Transformers-XL

Chiyu Zhang
March 8th 2019

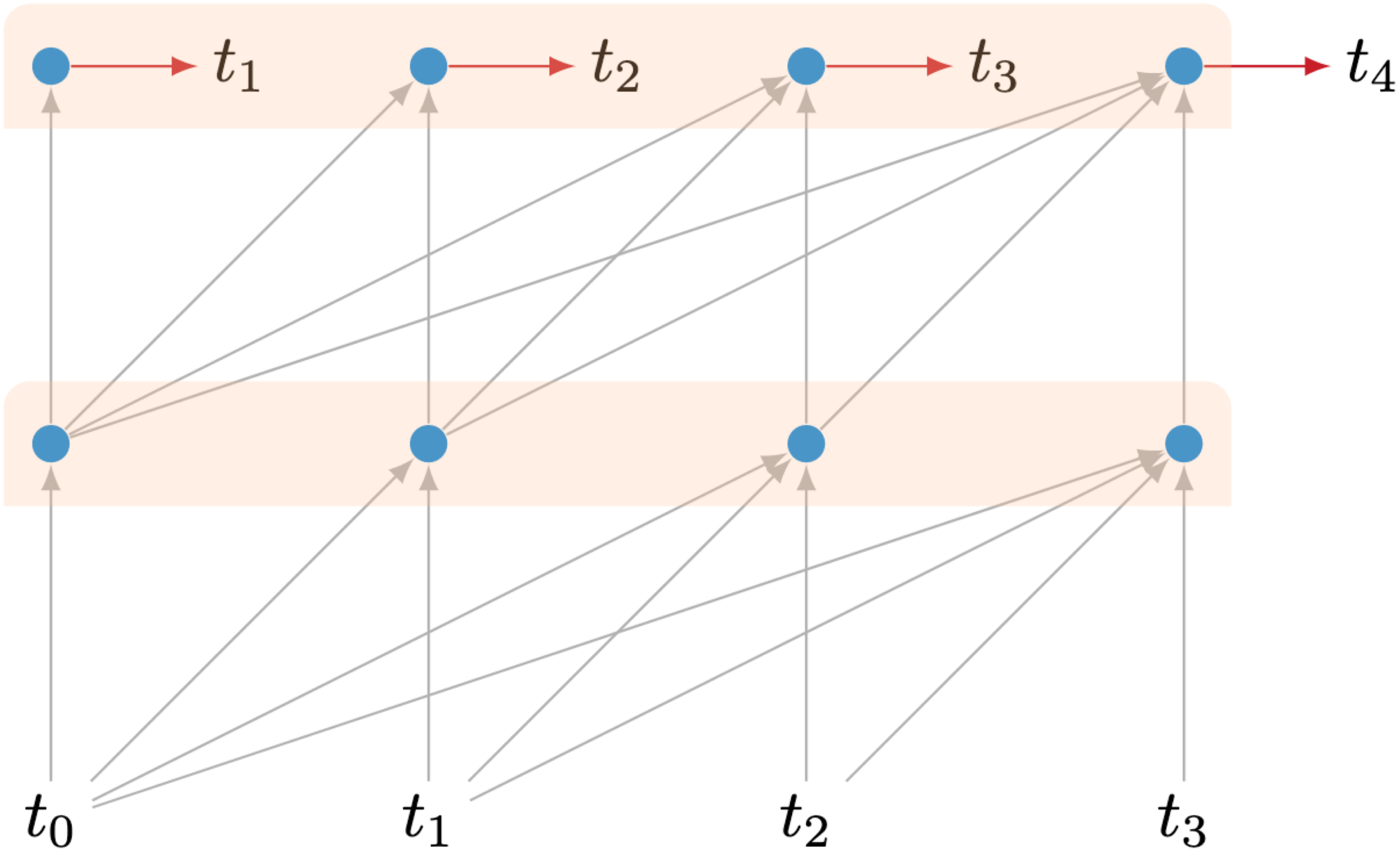
Problem statement

- Attention can only deal with fixed-length context
- Those fixed length are created by chunking up sentences causing context fragmentation

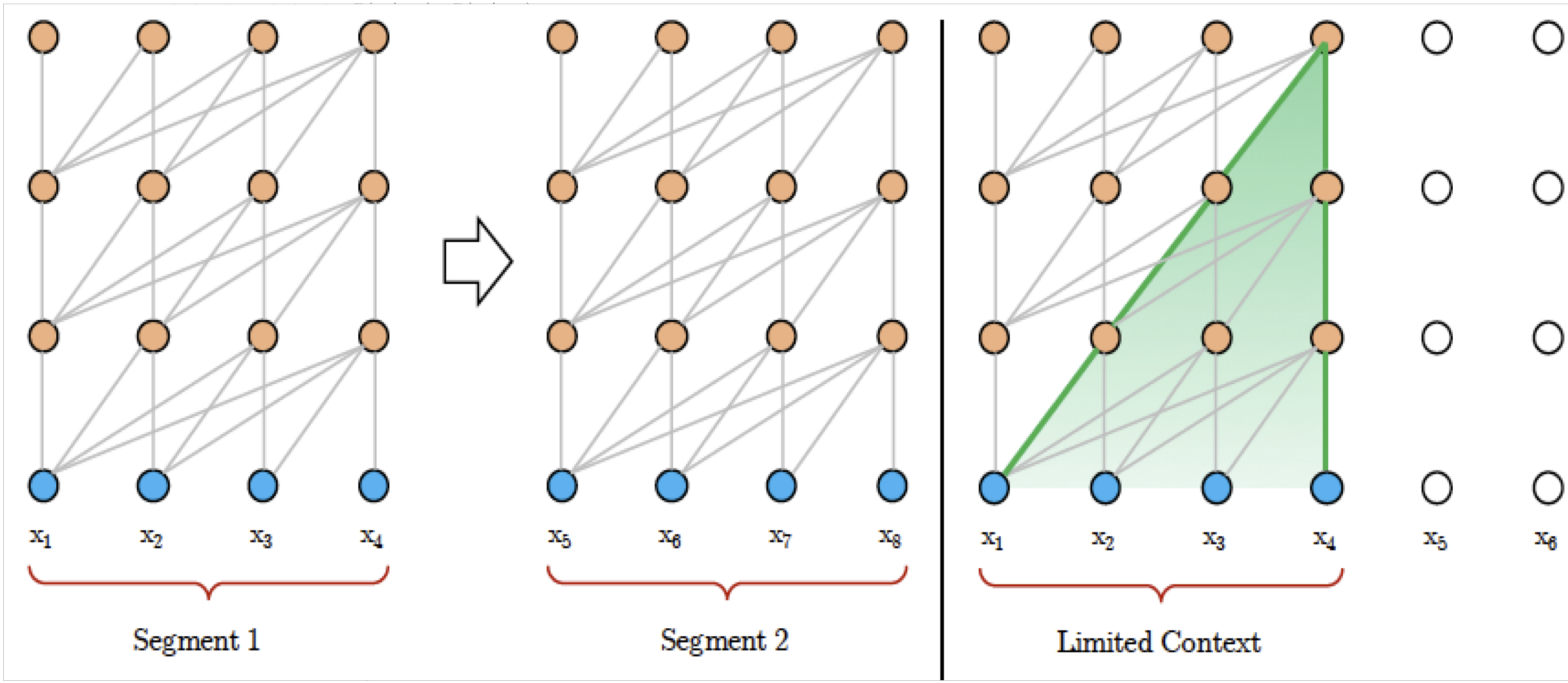
Solution:

- Introduce recurrence into deep self-attention networks

Vanilla training



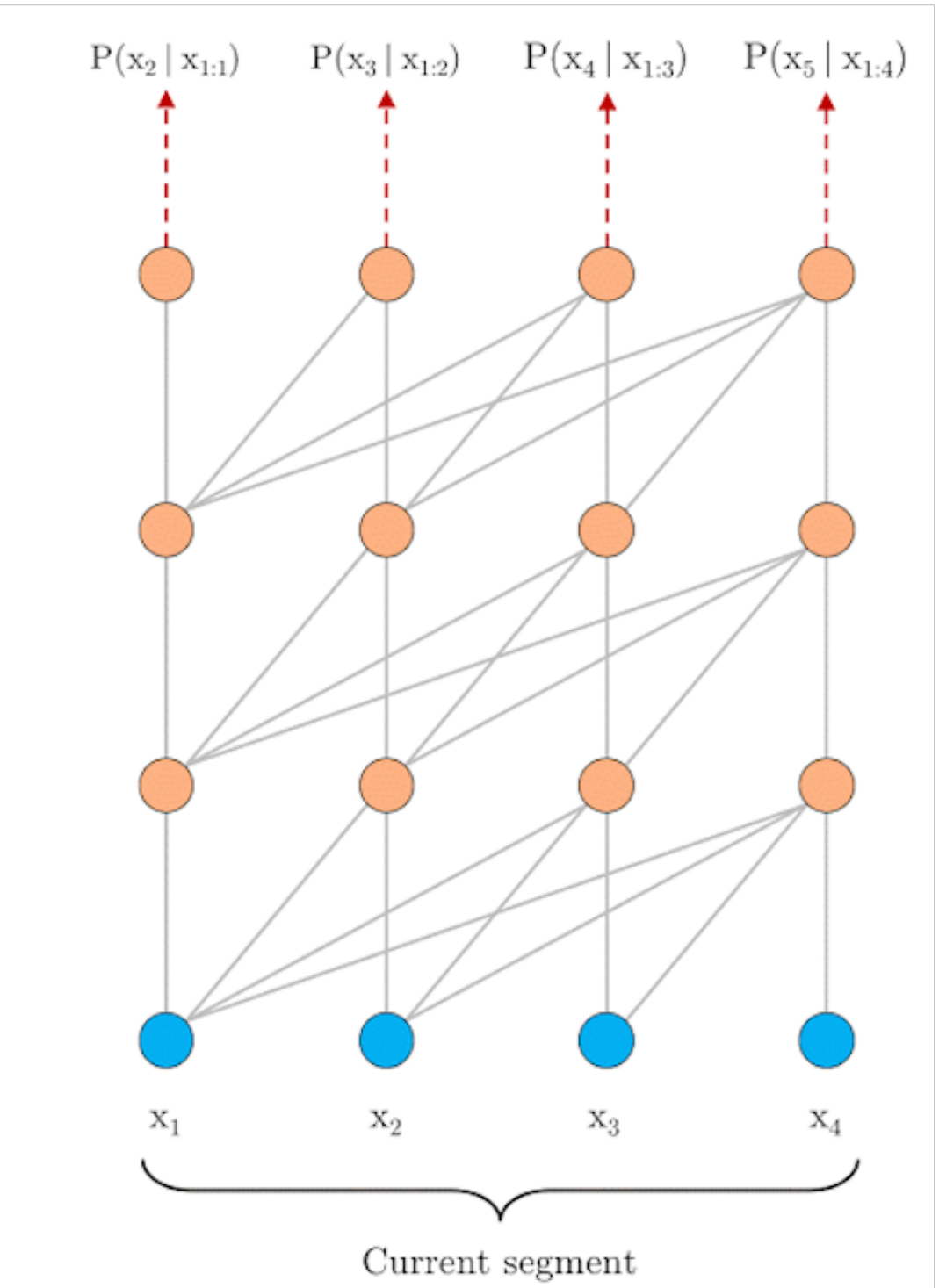
Vanilla training



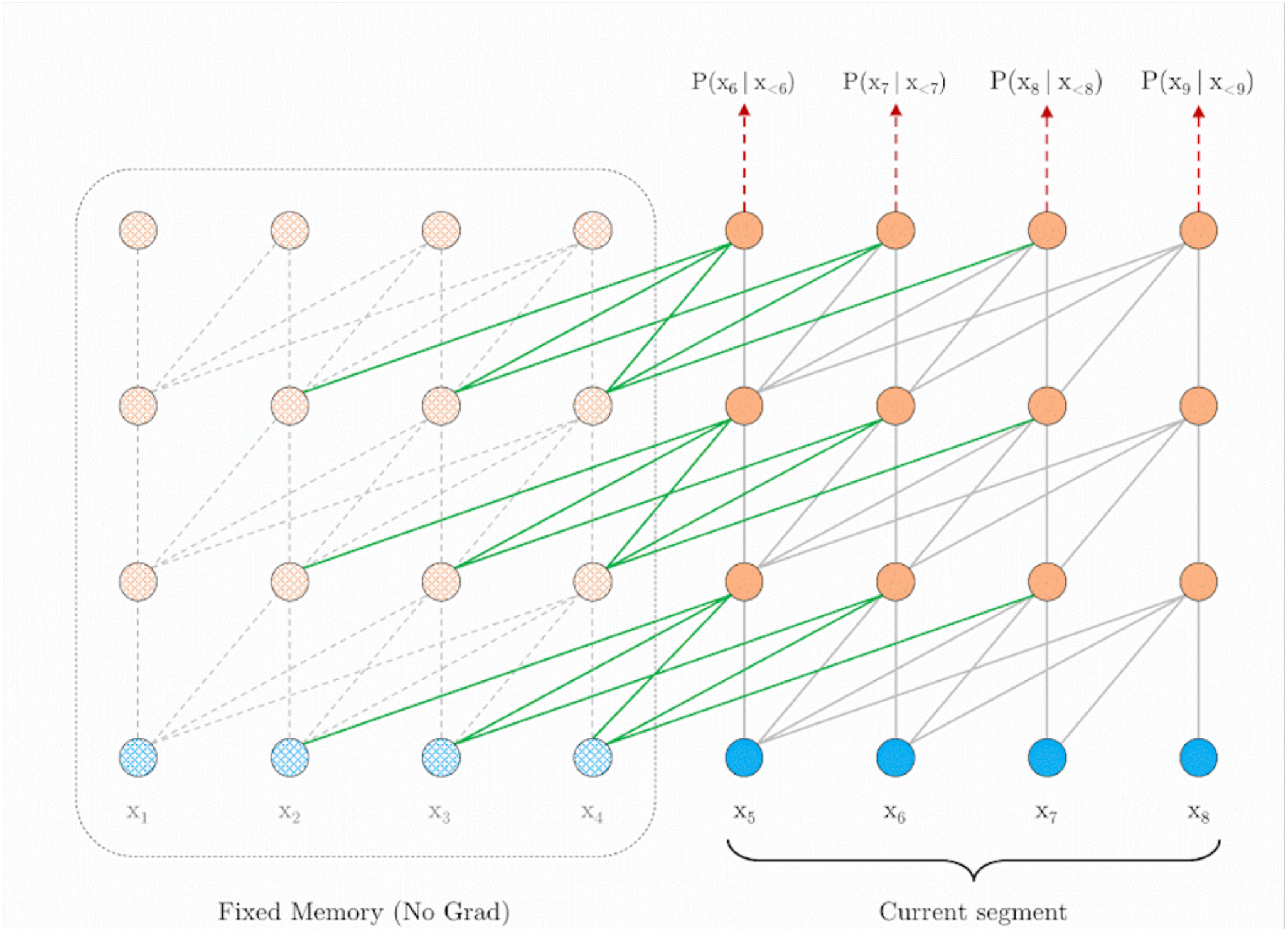
Vanilla training

- Information never flows across segments
- largest possible dependency is bound by sequence length
- longer sentences/sequences get split up causing context fragmentation

