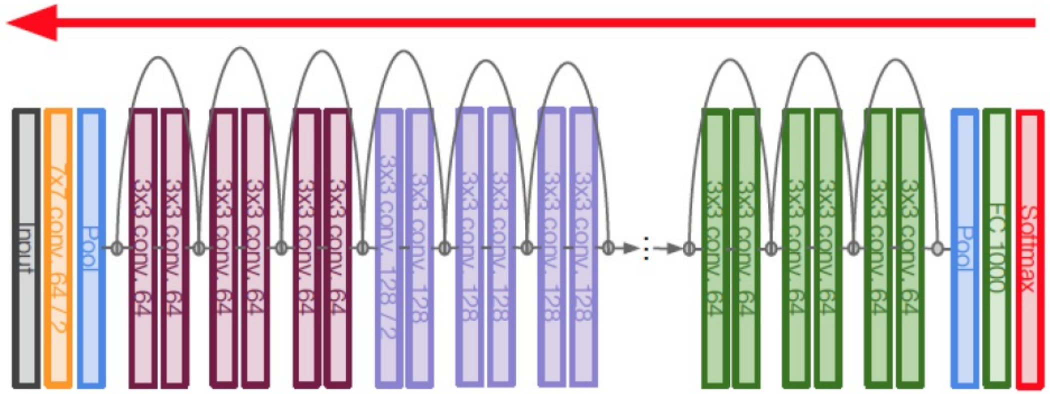
- Only elementwise multiplication and addition, no matrix multiply by W

Key property: Uninterrupted gradient flow!



Sounds familiar?

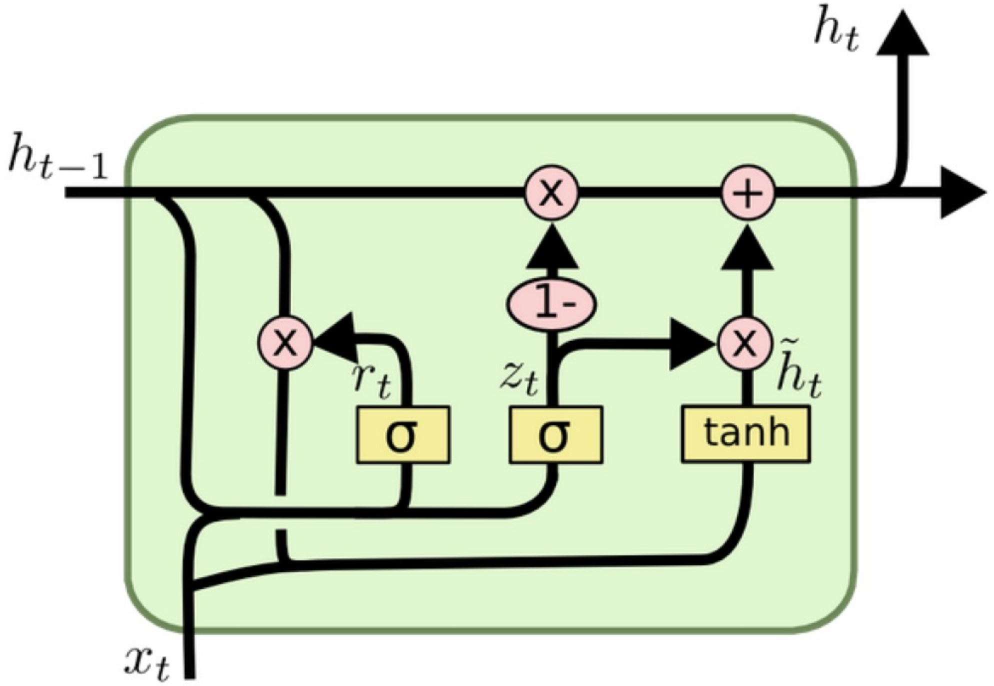
Similar to ResNets!



Credit: Fei-Fei

Gated Recurrent Unit [4]

- LSTM popular variant: Gated Recurrent Unit (GRU) [Cho et al., 2014]
  - Combines forget and input gates into a single "update gate"
  - Merges the cell state and hidden state



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

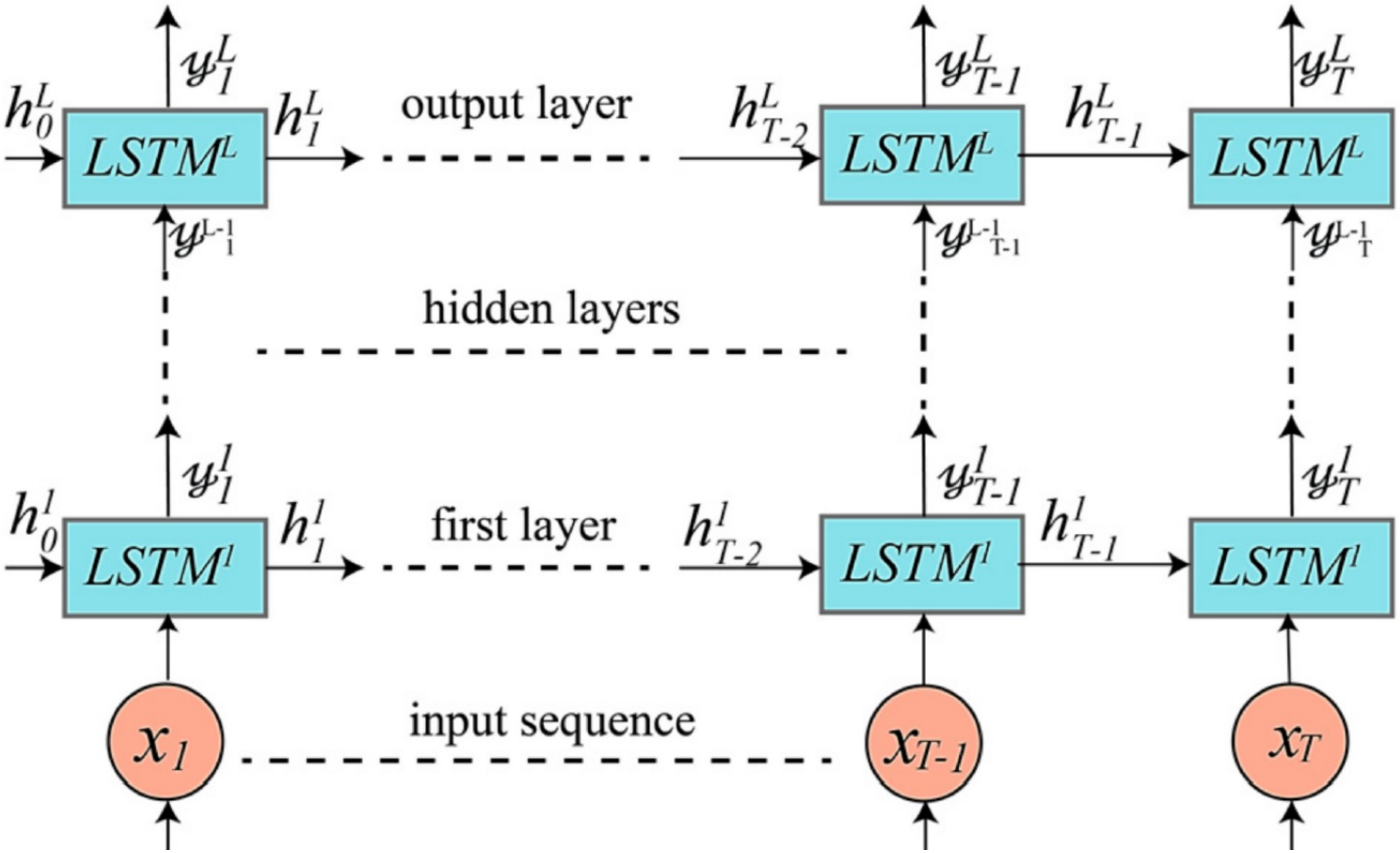
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Simpler than LSTM, generally slightly inferior performances

[4] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. EMNLP 2014.

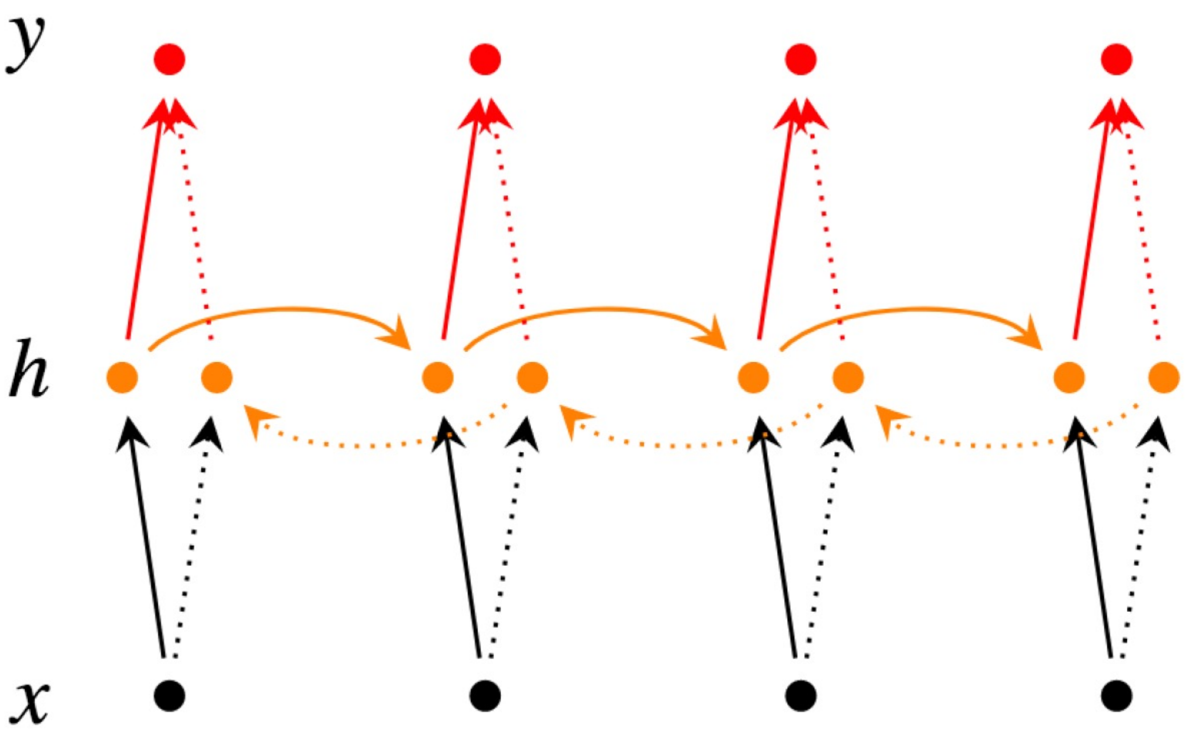
Deep RNNs

- Stacking RNN/ LSTM layers $\Rightarrow$ learning more complex features
- Deep LSTM: very powerful, especially when stacked and made even deeper and if you have lots and lots of data



Bi-directionnal RNNs

- For classification, incorporate information from words both preceding and following



$$\overrightarrow{h}_t = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b})$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})$$

$$y_t = g(U[\overrightarrow{h}_t;\overleftarrow{h}_t]+c)$$

$h = [\overrightarrow{h};\overleftarrow{h}]$ now represents (summarizes) the past and future around a single token.

Outline

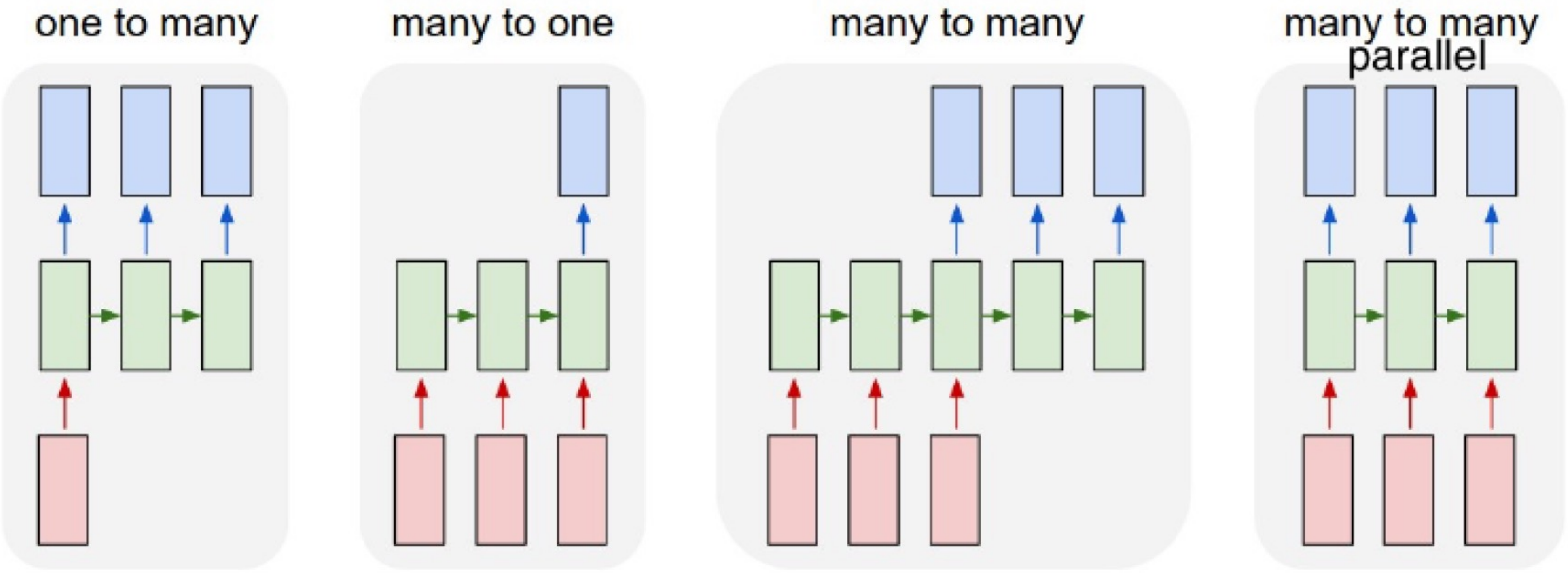
I. Recurrent Neural Networks (RNNs)

  a) Vanilla RNNs
  b) RNN training
  c) Specific RNN architectures
  d) RNN Applications

II. Attention models & transfomers

RNN applications

- **RNN: mapping input sequence $\{x_t\}_{t\in\{1;T\}}$ into $\{y_t\}_{t\in\{1;T\}}$**
- **Different tasks $\Leftrightarrow$ different mappings**



Language models, Image captioning

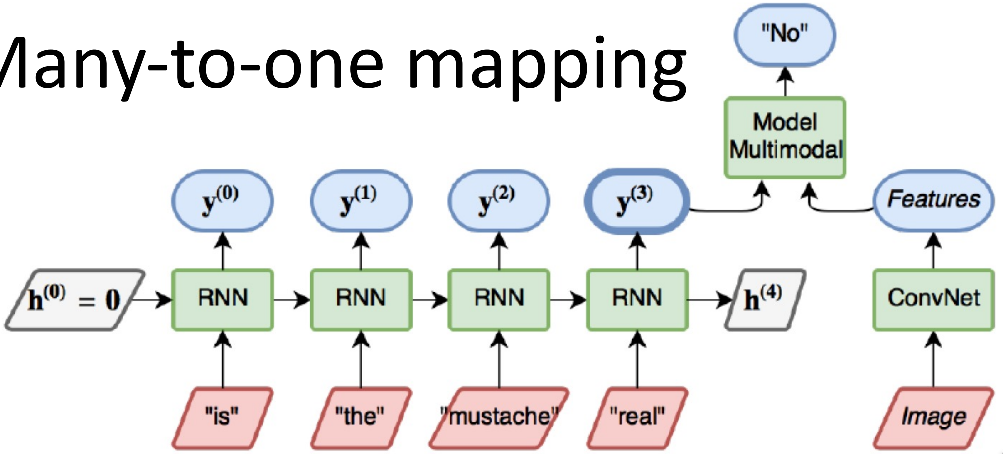
Sentiment classification, Visual Question Answering (VQA)

Translation, char nn

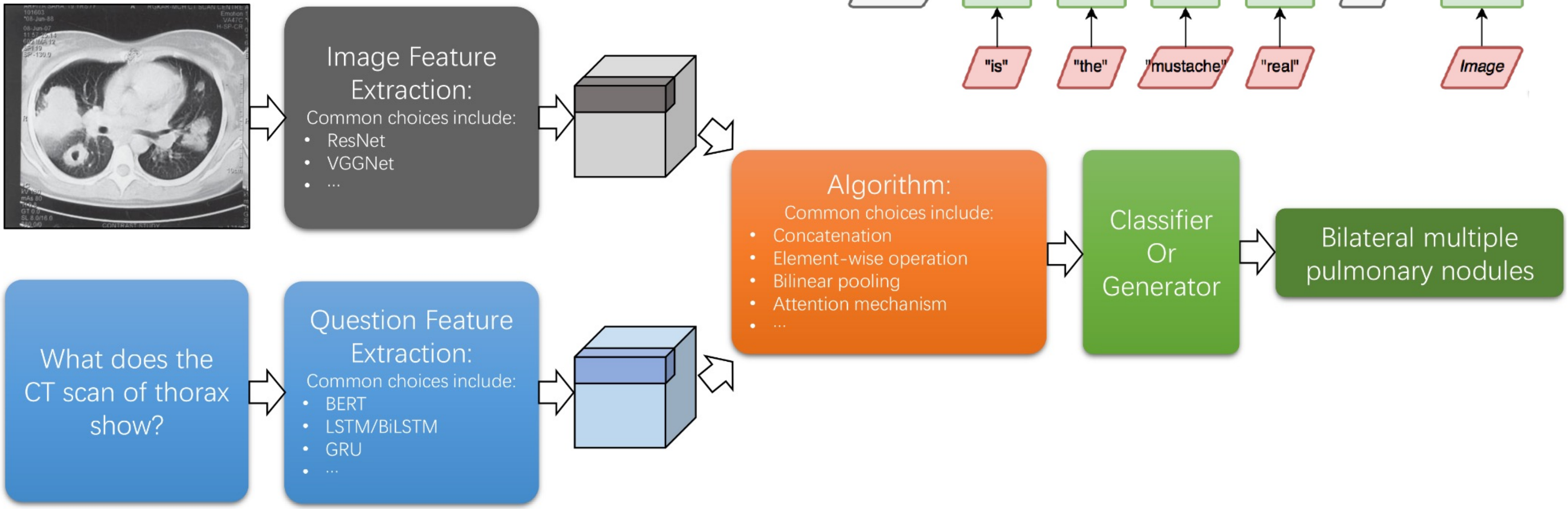
video classification

Medical Visual Question answering (MVQA) [5]

