Long Document Summarization

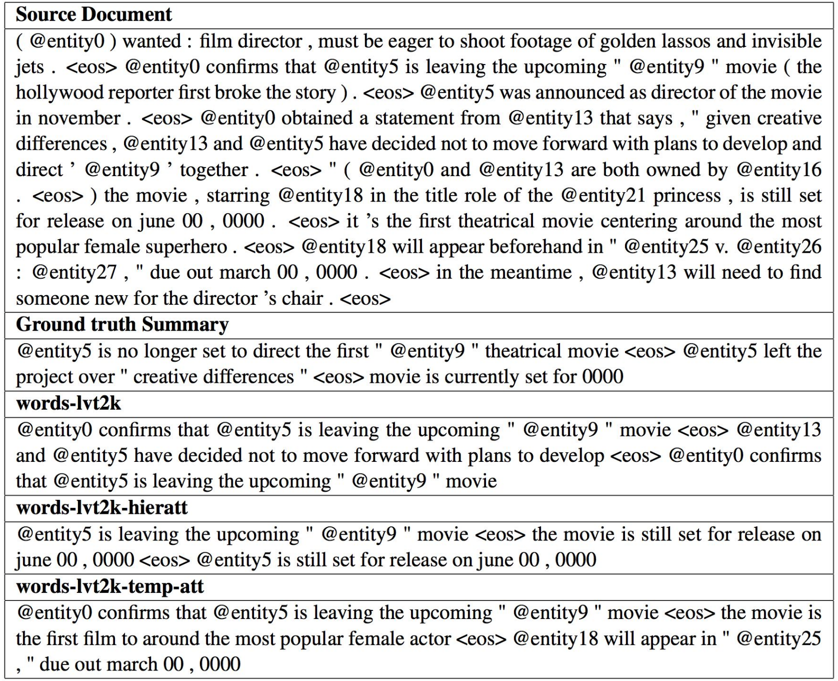
Qi Zeng
qizeng2@illinois.edu

Overview: Long-doc Summarization

- Challenges
- Methods
    - X-formers: *Efficient Transformers: A Survey*
    - Evaluation: *Long Range Arena: A Benchmark for Efficient Transformers*
    - HEPOS: *Efficient Attentions for Long Document Summarization*
    - Skyformer: *Remodel Self-Attention with Gaussian Kernel and Nystr"om Method*
    - Skeinformer: *Sketching as a Tool for Understanding and Accelerating Self-attention for Long Sequences*
- Resources
    - Datasets and tools

Introduction

Quick Recap: Short-doc Summarization



Source Document
(@entity0) wanted : film director , must be eager to shoot footage of golden lassos and invisible jets . <eos> @entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie (the hollywood reporter first broke the story) . <eos> @entity5 was announced as director of the movie in november . <eos> @entity0 obtained a statement from @entity13 that says , " given creative differences , @entity13 and @entity5 have decided not to move forward with plans to develop and direct ' @entity9 ' together . <eos> " (@entity0 and @entity13 are both owned by @entity16 . <eos>) the movie , starring @entity18 in the title role of the @entity21 princess , is still set for release on june 00 , 0000 . <eos> it 's the first theatrical movie centering around the most popular female superhero . <eos> @entity18 will appear beforehand in " @entity25 v. @entity26 : @entity27 , " due out march 00 , 0000 . <eos> in the meantime , @entity13 will need to find someone new for the director 's chair . <eos>
Ground truth Summary
@entity5 is no longer set to direct the first " @entity9 " theatrical movie <eos> @entity5 left the project over " creative differences " <eos> movie is currently set for 0000
words-lvt2k
@entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie <eos> @entity13 and @entity5 have decided not to move forward with plans to develop <eos> @entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie
words-lvt2k-hieratt
@entity5 is leaving the upcoming " @entity9 " movie <eos> the movie is still set for release on june 00 , 0000 <eos> @entity5 is still set for release on june 00 , 0000
words-lvt2k-temp-att
@entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie <eos> the movie is the first film to around the most popular female actor <eos> @entity18 will appear in " @entity25 , " due out march 00 , 0000

CNN/Daily Mail dataset (Nallapati et al, CONLL 2016)

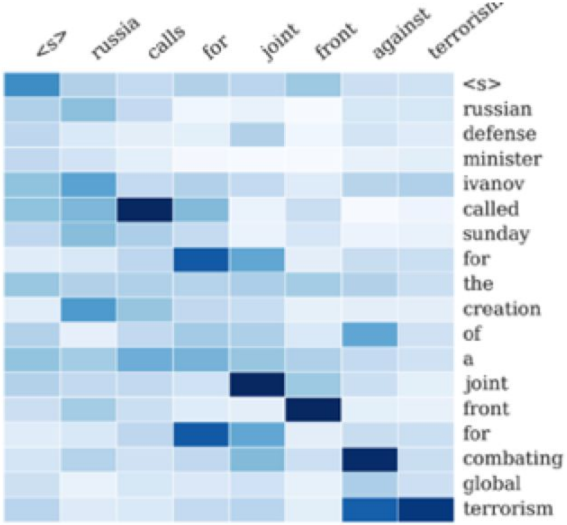


Figure 1: Example output of the attention-based summarization (ABS) system. The heatmap represents a soft alignment between the input (right) and the generated summary (top). The columns represent the distribution over the input after generating each word.

Neural attention model for sentence summarization (Rush et al., EMNLP 2015)

Challenges: From short-doc to long-doc

- General challenges for long sequence processing
  - Computation complexity due to length of sequence
  - The inter-dependency among processing units
  - The difficulty of information aggregation across long context
  - The lack of available datasets due to costly annotation
- Task-Specific challenges
  - **Summarization**: encoder-decoder attentions collaborate and dynamically pinpoint salient content in the source as the summary is decoded (Huang et al, NAACL 2021)
  - **Machine Translation**: existing approaches simply introduce the representations of context sentences without explicitly characterizing the inter-sentence reasoning process; and feed ground-truth target contexts as extra inputs at the training time, thus facing the problem of exposure bias (Zhang et al, NAACL 2021)
  - **Document Ranking**: remote connections between terms are ignored or captured by simple patterns (Zhou et al, 2021)
  - **Event Argument Extraction**: Sentence-level argument extraction approaches cannot handle the cross-sentence trigger-argument distribution and the existence of multiple events within one document (our paper in submission)

Methods

Method: Long-Doc Summarization

- Retrieval
- Graph
- Memory
- Hierarchy
- Attention

X-former: Transformer Acceleration Methods

- **Pruning redundant attention head**

Analyzing multi-head self-attention:Specialized heads do the heavy lifting, the rest can be pruned. Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, Ivan Titov. https://arxiv.org/abs/1905.09418

Are sixteen heads really better than one? Paul Michel, Omer Levy, Graham Neubig. https://arxiv.org/abs/1905.10650

- **model size reduction with knowledge distillation**

Tinybert: Distilling BERT for natural language understanding. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, Qun Liu. https://arxiv.org/abs/1909.10351