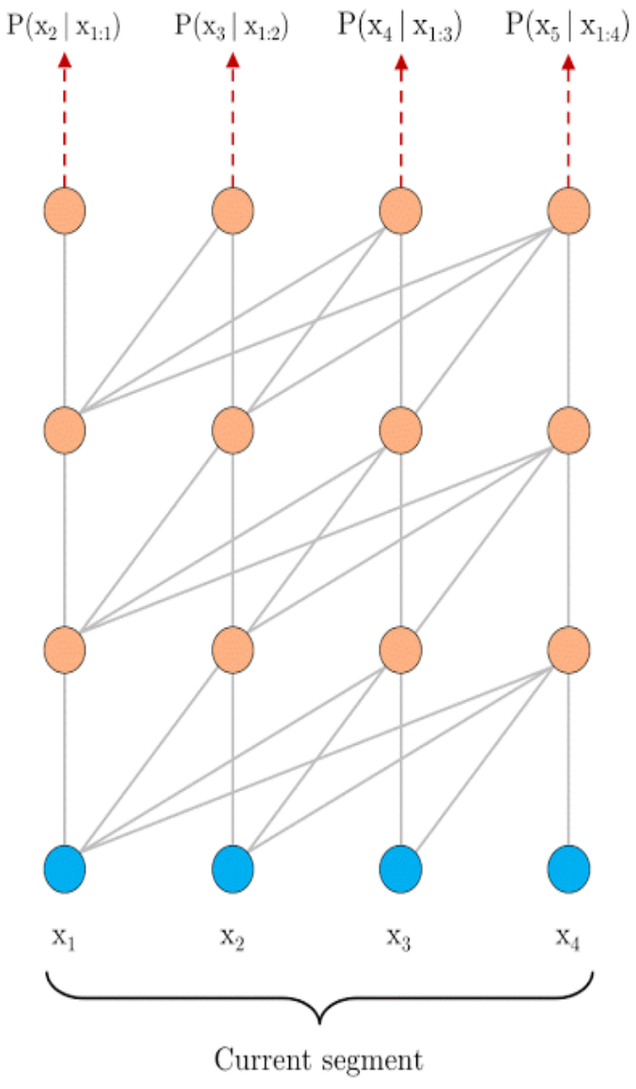
transformer-XL training



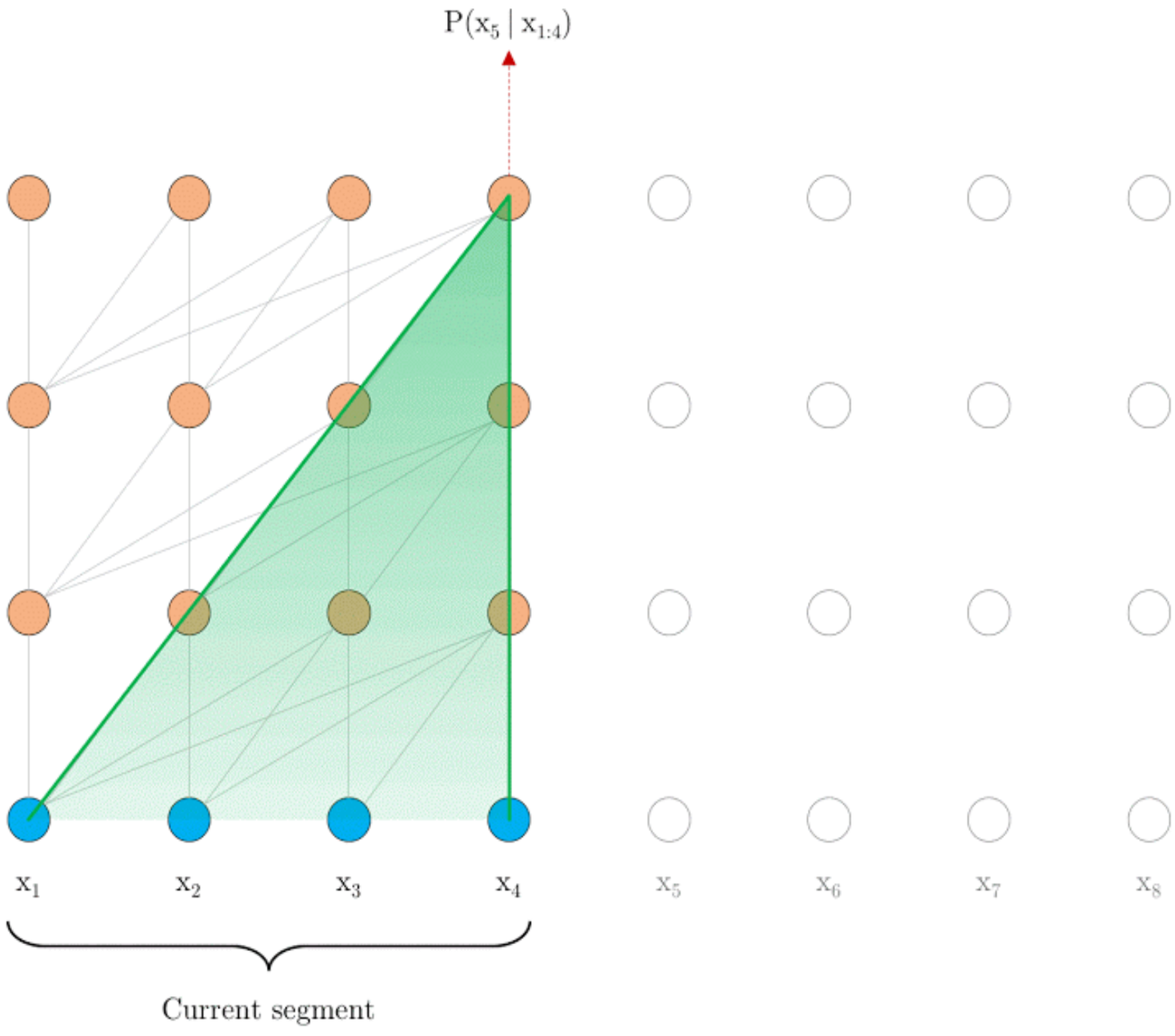
transformer-XL training



transformer-XL training



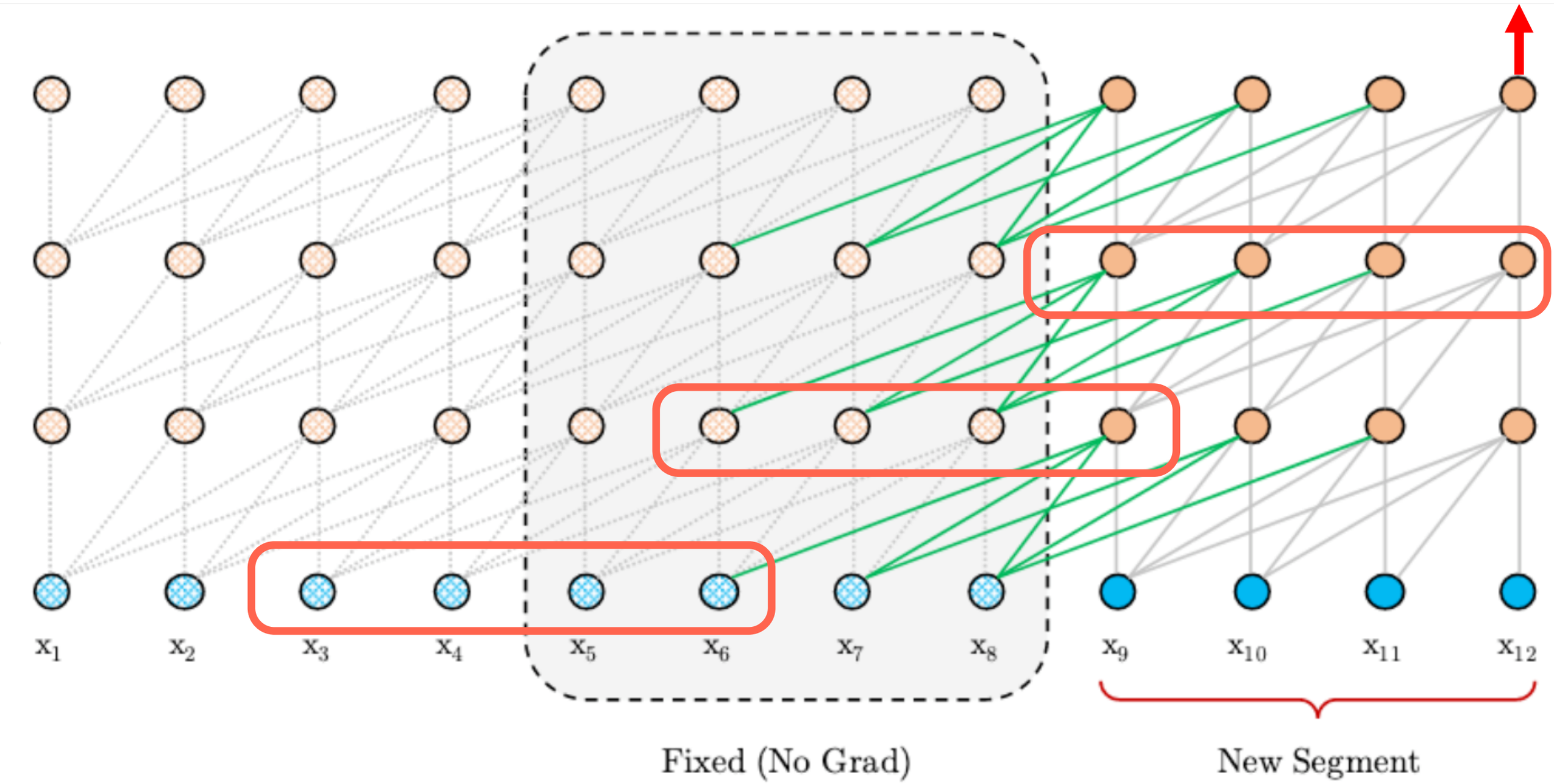