Many-to-one mapping



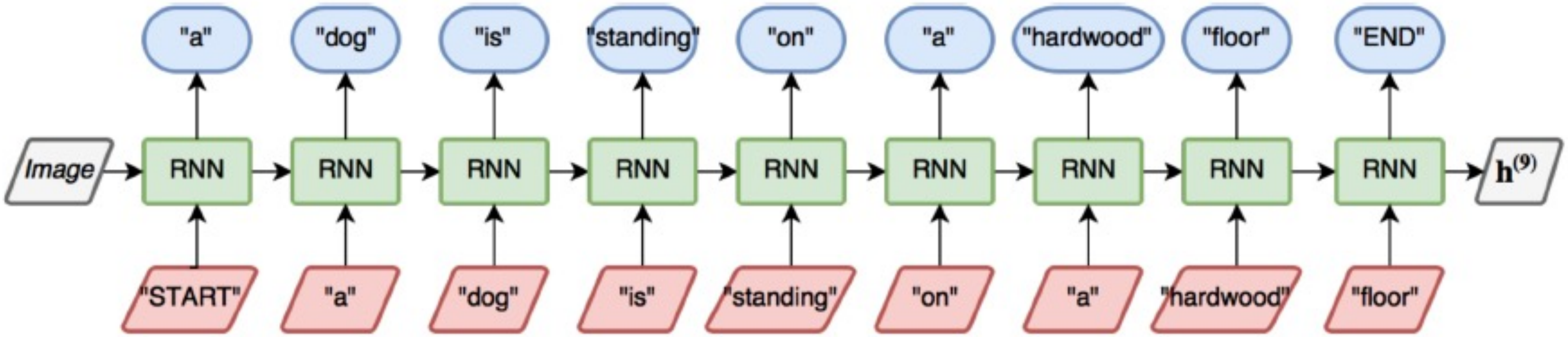
General architecture

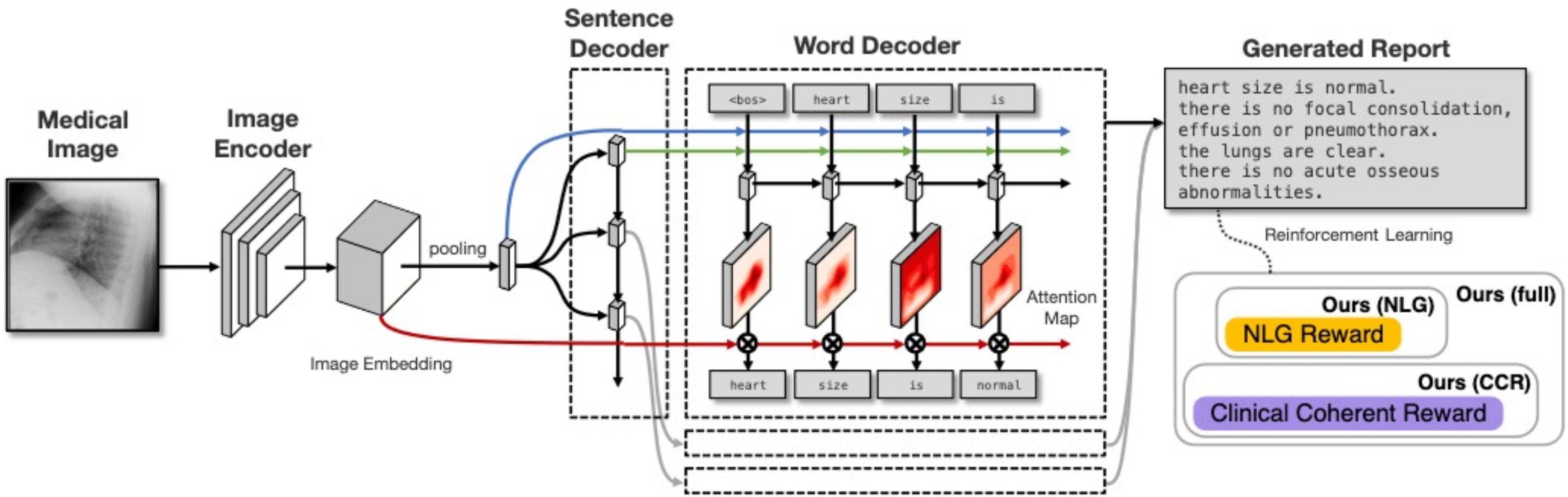


[5] From Z. Lin, D. Zhang, Q. Tao, D. Shi, G. Haffari, Q. Wu, M. He, Z. Ge. Medical Visual Question Answering: A Survey. ArXiv, 2023

Image Captioning: medical report generation [6]

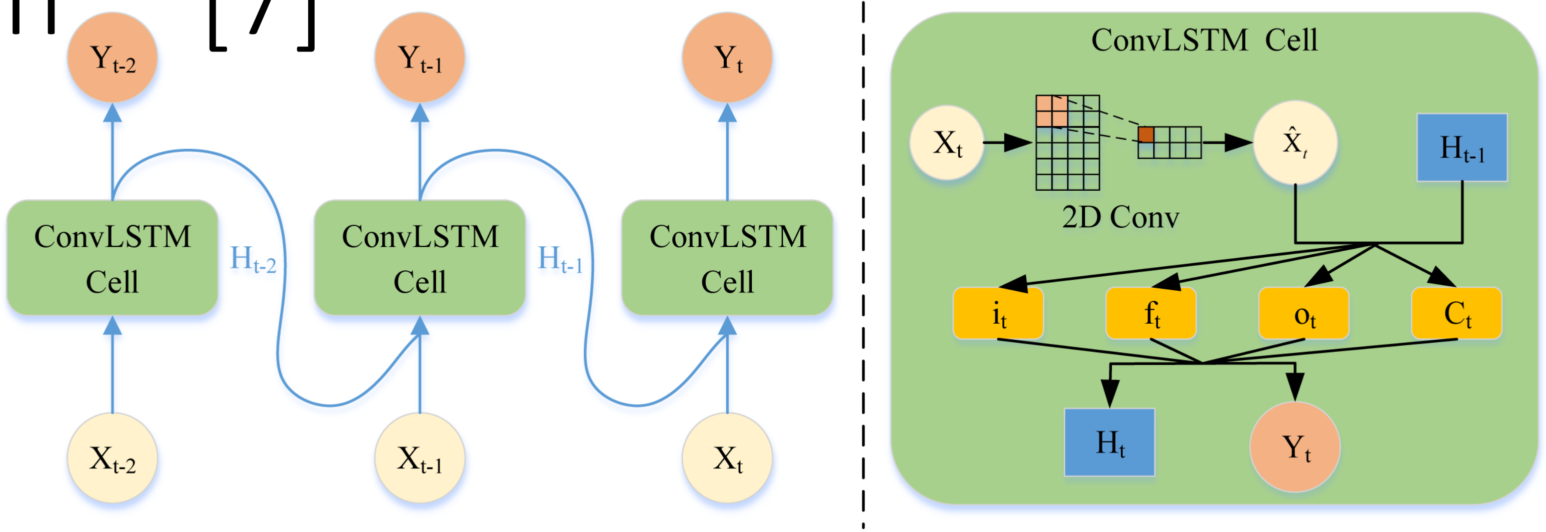
One-to-many mapping





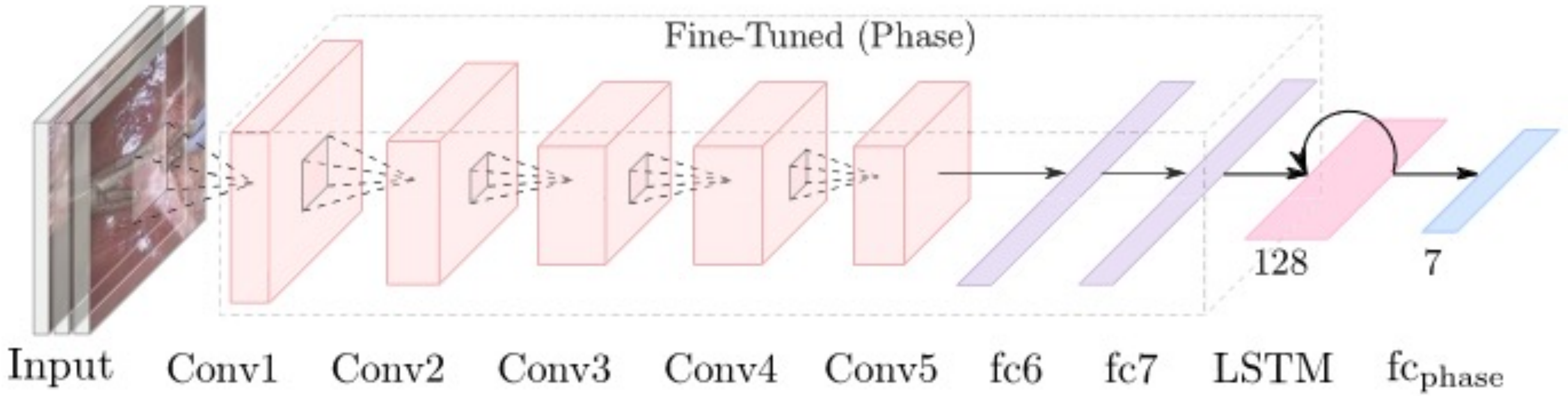
[6] From G. Liu, T.M. Harry Hsu, M. McDermott, W. Boag, W.H. Weng, P. Szolovits, M. Ghassemi. Clinically Accurate Chest X-Ray Report Generation. ArxiV, 2019.

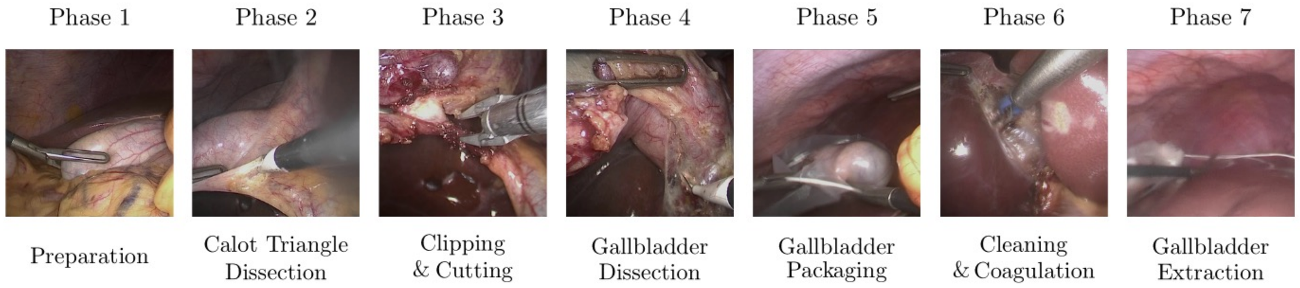
Video classification [7]



- ConvLSTM [Shi et. al. 2015]
  - generic module for video classification
  - Many-to-many mapping

- Application to Surgical Phase Recognition





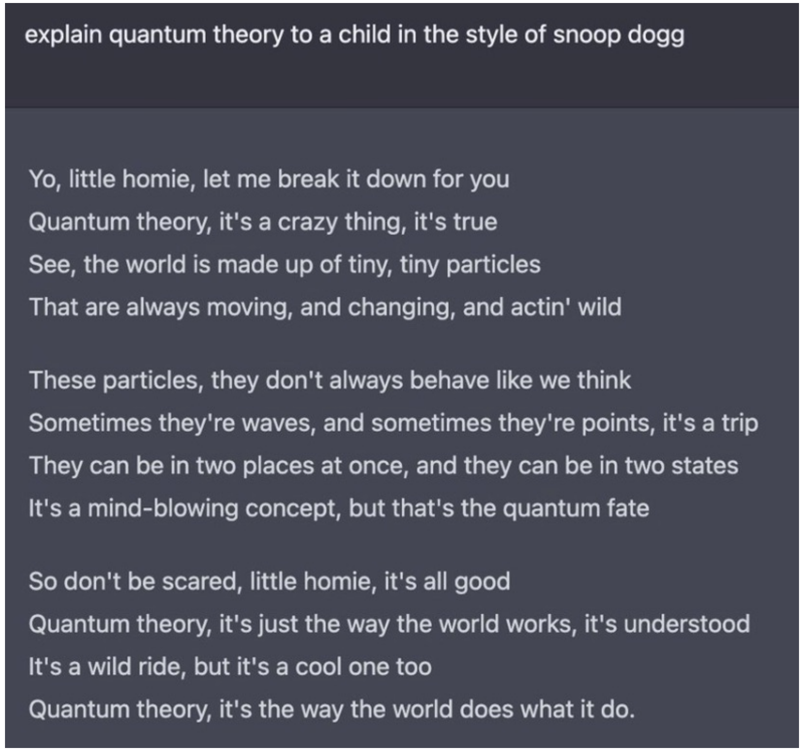
[7] G. Yengera, D. Mutter, J. Marescaux, N. Padoy. Less is More: Surgical Phase Recognition with Less Annotations through Self-Supervised Pre-training of CNN-LSTM Networks. ArXiv, 2019.

Outline

I. Recurrent Neural Networks (RNNs)
II. Attention models & transfomers
   a) Context
   b) Transformer block
   c) Medical image segmention

Transformers everywhere since 2017

NLP: BERT, GPT-3/4, Chat-GPT, *etc*

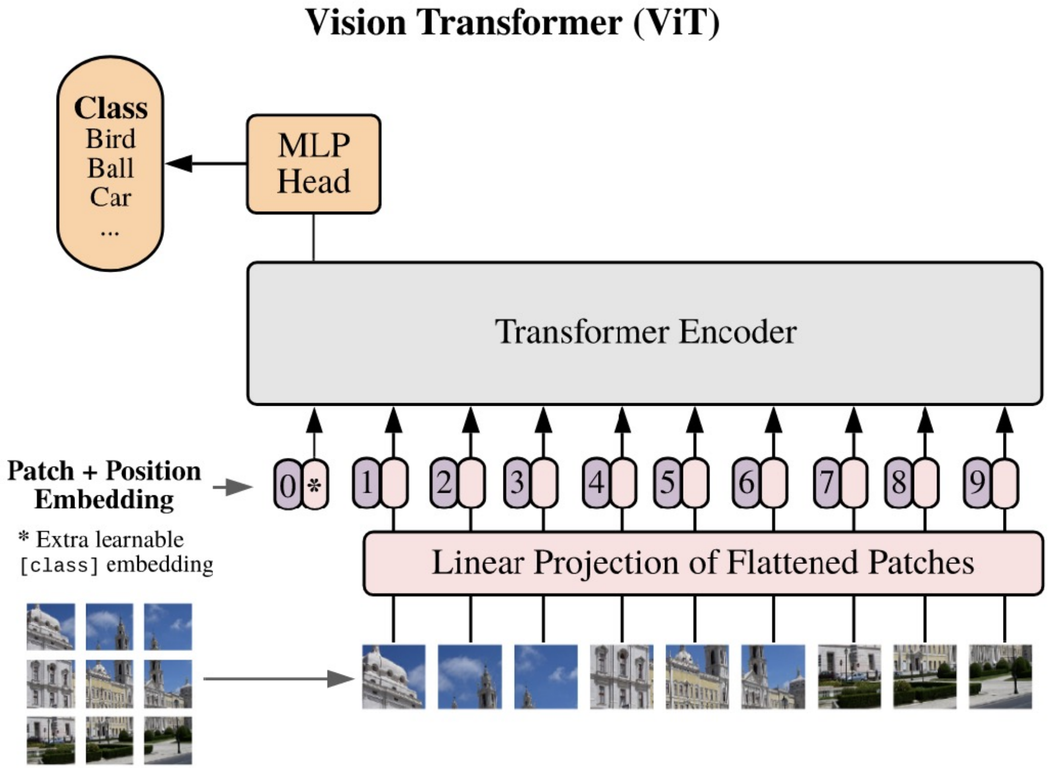

explain quantum theory to a child in the style of snoop dogg

Yo, little homie, let me break it down for you
Quantum theory, it's a crazy thing, it's true
See, the world is made up of tiny, tiny particles
That are always moving, and changing, and actin' wild

These particles, they don't always behave like we think
Sometimes they're waves, and sometimes they're points, it's a trip
They can be in two places at once, and they can be in two states
It's a mind-blowing concept, but that's the quantum fate

So don't be scared, little homie, it's all good
Quantum theory, it's just the way the world works, it's understood
It's a wild ride, but it's a cool one too
Quantum theory, it's the way the world does what it do.

