Transformer-XL & Longformer

- **Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context**
  Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov
- **Longformer: The Long-Document Transformer**
  Iz Beltagy, Matthew E. Peters, Arman Cohan

중급 7조
김영협, 박종일, 오경원, 오지현

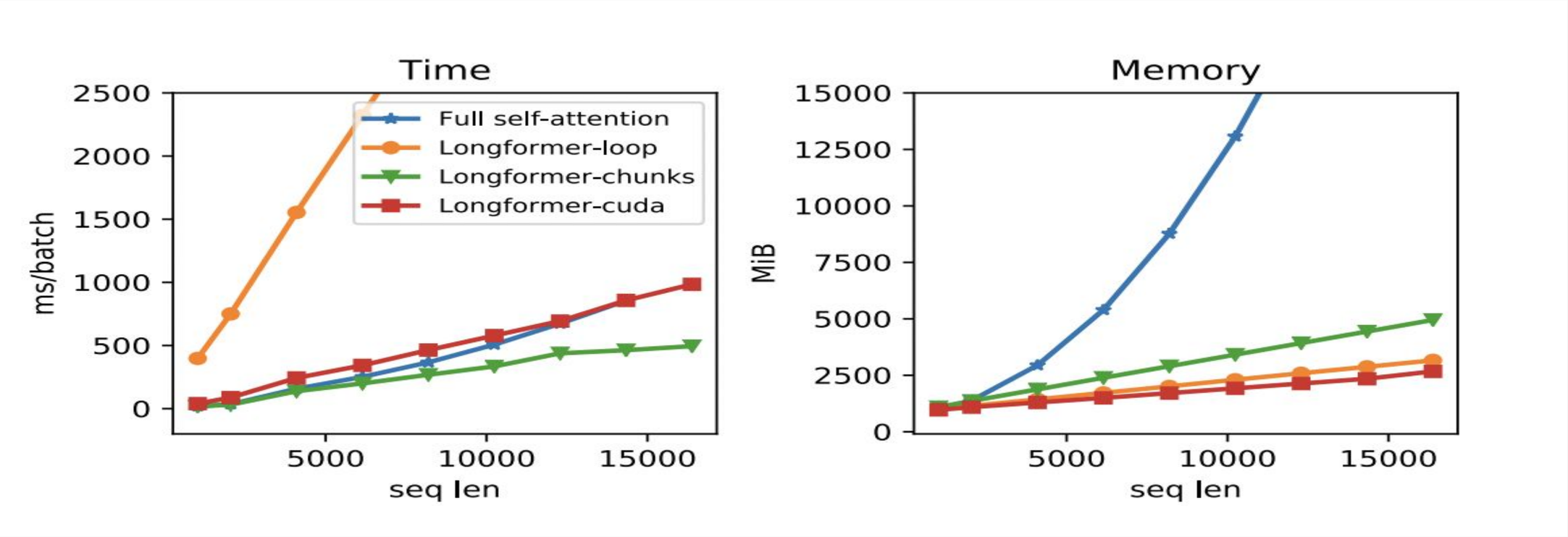
Contents

1. Abstract & Introduction

- **Why?**
- **Attention & Transformer**
- 문제해결 접근 방식 및 차이점
- **Related Work**

Why?

- **Too expensive to process from long sequences**
  - Full Attention is not only memory inefficient, but also computationally inefficient



Why?

- **Transformer의 Attention : 고정된 크기의 Input 값 (dmodel : 512)**
  - 고정된 Token Segment의 input
- **Recurrent to Self-attention Complexity per Layer : n to n2**

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Why?

- **LM : Longer-term dependency problem**
  - Gradient flow in recurrent nets: the difficulty of learning long-term dependencies : Gradient Vanishing and Explosion
  - LSTM language models use 200 context words on average

- **Transformer : Context Fragmentation**
  - Computing the hidden states from scratch for each new segments with fixed length : the model lacks necessary contextual information

문제해결 접근 방식

- **Transformer-XL**
  - Capture **longer-term dependency**
  - Resolve **context fragmentations**

- **Longformer**
  - **Drop-in replacement** for the standard self-attention and combines a local windowed attention with a task motivated global attention.

Related Work

- **Transformer-XL**
  - Truncated Backpropagation Through Time (Truncated BPTT)
    - Training technique for RNN based models
    - Hidden states from the previous batch are passed forward to the current batch
  - Character Transformer Model*
    - Apply transformer to character-level language model
    - Sequence Length of Word Token < Sequence Length of Character Token

**Al-Rhou et al 2018, Character-Level Language Modeling with Deeper Self-Attention*

Related Work

- **Longformer**
  - Long-Document Transformers
  - Task-specific Models for Long Documents