Fastbert: a self-distilling BERT with adaptive inference time. Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, Qi Ju. https://arxiv.org/abs/2004.02178

Distilling task-specific knowledge fromBERT into simple neural networks. Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, Jimmy Lin. https://arxiv.org/abs/1903.12136

- **Attention Approximation/Acceleration**

Skyformer: Remodel Self-Attention with Gaussian Kernel and Nystr"om Method. Yifan Chen*, Qi Zeng*, Heng Ji, Yun Yang. https://arxiv.org/abs/2111.00035

Sketching as a Tool for Understanding and Accelerating Self-attention for Long Sequences. Yifan Chen*, Qi Zeng*, Dilek Hakkani-Tur, Di Jin, Heng Ji, Yun Yang https://arxiv.org/pdf/2112.05359.pdf

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

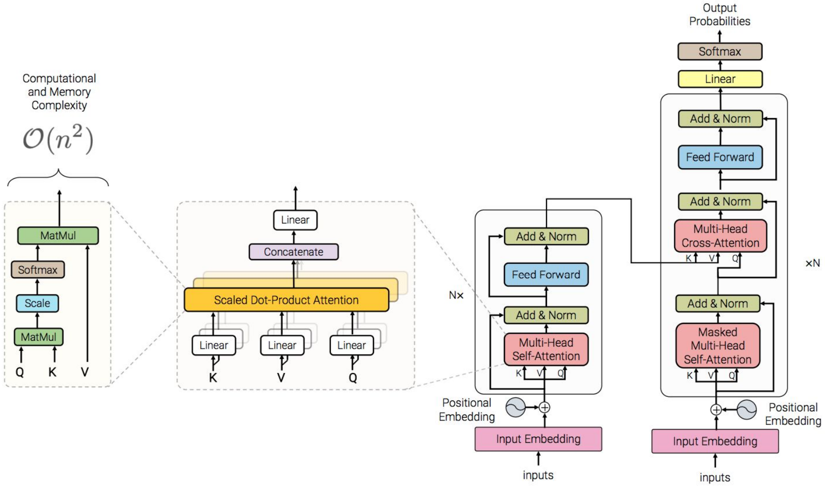
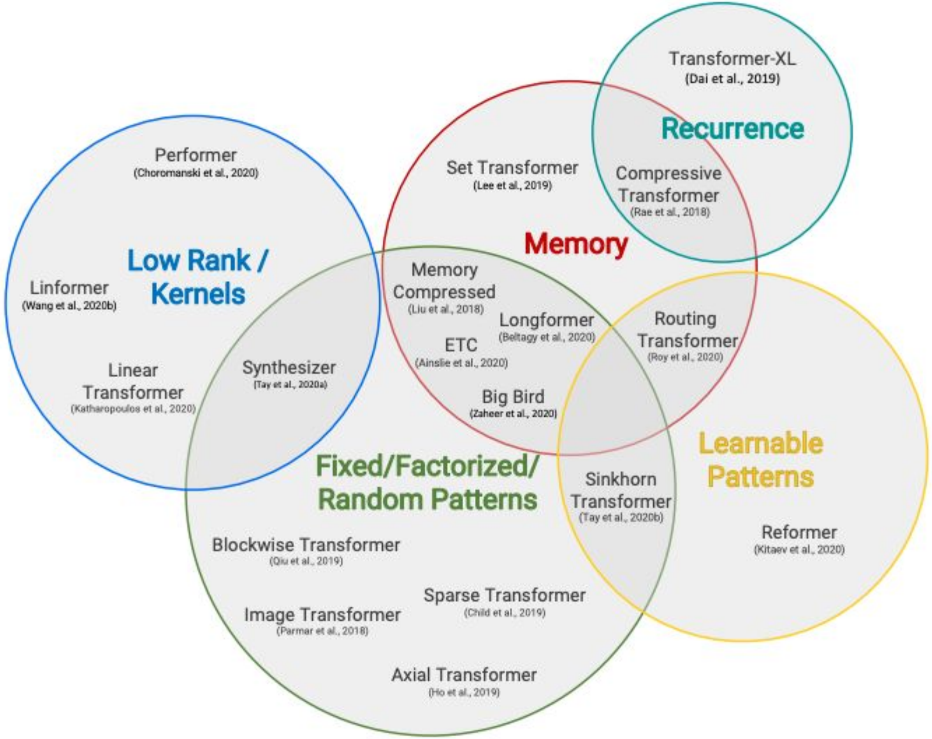


Figure 1: Architecture of the standard Transformer (Vaswani et al., 2017)

X-former: fast attention for efficient transformers



Transformer-XL
(Dai et al., 2019)

Recurrence

Compressive Transformer
(Rae et al., 2018)

Set Transformer
(Lee et al., 2019)

Memory

Performer
(Choromanski et al., 2020)

Low Rank / Kernels

Linformer
(Wang et al., 2020b)

Linear Transformer
(Katharopoulos et al., 2020)

Synthesizer
(Tay et al., 2020a)

Memory Compressed
(Liu et al., 2018)

Longformer
(Beltagy et al., 2020)

ETC
(Ainslie et al., 2020)

Big Bird
(Zaheer et al., 2020)

Routing Transformer
(Roy et al., 2020)

Learnable Patterns

Sinkhorn Transformer
(Tay et al., 2020b)

Reformer
(Kitaev et al., 2020)

Fixed/Factorized/ Random Patterns

Blockwise Transformer
(Qiu et al., 2019)

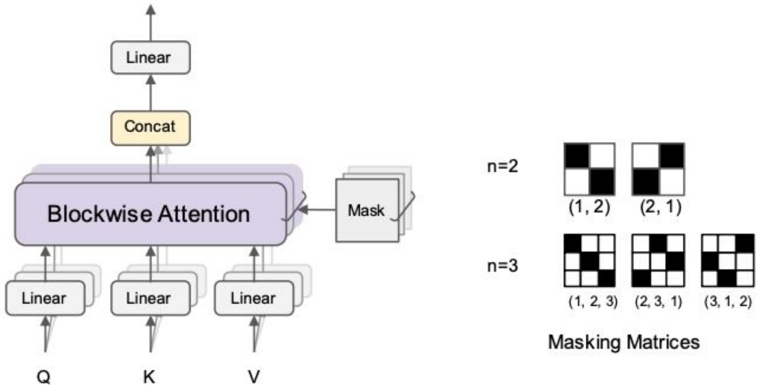
Sparse Transformer
(Child et al., 2019)

Image Transformer
(Parmar et al., 2018)