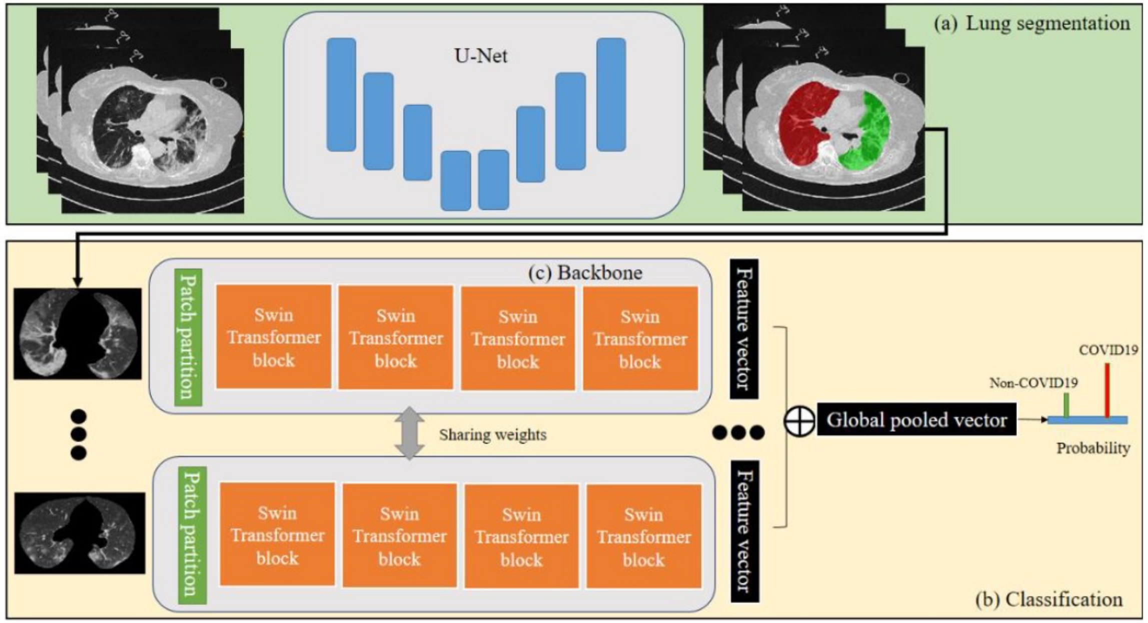
Vision since ’21: Vision Image Transformer (ViT)



Transformer in medical image analysis

Used in various contexts and tasks

- Image classification, detection, *e.g.* COVID, Semantic segmentation
- Image Registration
- Image Generation
- Im-2-im translation

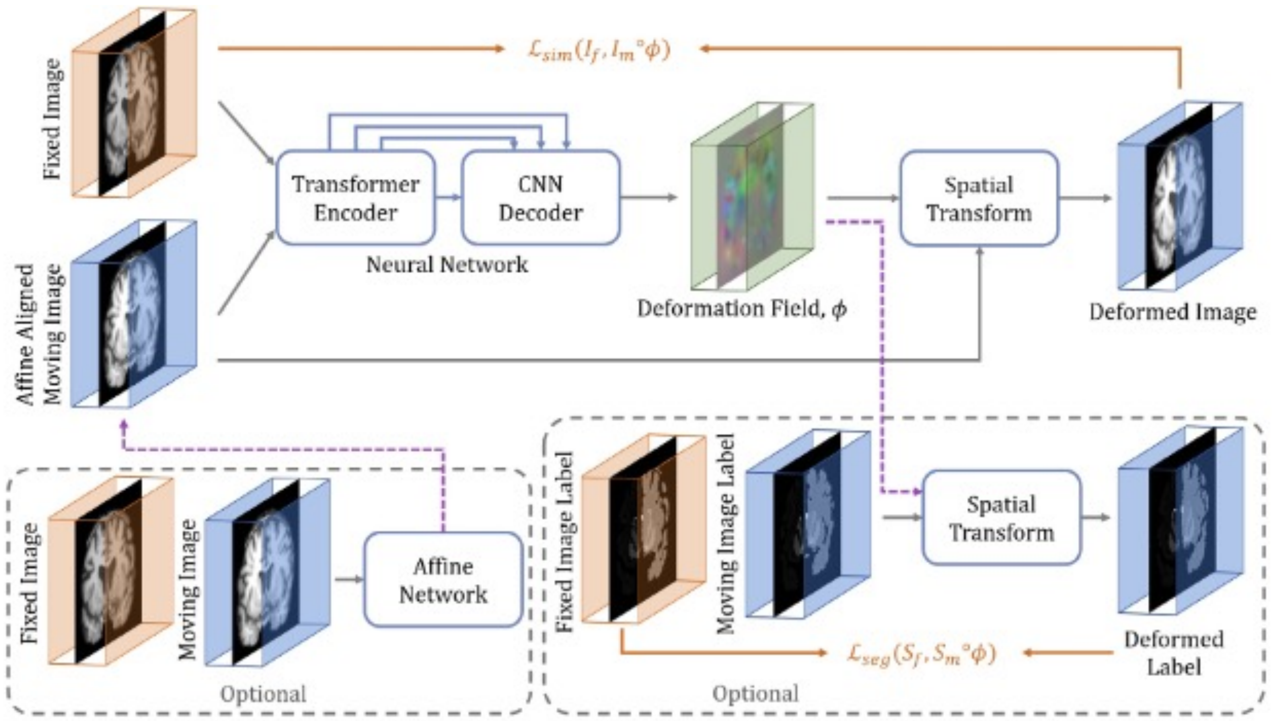


Zhang L, Wen Y. Mia-cov19d: A transformer-based framework for covid19 classification in chest cts. arXiv, 2021.

Transformer in medical image analysis

Used in various contexts and tasks

- Image classification, detection, *e.g.* COVID, Semantic segmentation
- Image Registration
- Image Generation
- Im-2-im translation

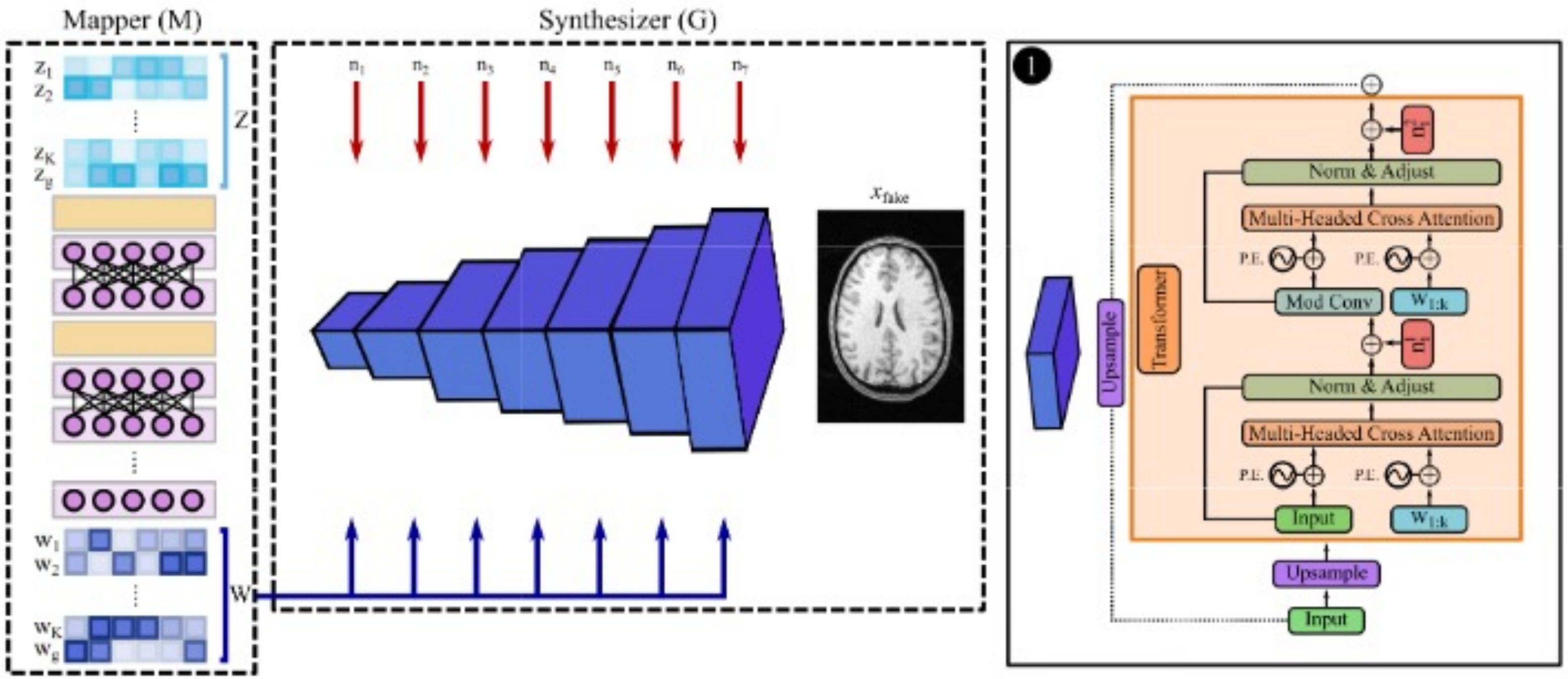


Chen J, Du Y, He Y, et al. Transmorph: Transformer for unsupervised medical image registration. Medical Image Analysis, 2022.

Transformer in medical image analysis

Used in various contexts and tasks

- Image classification, detection, *e.g.* COVID, semantic segmentation
- Image Registration
- Image Generation
- Im-2-im translation

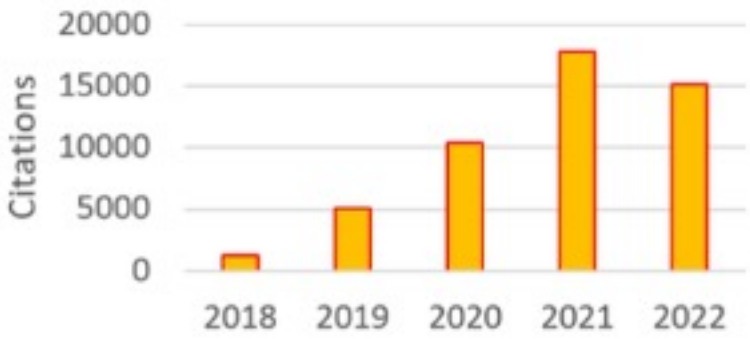


Korkmaz Y, Dar SU, Yurt M, et al. Unsupervised MRI reconstruction via zero-shot learned adversarial transformers. IEEE TMI, VOL. 41, NO. 7, JULY 2022

Focus on this talk

- Paper on transformer every day…

(a) Citations of Transformer papers in recent years



(b) Number of papers published in the last 12 months that contain "Action Recognition" + ("Transformer" OR "Attention") in their titles

- By no means exhaustive literature review


Intelligent Medicine 3 (2023) 59–78

Contents lists available at ScienceDirect

Intelligent Medicine

journal homepage: www.elsevier.com/locate/imed

ELSEVIER

Review

Transformers in medical image analysis

Kelei He[1,2,#], Chen Gan[2,#], Zhuoyuan Li[1,2,#], Islem Rekik[3,4,#], Zihao Yin[2], Wen Ji[2], Yang Gao[2,5], Qian Wang[6,*], Junfeng Zhang[1,2,*], Dinggang Shen[6,7,8,*]

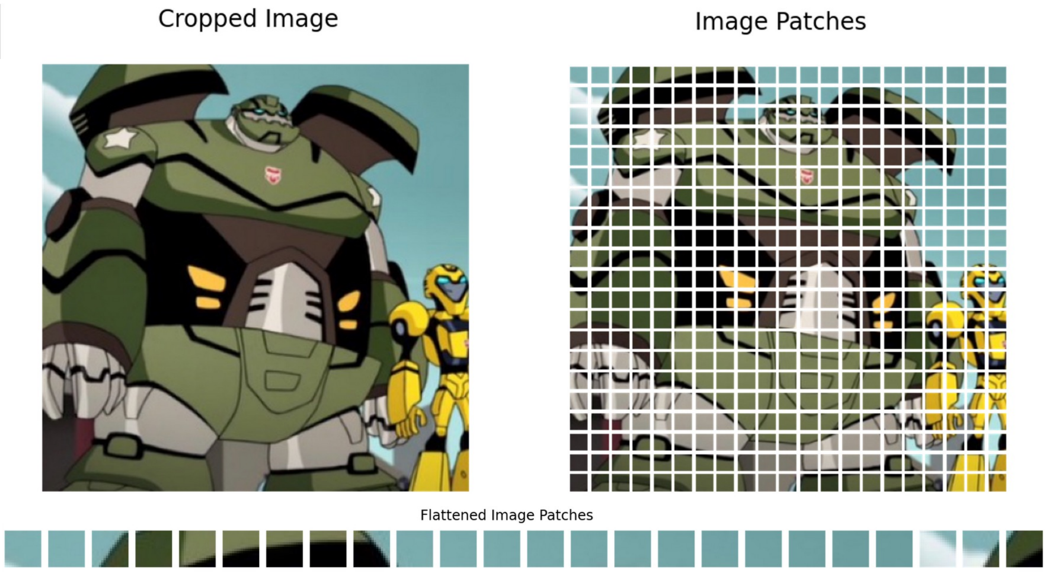
Outline

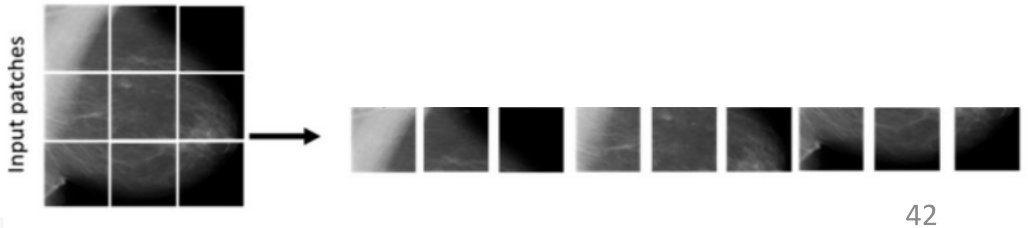
I. Recurrent Neural Networks (RNNs)
II. Attention models & transfomers
  a) Context
  b) Transformer block
  c) Medical image segmention

From sequence to set

- A sequence of elements → a **set** of tokens**, no order**
  - Token: primitives, elementary elements of data
    - Text: token are e.g. words
    - Image: token are e.g. patches

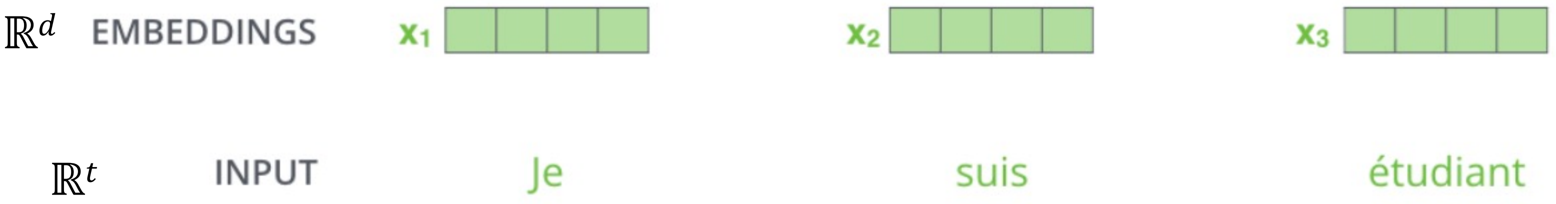
Text	Tokenization
"Hello I love you" ⇨	"Hello", "I", "love", "you"
	"love", "Hello", "I", "you"
	"I", "love", "Hello", "you"

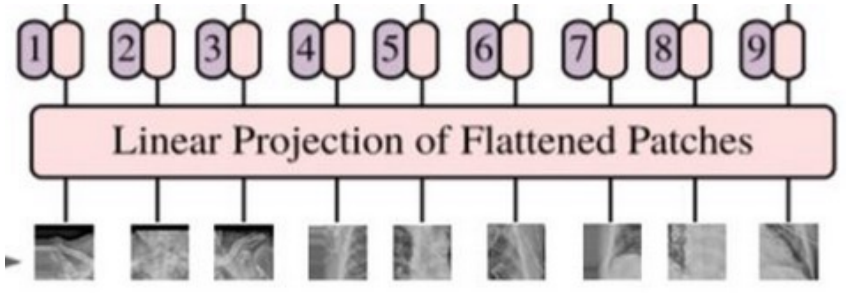




Input embedding

- Token: input vector in $\mathbb{R}^t$
    - Word: t =|V|, V vocabulary
    - Image patch: t = s^2, where s is the patch size
- Input embedding: linear projection $\mathbb{R}^t \rightarrow \mathbb{R}^d$: $e_i = x_i W^e$





Positional encoding

- Sequence → set of token:
  - Permutation invariant
  - Loosing structural information from data
- Recovering structure: **positional encoding (PE)**
  - Mapping token position t to a vector $\mathbf{p}_t \in \mathbb{R}^d$
  - Seminal PE: sinusoidal

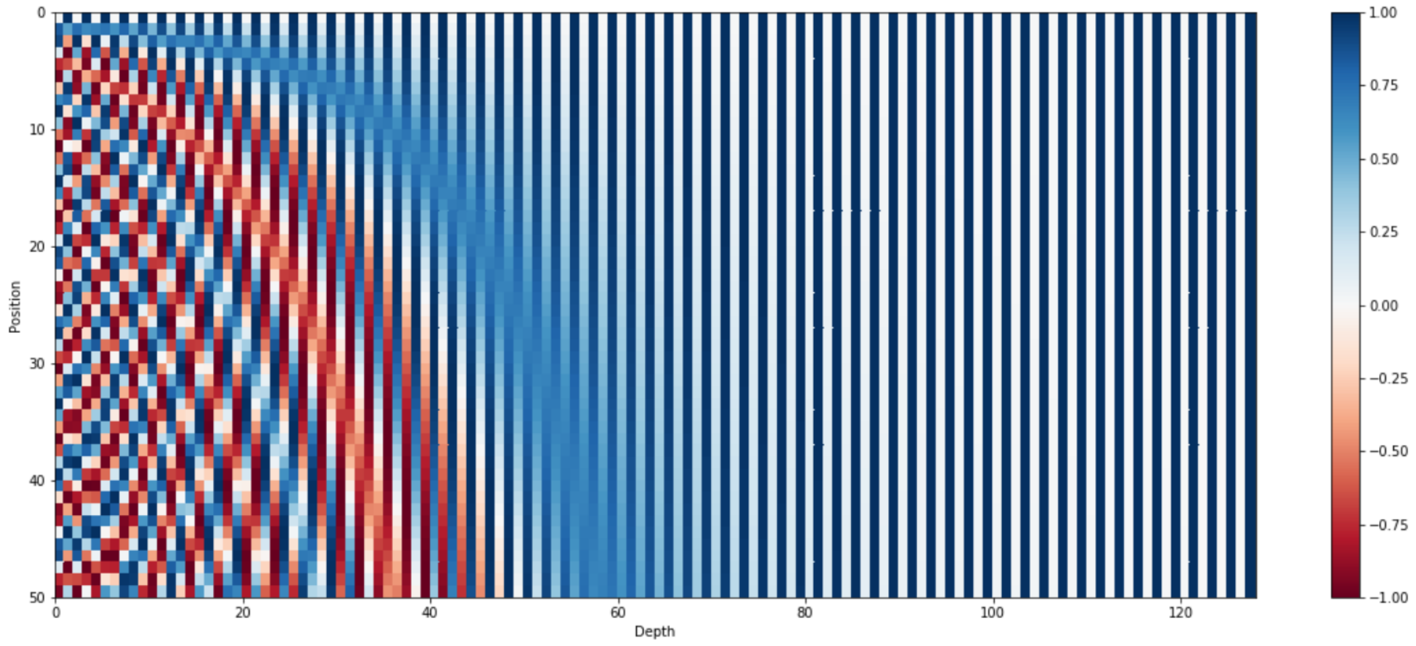
$$\vec{p}^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases}$$

$$\omega_k = \frac{1}{10000^{2k/d}}$$

$$\vec{p} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Sinusoidal positional encoding