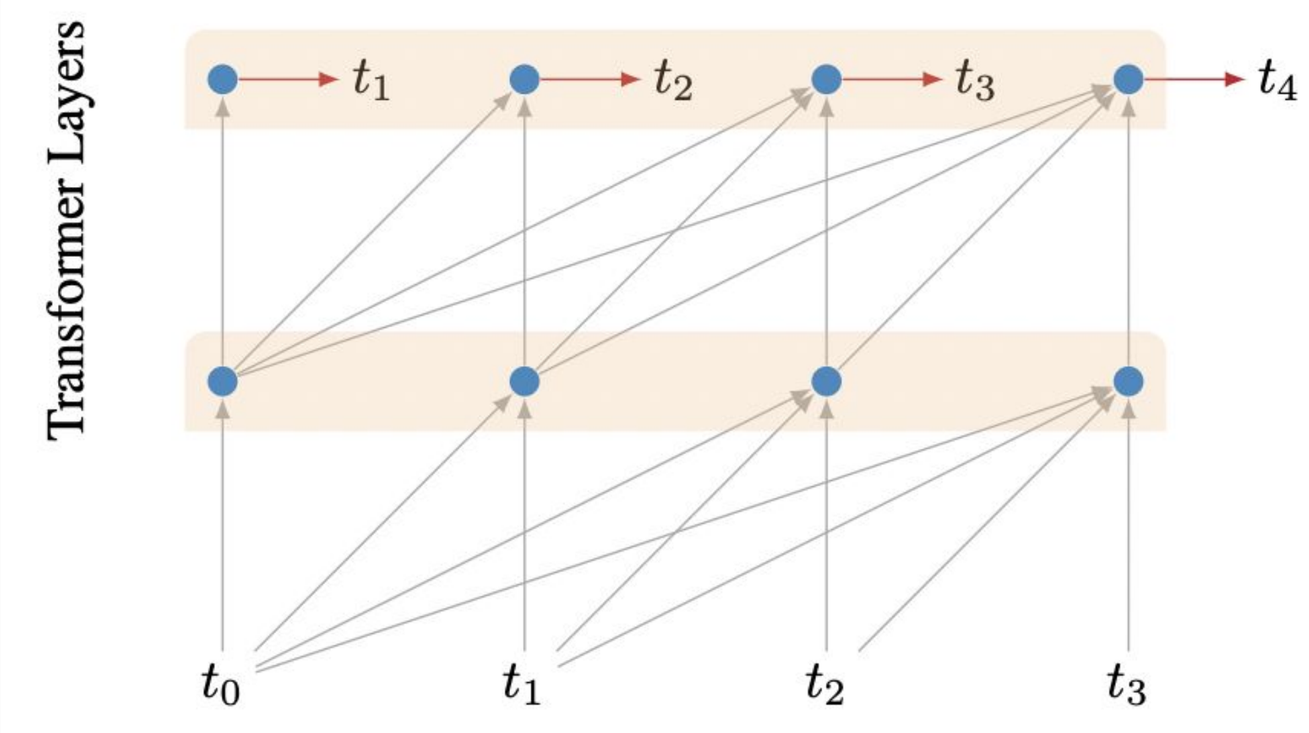
Model	attention matrix	char-LM	other tasks	pretrain
Transformer-XL (2019)	ltr	yes	no	no
Adaptive Span (2019)	ltr	yes	no	no
Compressive (2020)	ltr	yes	no	no
Reformer (2020)	sparse	yes	no	no
Sparse (2019)	sparse	yes	no	no
Routing (2020)	sparse	yes	no	no
BP-Transformer (2019)	sparse	yes	MT	no
Blockwise (2019)	sparse	no	QA	yes
Our Longformer	sparse	yes	multiple	yes

2. Model Architecture

- **Transformer-XL**
  - Vanilla Transformer LM
  - Segment-Level Recurrence with State Reuse
  - Relative Positional Encodings
- **Longformer**
  - Attention Pattern
    - Sliding Window
    - Dilated Sliding Window
    - Global Attention

Vanilla Training

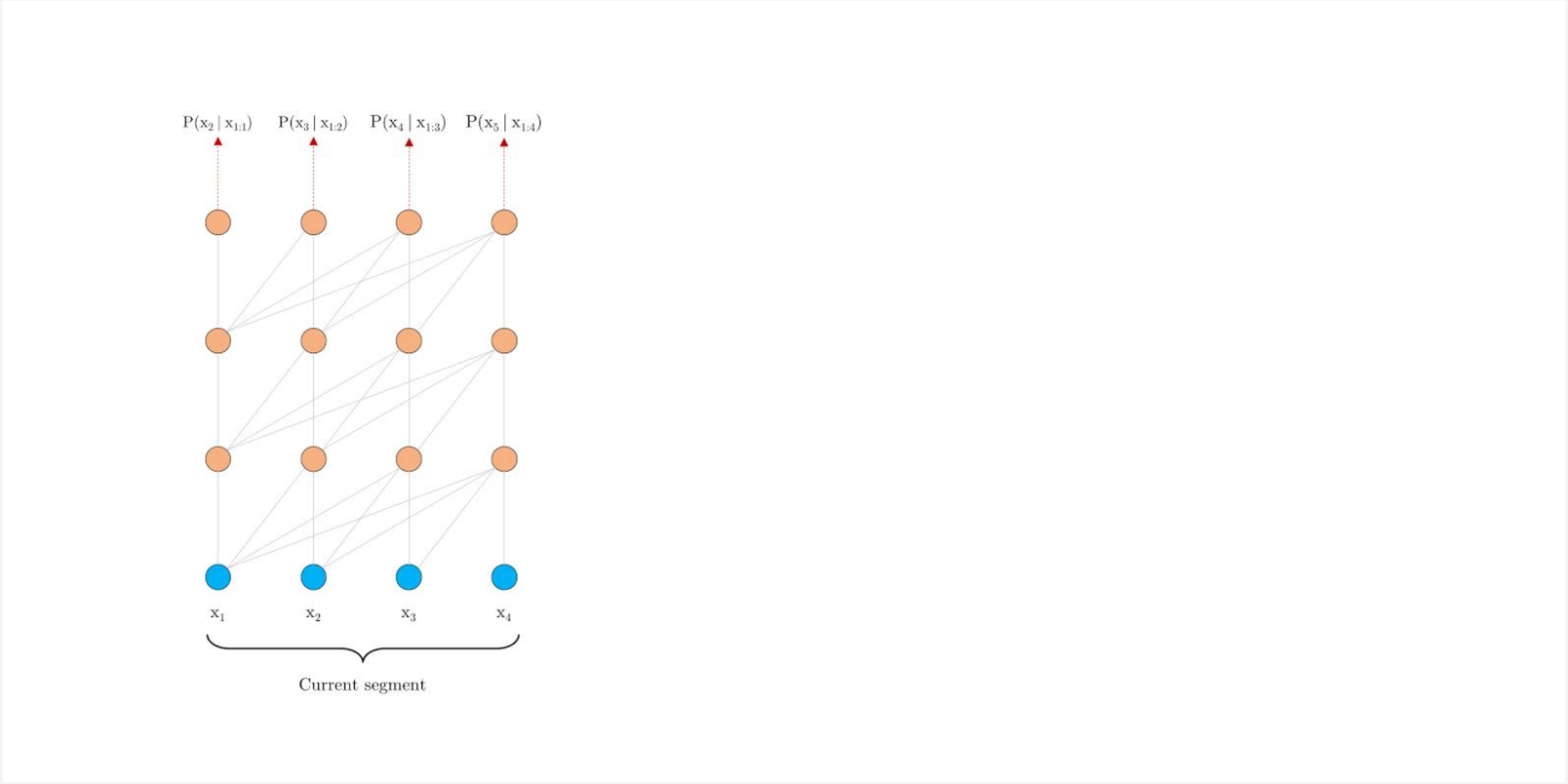
- Apply transformer to language modeling*



*Al-Rhou et al 2018, Character-Level Language Modeling with Deeper Self-Attention