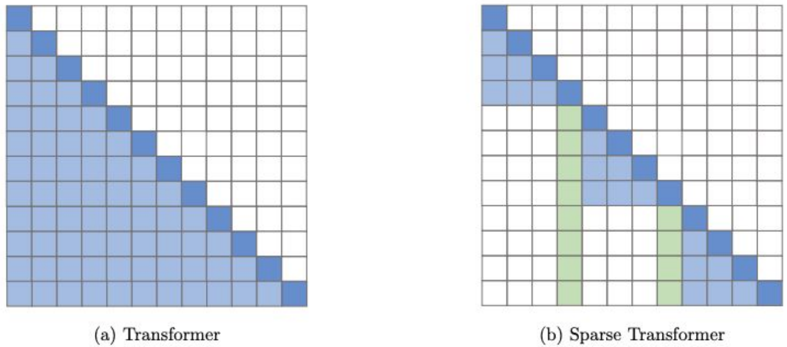
Axial Transformer
(Ho et al., 2019)

Fixed Patterns (FP)

- Limit the field of view to fixed, predefined patterns



Blockwise Pattern: BlockBERT [8]

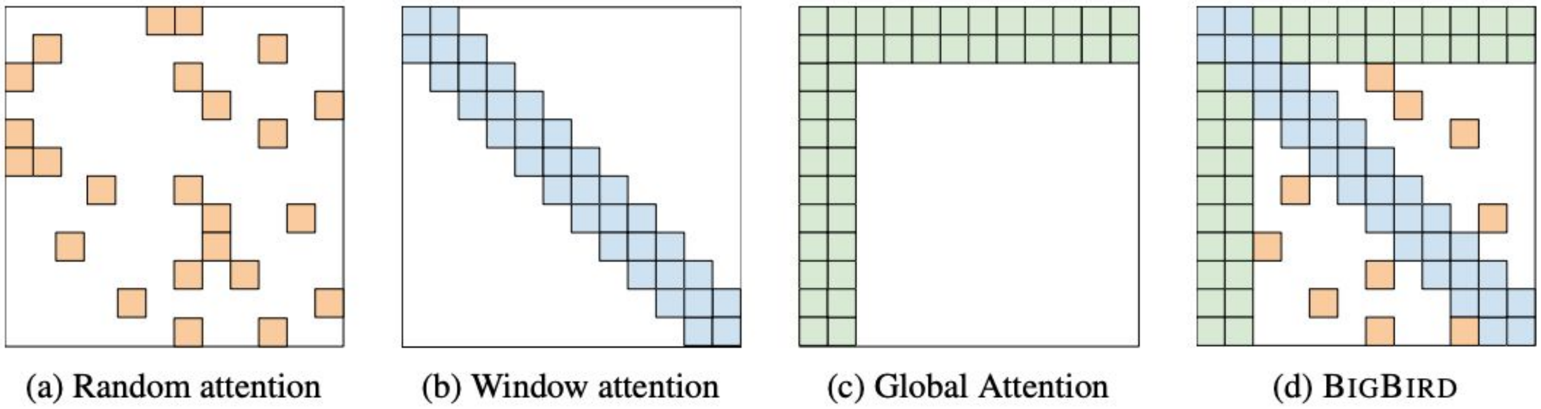

(a) Transformer

(b) Sparse Transformer

Strided/Dilated Pattern: Sparse Transformer [9]

Combination of Patterns (CP)

- Combine two or more distinct access patterns



(a) Random attention

(b) Window attention

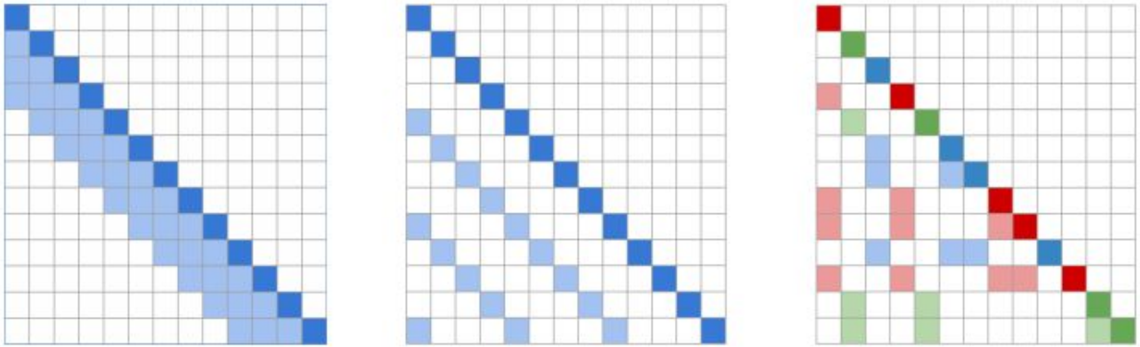
(c) Global Attention

(d) BIGBIRD

BIGBIRD [12]

Learnable Patterns (LP)

- Learn the access pattern in a data-driven fashion
- Determine the token relevance and assign tokens to buckets or clusters

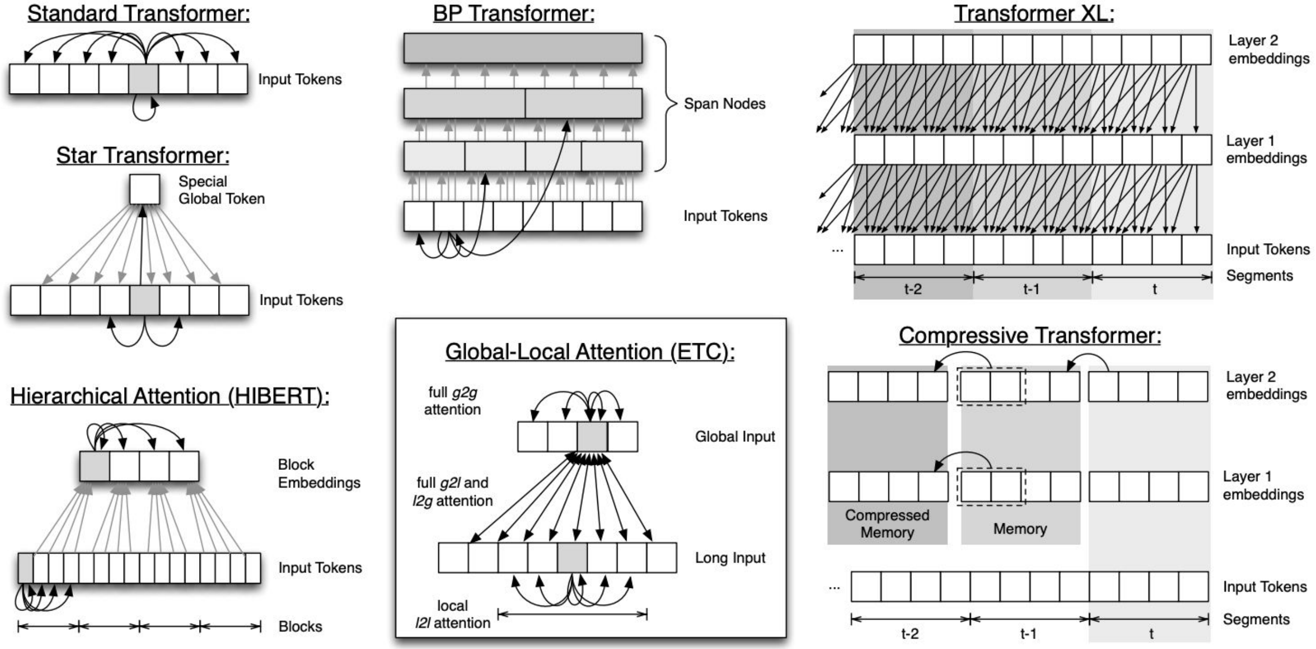


(a) Local attention (b) Strided attention (c) Routing attention

Routing Transformers [14]

