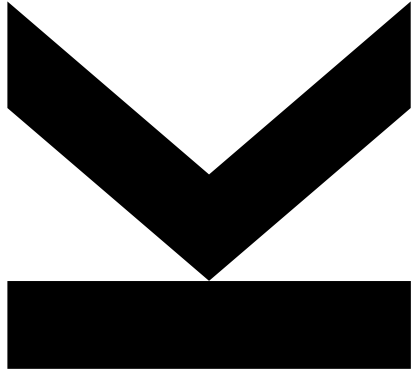


xLSTM

A Recurrent Large Language Model



PhD Seminar Talk

Maximilian Beck, beck@ml.jku.at,  [maxmbeck](https://twitter.com/maxmbeck)

Joint work with Korbinian Pöppel, Markus Spanring, Oleksandra Prudnikova, Michael Kopp, Johannes Brandstetter and Sepp Hochreiter

Institute for Machine Learning, October 2023

Motivation

- Successes of Transformer with Self-Attention:
 - ChatGPT, Diffusion Models, Computer Vision, Climate Modelling, ...
- Drawbacks of Self-Attention:
 - Scales quadratically with sequence length
 - Very expensive (ChatGPT could cost 700k \$ per day [1])
- Many have tried to improve Self-Attention:
 - Approximation based: Approximating the softmax or the attention matrix
 - State Space Models based: H3, Hyena
 - RWKV: Linear recurrence
- Our work: xLSTM

Outline

- The competitor: Self-Attention and the Transformer architecture
- The contenders: xLSTM Family
- The competitions: Outline of the Experiments

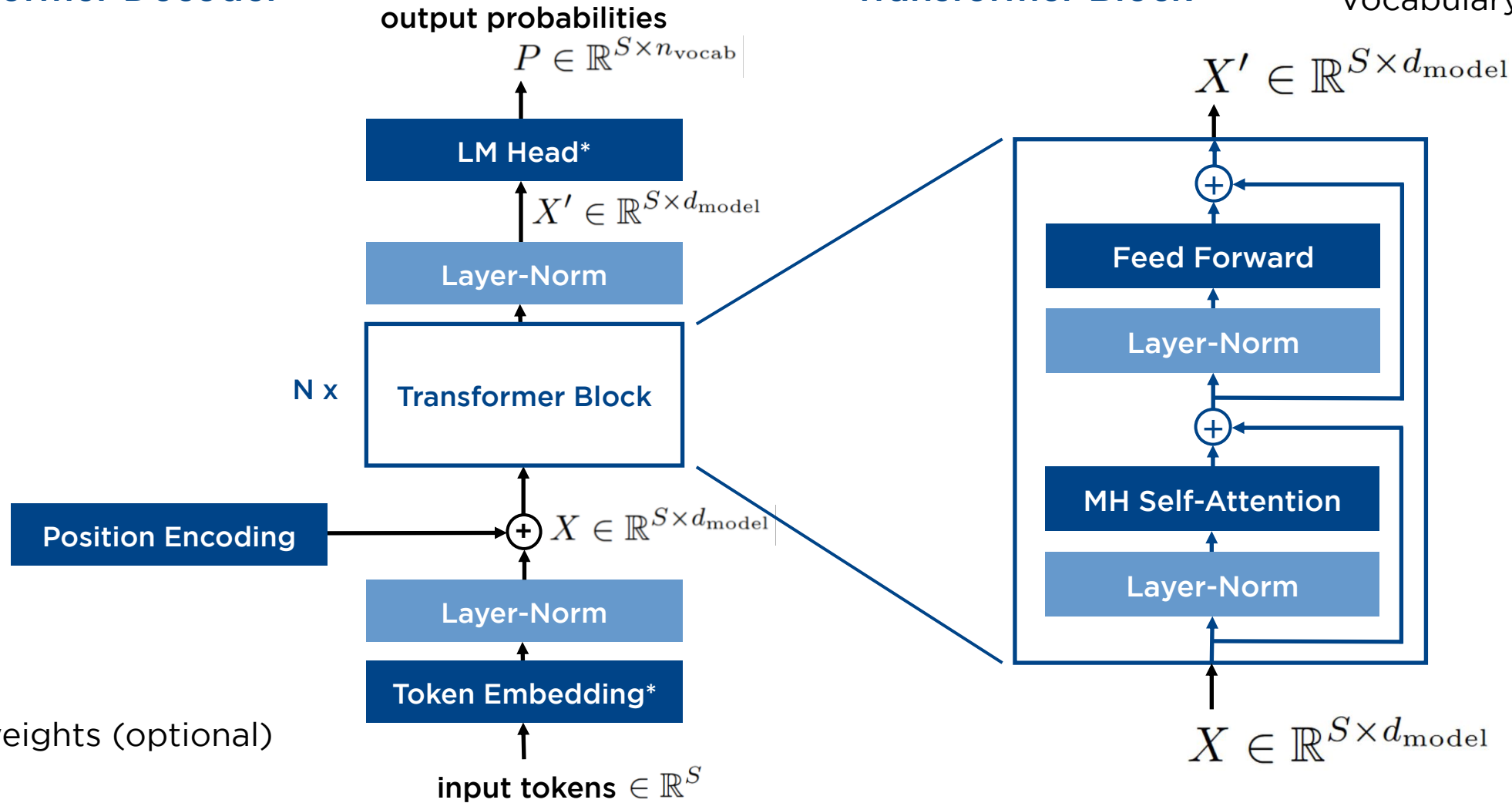
The competitor: Self-Attention and the Transformer architecture

Transformer Decoder Only Architecture

Sequence length: S
Embedding dim.: d_{model}
Vocabulary size: n_{vocab}

Transformer Decoder

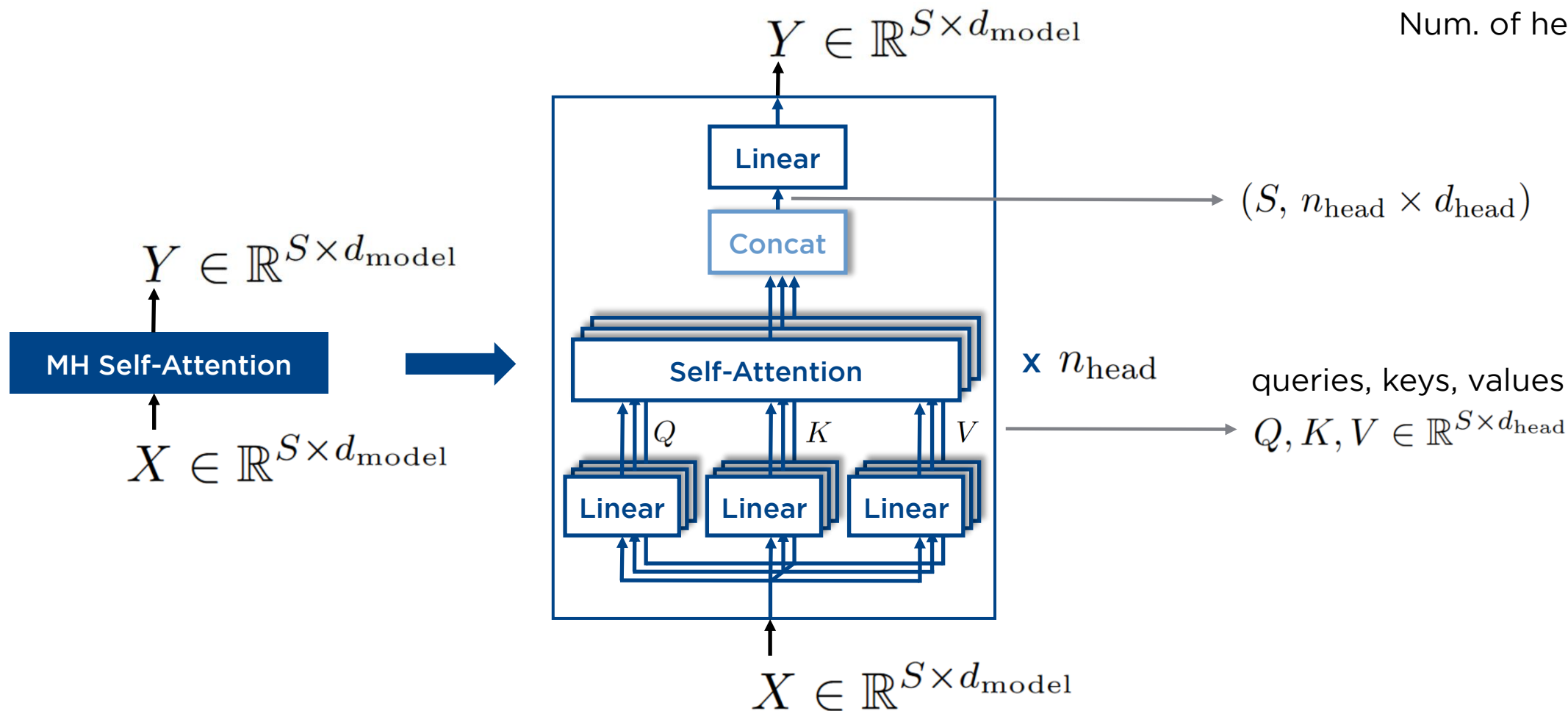
Transformer Block



*tied weights (optional)