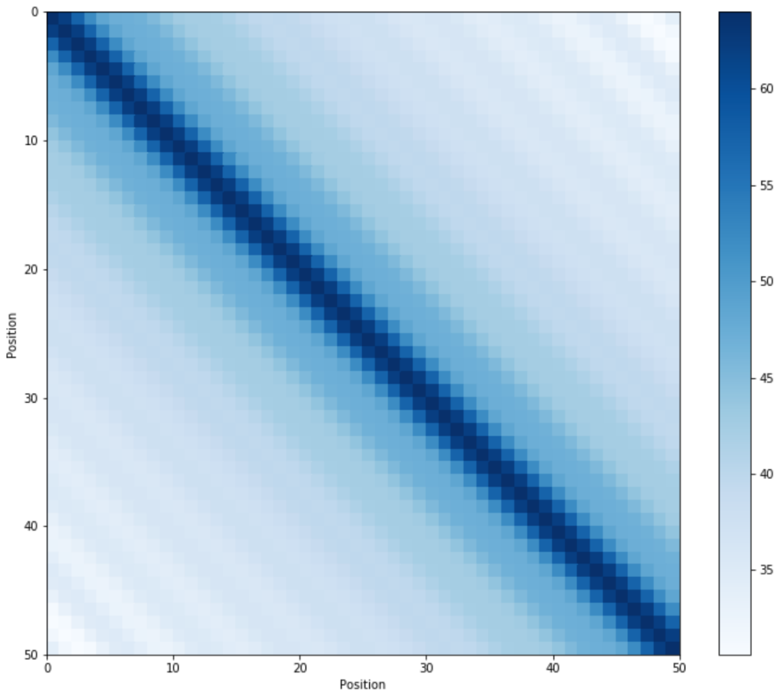
- Unique vector $\mathbf{p}_t$ for each position t
- $p_t(i) \in [-1;1]$: natural normalization



d=128, max length of token set = 50

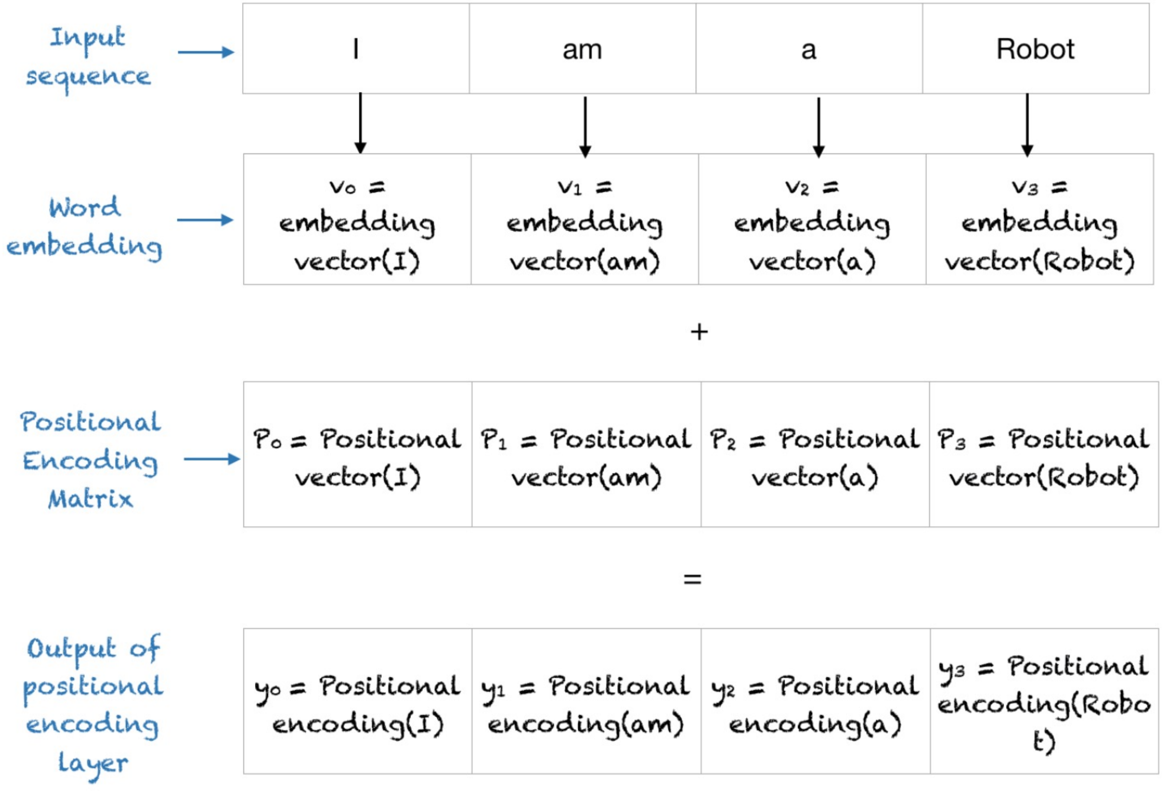
- Models relative position
- Positional similarity:

$$K = PPt$$



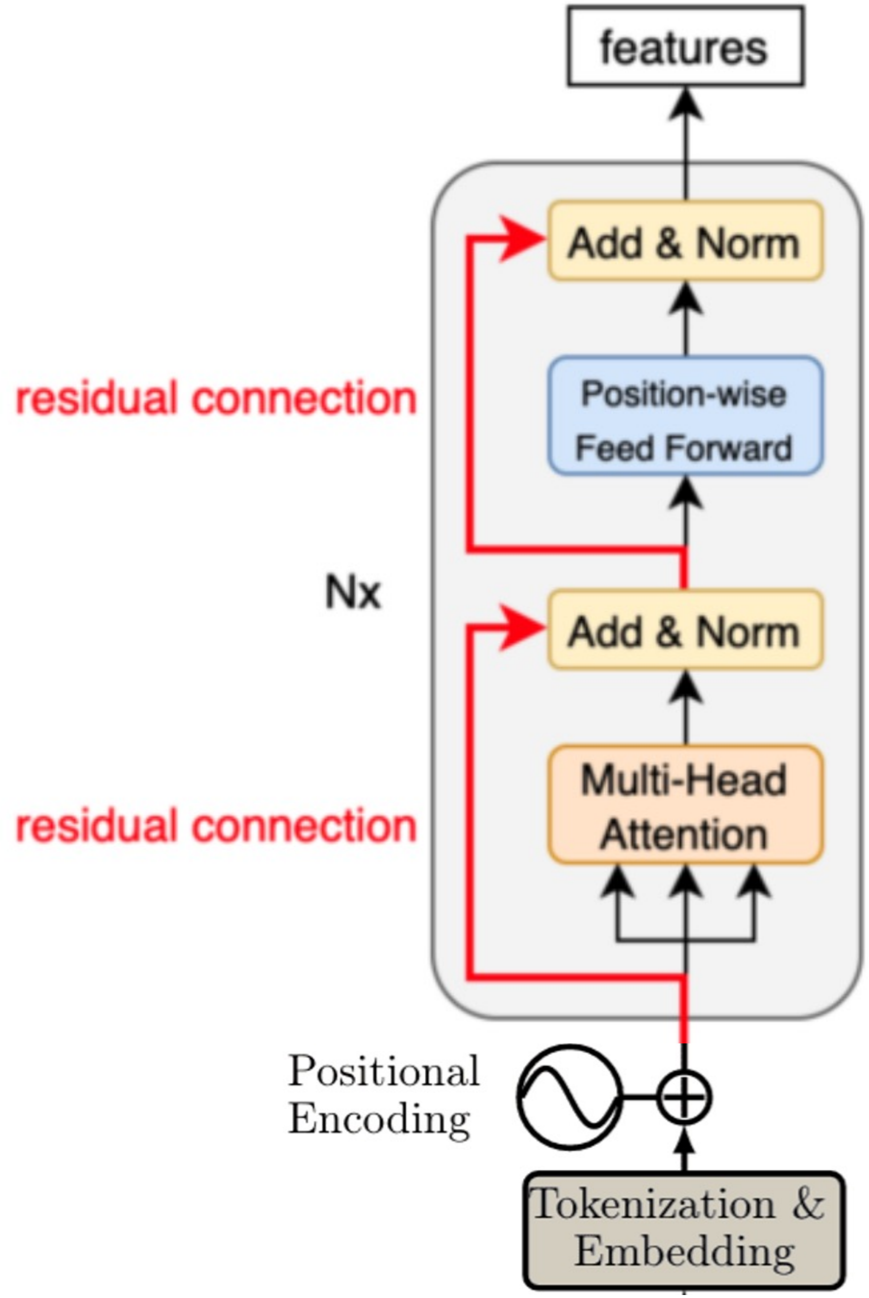
Positional encoding

- Other possible encoding, can be learned
- Final embedding :



=> Input of transformer!

Transformer [8] : the encoder

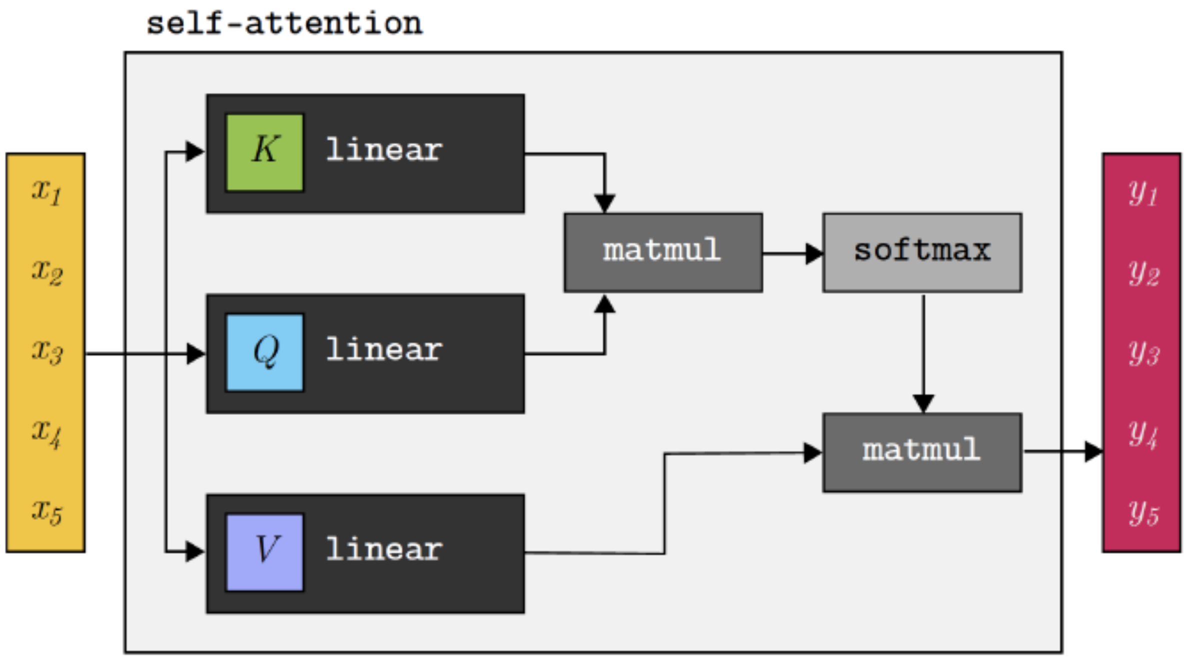


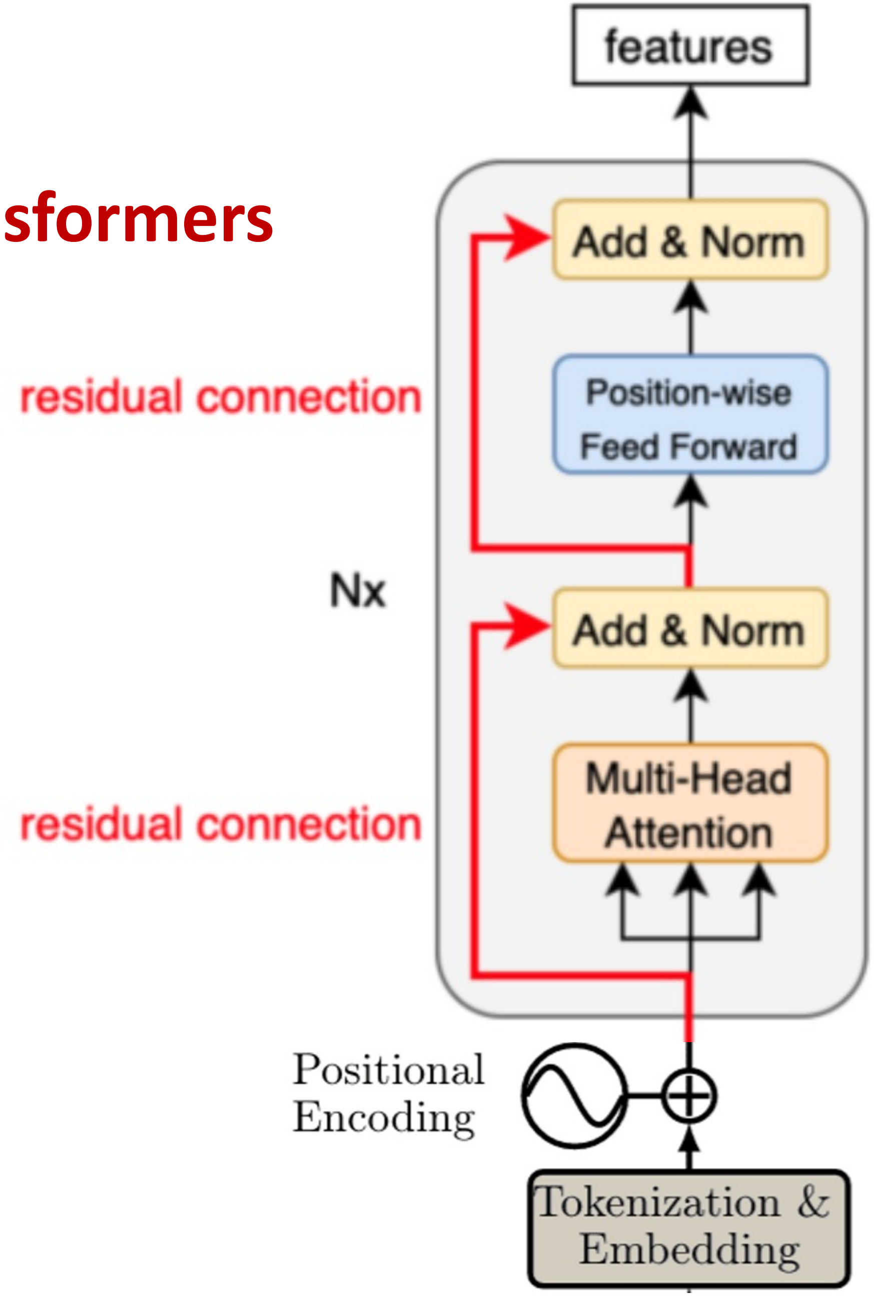
- A stack a N transformer blocks
  - Input a set of embedded tokens
  - Output: a set of re-embedded tokens

[8] Attention Is All You Need. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. NeurIPS 2017.

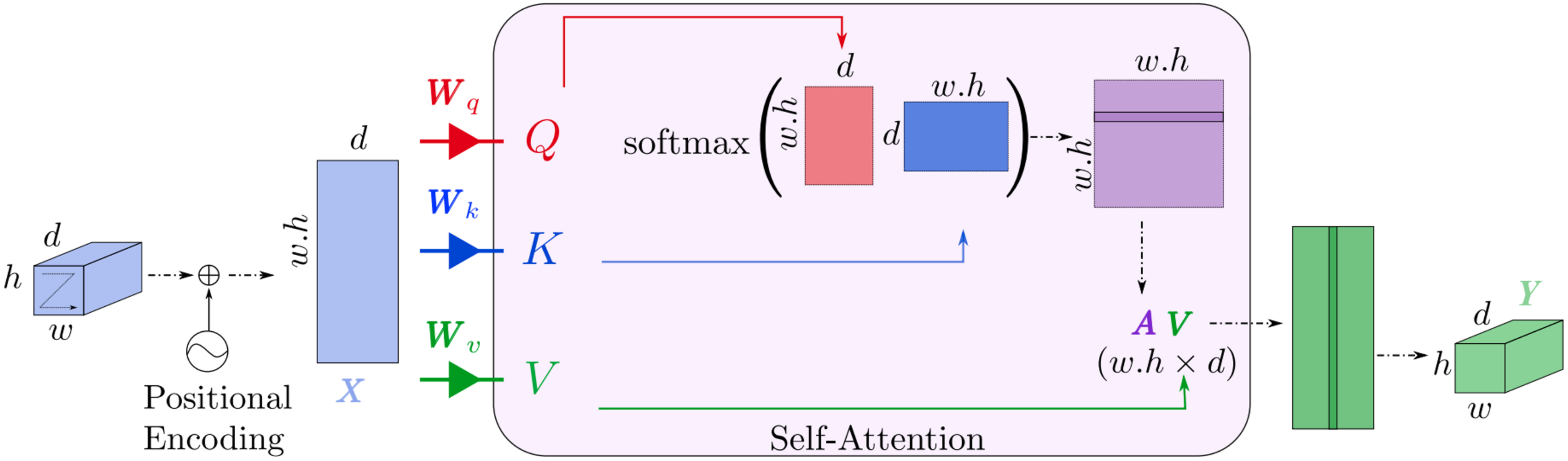
Transformer: self attention

- **The most important and specific module in transformers**
- Project the input set into 3 sets
  - Query: sought info
  - Key: context elements
  - Value: retrieved





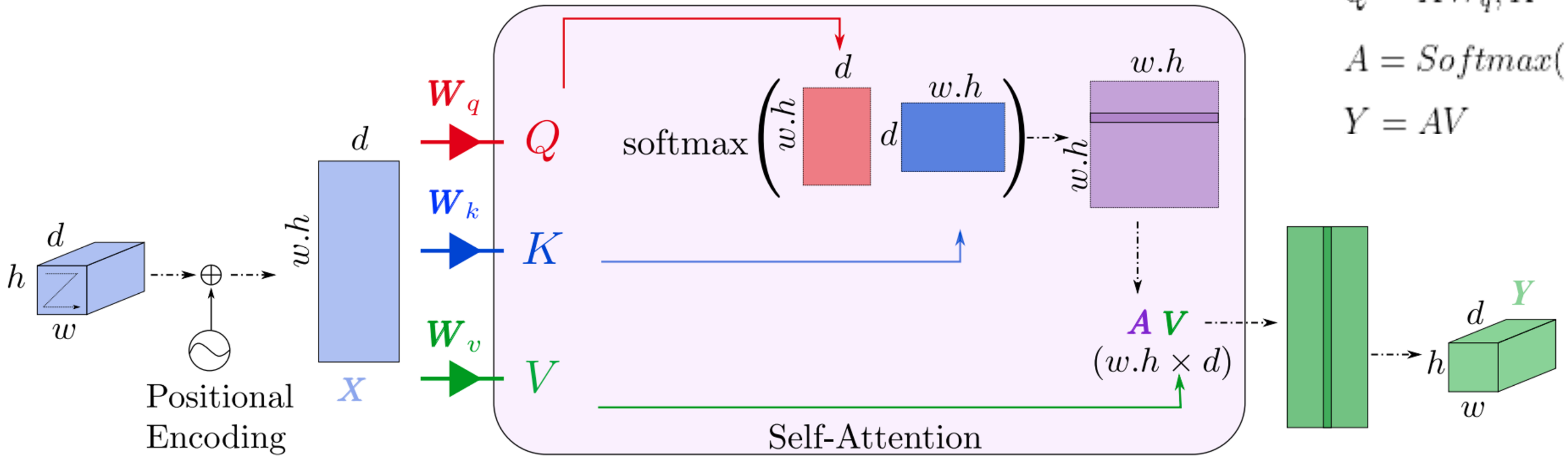
Self-attention



$$X \in \mathbb{R}^{wh \times d}, W_q \in \mathbb{R}^{d \times d}, W_k \in \mathbb{R}^{d \times d}, W_v \in \mathbb{R}^{d \times d}$$
$$Q = XW_q, K = XW_k, V = XW_v$$
$$A = Softmax(\frac{QK^T}{\sqrt{d}})$$
$$Y = AV$$