Vanilla Training

- Information never flows across segments in either forward or backward pass

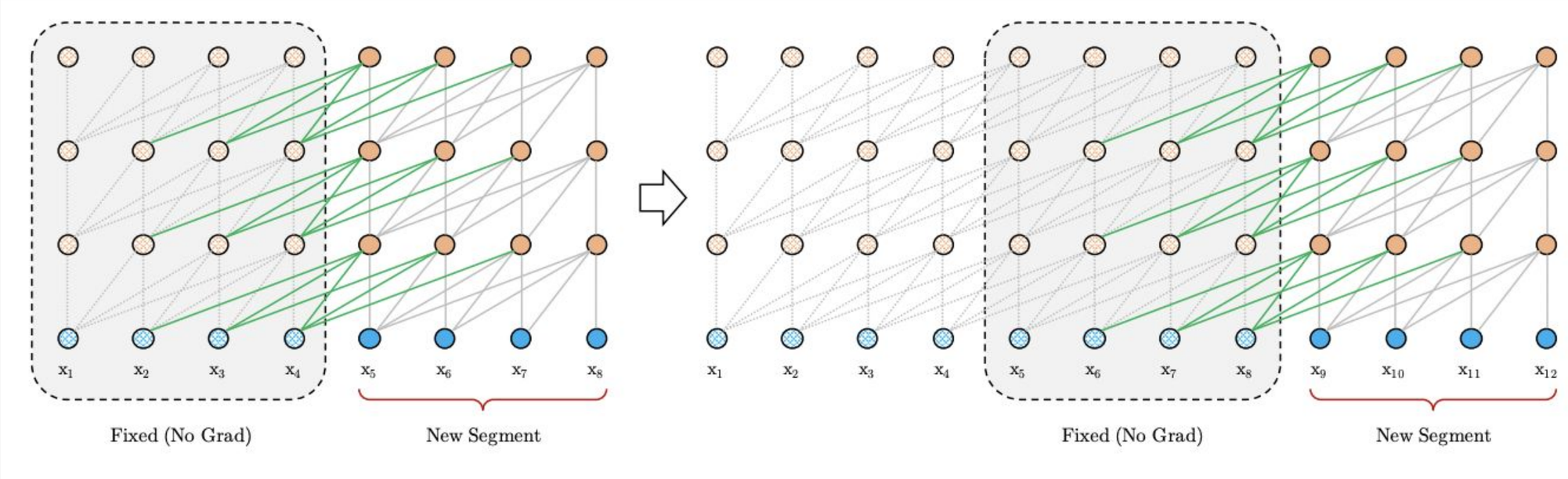


Vanilla Transformer with a fixed-length context at training time.

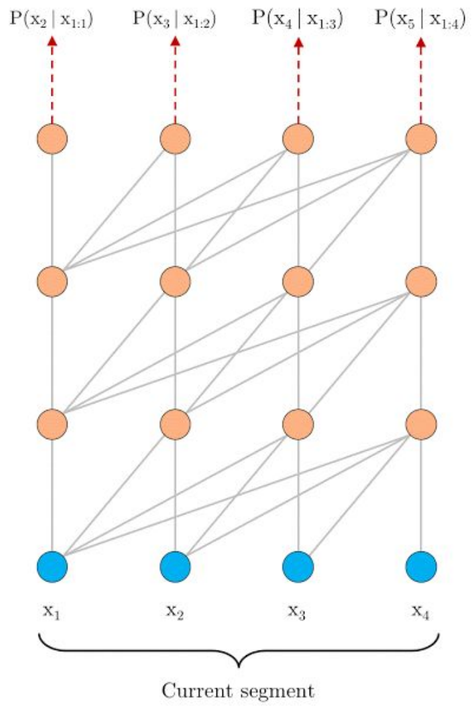
**Google AI blog, Transformer-XL: Unleashing the Potential of Attention Models*

Transformer-XL Training

- **Segment-Level Recurrence with State Reuse**
  - During training, the hidden state sequence computed for the previous segment is fixed and cached to be reused as an extended context
  - It increases the largest possible dependency by using contextual information from several previous segments



Transformer-XL Training



Transformer-XL with segment-level recurrence at training time.

**Google AI blog, Transformer-XL: Unleashing the Potential of Attention Models*

Transformer-XL Training

- **Relative Positional Encodings**
  - When reuse old hidden states, how can we define positional encodings?
  - It is enough to know **relative distance** between key vector and query vector (i - j)
    - Standard Transformer

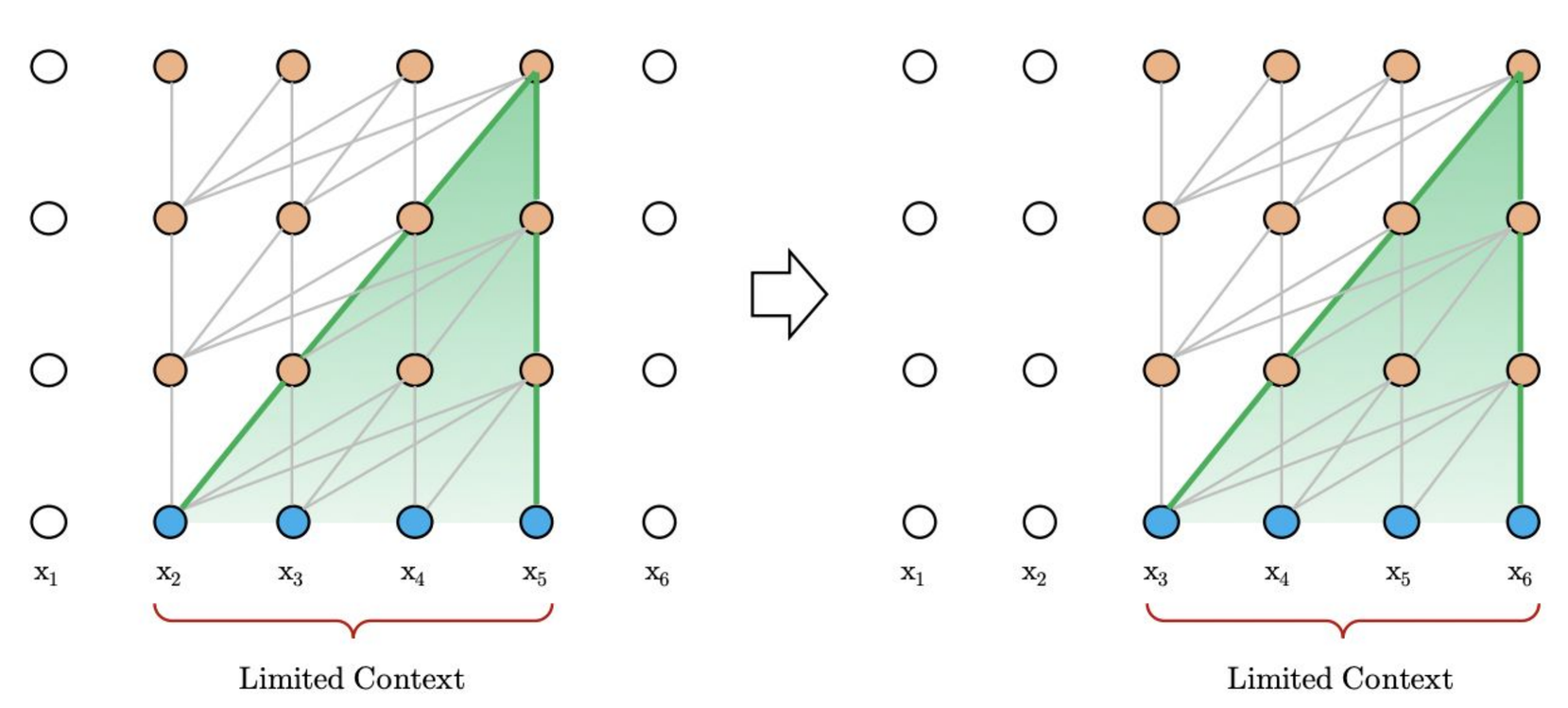
$$\mathbf{A}^{\text{abs}}_{i,j} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top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