Multi-Head Attention

Sequence length: S
Embedding dim.: d_{model}
Head dimension: d_{head}
Num. of heads: n_{head}



Self-Attention

$$\text{SelfAttention}(Q, K, V) = \left(\text{softmax} \left(\frac{QK^\top}{\sqrt{d_{\text{head}}}} \right) \odot D \right) V$$

Sequence length: S

Head dimension: d_{head}

$$Q, K, V \in \mathbb{R}^{S \times d_{\text{head}}}$$

$$D = \begin{pmatrix} 1 & 0 & \dots \\ 1 & 1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \in \mathbb{R}^{S \times S}$$

$$q_t, k_t, v_t \in \mathbb{R}^{d_{\text{head}}}$$

$$\begin{aligned} \cancel{\text{softmax}}(QK^\top) V &= \cancel{\text{softmax}} \left(\begin{pmatrix} q_1^\top \\ \vdots \\ q_S^\top \end{pmatrix} \begin{pmatrix} k_1 & \dots & k_S \end{pmatrix} \right) \begin{pmatrix} v_1^\top \\ \vdots \\ v_S^\top \end{pmatrix} \\ &= \cancel{\text{softmax}} \begin{pmatrix} q_1^\top k_1 & q_1^\top k_2 & \dots & q_1^\top k_S \\ q_2^\top k_1 & q_2^\top k_2 & \dots & q_2^\top k_S \\ \vdots & \vdots & \ddots & \vdots \\ q_S^\top k_1 & q_S^\top k_2 & \dots & q_S^\top k_S \end{pmatrix} \begin{pmatrix} v_1^\top \\ v_2^\top \\ \vdots \\ v_S^\top \end{pmatrix} \end{aligned}$$

=

C

$(S \times S)$

combination matrix

V

$(S \times d_{\text{head}})$

values

The QKV Product

$$\begin{aligned} V' &= (Q K^\top) V \\ &= \left(\begin{pmatrix} q_1^\top \\ \vdots \\ q_S^\top \end{pmatrix} (k_1 \quad \cdots \quad k_S) \right) \begin{pmatrix} v_1^\top \\ \vdots \\ v_S^\top \end{pmatrix} \\ &= \begin{pmatrix} q_1^\top k_1 & q_1^\top k_2 & \cdots & q_1^\top k_S \\ q_2^\top k_1 & q_2^\top k_2 & \cdots & q_2^\top k_S \\ \vdots & \vdots & \ddots & \vdots \\ q_S^\top k_1 & q_S^\top k_2 & \cdots & q_S^\top k_S \end{pmatrix} \begin{pmatrix} v_1^\top \\ v_2^\top \\ \vdots \\ v_S^\top \end{pmatrix} \end{aligned}$$

$$\begin{pmatrix} v_1'^\top \\ v_2'^\top \\ \vdots \\ v_S'^\top \end{pmatrix} = \begin{pmatrix} (q_1^\top k_1)v_1^\top + (q_1^\top k_2)v_2^\top + \cdots + (q_1^\top k_S)v_S^\top \\ (q_2^\top k_1)v_1^\top + (q_2^\top k_2)v_2^\top + \cdots + (q_2^\top k_S)v_S^\top \\ \vdots \\ (q_S^\top k_1)v_1^\top + (q_S^\top k_2)v_2^\top + \cdots + (q_S^\top k_S)v_S^\top \end{pmatrix}$$

Sequence length: S

Head dimension: d_{head}

$$Q, K, V \in \mathbb{R}^{S \times d_{\text{head}}}$$

$$D = \begin{pmatrix} 1 & 0 & \cdots \\ 1 & 1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \in \mathbb{R}^{S \times S}$$

$$q_t, k_t, v_t \in \mathbb{R}^{d_{\text{head}}}$$

With causal masking:

$$\longrightarrow V' = (Q K^\top \odot D) V$$

The QKDV Product

Sequence length: S

Head dimension: d_{head}

$$Q, K, V \in \mathbb{R}^{S \times d_{\text{head}}}$$

$$D = \begin{pmatrix} 1 & 0 & \dots \\ 1 & 1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \in \mathbb{R}^{S \times S}$$

$$q_t, k_t, v_t \in \mathbb{R}^{d_{\text{head}}}$$

$$\begin{aligned} V' &= (Q K^\top \odot D) V \\ \begin{pmatrix} v_1'^\top \\ v_2'^\top \\ \vdots \\ v_S'^\top \end{pmatrix} &= \begin{pmatrix} (q_1^\top k_1) v_1^\top \\ (q_2^\top k_1) v_1^\top + (q_2^\top k_2) v_2^\top \\ \vdots \\ (q_S^\top k_1) v_1^\top + (q_S^\top k_2) v_2^\top + \dots + (q_S^\top k_S) v_S^\top \end{pmatrix} \\ &= \begin{pmatrix} q_1^\top (k_1 v_1^\top) \\ q_2^\top (k_1 v_1^\top + k_2 v_2^\top) \\ \vdots \\ q_S^\top (k_1 v_1^\top + k_2 v_2^\top + \dots + k_S v_S^\top) \end{pmatrix} \end{aligned}$$

**Associative property
of matrix multiplication:**

$$(q_i^\top k_j) v_j^\top = q_i^\top (k_j v_j^\top)$$

**This is one core idea
of Linear Attention!**

 **Compact form for every time step:**

$$v_t'^\top = q_t^\top \sum_{j=1}^t k_j v_j^\top$$

The Softmax

For a vector $x = [x_1, x_2, \dots, x_{d-1}, x_d] \in \mathbb{R}^d$

we can compute the softmax with

$$(\text{softmax}(x))_i = \frac{e^{x_i}}{e^{x_1} + e^{x_2} + \dots + e^{x_d}} = \frac{e^{x_i}}{n}$$

QKVD Product with Softmax (i.e. Self-Attention):

$$v_t'^{\top} = \sum_{i=1}^t \frac{e^{q_t^{\top} k_i} v_i^{\top}}{e^{q_t^{\top} k_1} + e^{q_t^{\top} k_2} + \dots + e^{q_t^{\top} k_t}} = \sum_{i=1}^t \frac{e^{q_t^{\top} k_i}}{\sum_{j=1}^t e^{q_t^{\top} k_j}} v_i^{\top}$$

Sequence length: S

Head dimension: d_{head}

queries, keys, values

$q_t, k_t, v_t \in \mathbb{R}^{d_{\text{head}}}$

The Softmax

For a vector $x = [x_1, x_2, \dots, x_{d-1}, x_d] \in \mathbb{R}^d$

we can compute the softmax with

$$(\text{softmax}(x))_i = \frac{e^{x_i}}{e^{x_1} + e^{x_2} + \dots + e^{x_d}} = \frac{e^{x_i}}{n}$$

Not numerically stable!