Memory (M)

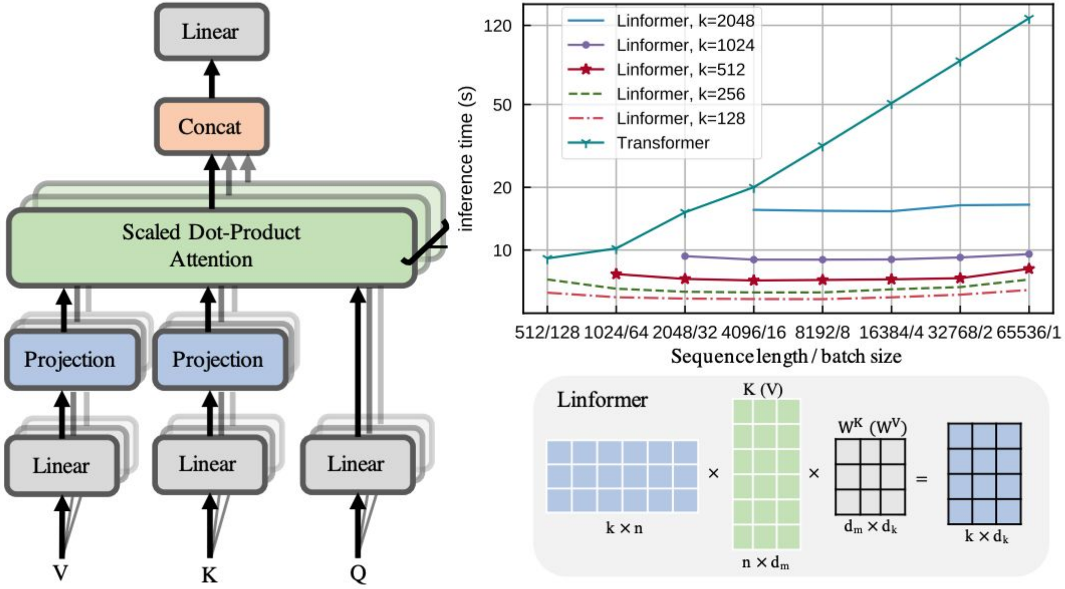
- Access multiple tokens at once with Global Memory



ETC [15]

Low-Rank Methods (LR)

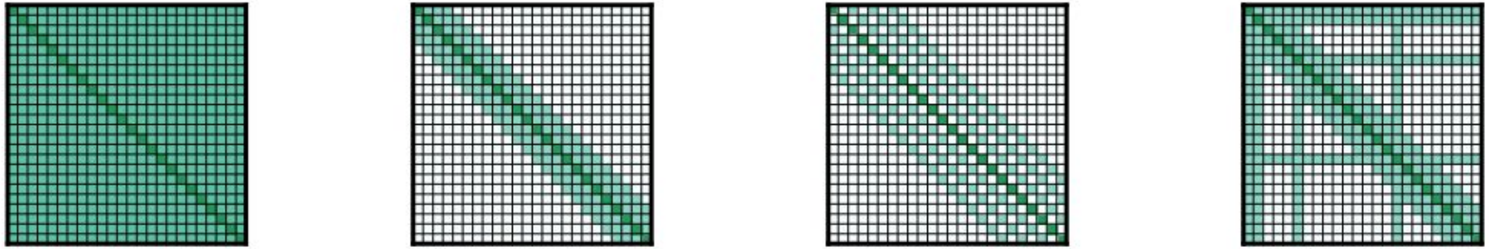
- Assume low-rank structure in the N×N matrix



Linformer [16]

X-former: Longformer

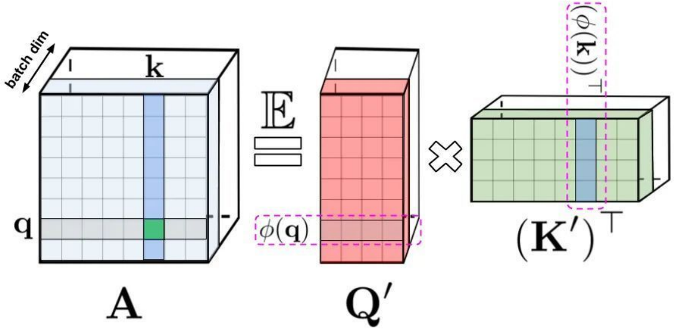
- Combines a local windowed attention with a task motivated global attention
  - Motivation of global attention: the optimal input representation varies by task

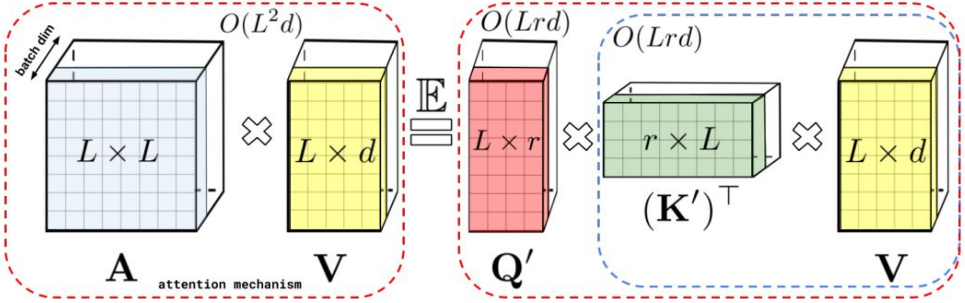


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

X-former: Performer

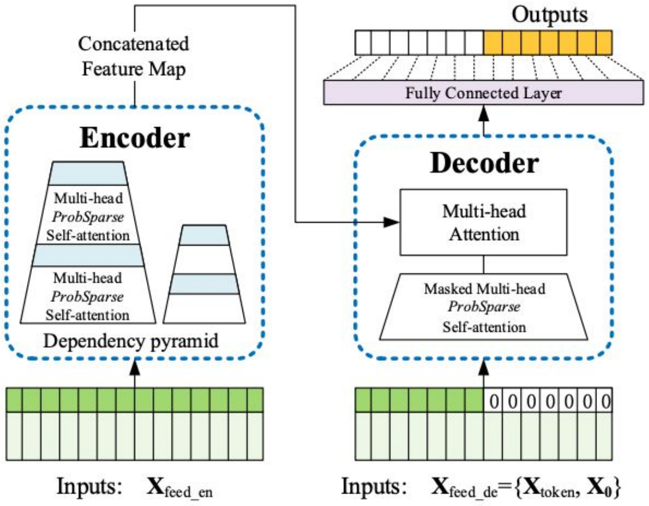
- FAVOR+ : Fast Attention Via positive Orthogonal Random features
  - FA $A = exp(QK^T) \approx \phi(Q)\phi(K)^T = Q^{'}(K^{'})^T$
  - R+ $\phi(\mathbf{x}) = \frac{h(\mathbf{x})}{\sqrt{m}}(f_1(\omega_1^\top \mathbf{x}), ..., f_1(\omega_m^\top \mathbf{x}), ..., f_l(\omega_1^\top \mathbf{x}), ..., f_l(\omega_m^\top \mathbf{x}))$
  - O entangle different random samples ω1, ..., ωm to be exactly orthogonal