    - This Paper(Transformer-XL)

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

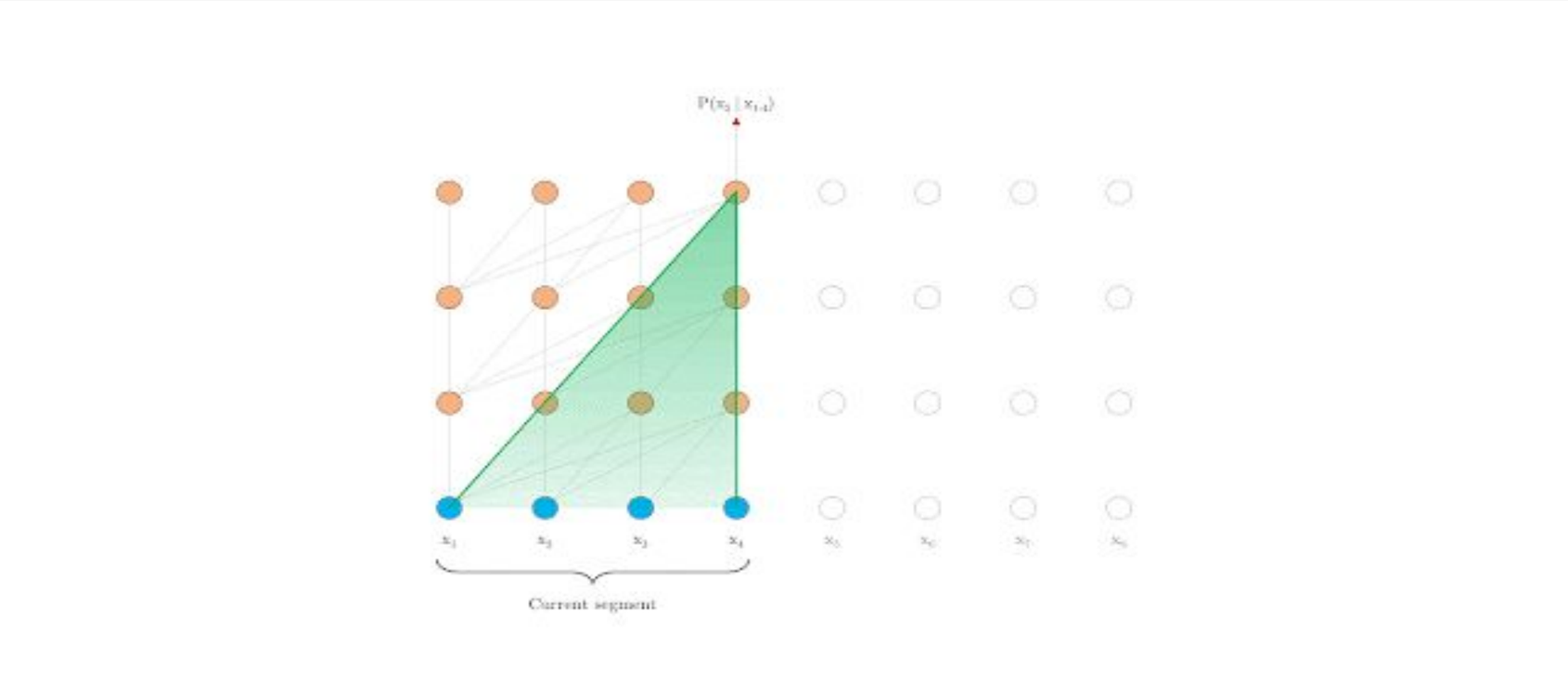
Vanilla Transformer Prediction

- During evaluation, vanilla model consumes a segment to **make only one prediction at the last position**, which is extremely expensive



Vanilla Transformer Prediction

- During evaluation, vanilla model consumes a segment to **make only one prediction at the last position**, which is extremely expensive