Self-attention: conclusion

$$X \in \mathbb{R}^{wh \times d}, W_q \in \mathbb{R}^{d \times d}, W_k \in \mathbb{R}^{d \times d}, W_v \in \mathbb{R}^{d \times d}$$
$$Q = XW_q, K = XW_k, V = XW_v$$
$$A = Softmax(\frac{QK^T}{\sqrt{d}})$$
$$Y = AV$$



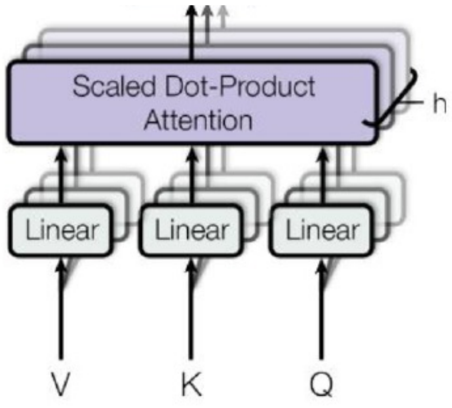
- Each token y_i in Y: computed a linear combination of v_i
  - Enables to model **global interactions** between v_i tokens: full contextual information
  - $\neq$ ConvNets in vision, interactions limited by the size of the receptive field
  - $\neq$ RNNs for sequence processing, interactions limited by vanishing gradients
- **Self attention: $O(N^2)$ complexity**
  - Expensive (or impossible) for large N

Multi-headed attention

- High-level idea: multiple self-attention in parallel

- Each head: attend to different parts

- Combine the heads' outputs

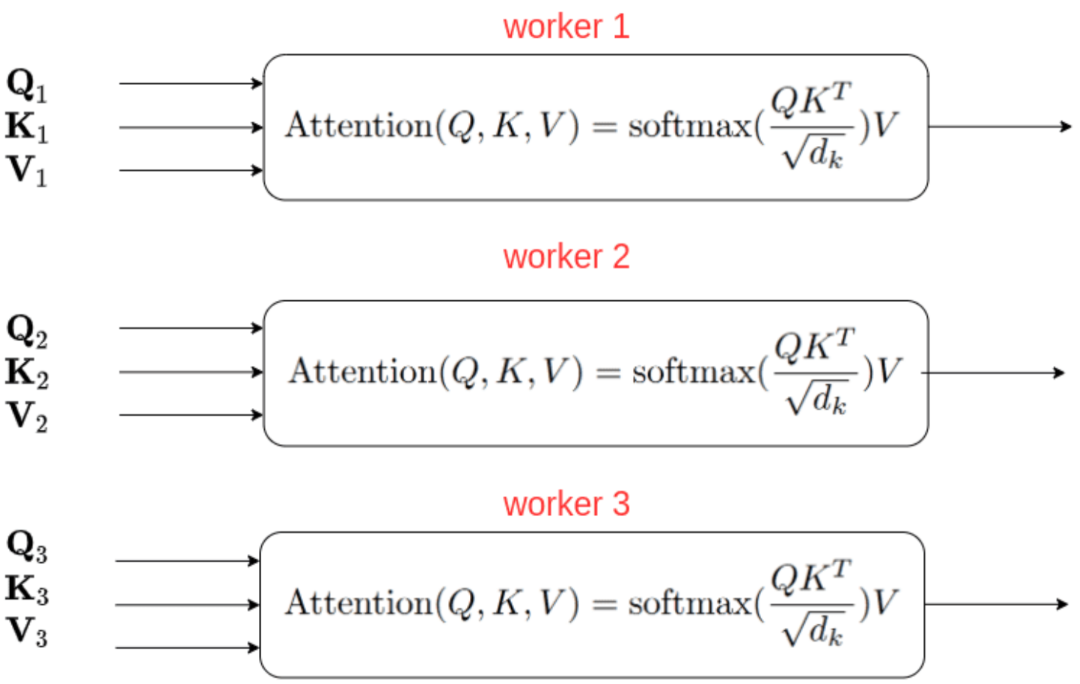


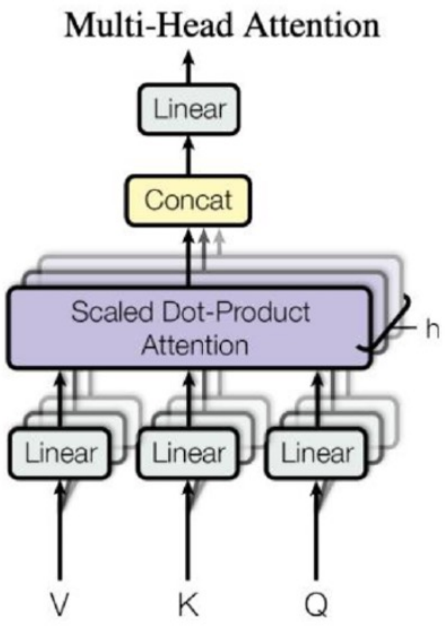
[Vaswani et al. 2017]

Wizards of the Coast, Artist: Todd Lockwood

Credit: Anna Goldie

Multi-headed attention



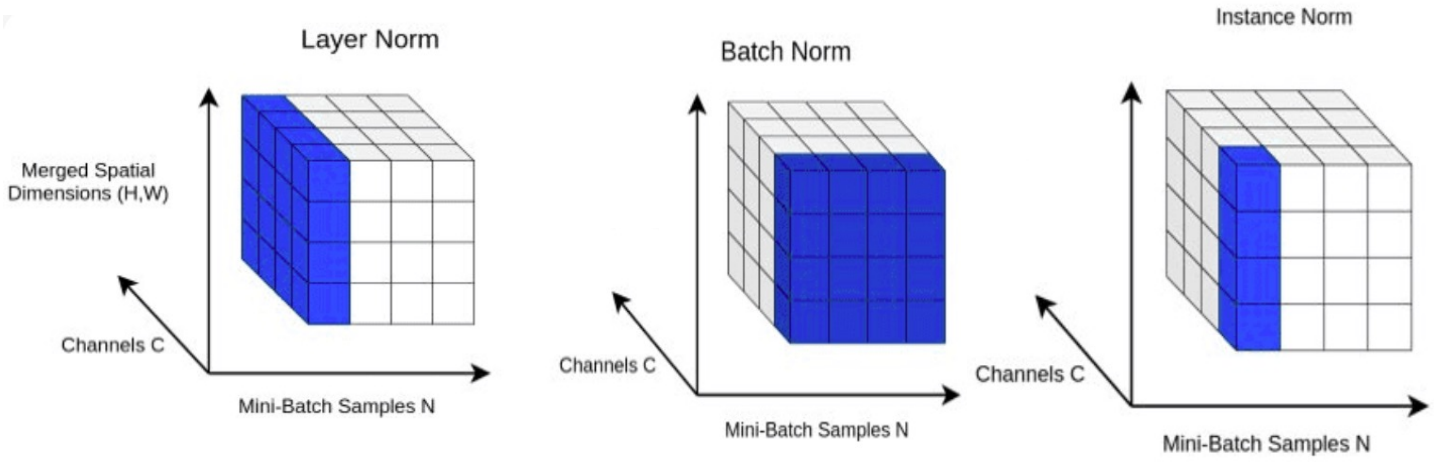


[Vaswani et al. 2017]

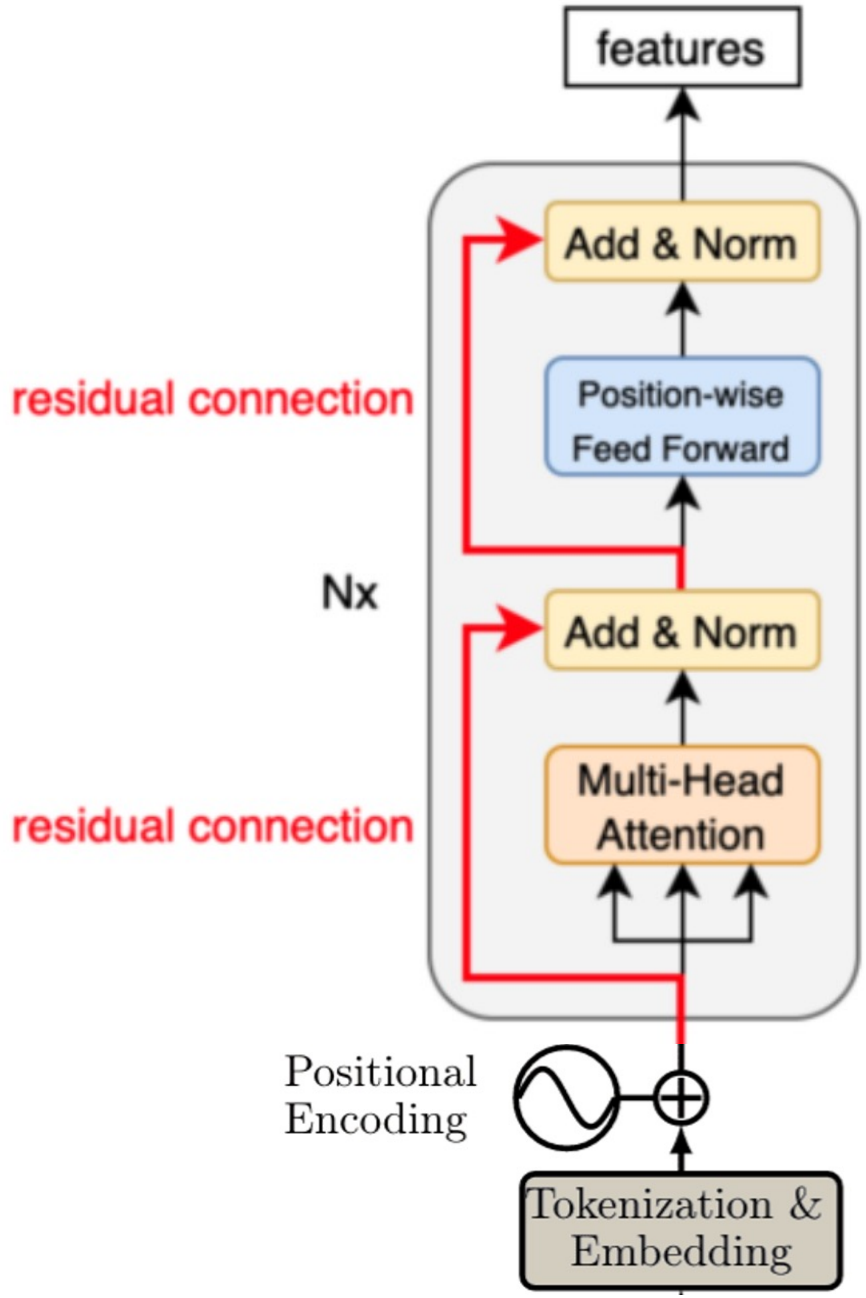
- Concatenate the heads' outputs
- Use a linear layer: desired output size

Layer normalization

- Normalization on joint channel and spatial dimensions

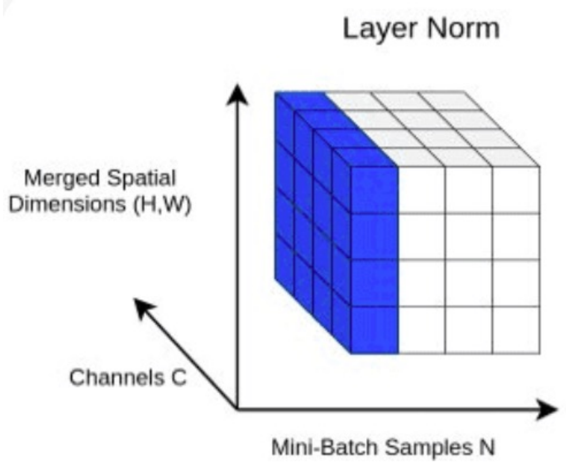


  - Stabilize training, faster convergence



Layer normalization

- Normalization on joint channel and spatial dimensions



$$\mu_n = \frac{1}{K}\sum_{k=1}^{K} x_{nk}$$

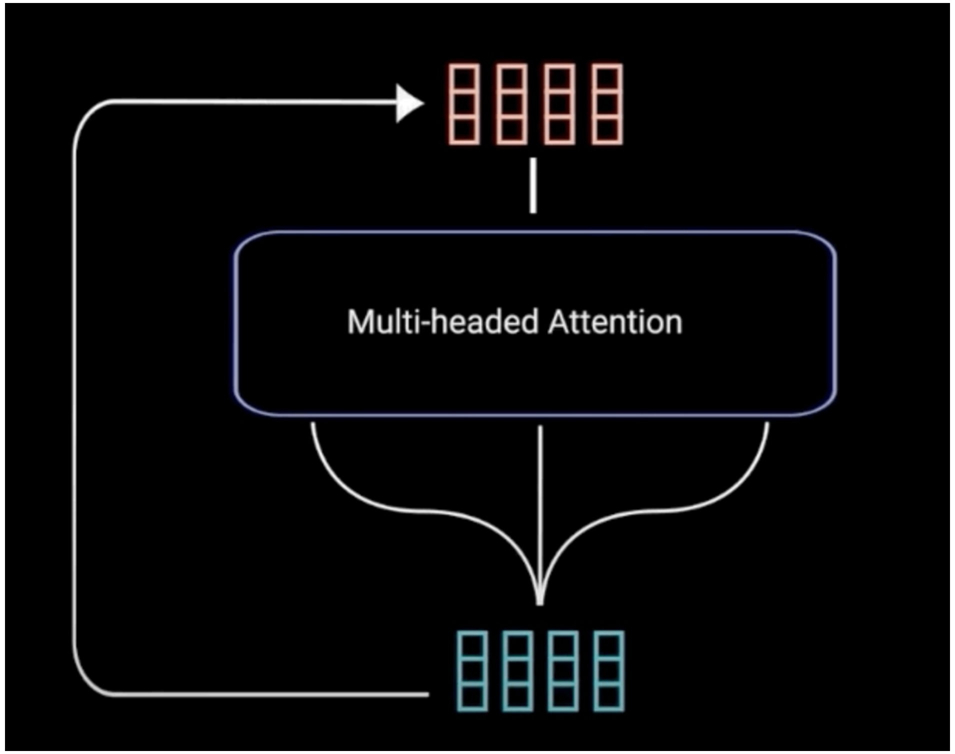
$$\sigma_n^2 = \frac{1}{K}\sum_{k=1}^{K} (x_{nk} - \mu_n)^2$$

$$\hat{x}_{nk} = \frac{x_{nk} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}, \hat{x}_{nk} \in R$$

$$\mathrm{LN}_{\gamma,\beta}(x_n) = \gamma \hat{x}_n + \beta, x_n \in R^K$$

β, γ learnable parameters

Layer normalization + residual connections



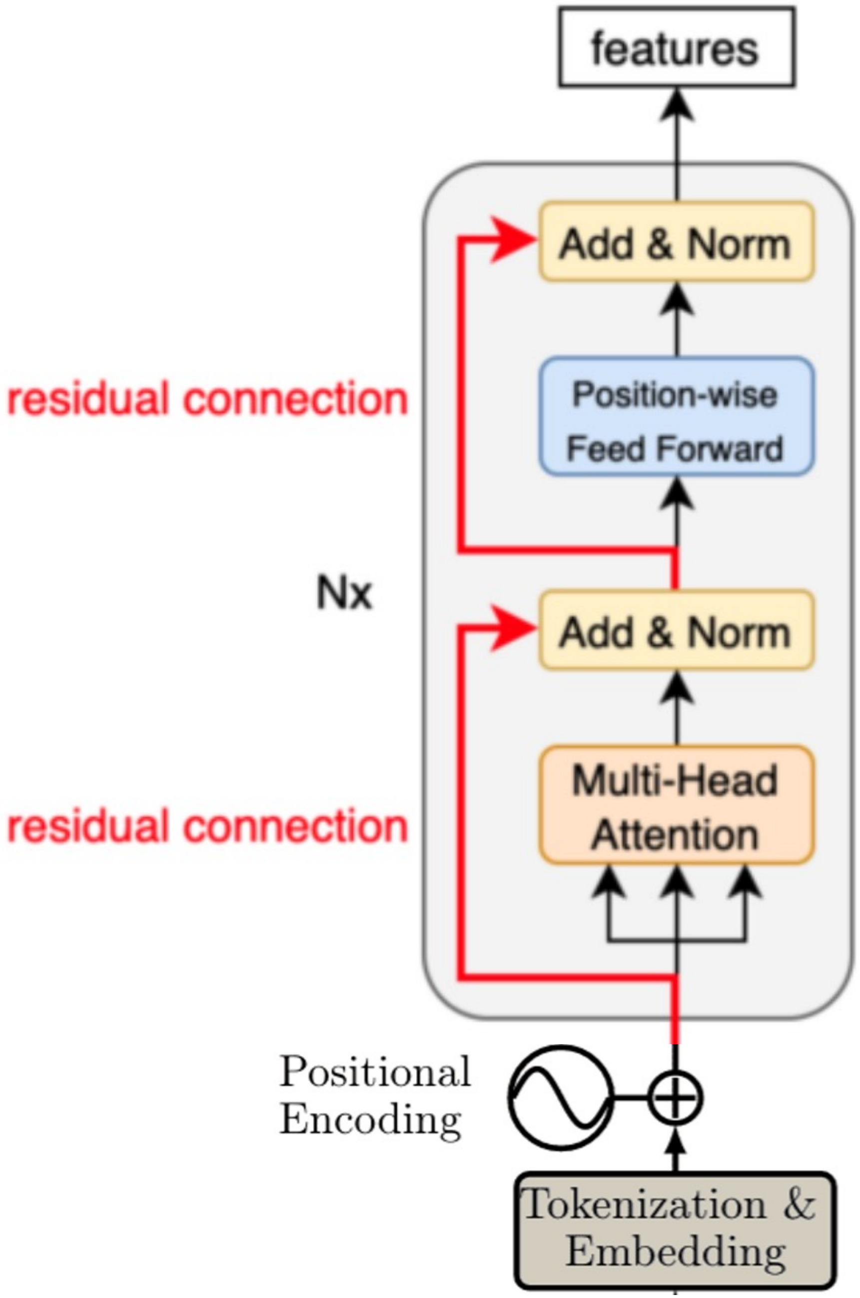
Residual connections

- Better gradient flow (vanishing gradients)
- Leverage input encoding, *e.g.* PE

Feed-Forward Network (FFN)