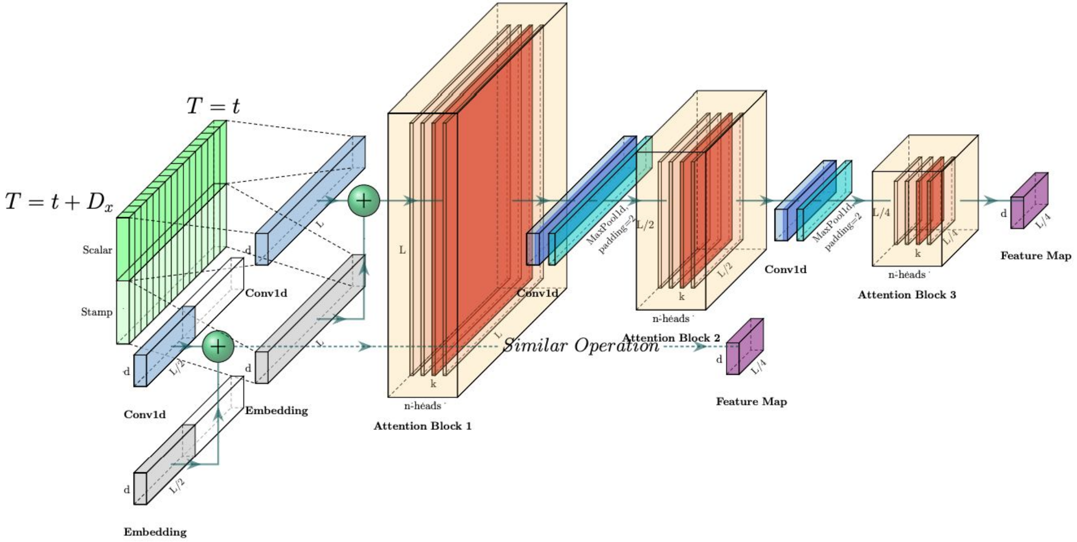




X-former: Informer

- ProbSparse self-attention
- Self-attention distillation


Outputs
Concatenated Feature Map
Fully Connected Layer
Encoder
Multi-head ProbSparse Self-attention
Multi-head ProbSparse Self-attention
Dependency pyramid
Decoder
Multi-head Attention
Masked Multi-head ProbSparse Self-attention
0 0 0 0 0 0 0
Inputs: $\mathbf{X}_{\text{feed\_en}}$
Inputs: $\mathbf{X}_{\text{feed\_de}}=\{\mathbf{X}_{\text{token}}, \mathbf{X}_{\mathbf{0}}\}$

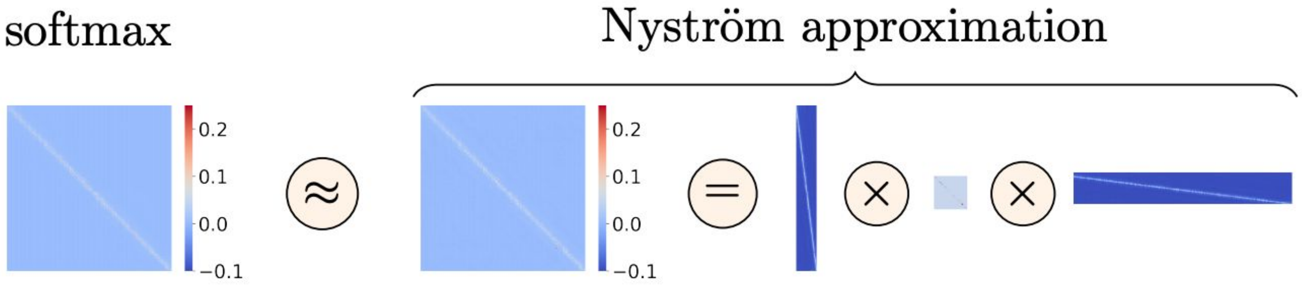


X-former: Nyströmformer

- Nyström Method is widely adopted for matrix approximation
- Sample a subset of columns and rows
    - Rewrite attention matrix, approximate S via the basic quadrature technique of the Nyström Method
    - Challenge
    - Using landmarks

$$S = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right) = \begin{bmatrix} A_S & B_S \\ F_S & C_S \end{bmatrix} \quad \hat{S} = \begin{bmatrix} A_S & B_S \\ F_S & F_S A_S^+ B_S \end{bmatrix} = \begin{bmatrix} A_S \\ F_S \end{bmatrix} A_S^+ \begin{bmatrix} A_S & B_S \end{bmatrix} \quad \hat{S} = \left[\text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)\right]_{n\times m} A_S^+ \left[\text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)\right]_{m\times n}$$

$$\hat{S} = \text{softmax}\left(\frac{Q\tilde{K}^T}{\sqrt{d_q}}\right)\left(\text{softmax}\left(\frac{\tilde{Q}\tilde{K}^T}{\sqrt{d_q}}\right)\right)^+ \text{softmax}\left(\frac{\tilde{Q}K^T}{\sqrt{d_q}}\right)$$