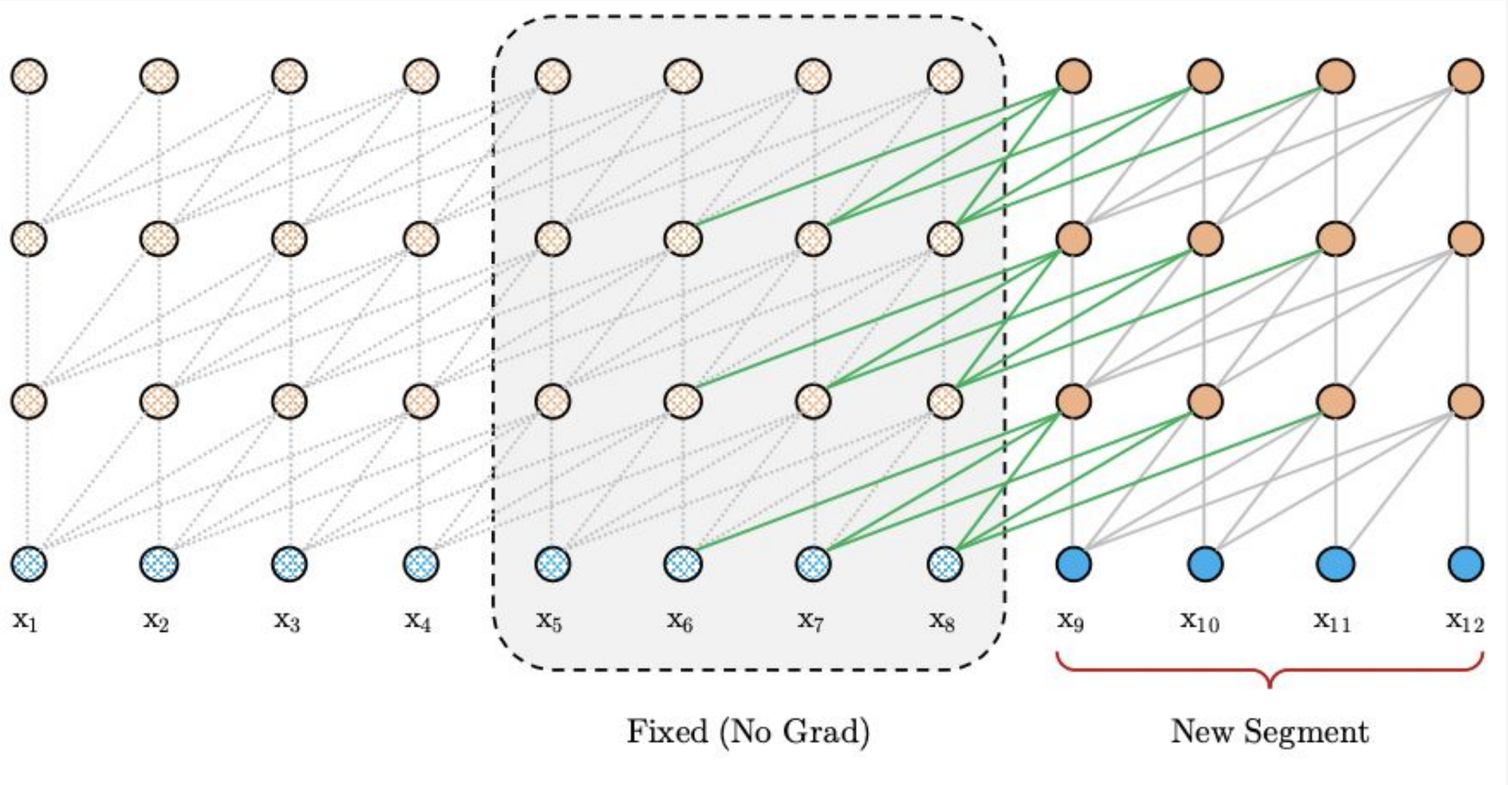


Vanilla Transformer with a fixed-length context at evaluation time.

**Google AI blog, Transformer-XL: Unleashing the Potential of Attention Models*

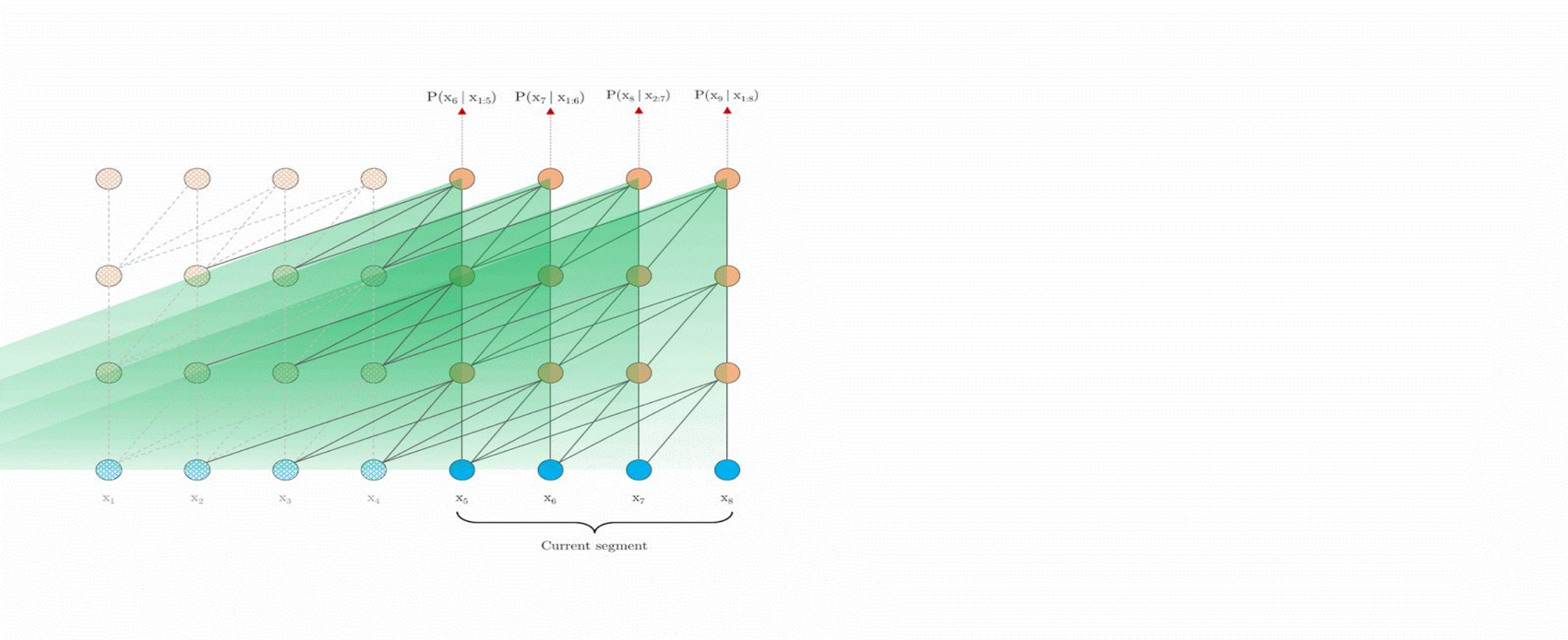
Transformer-XL Prediction

- Transformer-XL **uses representations (memory) form previous segments** instead of computing from scratch



Transformer-XL Prediction

- Transformer-XL **uses representations (memory) form previous segments** instead of computing from scratch