Vanilla prediction



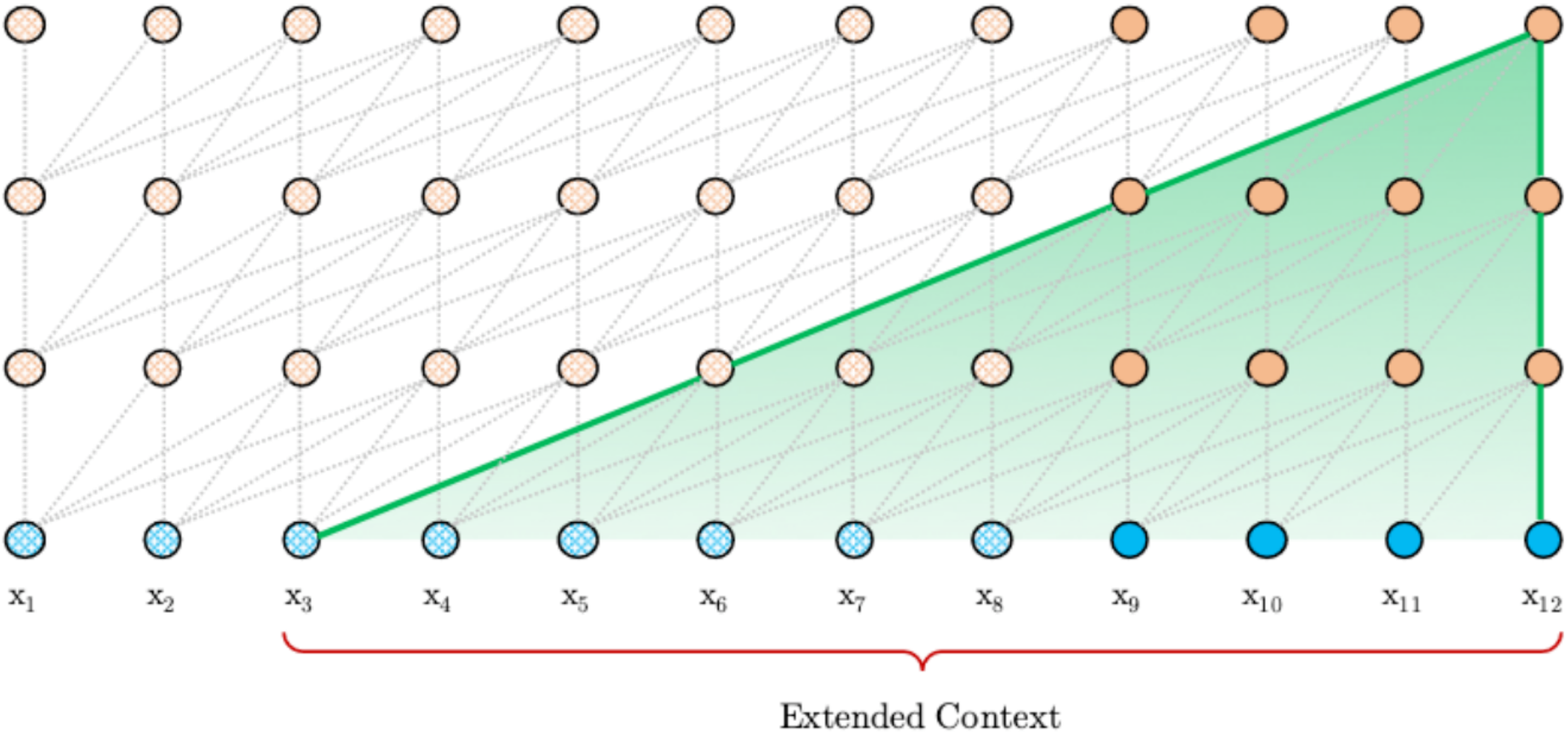
Vanilla prediction

- Consumes one segment at a time, but only does one prediction
- Relieves context fragmentation
- Extremely expensive