Numerically stable softmax:

$$(\text{softmax}(x))_i = \frac{e^{x_i - \max_{k=1}^d x_k}}{e^{x_1 - \max_{k=1}^d x_k} + \dots + e^{x_d - \max_{k=1}^d x_k}} = \frac{e^{x_i - m}}{n}$$

Numerically stable!



Stable softmax requires three passes over the vector x !
(1. Computing m , 2. Computing n , 3. Computing $\text{softmax}(x)$)

The “Online Softmax”

Numerically stable softmax:

vector: $x = [x_1, x_2, \dots, x_{d-1}, x_d] \in \mathbb{R}^d$

$$(\text{softmax}(x))_i = \frac{e^{x_i - \max_{k=1}^d x_k}}{e^{x_1 - \max_{k=1}^d x_k} + \dots + e^{x_d - \max_{k=1}^d x_k}} = \frac{e^{x_i - m}}{n}$$

Algorithm 1 ”Online Softmax”

```
1: procedure ONLINENORMALIZERsoftmax( $x$ )
2:    $m_0 = -\inf$ 
3:    $n_0 = 0$ 
4:   for  $i = 1$  to  $d$  do
5:      $m_i = \max(m_{i-1}, x_i)$ 
6:      $n_i = n_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$ 
7:   end for
8:   for  $i = 1$  to  $d$  do
9:      $(\text{softmax}(x))_i = \frac{e^{x_i - m_d}}{n_d}$ 
10:  end for
11: end procedure
```

Online Softmax requires only 2 passes over x !

Can also be used for parallel computation of the softmax on a GPU:

$$m = \max(m_A, m_B),$$

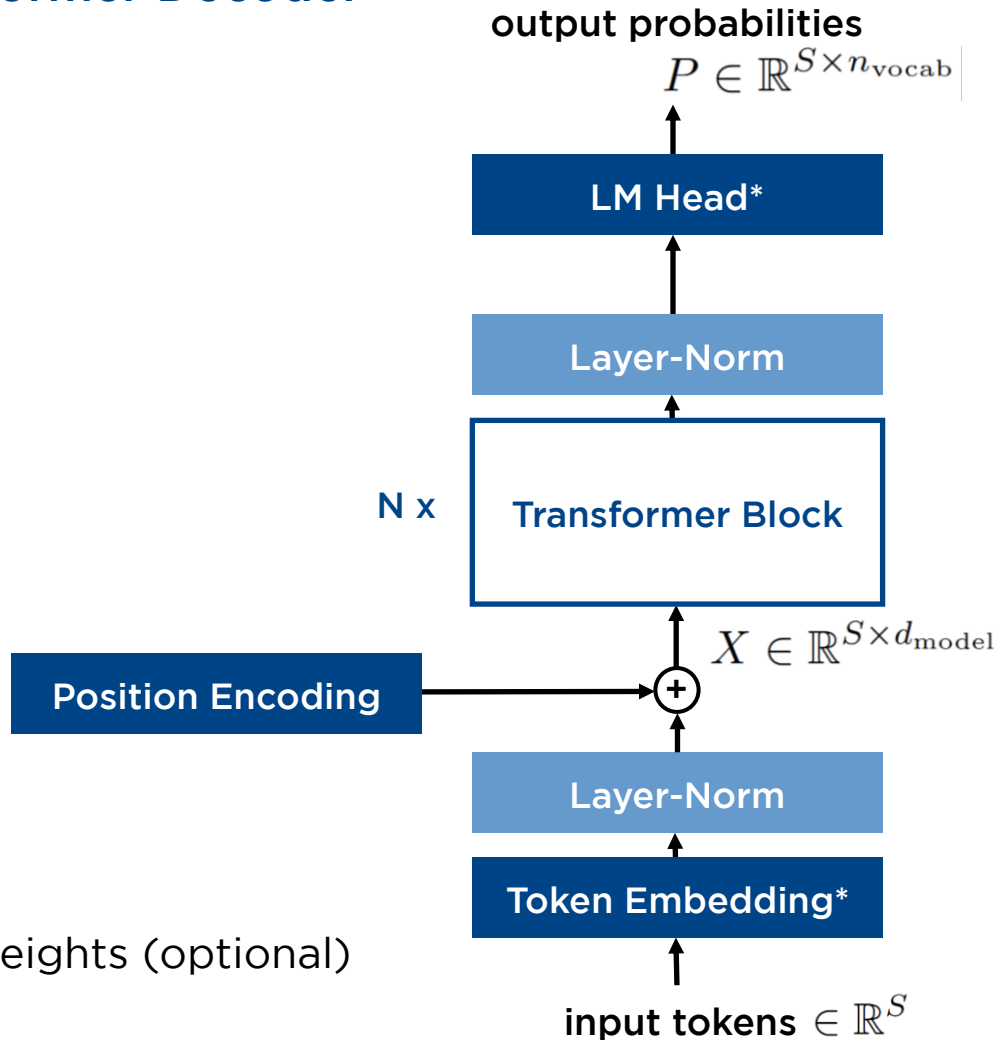
$$n = n_A \cdot e^{m_A - \max(m_A, m_B)} + n_B \cdot e^{m_B - \max(m_A, m_B)}$$

- This trick is used in Flash Attention!
- We also use it for the xLSTM!

The contenders: xLSTM Family

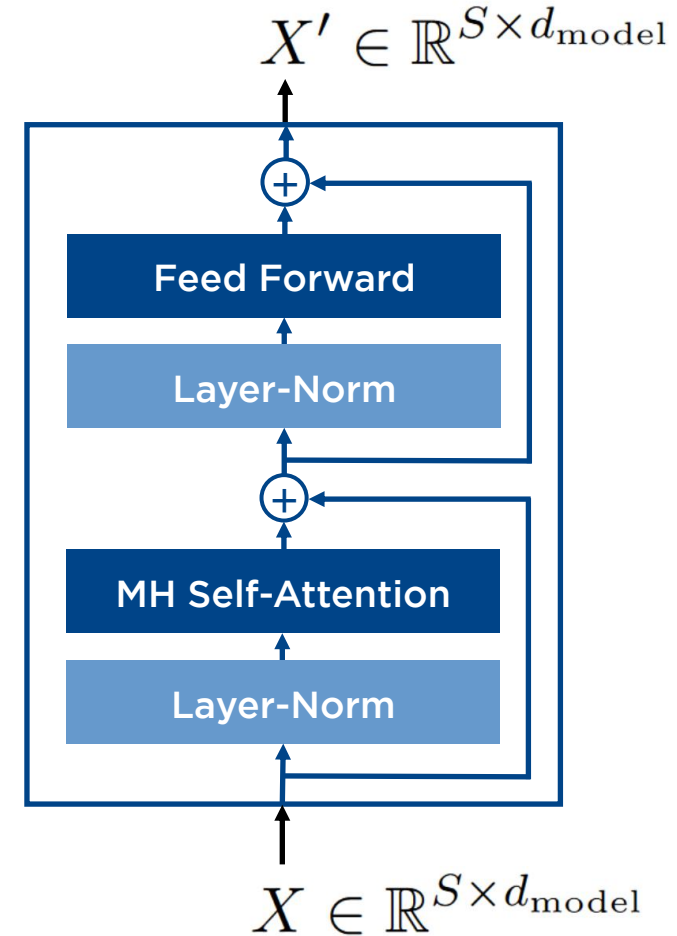
Transformer Decoder Only Architecture

Transformer Decoder



*tied weights (optional)