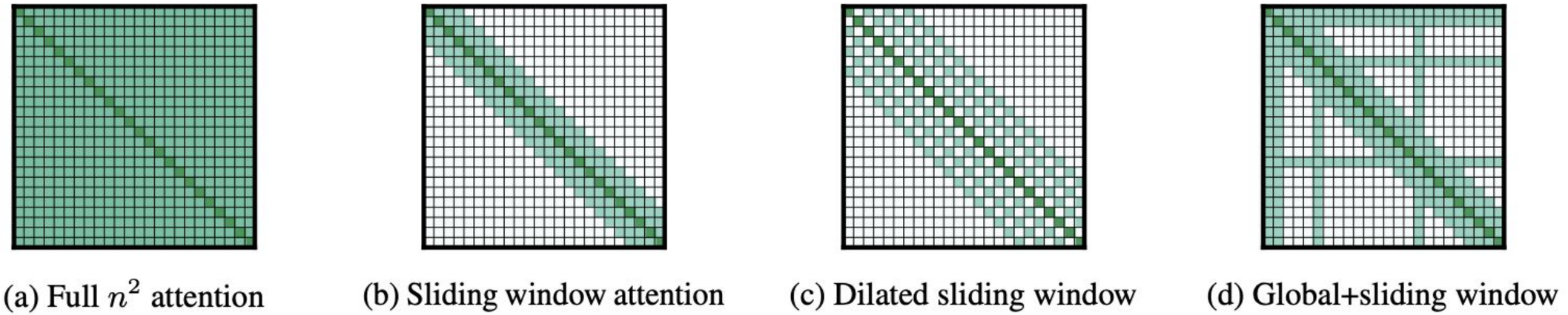


Transformer-XL with segment-level recurrence at evaluation time.

**Google AI blog, Transformer-XL: Unleashing the Potential of Attention Models*

Longformer

- **Attention Methods**
  - The original Transformer model has a self-attention component with **O(n^2)** time and memory complexity where n is the input sequence length.

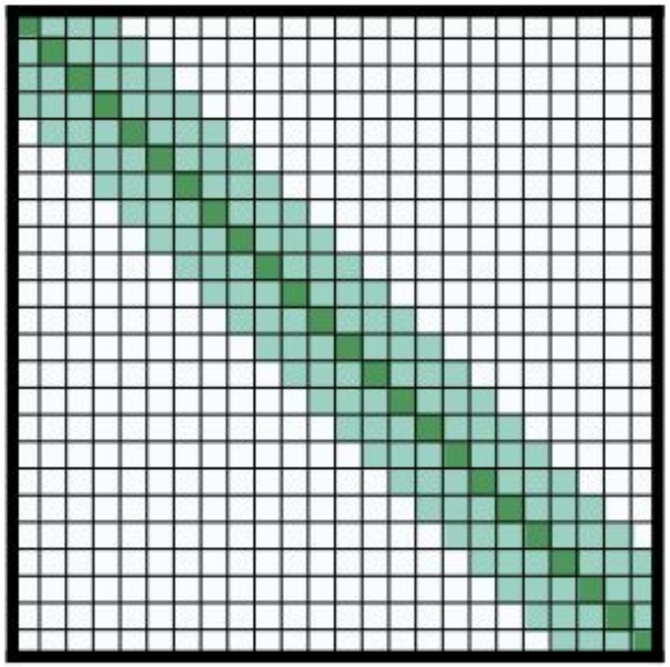


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

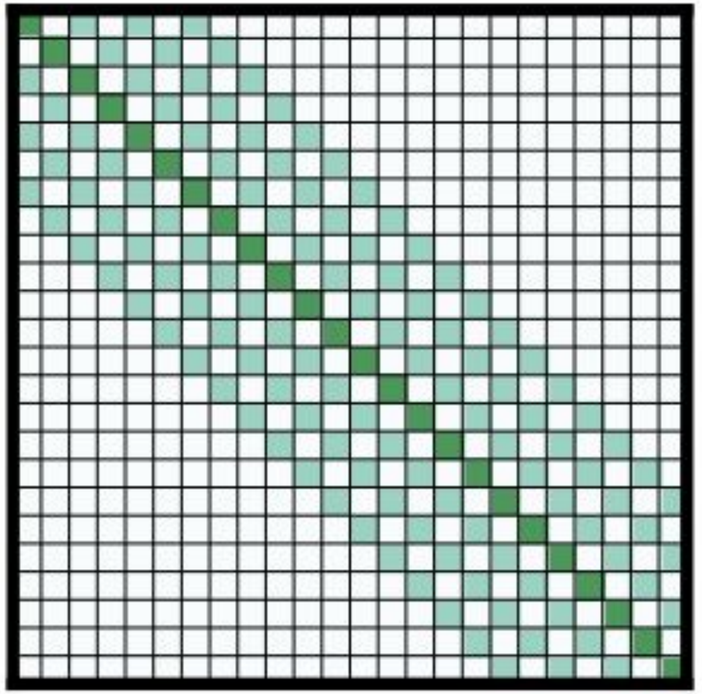
Attention Pattern

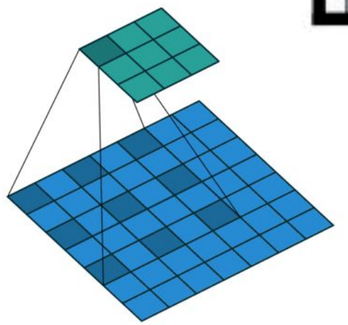
- **Sliding Window**
  - Given a fixed window size w, each token attends to **½ * w** tokens on each side
  - The computation complexity : **O(n x w)**
  - Receptive field : **ℓ * w**