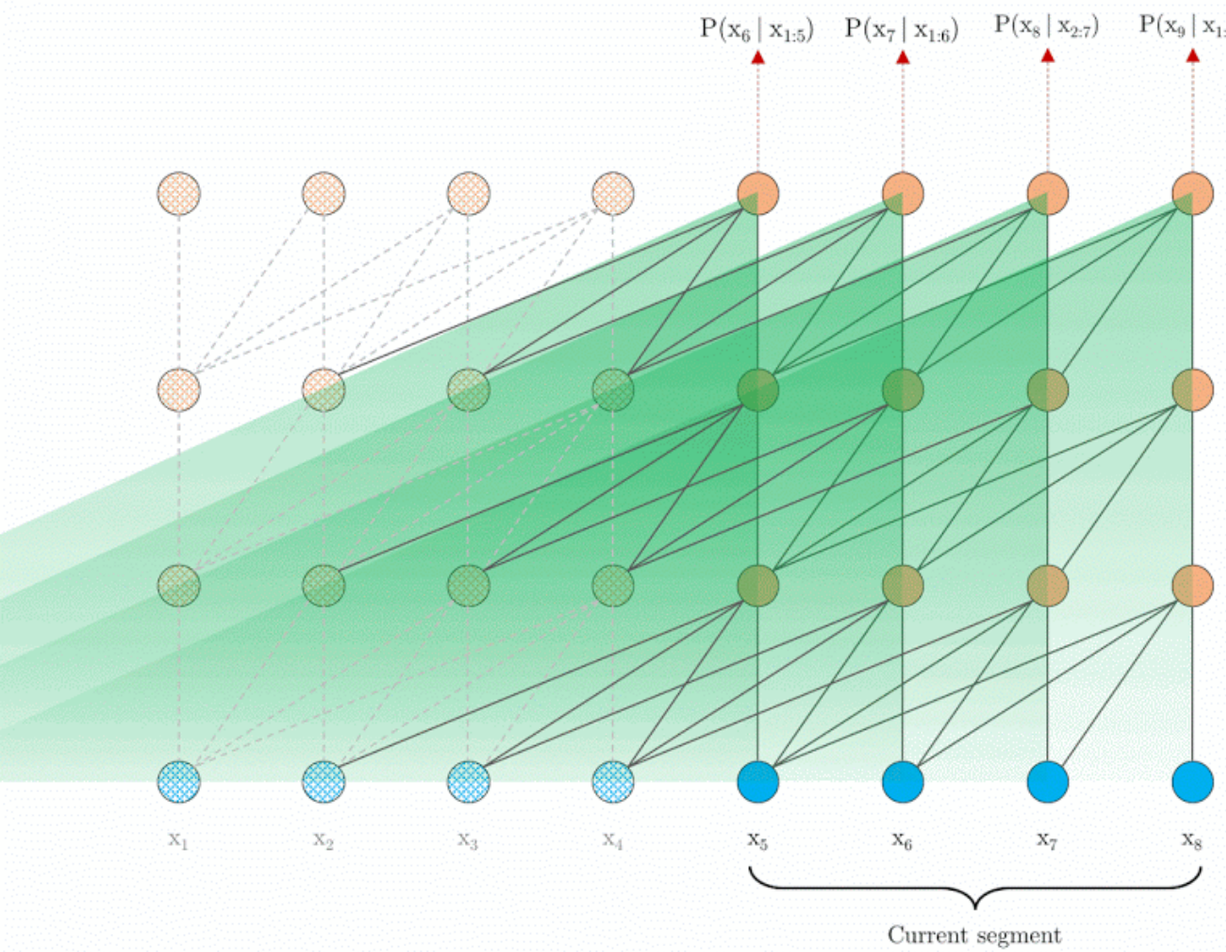
Transformer-XL prediction



Transformer-XL prediction



Transformer-XL prediction



In formula

$$\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = [\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}], \qquad \text{(extended context)}$$

$$\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top}, \qquad \text{(query, key, value vectors)}$$

$$\mathbf{h}_{\tau+1}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right). \qquad \text{(self-attention + feed-forward)}$$

$\mathbf{h}_{\tau}^{n} \in \mathbb{R}^{L \times d}$ ***n*th layer hidden state for segment τ**

Using memory

$$
\begin{aligned}
\widetilde{\mathbf{h}}_{\tau+1}^{n-1} &= [\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}], && \text{(extended context)} \\
\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} &= \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top}, && \text{(query, key, value vectors)} \\
\mathbf{h}_{\tau+1}^{n} &= \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right). && \text{(self-attention + feed-forward)}
\end{aligned}
$$