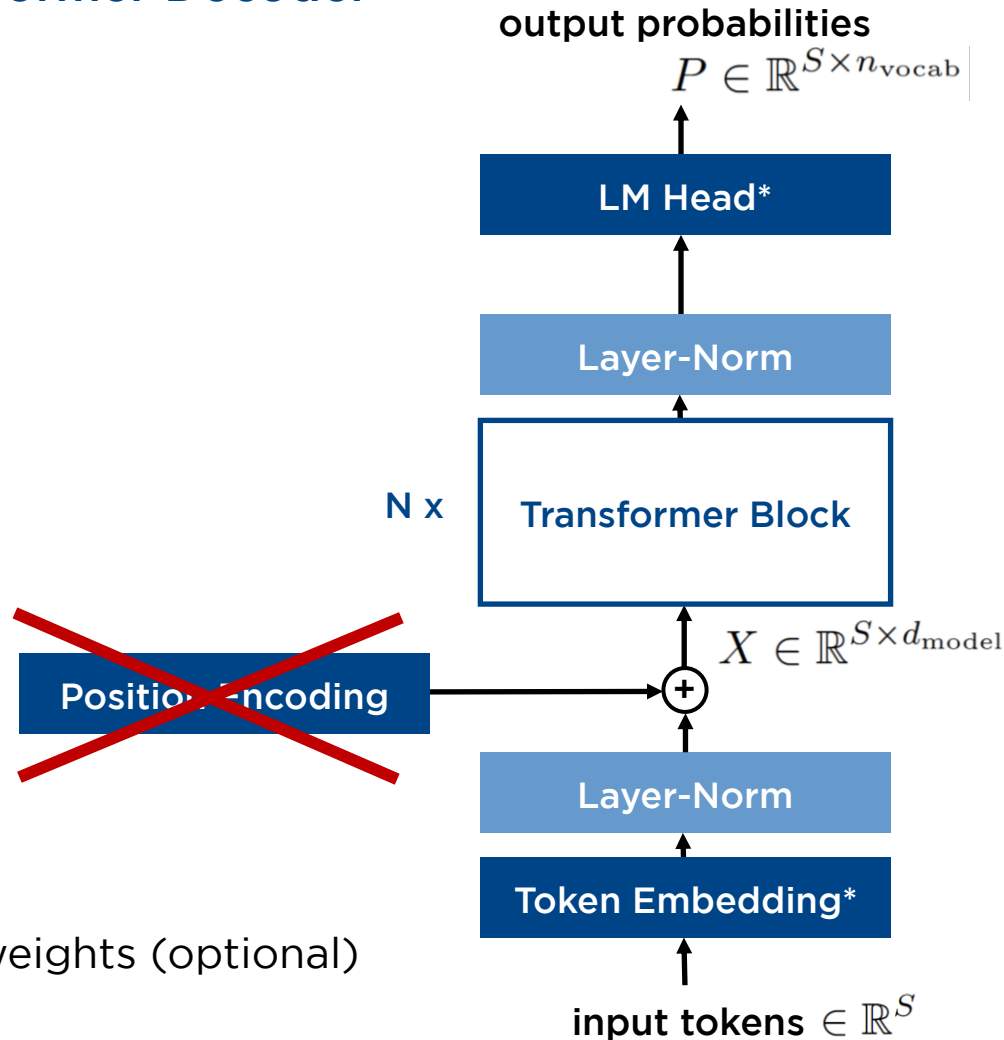
Transformer Block



Sequence length: S
Embedding dim.: d_{model}
Vocabulary size: n_{vocab}

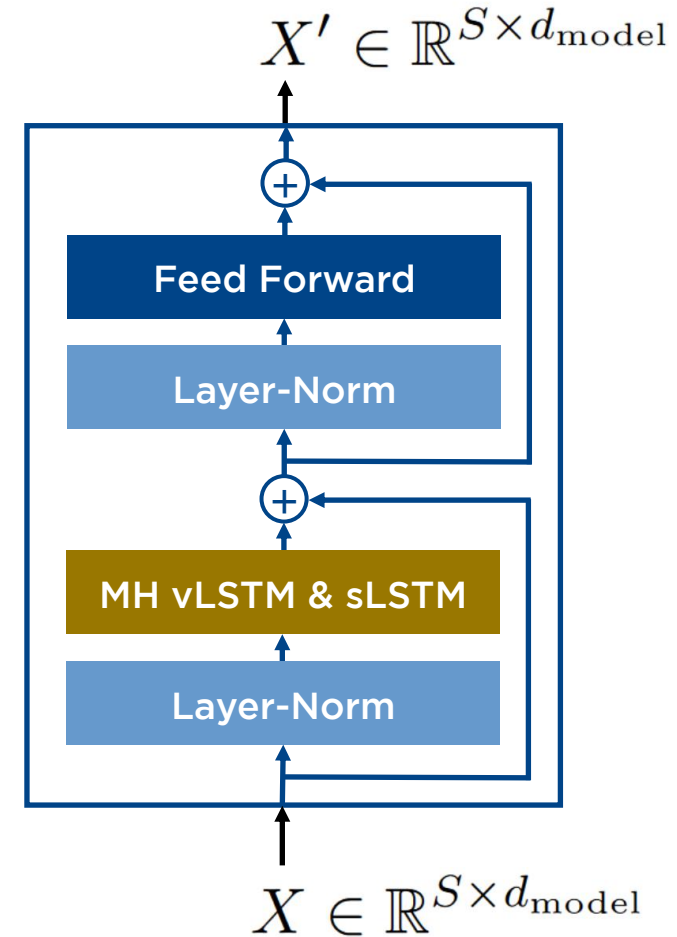
Transformer Decoder Only Architecture

Transformer Decoder



*tied weights (optional)

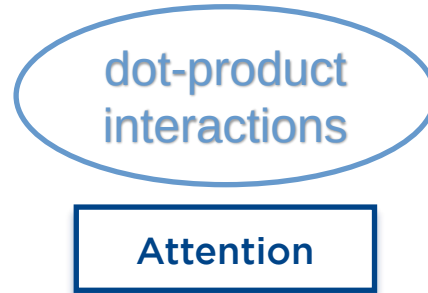
Transformer Block



Sequence length: S
Embedding dim.: d_{model}
Vocabulary size: n_{vocab}

The xLSTM Family Tree

The xLSTM Family Tree has two branches:



The original LSTM

Input / Embedding dim.: d_{model}

Hidden / Head dimension: d_{head}

Four Gates:

forget gate $f_t \in \mathbb{R}^{d_{\text{head}}}$

input gate $i_t \in \mathbb{R}^{d_{\text{head}}}$

cell gate $z_t \in \mathbb{R}^{d_{\text{head}}}$

output gate $o_t \in \mathbb{R}^{d_{\text{head}}}$

Input & Recurrent weights:

$W_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$ for $g \in \{f, i, z, o\}$

$R_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

$b_g \in \mathbb{R}^{d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

Two States:

memory cell state: $c_t \in \mathbb{R}^{d_{\text{head}}}$

hidden state: $h_t \in \mathbb{R}^{d_{\text{head}}}$

$$g_t = x_t W_g^\top + h_t R_g^\top + b_g \quad \text{for } g \in \{f, i, z, o\} \in \mathbb{R}^{d_{\text{head}}}$$

Forward updates:

$$c_{t+1} = \sigma(f_t) \odot c_t + \sigma(i_t) \odot \tanh(z_t)$$

$$h_{t+1} = \sigma(o_t) \odot \tanh(c_{t+1})$$

$$\tilde{f}_t = \sigma(f_t), \quad \tilde{i}_t = \sigma(i_t)$$

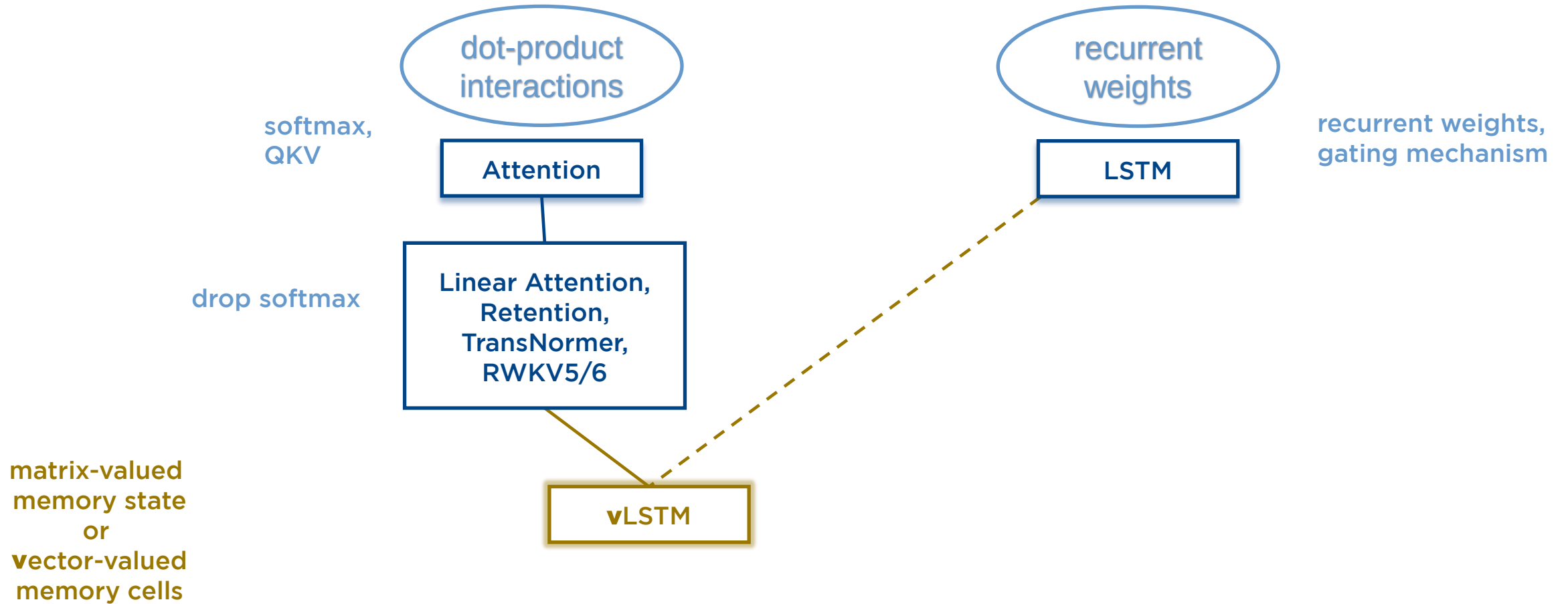


$$\tilde{z}_t = \tanh(z_t), \quad \tilde{o}_t = \sigma(o_t)$$

$$c_{t+1} = \tilde{f}_t \odot c_t + \tilde{i}_t \odot \tilde{z}_t$$

$$h_{t+1} = \tilde{o}_t \odot \tanh(c_{t+1})$$

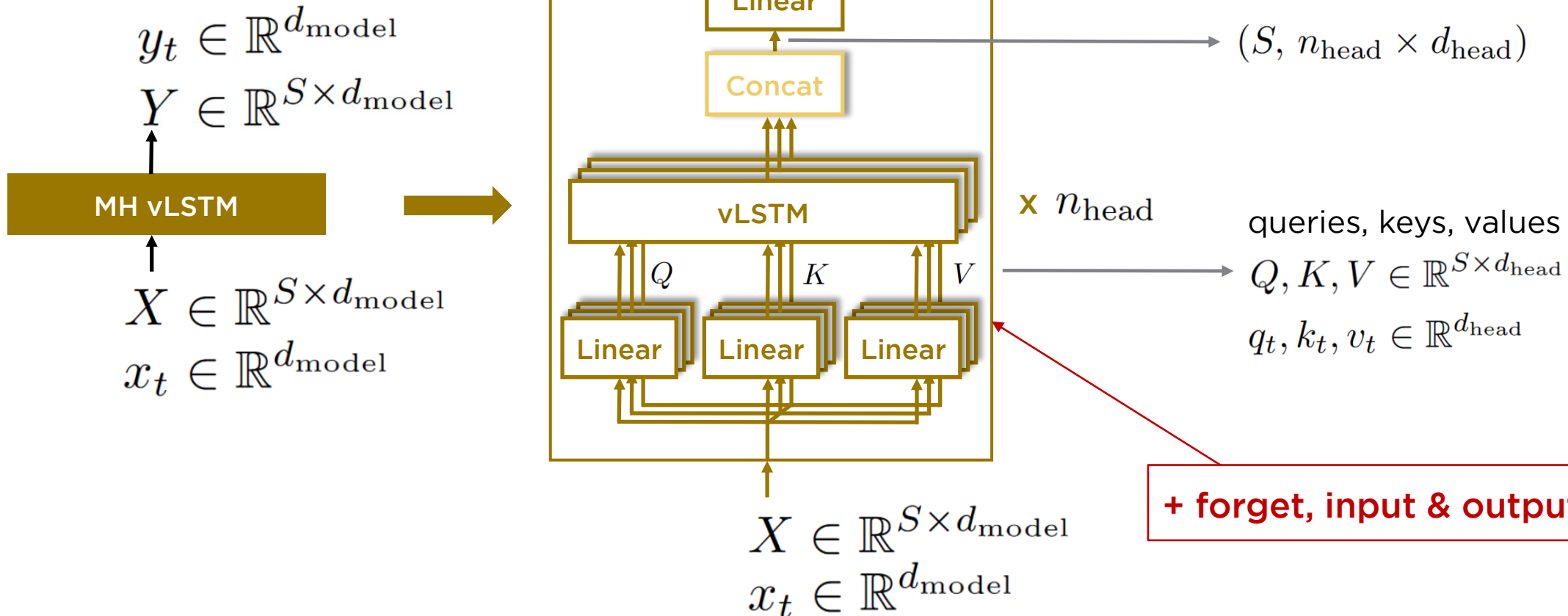
The xLSTM Family Tree



Multi-Head vLSTM

vLSTM has a
parallel & recurrent form

Sequence length: S
 Embedding dim.: d_{model}
 Head dimension: d_{head}
 Num. of heads: n_{head}



The vLSTM – Recurrent Formulation

Embedding dim.: d_{model}

Head dimension: d_{head}

Three Gates:

forget gate $f_t = x_t W_f^\top + b_f \in \mathbb{R}$

input gate $i_t = x_t W_i^\top + b_i \in \mathbb{R}$

output gate $o_t = x_t W_o^\top + b_o \in \mathbb{R}^{d_{\text{head}}}$

Input projections:

queries $q_t \in \mathbb{R}^{d_{\text{head}}}$

keys $k_t \in \mathbb{R}^{d_{\text{head}}}$

values $v_t \in \mathbb{R}^{d_{\text{head}}}$

Three States:

memory cell state $c_t \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$

normalizer state $n_t \in \mathbb{R}^{d_{\text{head}}}$

hidden state $h_t \in \mathbb{R}^{d_{\text{head}}}$

Forward updates:

$$c_{t+1} = \sigma(f_t) \odot c_t + e^{i_t} \odot k_t v_t^\top$$

$$n_{t+1} = \sigma(f_t) \odot n_t + e^{i_t} \odot k_t$$

$$h_{t+1}^\top = \sigma(o_t) \odot \frac{q_t^\top c_{t+1}}{q_t^\top n_{t+1}}$$

$$\tilde{f}_t = \sigma(f_t), \quad \tilde{i}_t = e^{i_t}$$

$$\tilde{o}_t = \sigma(o_t)$$

$$c_{t+1} = \tilde{f}_t \odot c_t + \tilde{i}_t \odot k_t v_t^\top$$

$$n_{t+1} = \tilde{f}_t \odot n_t + \tilde{i}_t \odot k_t$$

$$h_{t+1}^\top = \tilde{o}_t \odot \frac{q_t^\top c_{t+1}}{q_t^\top n_{t+1}}$$

Unstable due to sum of dot-products $\longrightarrow h_{t+1}^\top = \sigma(o_t) \odot \frac{q_t^\top c_{t+1}}{\max(|q_t^\top n_{t+1}|, 1.0)}$

The vLSTM – Parallel Formulation

Sequence length: S

Embedding dim.: d_{model}

Head dimension: d_{head}

Gate Matrices:

forget gates $\hat{F} = XW_f^\top + b_f \in \mathbb{R}^{S \times 1}$

input gates $\hat{I} = XW_i^\top + b_i \in \mathbb{R}^{S \times 1}$

output gates $O = XW_o^\top + b_o \in \mathbb{R}^{S \times d_{\text{head}}}$

Input projections:

queries $Q \in \mathbb{R}^{S \times d_{\text{head}}}$

keys $K \in \mathbb{R}^{S \times d_{\text{head}}}$

values $V \in \mathbb{R}^{S \times d_{\text{head}}}$

States:

hidden states $H \in \mathbb{R}^{S \times d_{\text{head}}}$

$$D = \tilde{F} \odot \exp(I) \in \mathbb{R}^{S \times S}$$

$$\tilde{F} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ \sigma(f_2) & 1 & 0 & \dots & 0 \\ \sigma(f_2)\sigma(f_3) & \sigma(f_3) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \sigma(f_2)\sigma(f_3)\dots\sigma(f_S) & \sigma(f_3)\dots\sigma(f_S) & \dots & \sigma(f_S) & 1 \end{pmatrix} \in \mathbb{R}^{S \times S}$$