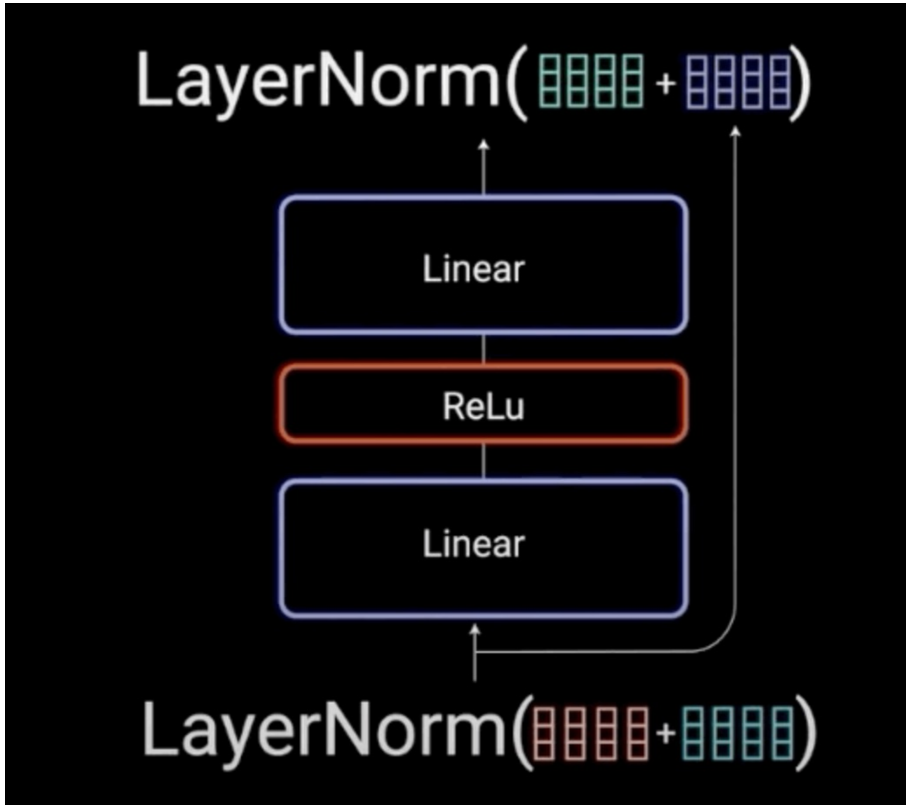
- Position-wise FFN: applied to each token separately and identically

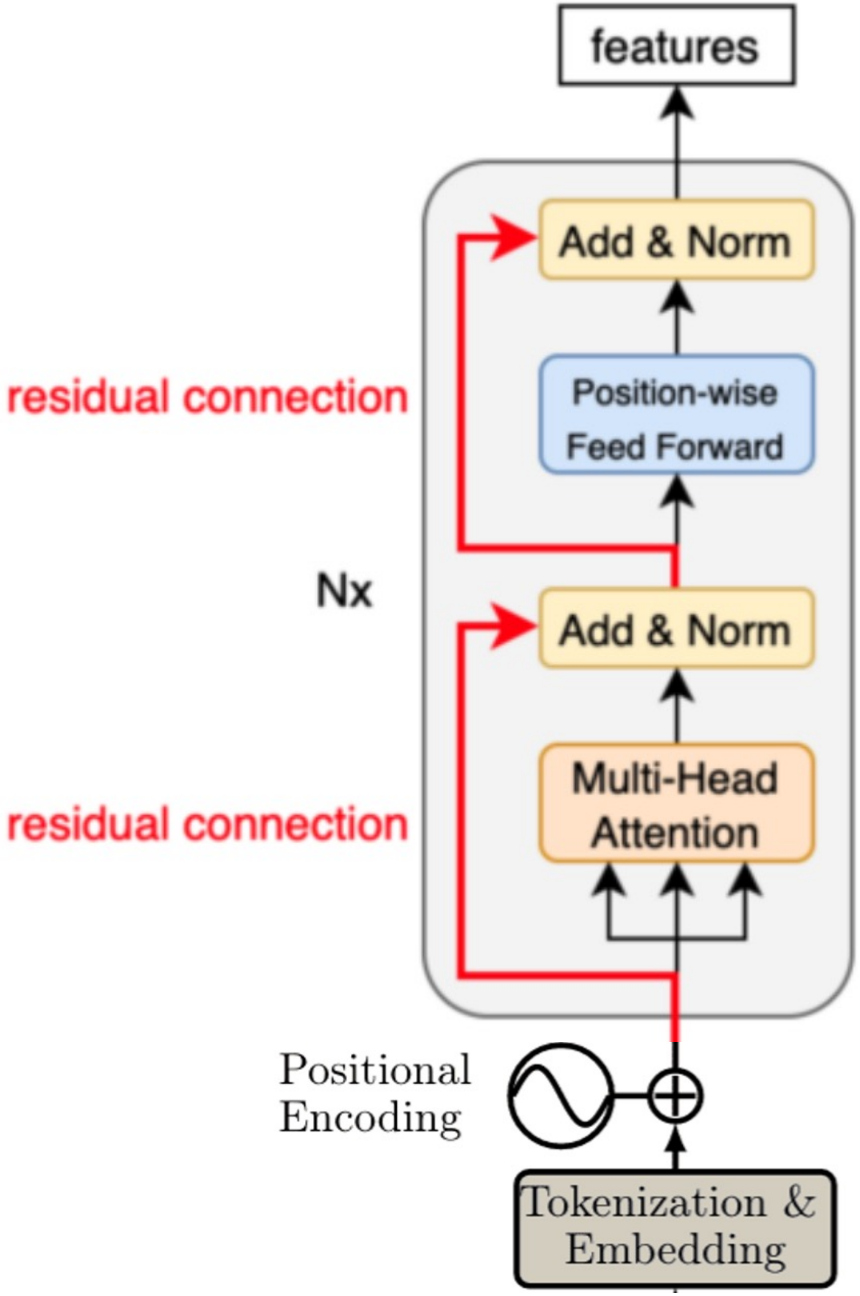
$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$



FNN + residual Layer Norm

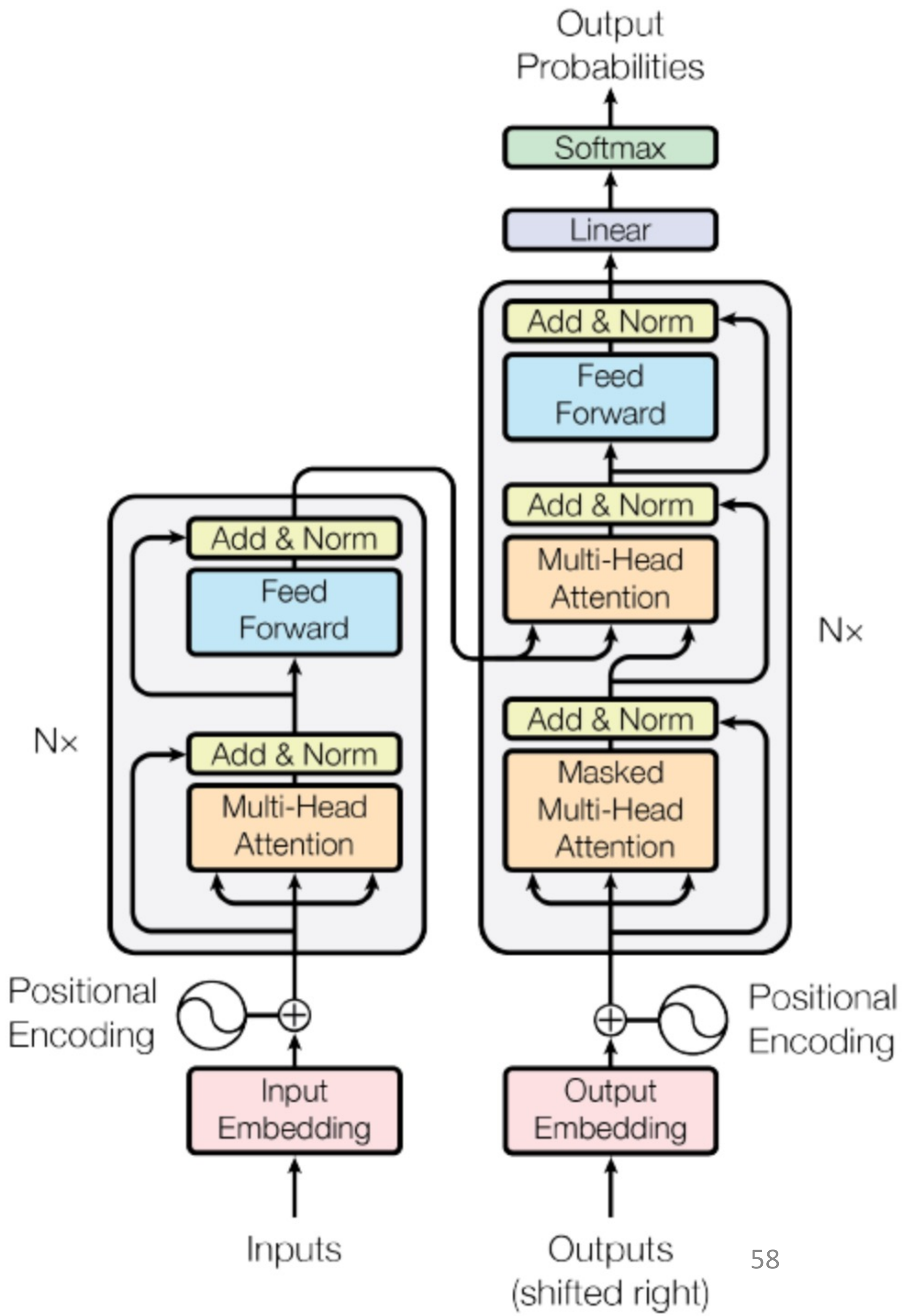
$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$





Transformer: conclusion

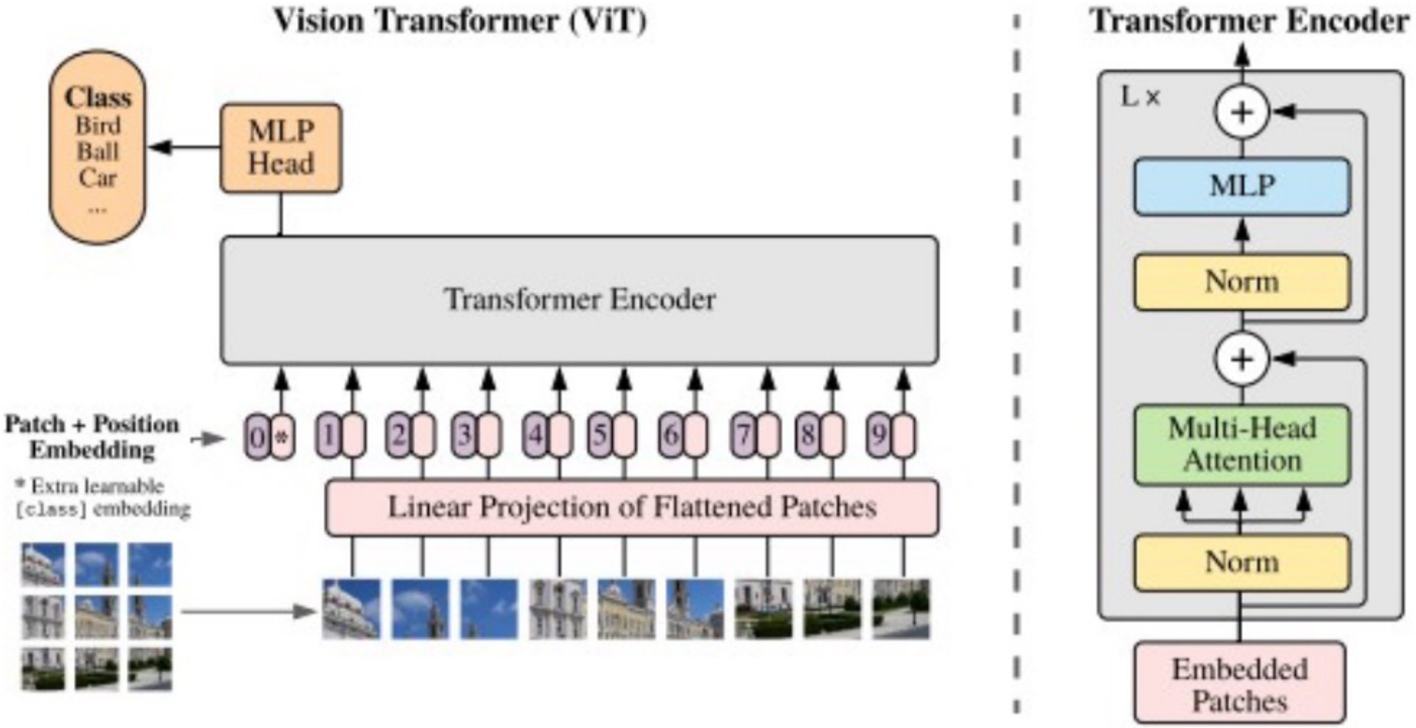
- Importance of attention: global interactions between tokens
- On the other hand relaxes inductive biases
  - e.g. ConvNets translation equivariant
    - vs transformers permutation equivariant
  - More flexibility to learn adequate mapping
  - Needs more data



Outline

I. Recurrent Neural Networks (RNNs)
II. Attention models & transfomers
   a) Context
   b) Transformer block
   c) Medical image segmention

Vision Image Transformer (ViT) [9]



- Direct application of transformer's encoder for images
- Learned on JFT (300.10^6 images)
- Extra learnable token: used for class prediction
  - "Learned" pooling wrt visual tokens

[9] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby. ICLR 2020.

Transformer in segmentation

Vanilla idea: use ViT encoder, add a classification layer for each patch (token) => SETR [10]

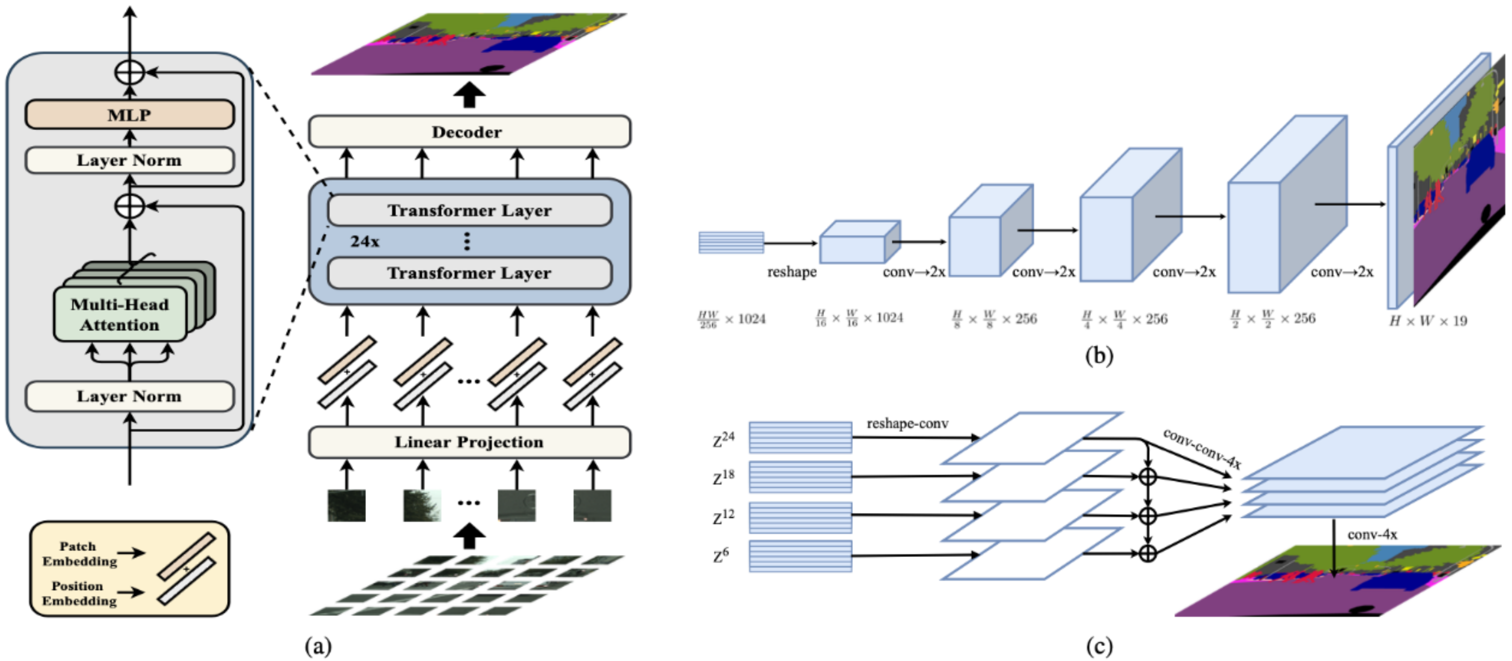
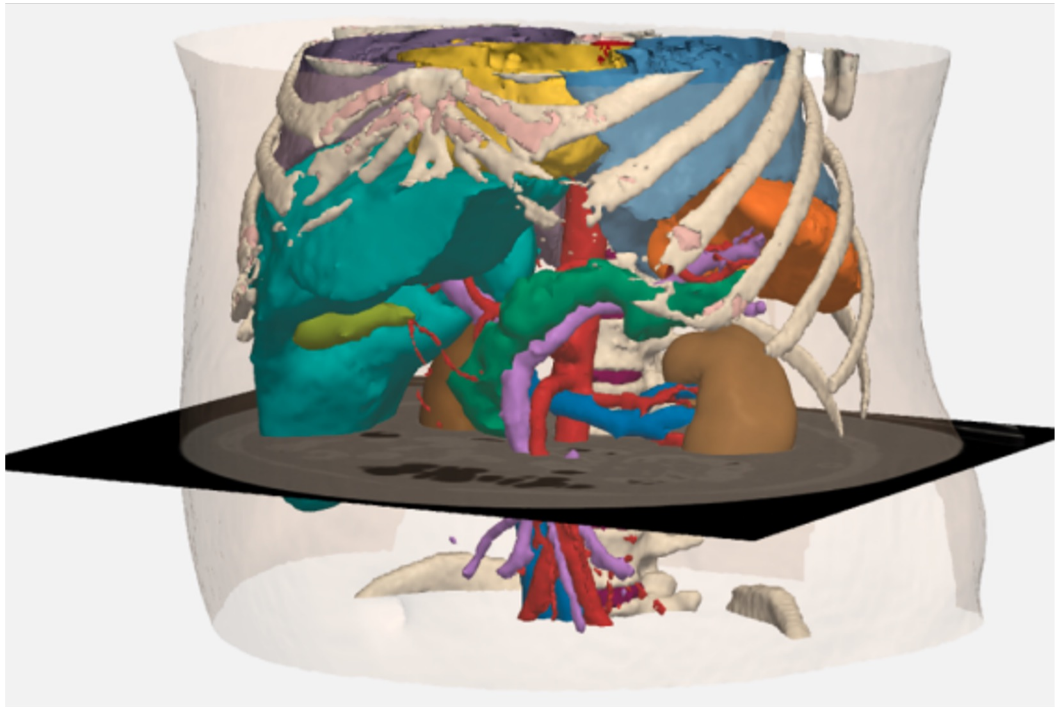


Figure 3: SETR architecture and its variants adapted from [5]. (a) SETR consists of a standard Transformer. (b) *SETR-PUP* with a progressive up-sampling design. (c) *SETR-MLA* with a multi-level feature aggregation.