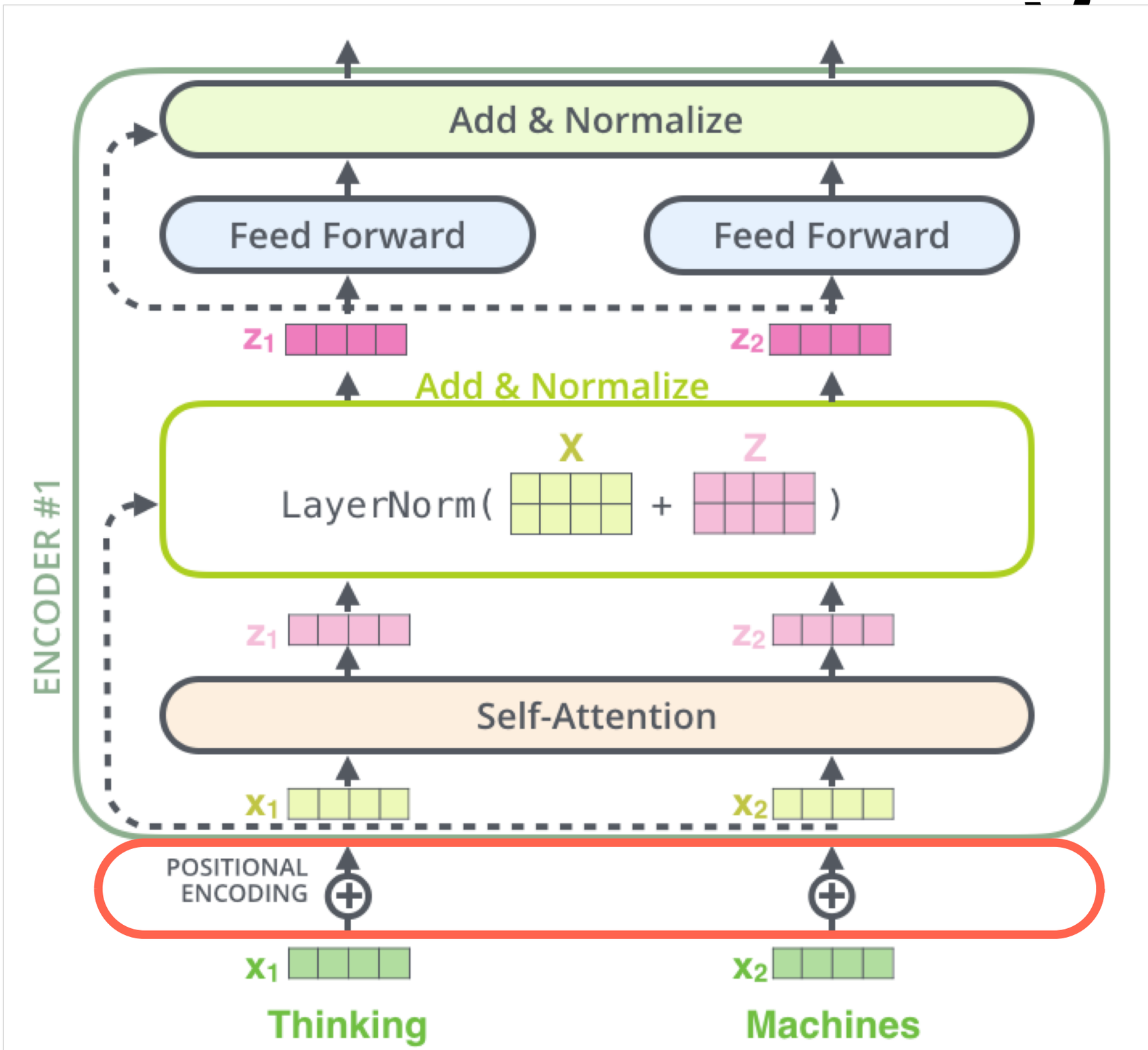
$$\mathbf{m}_{\tau}^{n} \in \mathbb{R}^{M \times d}$$

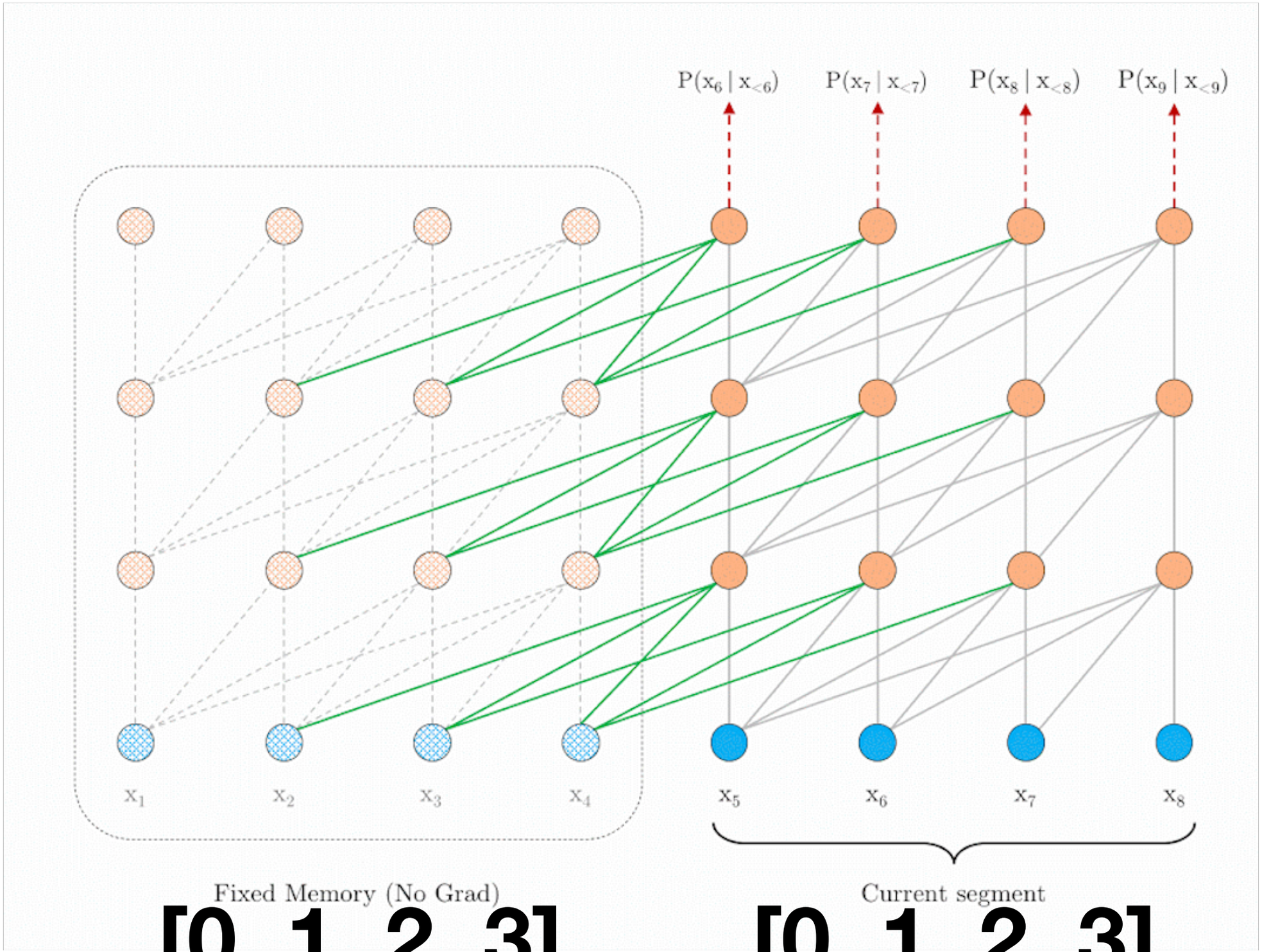
- **Cache predefined length-*M* old hidden:**
  - **Training M sequence length**
  - **Evaluation M is multiple of sequence length**

$$\widetilde{\mathbf{h}}_{\tau}^{n-1} = \left[\mathrm{SG}\left(\mathbf{m}_{\tau}^{n-1}\right) \circ \mathbf{h}_{\tau}^{n-1}\right]$$

Position wise embeddings

Vanilla embeddings




$P(x_6 \mid x_{<6})$
$P(x_7 \mid x_{<7})$
$P(x_8 \mid x_{<8})$
$P(x_9 \mid x_{<9})$
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
Fixed Memory (No Grad)
Current segment

[0, 1, 2, 3] **[0, 1, 2, 3]**

$$\mathbf{E}_{\mathbf{s}_\tau} \in \mathbb{R}^{L \times d}$$ **Word embeddings**

$$\mathbf{U}_{1:L}$$ **Position wise embeddings**

$$\mathbf{h}_\tau = f(\mathbf{h}_{\tau-1}, \mathbf{E}_{\mathbf{s}_\tau} + \mathbf{U}_{1:L})$$

$$\mathbf{h}_{\tau+1} = f(\mathbf{h}_\tau, \mathbf{E}_{\mathbf{s}_{\tau+1}} + \mathbf{U}_{1:L})$$