Forward updates:

$$H = \sigma(O) \odot C V, \quad \text{where } C = \frac{\tilde{C}}{\sum_{j=1}^S \tilde{C}_{ij}}, \text{ and } \tilde{C} = QK^\top \odot D$$

$$\tilde{F}_{ij} = \begin{cases} 0 & \text{for } j > i \\ 1 & \text{for } j = i \\ \prod_{k=j+1}^i \sigma(f_k) & \text{for } j < i \end{cases}$$

$$I = \begin{pmatrix} i_1 & 0 & \dots & 0 \\ i_1 & i_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ i_1 & i_2 & \dots & i_S \end{pmatrix} \in \mathbb{R}^{S \times S}$$

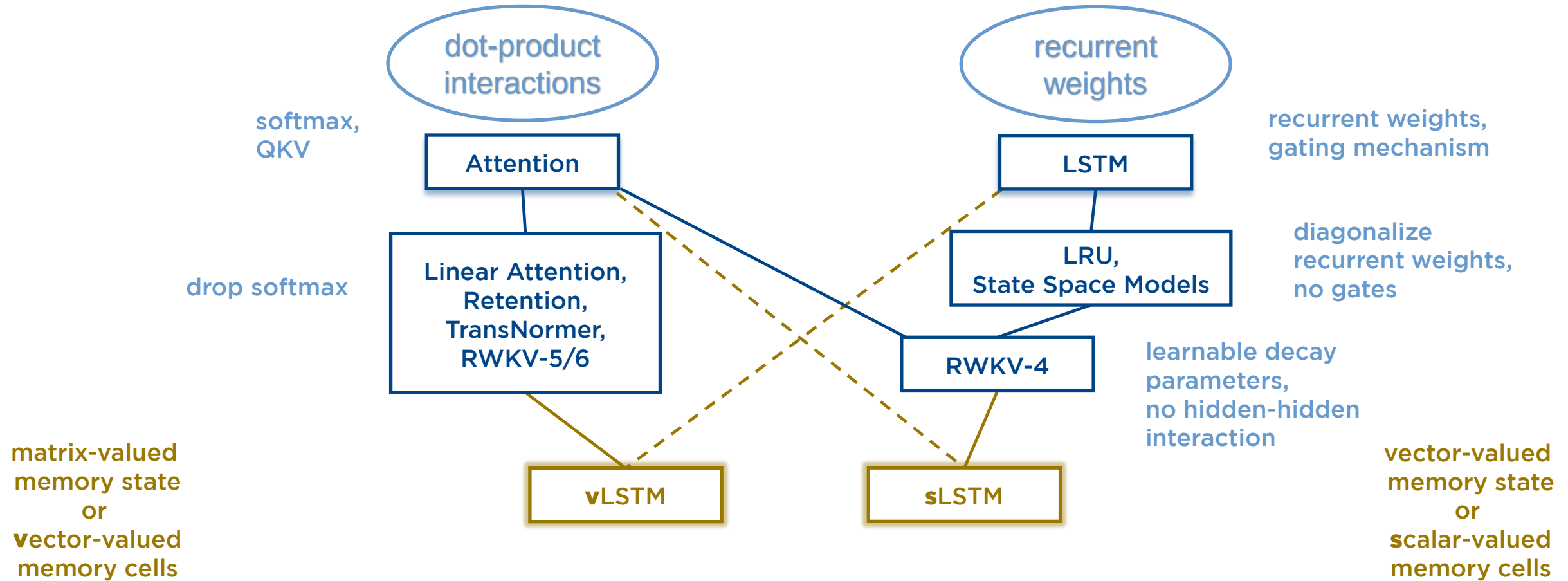
Unstable due to sum of dot-products

$$\longrightarrow C = \frac{\tilde{C}}{\max\left(\left|\sum_{j=1}^S \tilde{C}_{ij}\right|, 1.0\right)}$$

The vLSTM – Intuition

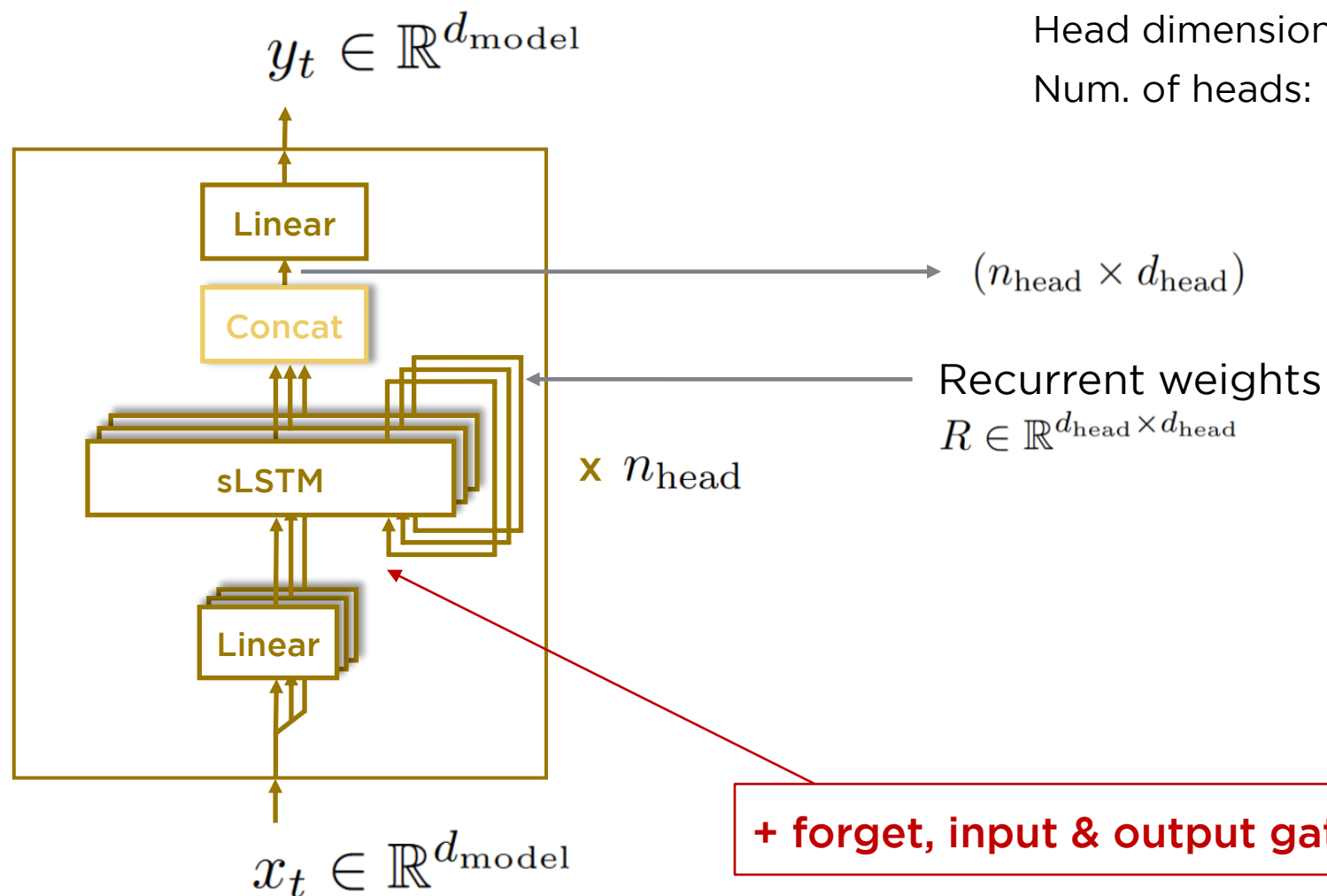
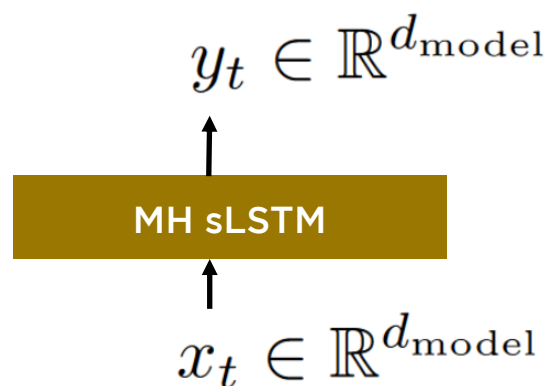
- Compared to **Linear Attention**:
 - vLSTM does not use feature functions on keys and queries.
 - vLSTM is more like Retention in that sense.
- Compared to **Retention**:
 - vLSTM is like Retention, but retention only uses fixed decay factors.
 - Retention has imaginary parameterization of Q and K.
 - We use a gating mechanism like LSTM with exponential input gate.
- With these gates we increase the non-linearity.
- The hope is that we achieve the performance and an information routing like the softmax.