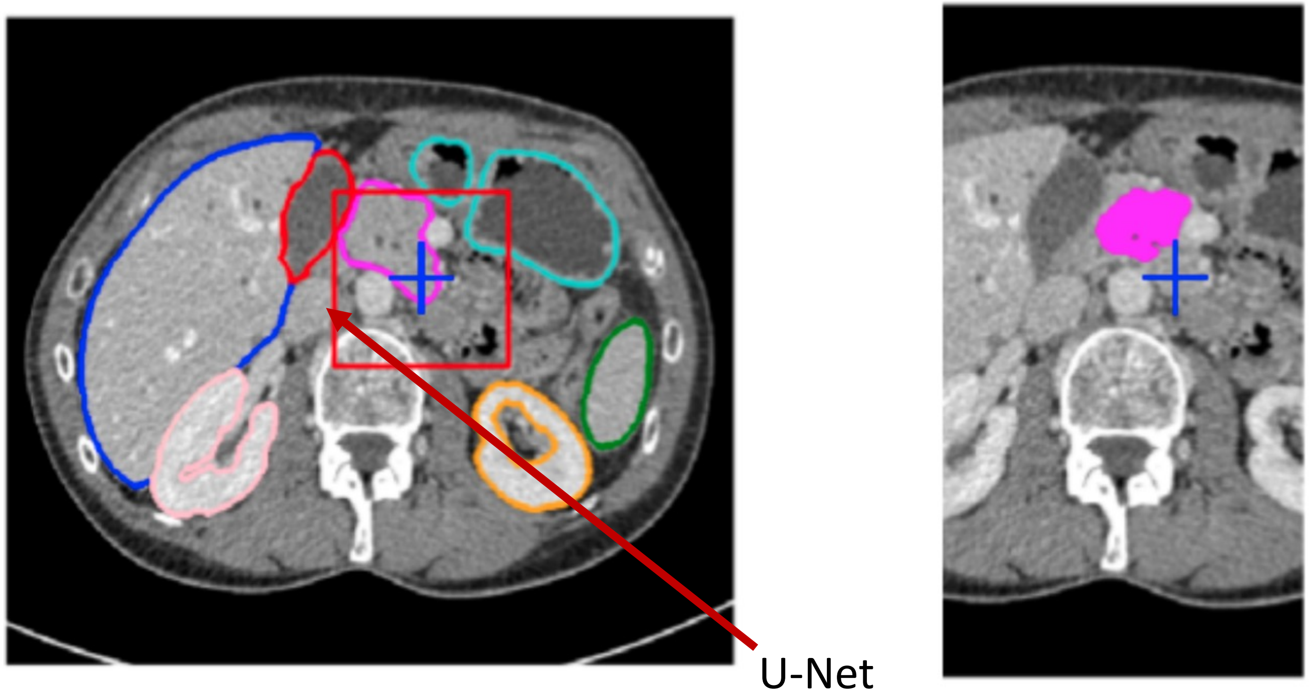
[10] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. CVPR 2021.

Transformer in medical image segmentation

Organ segmentation



Ex: Pancreas segmentation



a) Ground Truth

c) U-Net

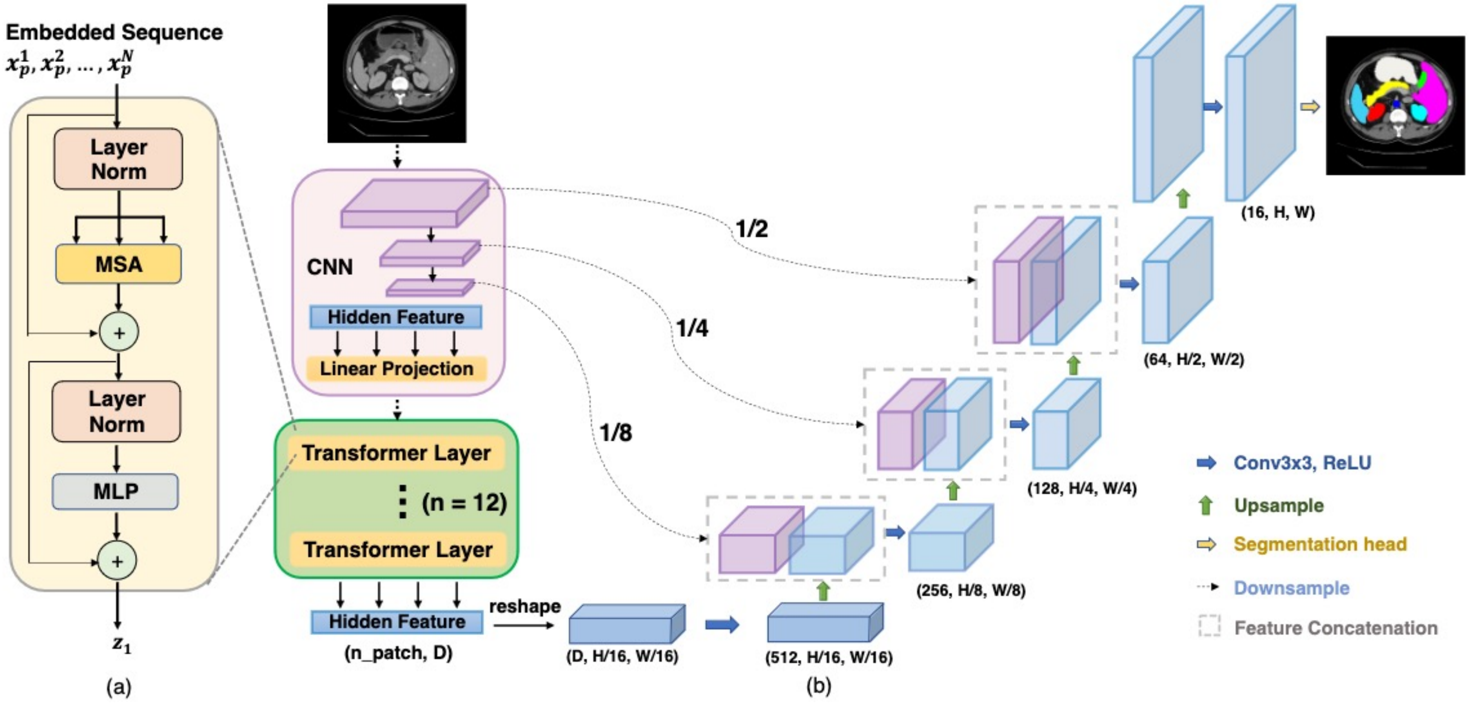
<u>Long-range dependencies:</u> crucial context in segmentation

<u>Main challenge of transformers:</u> self attention complexity - $O(N^2)$

- Expensive (or impossible) for large N => critical for large 2D images, 3D volumes

Hybrid conv / transformers architectures

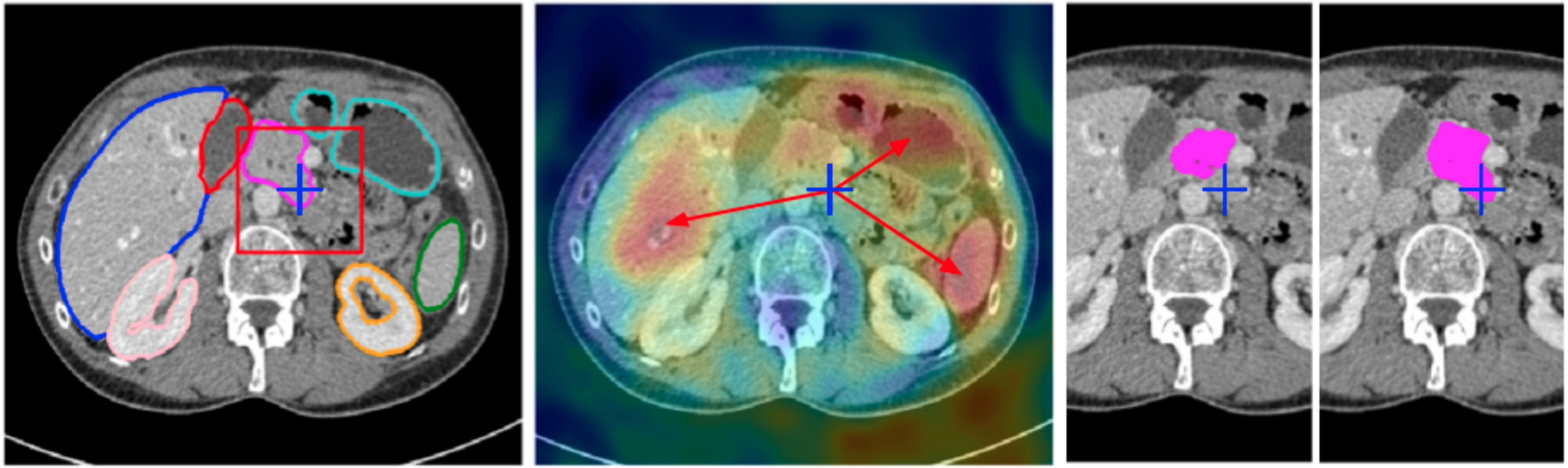
- Trans U-Net [11], U-Transformer [12]: seminal works for using transformers in medical image segmentation
- Adding self-attention on U-Net's bottleneck
  - Inspired from non-local networks [13]



Trans U-Net architecture

[11] TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. J. Chen et.al. arXiv, Feb 2021.
[12] U-Net Transformer: Self and Cross Attention for Medical Image Segmentation. O. Petit, N. Thome, C. Rambour, L. Soler. arXiv, March 2021.
[13] Non-local Neural Networks. X. Wang, R. Girshick, A. Gupta, K. He. CVPR 2018.

U-Transformer [12]



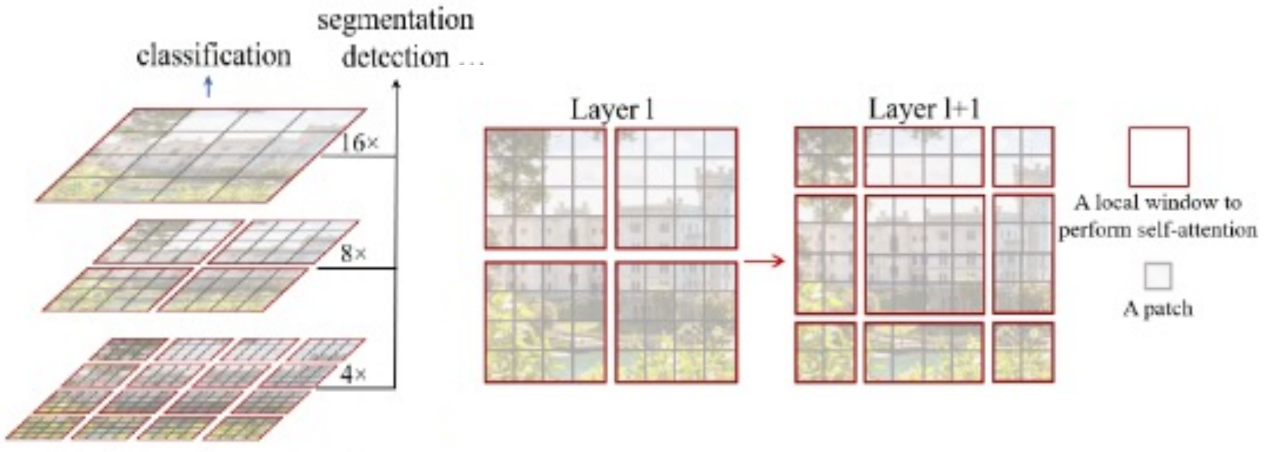
a) Ground Truth b) Attention map c) U-Net d) U-Transformer

[12] U-Net Transformer: Self and Cross Attention for Medical Image Segmentation. O. Petit, N. Thome, C. Rambour, L. Soler. arXiv, March 2021.

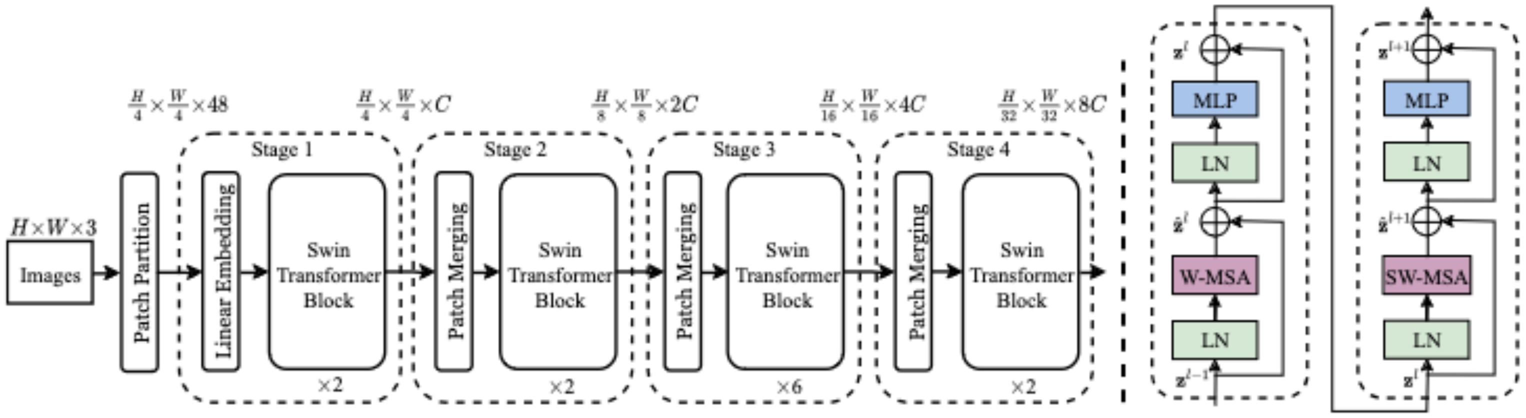
Full transformers in segmentation

Motivation: breaking self-attention complexity

- Swin-Transformer [14]
  - Multi-resolution transformer
    - Local attention in lower-layers
      - Shifted windows at layers l/l+1
    - Patch merging => larger receptive field



(a) Swin Transformer (ours)



(a) Architecture (b) Two Successive Swin Transformer Blocks

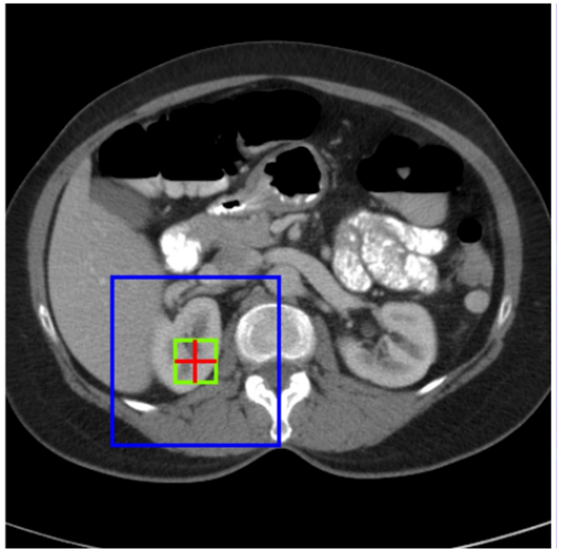
[14] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo. ICCV 2021

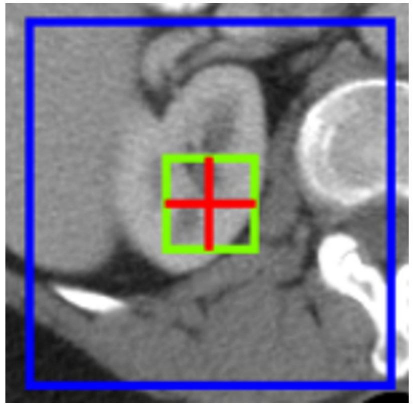
3D medical image segmentation

<u>Challenge: </u>input volume size

•Intractable memory requirements (180Gb for U-Net with image 512x512x256

•**Common strategy:** train on 3 crops



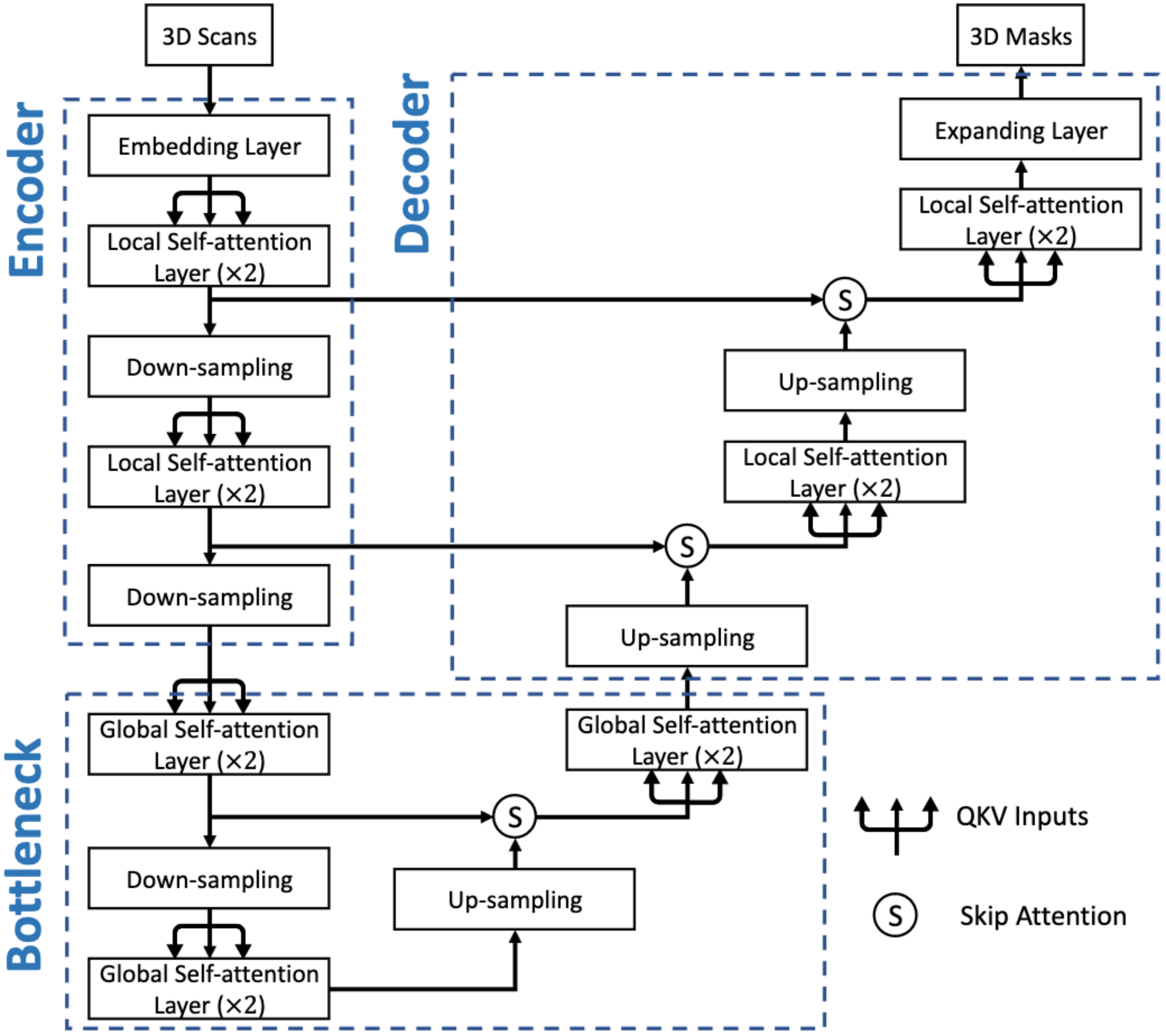


- **Full context lost**
- **Even on patch: full context challenging!**

nn-Former [15]