Attention Pattern

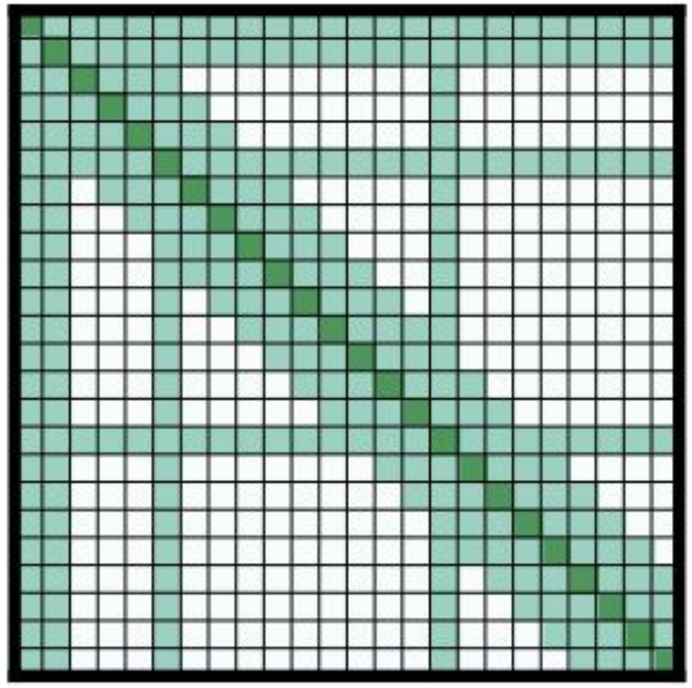
- **Dilated Sliding Window**
  - **Window를 dilated size만큼 확장**시켜 사용
  - Receptive field : **ℓ * d * w**
  - In multi-headed attention, **each attention head computes a different attention score**
  - Found settings with **different dilation configurations per head improves performance** by allowing some heads without dilation to focus on **local context**, while others with dilation focus on **longer context**





Attention Pattern

- **Global Attention**
  - In our case, the windowed and dilated attention are **not flexible enough to learn task-specific representations**. **Add “global attention”** on few pre-selected input locations
  - Sliding window attention with global attention at a few tokens at custom locations
  - The computation complexity : **O(n)**
  - It is an easy way to add inductive bias to the model’s attention, and it is much simpler than existing task specific approaches



3. Model Setup & Experimental Results

- **Transformer-XL**
  - Dataset / Evaluation
  - Main Results

- **Longformer**
  - Autoregressive Language Modeling
  - Pretraining and Fine-tuning
  - Tasks
  - Longformer-Encoder-Decoder(LED)

Transformer-XL Experiments

- **Dataset**
  - WikiText-103*
    - **Word-level** dataset with long-term dependency
    - 103M training tokens from 28K articles, average length of 3.6K tokens per article
  - enwiki-8
    - 100M bytes of unprocessed Wikipedia text
  - text-8
    - 100M processed Wikipedia **characters**
  - One Billion Word
    - Shuffled sentences (**No long-term dependency**)

Transformer-XL Experiments

- **Evaluation**
  - Perplexity (PPL)
  - Bit Per Character (BPC)
    - Character-level perplexity
  - **Relative Effective Context Length (RECL)**
    - Effective Context Length* : longest length to which increasing the context span would lead to a gain more than a threshold
    - RECL : relative improvement over the best short context model

Transformer-XL Experiments

- **Main Results**

Model	#Param	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) - QRNN	151M	33.0
Rae et al. (2018) - Hebbian + Cache	-	29.9
Ours - Transformer-XL Standard	151M	**24.0**
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

Table 1: Comparison with state-of-the-art results on WikiText-103. $^{\diamond}$ indicates contemporary work.

Model	#Param	bpc
Ha et al. (2016) - LN HyperNetworks	27M	1.34
Chung et al. (2016) - LN HM-LSTM	35M	1.32
Zilly et al. (2016) - RHN	46M	1.27
Mujika et al. (2017) - 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) - 12L Transformer	44M	1.11
Ours - 12L Transformer-XL	41M	**1.06**
Al-Rfou et al. (2018) - 64L Transformer	235M	1.06
Ours - 18L Transformer-XL	88M	1.03
Ours - 24L Transformer-XL	277M	**0.99**

Table 2: Comparison with state-of-the-art results on enwik8.

Model	#Param	PPL
Shazeer et al. (2014) - Sparse Non-Negative	33B	52.9
Chelba et al. (2013) - RNN-1024 + 9 Gram	20B	51.3
Kuchaiev and Ginsburg (2017) - G-LSTM-2	-	36.0
Dauphin et al. (2016) - GCNN-14 bottleneck	-	31.9
Jozefowicz et al. (2016) - LSTM	1.8B	30.6
Jozefowicz et al. (2016) - LSTM + CNN Input	1.04B	30.0
Shazeer et al. (2017) - Low-Budget MoE	~5B	34.1
Shazeer et al. (2017) - High-Budget MoE	~5B	28.0
Shazeer et al. (2018) - Mesh Tensorflow	4.9B	24.0
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	0.46B	24.1
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	1.0B	23.7
Ours - Transformer-XL Base	0.46B	23.5
Ours - Transformer-XL Large	0.8B	**21.8**

Table 4: Comparison with state-of-the-art results on One Billion Word. $^{\diamond}$ indicates contemporary work.

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

Table 3: Comparison with state-of-the-art results on text8.

Method	PPL
Ours	**25.2**
With Shaw et al. (2018) encodings	25.7
Without recurrence	27.1

Table 7: Ablation study on One Billion Word, a dataset without long-term dependency.

Transformer-XL Experiments

- **Ablation Study**
  - How fast

Attn Len	How much Al-Rfou et al. (2018) is slower
3,800	1,874x
2,800	1,409x
1,800	773x
800	363x

Table 9: Slowdown in terms of running time during evaluation. Evaluation is based on per-token time on one GPU.

  - How long

Model	$r = 0.1$	$r = 0.5$	$r = 1.0$
Transformer-XL 151M	**900**	**800**	**700**
QRNN	500	400	300
LSTM	400	300	200
Transformer-XL 128M	**700**	**600**	**500**
- use Shaw et al. (2018) encoding	400	400	300
- remove recurrence	300	300	300
Transformer	128	128	128

Table 8: Relative effective context length (RECL) comparison. See text for the definition of RECL and r. The first three models and the last four models are compared as two *model groups* when we calculate RECL (RECL is computed on a model group rather than a single model). Each group has the same parameter budget.