The xLSTM Family Tree



Multi-Head sLSTM

sLSTM has a recurrent form only



Sequence length: S
Embedding dim.: d_{model}
Head dimension: d_{head}
Num. of heads: n_{head}

The sLSTM & sLSTMhin

Embedding dim.: d_{model}

Head dimension: d_{head}

Four Gates:

forget gate $f_t \in \mathbb{R}^{d_{\text{head}}}$

input gate $i_t \in \mathbb{R}^{d_{\text{head}}}$

cell gate $z_t \in \mathbb{R}^{d_{\text{head}}}$

output gate $o_t \in \mathbb{R}^{d_{\text{head}}}$

Input & Recurrent weights:

$W_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$ for $g \in \{f, i, z, o\}$

$R_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

$b_g \in \mathbb{R}^{d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

Three States:

memory cell state $c_t \in \mathbb{R}^{d_{\text{head}}}$

normalizer state $n_t \in \mathbb{R}^{d_{\text{head}}}$

hidden state $h_t \in \mathbb{R}^{d_{\text{head}}}$

Two versions:

sLSTM

$$f_t = x_t W_f^\top + h_t R_f^\top + b_f$$

$$i_t = x_t W_i^\top + h_t R_i^\top + b_i$$

$$z_t = x_t W_z^\top + h_t R_z^\top + b_z$$

$$o_t = x_t W_o^\top + h_t R_o^\top + b_o$$

sLSTMhin

$$f_t = x_t W_f^\top + b_f$$

$$i_t = h_t R_i^\top + b_i$$

$$z_t = x_t W_z^\top + b_z$$

$$o_t = x_t W_o^\top + b_o$$

Hiddens are fed
only into the
input gate

The sLSTM Recurrence

Embedding dim.: d_{model}
Head dimension: d_{head}

Four Gates:

forget gate $f_t \in \mathbb{R}^{d_{\text{head}}}$

input gate $i_t \in \mathbb{R}^{d_{\text{head}}}$

cell gate $z_t \in \mathbb{R}^{d_{\text{head}}}$

output gate $o_t \in \mathbb{R}^{d_{\text{head}}}$

Input & Recurrent weights:

$W_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$ for $g \in \{f, i, z, o\}$

$R_g \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

$b_g \in \mathbb{R}^{d_{\text{head}}}$ for $g \in \{f, i, z, o\}$

Three States:

memory cell state $c_t \in \mathbb{R}^{d_{\text{head}}}$

normalizer state $n_t \in \mathbb{R}^{d_{\text{head}}}$

hidden state $h_t \in \mathbb{R}^{d_{\text{head}}}$

Forward updates:

$$\begin{aligned}c_{t+1} &= \sigma(f_t) \odot c_t + e^{i_t} \odot z_t \\n_{t+1} &= \sigma(f_t) \odot n_t + e^{i_t} \\h_{t+1} &= \sigma(o_t) \odot \frac{c_{t+1}}{n_{t+1}}\end{aligned}$$

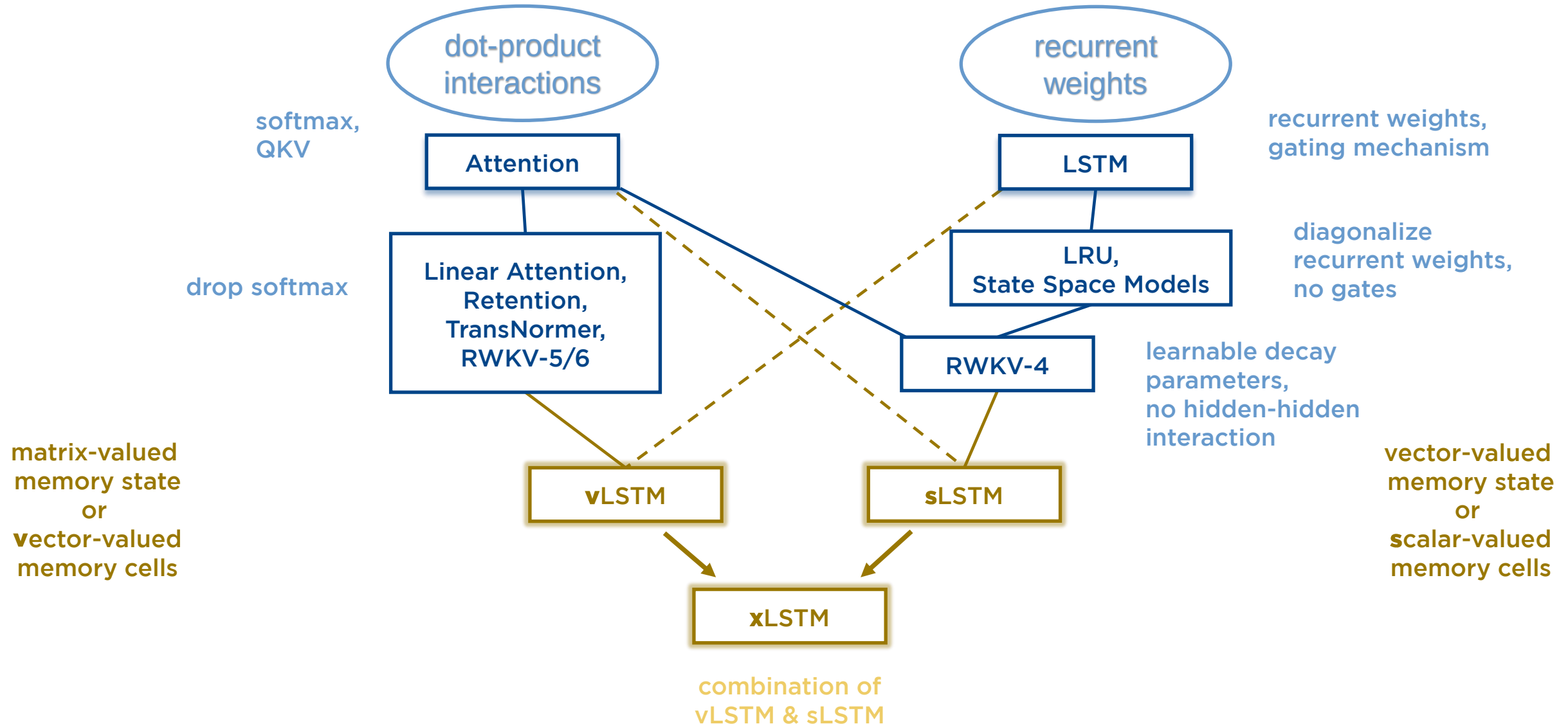
$$\tilde{f}_t = \sigma(f_t), \quad \tilde{i}_t = e^{i_t}$$