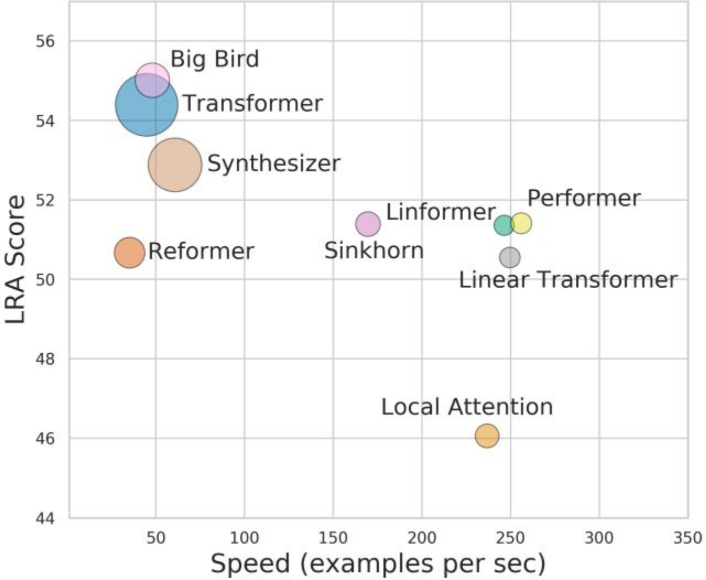


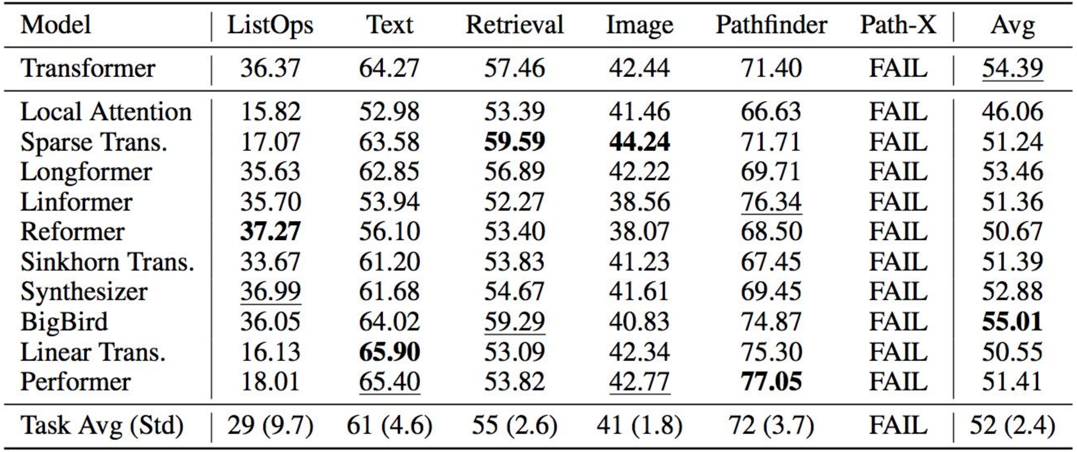
Long Range Arena: A Benchmark for Efficient Transformers

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, Donald Metzler

LRA: Evaluation Benchmark for X-formers

This paper proposes a systematic and unified benchmark specifically focused on evaluating model quality under long-context scenarios.



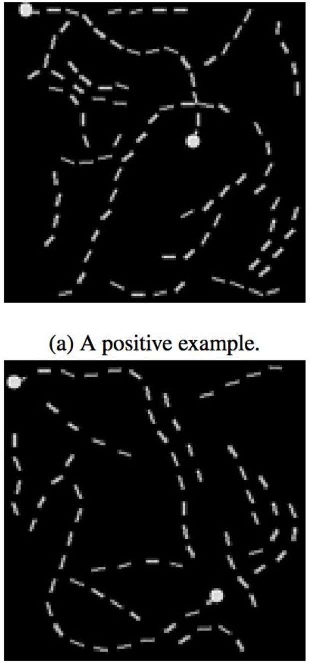


Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	FAIL	54.39
Local Attention	15.82	52.98	53.39	41.46	66.63	FAIL	46.06
Sparse Trans.	17.07	63.58	**59.59**	**44.24**	71.71	FAIL	51.24
Longformer	35.63	62.85	56.89	42.22	69.71	FAIL	53.46
Linformer	35.70	53.94	52.27	38.56	76.34	FAIL	51.36
Reformer	**37.27**	56.10	53.40	38.07	68.50	FAIL	50.67
Sinkhorn Trans.	33.67	61.20	53.83	41.23	67.45	FAIL	51.39
Synthesizer	36.99	61.68	54.67	41.61	69.45	FAIL	52.88
BigBird	36.05	64.02	59.29	40.83	74.87	FAIL	**55.01**
Linear Trans.	16.13	**65.90**	53.09	42.34	75.30	FAIL	50.55
Performer	18.01	65.40	53.82	42.77	**77.05**	FAIL	51.41
Task Avg (Std)	29 (9.7)	61 (4.6)	55 (2.6)	41 (1.8)	72 (3.7)	FAIL	52 (2.4)

LRA: Evaluation Benchmark for X-formers

Six classification tasks for transformer encoder

- **ListOps**: seq_len=2k
  - INPUT: [MAX 4 3 [MIN 2 3] 1 0 [MEDIAN 1 5 8 9, 2]]
    OUTPUT: 5
- **Byte-level Text Classification**: IMDb reviews, seq_len=4k
- **Byte-level Document Retrieval**: ANN dataset, seq_len=4k, two-tower setting
- **Image Classification**: CIFAR-10, seq_len=1k
- **Pathfinder** (long-range spatial dependency): seq_len=1k (32*32)
- **Pathfinder-X**: seq_len=16k (128*128)



(a) A positive example.

(b) A negative example.

LRA: Comparisons are made with 2-layer transformers

Why not larger or deeper transformers?

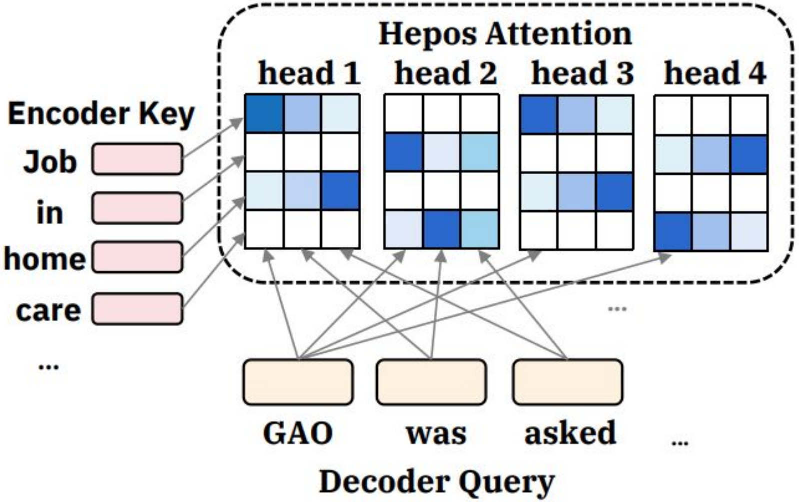
- Training a deep transformer from scratch requires large computational resources and much more data to converge, and therefore is not adopted by previous work.
- A shallow transformer structure, on the other hand, has been justified by previous work to be enough for fair comparison in attention acceleration performance.
- Pretrained models, like **BERT**, are trained for token-level text-based tasks, and are not suitable for image pixel sequences (as in Pathfinder and Image Classification), character sequences (as in Text Classification) and math operation sequences (as in ListOps).