Longformer Experiments

- **Autoregressive Language Modeling**
- **Pretraining and Fine-tuning**
- **Tasks**
- **Longformer-Encoder-Decoder(LED)**

Autoregressive Language Modeling

- **Experiment Setup**
  - we found that the model needs a large number of gradient updates to learn the local context first, before learning to utilize longer context
  - Staged training procedure
- **Results**

Model	#Param	Dev	Test
Dataset `text8`			
T12 (Al-Rfou et al., 2018)	44M	-	1.18
Adaptive (Sukhbaatar et al., 2019)	38M	1.05	1.11
BP-Transformer (Ye et al., 2019)	39M	-	1.11
Our Longformer	41M	1.04	**1.10**
Dataset `enwik8`			
T12 (Al-Rfou et al., 2018)	44M	-	1.11
Transformer-XL (Dai et al., 2019)	41M	-	1.06
Reformer (Kitaev et al., 2020)	-	-	1.05
Adaptive (Sukhbaatar et al., 2019)	39M	1.04	1.02
BP-Transformer (Ye et al., 2019)	38M	-	1.02
Our Longformer	41M	1.02	**1.00**

Table 2: *Small* model BPC on `text8` & `enwik8`

Model	#Param	Test BPC
Transformer-XL (18 layers)	88M	1.03
Sparse (Child et al., 2019)	≈100M	0.99
Transformer-XL (24 layers)	277M	0.99
Adaptive (Sukhbaatar et al., 2019)	209M	0.98
Compressive (Rae et al., 2020)	277M	0.97
Routing (Roy et al., 2020)	≈223M	0.99
Our Longformer	102M	0.99

Table 3: Performance of *large* models on `enwik8`

Autoregressive Language Modeling

- **Ablation Study**
  - Observe that increasing the window size from the bottom to the top layer leads to the best performance
  - Tab. 4 shows the impact of adding dilation. Adding some dilation to two heads leads to some improvement compared with no dilation at all

Model	Dev BPC
Decreasing w (from 512 to 32)	1.24
Fixed w (= 230)	1.23
Increasing w (from 32 to 512)	**1.21**
No Dilation	1.21
Dilation on 2 heads	**1.20**

Table 4: Top: changing window size across layers. Bottom: with/without dilation (@ 150K steps on phase1)

Pretraining and Fine-tuning

- **Results**

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738
+ copy position embeddings	1.957	1.597
+ 2K gradient updates	1.753	1.414
+ 65K gradient updates	1.705	1.358
Longformer (train extra pos. embed. only)	1.850	1.504

Table 5: MLM BPC for RoBERTa and various pre-trained Longformer configurations.

Tasks

- the evaluation datasets have contexts significantly longer than 512 wordpieces
- Baseline : RoBERTa

Wordpieces	WH	TQA	HQA	ON	IMDB	HY
avg.	1,535	6,589	1,316	506	300	705
95th pctl.	3,627	17,126	1,889	1,147	705	1,975

Table 6: Average and 95th percentile of context length of datasets in wordpieces. WH: WikiHop, TQA: TriviaQA, HQA: HotpotQA, ON: OntoNotes, HY: Hyperpartisan news

Model	WikiHop	TriviaQA	HotpotQA
Current* SOTA	78.3	73.3	**74.2**
Longformer-large	**81.9**	**77.3**	73.2

Table 8: Leaderboard results of Longformer-large at time of submission (May 2020). All numbers are F1 scores.

Tasks

- **Results**

	QA			Coref.	Classification	
Model	WikiHop	TriviaQA	HotpotQA	OntoNotes	IMDB	Hyperpartisan
RoBERTa-base	72.4	74.3	63.5	78.4	95.3	87.4
Longformer-base	**75.0**	**75.2**	**64.4**	**78.6**	**95.7**	**94.8**

Table 7: Summary of finetuning results on QA, coreference resolution, and document classification. Results are on the development sets comparing our Longformer-base with RoBERTa-base. TriviaQA, Hyperpartisan metrics are F1, WikiHop and IMDB use accuracy, HotpotQA is joint F1, OntoNotes is average F1.

Longformer-Encoder-Decoder(LED)

	R-1	R-2	R-L
Discourse-aware (2018)	35.80	11.05	31.80
Extr-Abst-TLM (2020)	41.62	14.69	38.03
Dancer (2020)	42.70	16.54	38.44
Pegasus (2020)	44.21	16.95	38.83
LED-large (seqlen: 4,096) (ours)	44.40	17.94	39.76
BigBird (seqlen: 4,096) (2020)	**46.63**	19.02	41.77
LED-large (seqlen: 16,384) (ours)	**46.63**	**19.62**	**41.83**

Table 11: Summarization results of Longformer-Encoder-Decoder (LED) on the arXiv dataset. Metrics from left to right are ROUGE-1, ROUGE-2 and ROUGE-L.

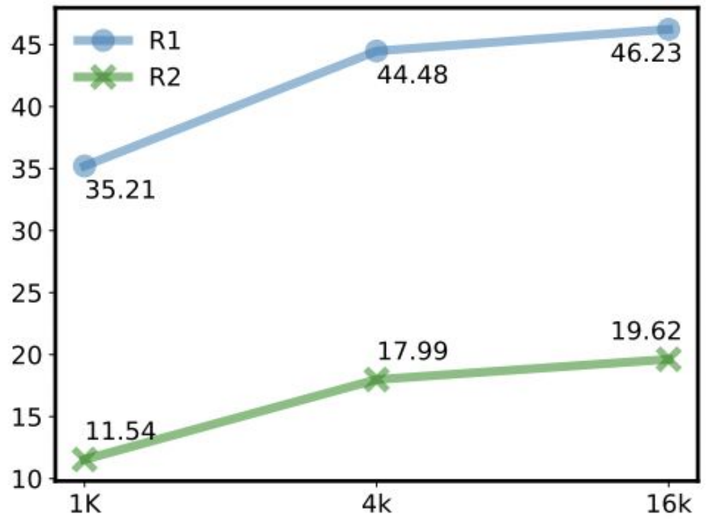


Figure 3: ROUGE-1 and ROUGE-2 of LED when varying the input size (arXiv validation set).

Summary

- **Transformer-XL**
  - Transformer model has weakness on **capturing long-term dependency** and **resolving context fragementations**
  - This paper suggests **segment-level recurrence** and **relative positional embedding** to solve above problem
  - Transformer-XL obtains **strong perplexity results**, models longer-term dependency than RNNs and Transformer, achieves substantial **speedup during evaluation**

Summary

- **Longformer**
  - **Drop-in replacement** for the standard self-attention and combines a local windowed attention with a task motivated global attention
  - Scalable for processing long documents and that makes it easy to perform a wide range of document-level NLP tasks without chunking/shortening the long input and without complex architecture to combine information across these chunks
  - Longformer achieves **SOTA results on the character-level language modeling** task of text8 and enwik8
  - **When pretrained**, Longformer **consistently outperforms RoBERTa on long document tasks** and sets **new SOTA results** on WikiHop and TriviaQA

Q & A

감사합니다.