Relative-position wise embeddings

Original self attention in other words

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

$\mathbf{E}_{\mathbf{s}_\tau} \in \mathbb{R}^{L \times d}$ is the word embedding sequence of $\mathbf{s}_\tau$

$\mathbf{U}_i$ *i*-th absolute position within a segment

$$\mathbf{A}^{\text{abs}}_{i,j} = q_i^\top k_j = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}$$

It is enough to know the relative distance between i and j

$\mathbf{R} \in \mathbb{R}^{L_{\max} \times d}$, where the i-th row $\mathbf{R}_i$ indicates a relative distance of i between two positions.

$$\mathbf{A}^{\text{rel}}_{i,j} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}$$

a. Content based

b. content-depend position bias

c. Global content bias

d. Global position bias

Extend Self attention

- Extend Qi*Ki by four terms:
  - Content Weight: the original score without the addition of the original positional encoding of course
  - Positional bias with respect to the current content (Qi).
  - A learned global content bias - vector for adjusting key importance
  - A learned global position bias

Transformer-XL

$$\mathbf{h}_\tau^0 := \mathbf{E}_{\mathbf{s}_\tau}$$

For $n = 1, \ldots, N$:

$$\widetilde{\mathbf{h}}_\tau^{n-1} = \left[\mathrm{SG}(\mathbf{m}_\tau^{n-1}) \circ \mathbf{h}_\tau^{n-1}\right]$$

$$\mathbf{q}_\tau^n, \mathbf{k}_\tau^n, \mathbf{v}_\tau^n = \mathbf{h}_\tau^{n-1}\mathbf{W}_q^{n\top}, \widetilde{\mathbf{h}}_\tau^{n-1}\mathbf{W}_{k,E}^{n\top}, \widetilde{\mathbf{h}}_\tau^{n-1}\mathbf{W}_v^{n\top}$$

$$\mathbf{A}_{\tau,i,j}^n = \mathbf{q}_{\tau,i}^{n\top}\mathbf{k}_{\tau,j}^n + \mathbf{q}_{\tau,i}^{n\top}\mathbf{W}_{k,R}^n\mathbf{R}_{i-j} + u^\top\mathbf{k}_{\tau,j} + v^\top\mathbf{W}_{k,R}^n\mathbf{R}_{i-j}$$

$$\mathbf{a}_\tau^n = \text{Masked-Softmax}(\mathbf{A}_\tau^n)\mathbf{v}_\tau^n$$

$$\mathbf{o}_\tau^n = \text{LayerNorm}(\text{Linear}(\mathbf{a}_\tau^n) + \mathbf{h}_\tau^{n-1})$$

$$\mathbf{h}_\tau^n = \text{Positionwise-Feed-Forward}(\mathbf{o}_\tau^n)$$

$W_{k,E}^n$ **Content based**

R_{i-j} **Relative position embedding**

u^T **Global position bias**

$W_{k,R}^n$ **Location based**

v^T **Global position bias**

Results

WikiText-103

Model	#Params	Validation PPL	Test PPL
Grave et al. (2016b) – LSTM	-	-	48.7
Bai et al. (2018) – TCN	-	-	45.2
Dauphin et al. (2016) – GCNN-8	-	-	44.9
Grave et al. (2016b) – LSTM + Neural cache	-	-	40.8
Dauphin et al. (2016) – GCNN-14	-	-	37.2
Merity et al. (2018) – 4-layer QRNN	151M	32.0	33.0
Rae et al. (2018) – LSTM + Hebbian + Cache	-	29.7	29.9
Ours – Transformer-XL Standard	151M	**23.1**	**24.0**
Baevski & Auli (2018) – adaptive input◇	247M	19.8	20.5
Ours – Transformer-XL Large	257M	**17.7**	**18.3**

103M training tokens from 28k articles, with 3.6k tokens per article

Attention Length 384 training, 1600 evaluation

16 layer and 10 heads of each layer

enwiki8

Model	#Params	Test bpc
Ha et al. (2016) – LN HyperNetworks	27M	1.34
Chung et al. (2016) – LN HM-LSTM	35M	1.32
Zilly et al. (2016) – Recurrent highway networks	46M	1.27
Mujika et al. (2017) – Large FS-LSTM-4	47M	1.25
Krause et al. (2016) – Large mLSTM	46M	1.24
Knol (2017) – cmix v13	-	1.23
Al-Rfou et al. (2018) – 12-layer Transformer	44M	1.11
Ours – 12-layer Transformer-XL	41M	**1.06**
Al-Rfou et al. (2018) – 64-layer Transformer	235M	1.06
Ours – 18-layer Transformer-XL	88M	1.03
Ours – 24-layer Transformer-XL	277M	**0.99**

Attention Length 784 training, 3200 evaluation

text8

Model	#Params	Test bpc
Cooijmans et al. (2016) – BN-LSTM	-	1.36
Chung et al. (2016) – LN HM-LSTM	35M	1.29
Zilly et al. (2016) – Recurrent highway networks	45M	1.27
Krause et al. (2016) – Large mLSTM	45M	1.27
Al-Rfou et al. (2018) – 12-layer Transformer	44M	1.18
Al-Rfou et al. (2018) – 64-layer Transformer	235M	1.13
Ours – 24-layer Transformer-XL	277M	**1.08**

Attention Length 784 training, 3200 test

Ablation study

Remark	Recurrence	Encoding	Loss	PPL init	PPL best	Attn Len
Transformer-XL (128M)	✓	Ours	Full	**27.02**	**26.77**	**500**
-	✓	Shaw et al. (2018)	Full	27.94	27.94	256
-	✓	Ours	Half	28.69	28.33	460
-	✗	Ours	Full	29.59	29.02	260
-	✗	Ours	Half	30.10	30.10	120
-	✗	Shaw et al. (2018)	Full	29.75	29.75	120
-	✗	Shaw et al. (2018)	Half	30.50	30.50	120
-	✗	Vaswani et al. (2017)	Half	30.97	30.97	120
Transformer (128M)†	✗	Al-Rfou et al. (2018)	Half	31.16	31.16	120
					23.09	**640**
Transformer-XL (151M)	✓	Ours	Full	23.43	23.16	450
					23.35	300