Efficient Attentions for Long Document Summarization

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, Lu Wang

NAACL 2021

HEPOS: Efficient Attentions for Long Document Summarization



Sketching as a Tool for Understanding and Accelerating Self-attention for Long Sequences

Yifan Chen*, Qi Zeng*, Dilek Hakkani-Tur, Di Jin, Heng Ji, Yun Yang

Skeinformer: Background

Revisit Self-Attention

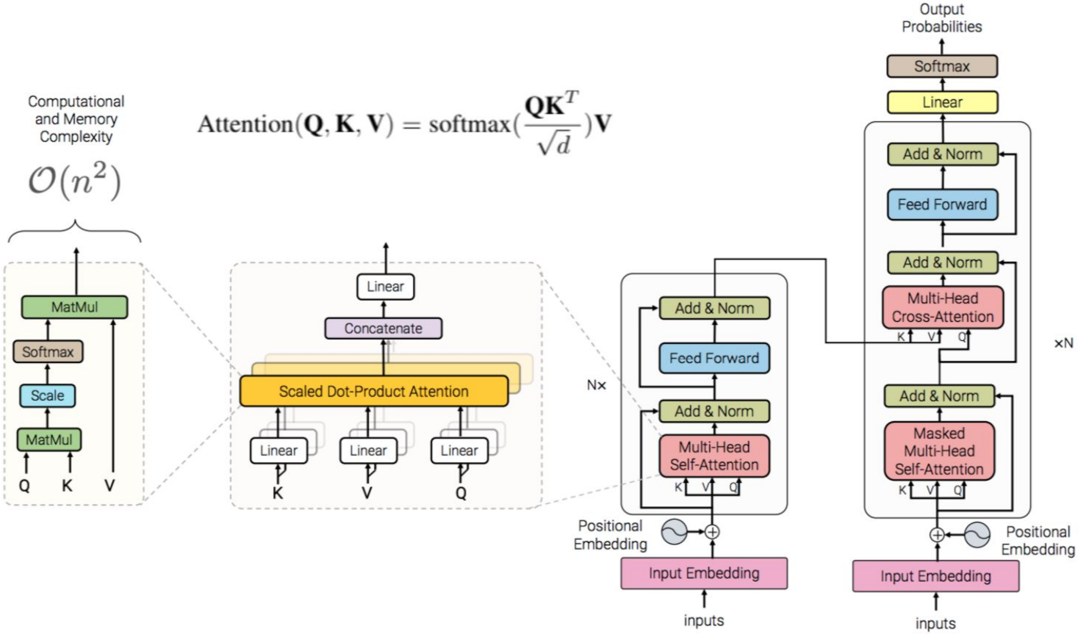


Figure 1: Architecture of the standard Transformer (Vaswani et al., 2017)

Skeinformer: Motivation - Sketching Framework

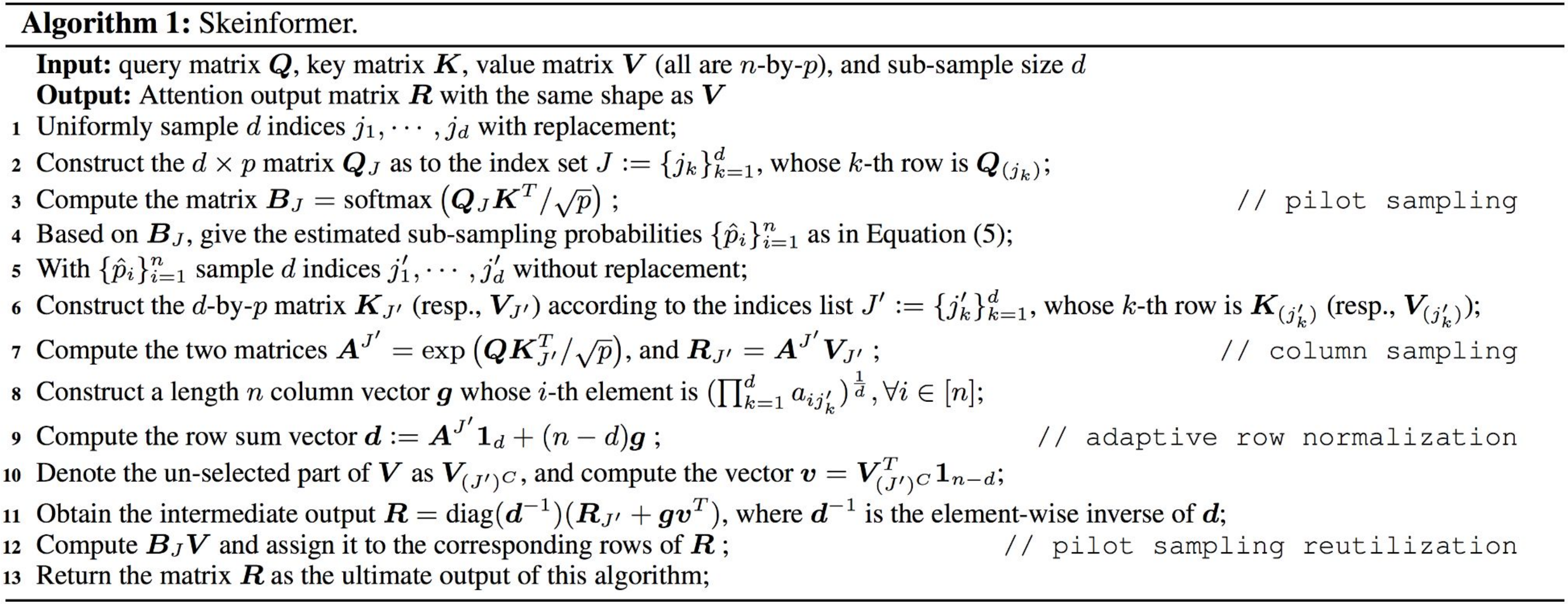
Linformer and Informer can be unified into the sketching framework:

- Sketching methods replace the original matrix B with its random sketch BS.
- Informer selects d important rows of $\boldsymbol{D}^{-1}\boldsymbol{A}$, which can be can be related to a sketched approximation $\boldsymbol{D}^{-1}\boldsymbol{S}\boldsymbol{S}^{T}\boldsymbol{A}$
- Linformer adds two linear projection layers, $((\boldsymbol{Q}\boldsymbol{K}^{T}/\sqrt{p})\boldsymbol{S})\ \boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{V}$ which can be related to a sketched approximation $\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{S}\boldsymbol{S}^{T}\boldsymbol{V}$

Skeinformer: Method

- Column Sampling in the sketching form $\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{S}\boldsymbol{S}^{T}\boldsymbol{V}$
  - Sampling probability in sub-sampling matrix S $p_i \propto \|(\boldsymbol{D}^{-1}\boldsymbol{A})^{(i)}\|_2 \|\boldsymbol{V}_{(i)}\|_2, \quad i = 1, 2, \ldots, n.$
  - Advantages:
    - This sketching form circumvents the computation burden of Gaussian sketching
    - This sketching form allows the incorporation of information from V
    - The construction of S theoretically guarantees the approximation performance in terms of Frobenius norm loss
- Adaptive Row Normalization $\hat{d}_{ii} = \sum_{k=1}^{d} a_{ij_k} + (n-d)(\prod_{k=1}^{d} a_{ij_k})^{\frac{1}{d}}$
  - Filling the unselected columns with the averaged selected columns
  - Advantage: It allows the whole value matrix V to participate in the computation, which will improve the efficiency of updating W_V
- Pilot Sampling Reutilization
  - Reproduce the d ros in the original self-attention output with an additional product