$$\tilde{z}_t = z_t, \quad \tilde{o}_t = \sigma(o_t)$$

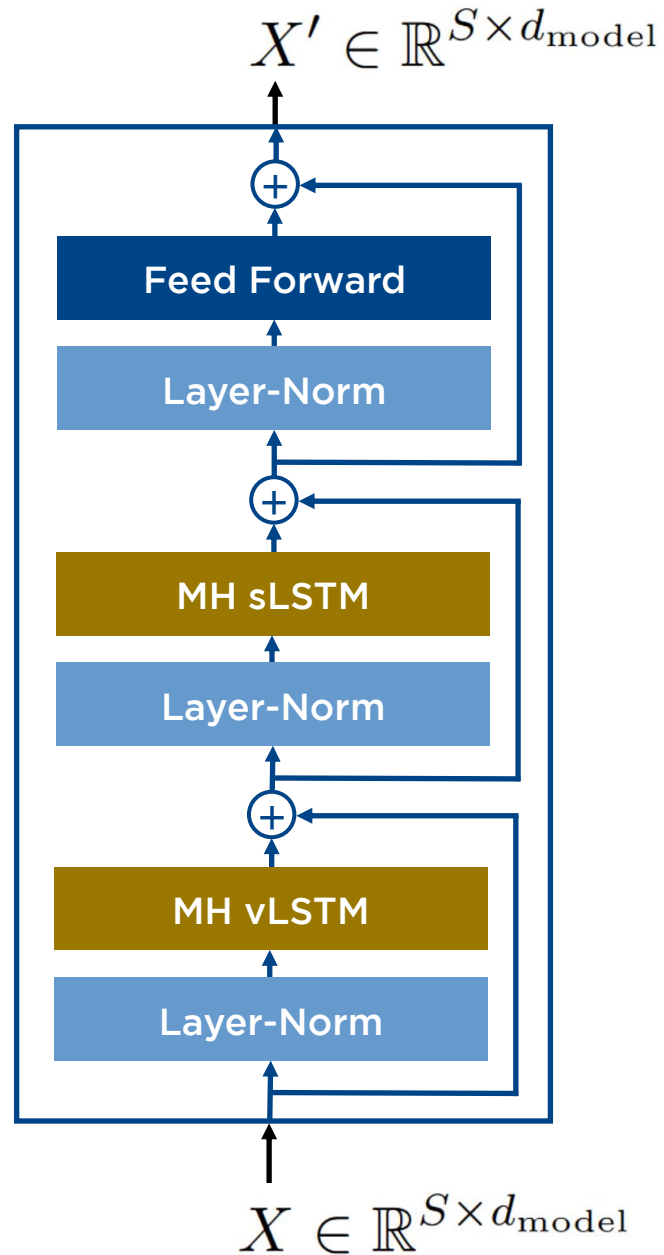
$$\begin{aligned}c_{t+1} &= \tilde{f}_t \odot c_t + \tilde{i}_t \odot \tilde{z}_t \\n_{t+1} &= \tilde{f}_t \odot n_t + \tilde{i}_t \\h_{t+1} &= \tilde{o}_t \odot \frac{c_{t+1}}{n_{t+1}}\end{aligned}$$

The xLSTM Family Tree



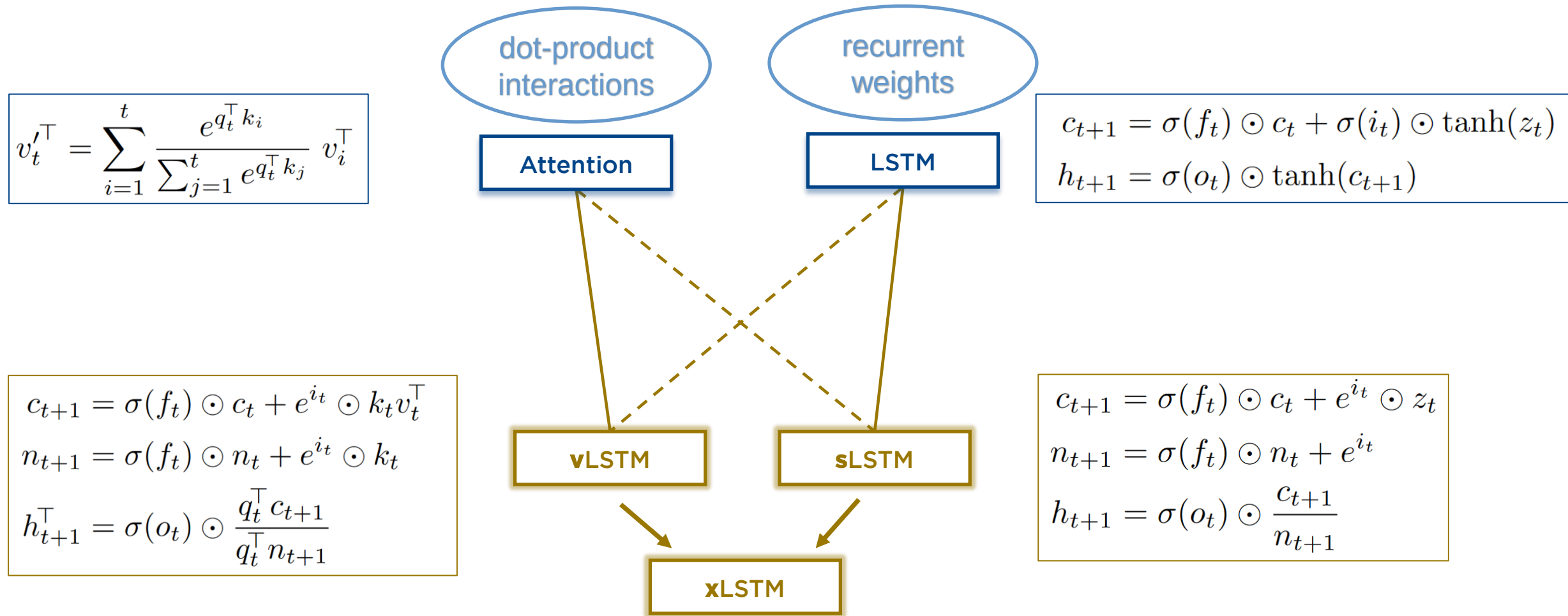
The xLSTM

xLSTM Block



Sequence length: S
Embedding dim.: d_{model}

The xLSTM Family – Summary



The competitions: Outline of the Experiments

Three competitions:

BL, CL & DFB-Pokal

- Language Modelling
- Synthetic Tasks
 - Motivated by H3, Hyena, Chomsky Hierarchy
- Long Range Modelling Capability
 - Long Range Arena Benchmark

Language Modelling – Planned Experiments

Datasets

- **SlimPajama**
 - 627B tokens
 - Cleaned and deduplicated version of RedPajama (Dataset to reproduce Llama2)

- **Subset of SlimPajama**
 - Single epoch or online training
- **Wikitext103**
 - 100M tokens from Wikipedia articles
 - Multiple epochs

Models

- Variants:
 - xLSTM, vLSTM, sLSTM
 - GPT2/3
- Sizes:
 - All: 125M, 350M, 760M, 1.3B
 - Best: as big as possible

- Variants:
 - Same as above
 - + all other baselines: (H3, Hyena, RWKV4, Linear Attention, Retention, etc.)
- Sizes:
 - all one size (e.g. 350M)

Metrics

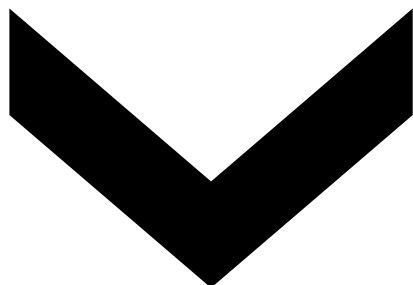
**Perplexity,
EleutherAI
Eval Harness**

Perplexity

Three competitions: BL, CL & DFB-Pokal



Thank You



PhD Seminar Talk

Maximilian Beck, beck@ml.jku.at,  [maxmbeck](https://twitter.com/maxmbeck)

Joint work with Korbinian Pöppel, Markus Spanring, Oleksandra Prudnikova, Michael Kopp, Johannes Brandstetter and Sepp Hochreiter

Institute for Machine Learning, October 2023