- Global self-attention in bottleneck
- Local self-attention in higher-resolution feature maps
  - ~ 3D Swin-Unet

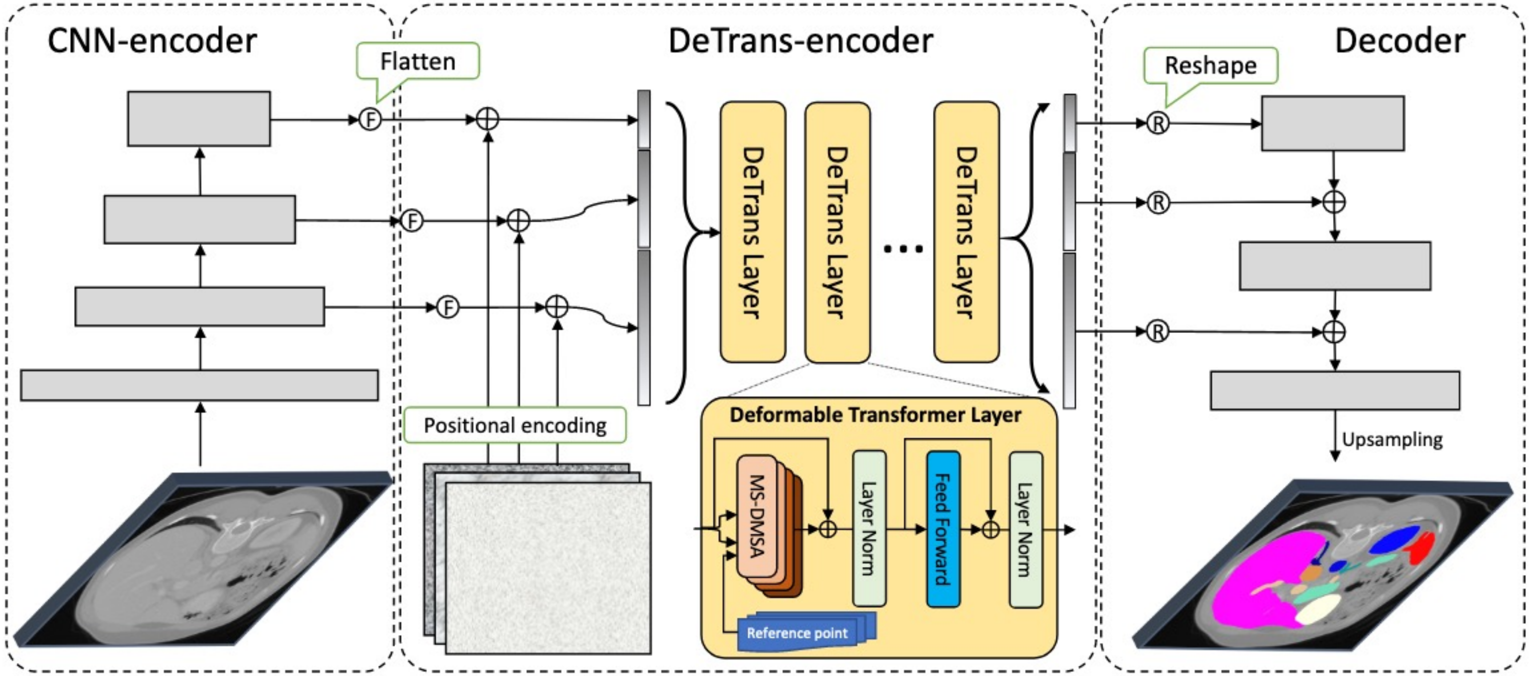
- **No context beyond patch**
- **No global context in high-resolution maps**



[15] nnFormer: Interleaved Transformer for Volumetric Segmentation. H.Y. Zhou, J. Guo, Y. Zhang, L. Yu, L. Wang, Y. Yu. Arxiv, September 2021.

CoTR: Convolutional NN and Transformer [16]

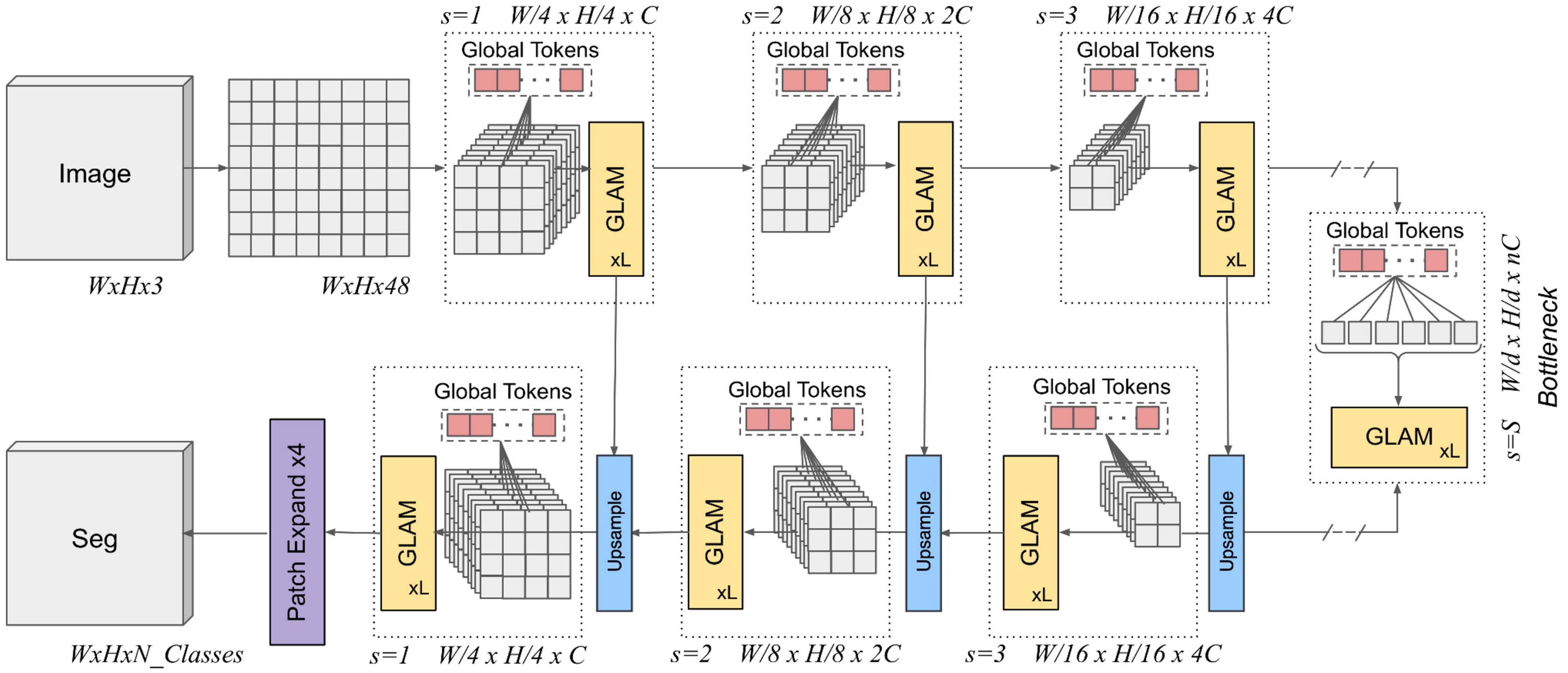
- **CoTr:** Conv encoder => flattened multi-scale feature
  - Deformable transformer encoder (DeTrans) in multi-res input
  - Several DeTrans layers, sent to conv decoder



[16] CoTr: Efficiently Bridging CNN and Transformer for 3D Medical Image Segmentation. Y. Xie, J. Zhang, C. Shen, Y. Xia. MICCAI 2021

Global attention in multi-resolution transformers (GLAM) [17]

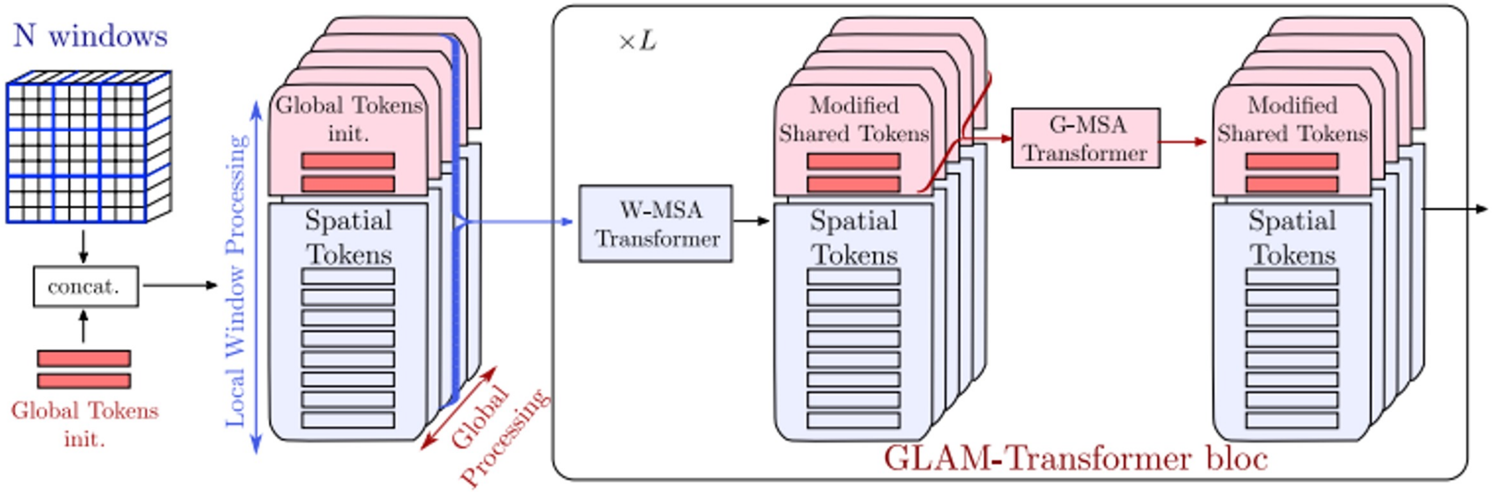
- GLAM block in any multi-resolution architecture (e.g. Swin, nn-Former)



[17] Full Contextual Attention for Multi-resolution Transformers in Semantic Segmentation. L. Themyr, C. Rambour, N. Thome, T. Collins, A. Hostettler. WACV 2023.

GLAM block

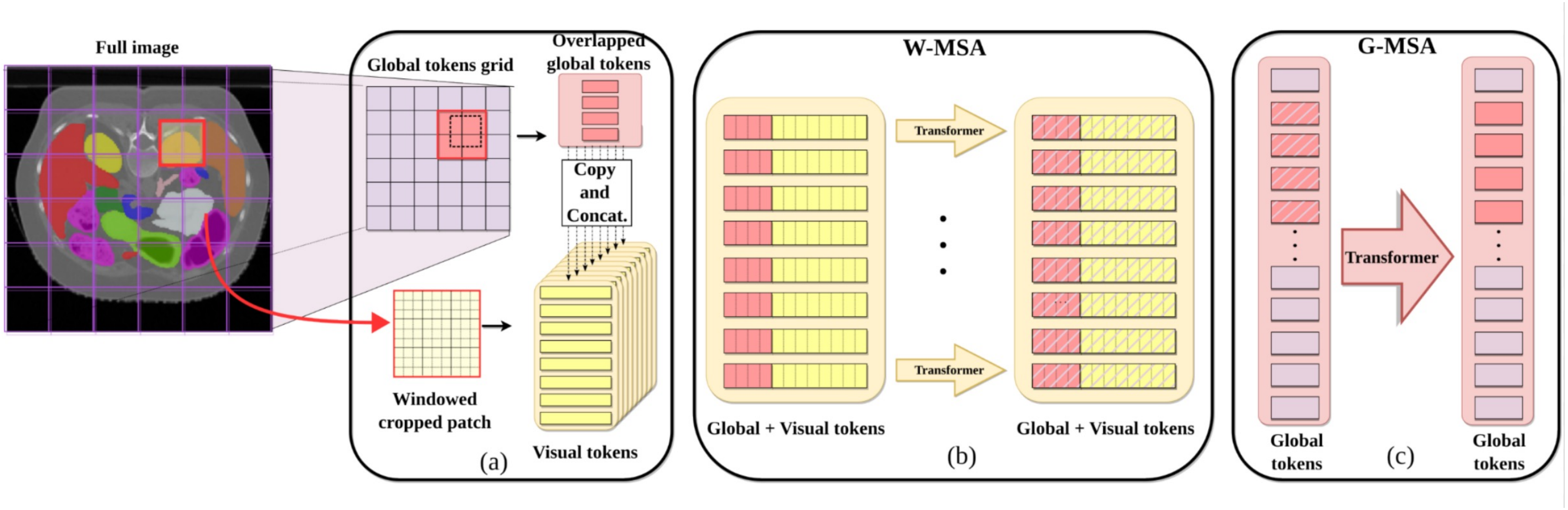
- Define learnable global tokens in each window, cf CLS in VIT
  - Window self-attention (W-MSA): attention between visual and global tokens
  - Global attention (G-MSA) between global token
- G-MSA: indirection between all visual tokens
  - Break computational complexity of full attention between visual token
  - But **enables full indirect interaction between them**



=> full attention even in high-resolution features!

FINE : Full resolutIoN mEmory transformer [18]

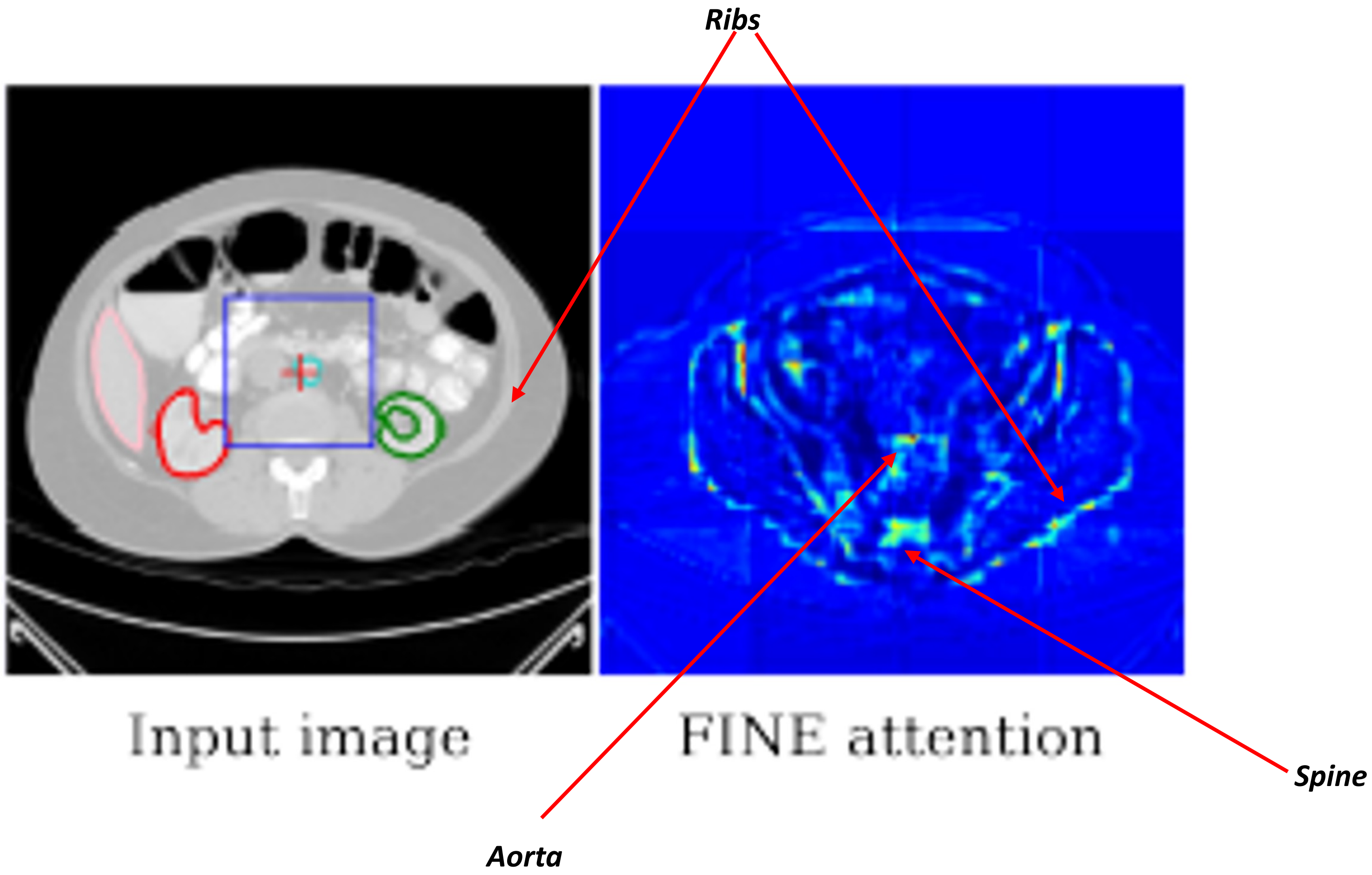
- Extends GLAM for full context modelling in 3D segmentation
  - Global tokens for the full volume



=> (indirect) **full interaction between all voxels!**

[18] Memory transformers for full context and high-resolution 3D Medical Segmentation. L. Themyr, C. Rambour, N. Thome, T. Collins, A. Hostettler. MLMI workshop, MICCAI 2022.

FINE results



Thank you for your attention!

Questions?