WikiText-103

Summary

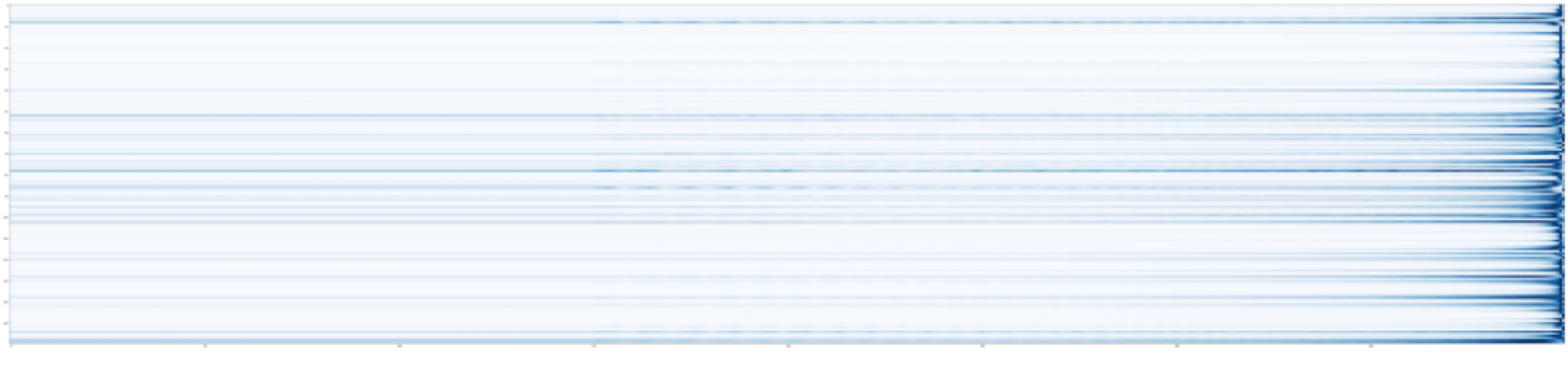
- Enable language modelling with self-attention architecture beyond a fixed length context. (Recurrence in purely self-attentive model)
- can learn longer dependency
  - 80% and 450% more than RNN and vanilla transformer
- 1,800 times faster than vanilla transformer

Attention Layers

Attention layers

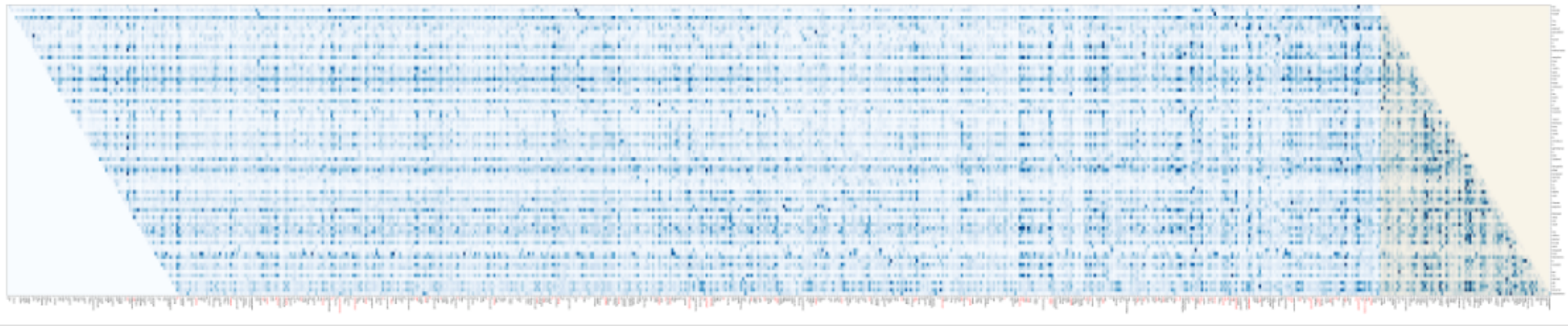
- Visualize attentions from for model on WikiText-103
  - 16 10-head transformer layers and memory length of 640

Attention layers



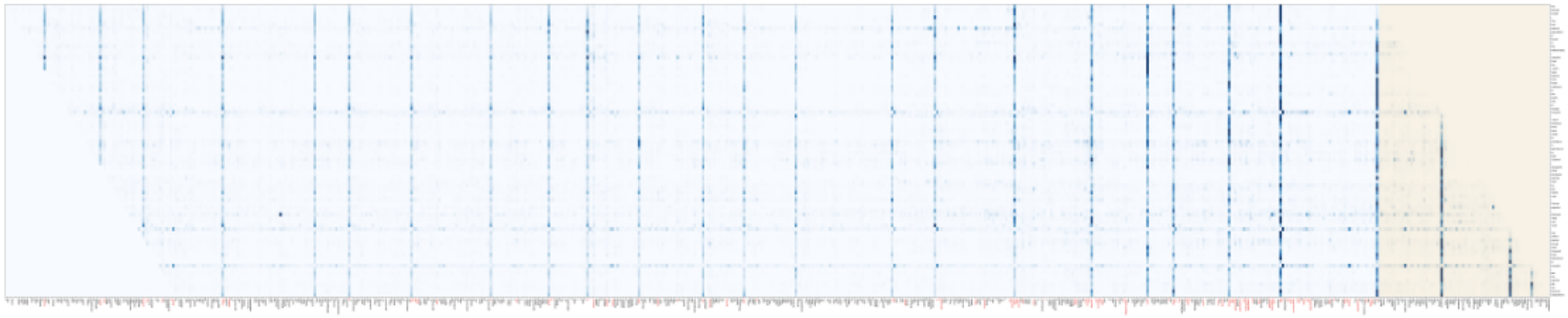
- Row attention head; Col relative locations (every 10 head 1 layer)
- Average attention over all tokens in the validation set
- trend focus nearby tokens (some exceptions)

Attention layers



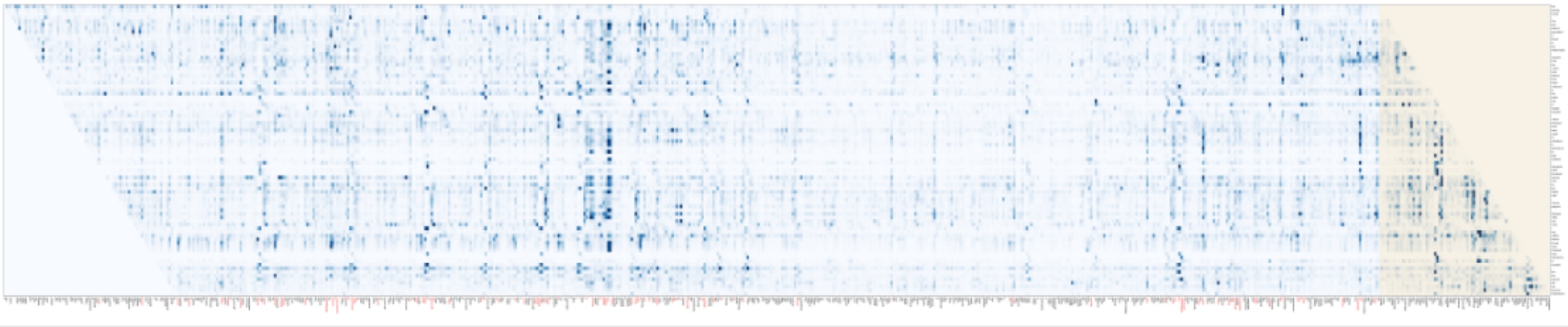
- Head 8 Layer 1:
    - Uniform attention over entire memory span - screen entire memory span

Attention layers



- Head 78 Layer 8th Layer (mid lvl layer)
  - sparse attention as information accumulates, the network focus on some particular positions

Attention layers



- Head 158 Layer 16 (last layer)
  - each target location has it’s own distinct focus
    - differs from layer 78 where focus is shared
    - layer 1 few locations attend more

External Ref:

Toronto Deep Learning Series (#TDLS):

https://tdls.a-i.science/events/2019-02-21/

Google Blog:
https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html