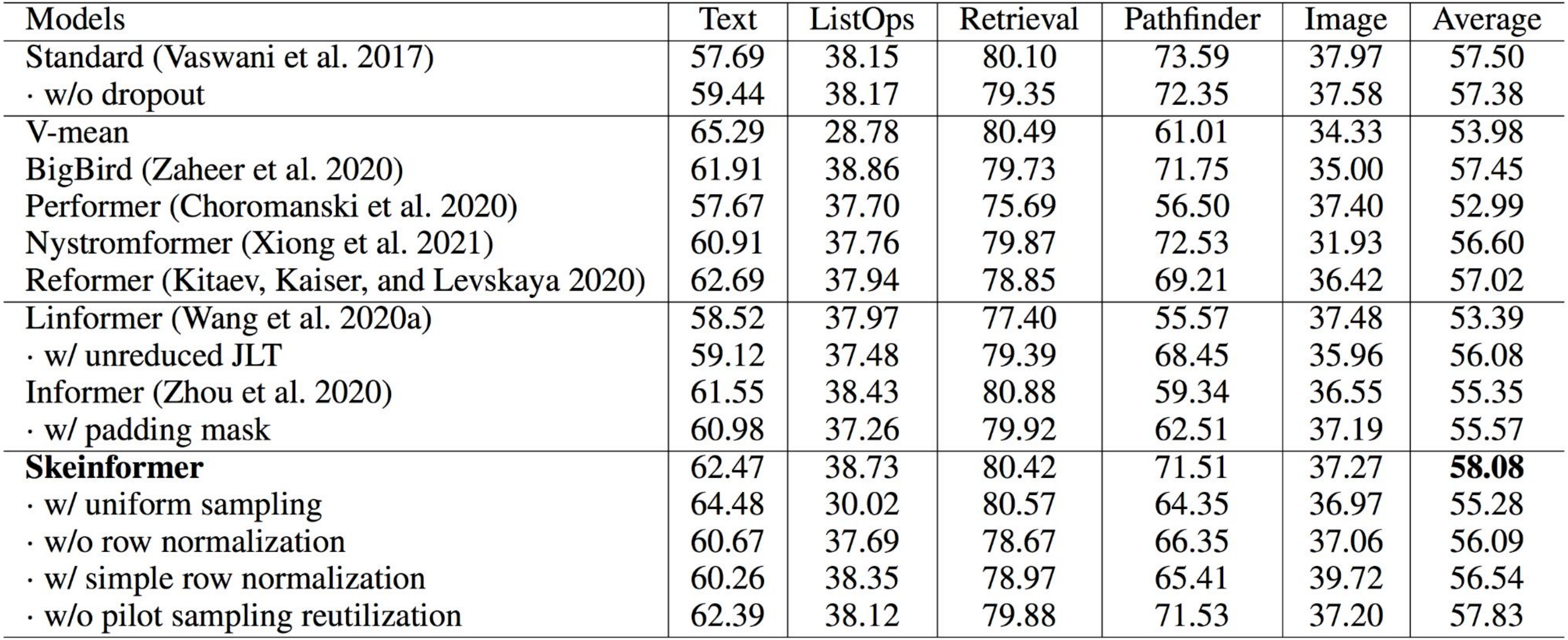
Skeinformer: Algorithm



Algorithm 1: Skeinformer.

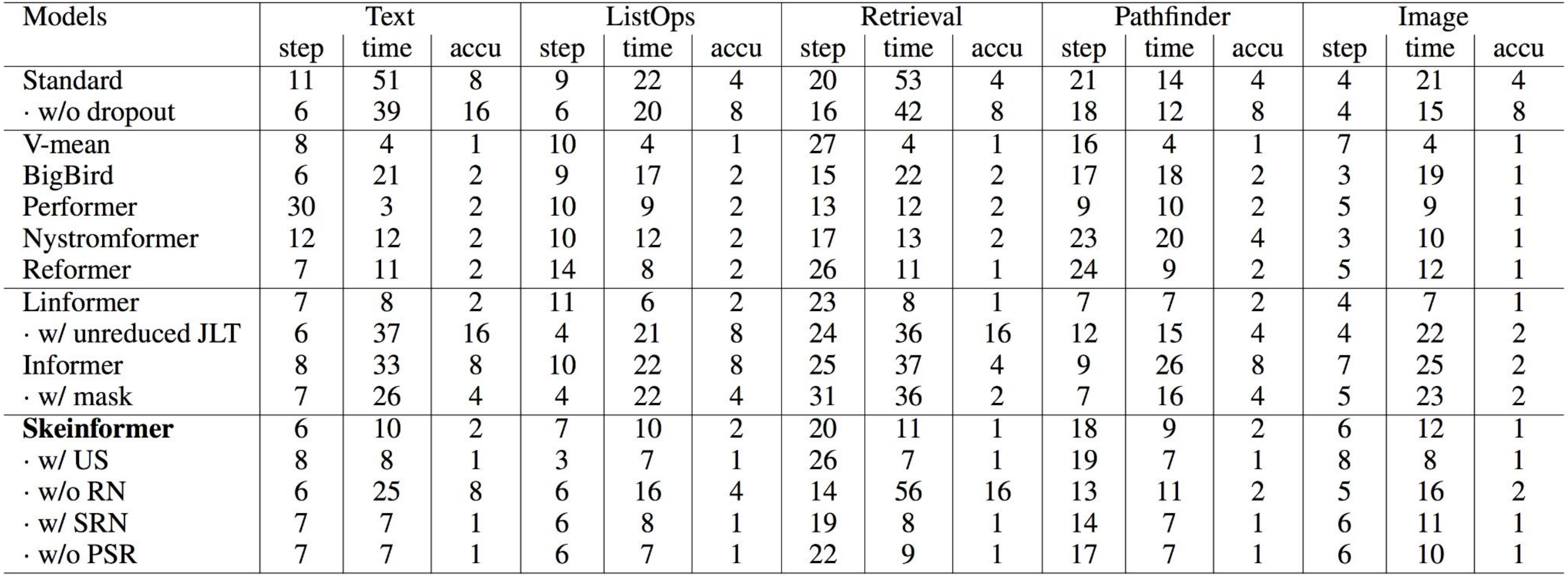
```
Input: query matrix $\boldsymbol{Q}$, key matrix $\boldsymbol{K}$, value matrix $\boldsymbol{V}$ (all are $n$-by-$p$), and sub-sample size $d$
Output: Attention output matrix $\boldsymbol{R}$ with the same shape as $\boldsymbol{V}$
Uniformly sample $d$ indices $j_1, \cdots, j_d$ with replacement;
Construct the $d \times p$ matrix $\boldsymbol{Q}_J$ as to the index set $J := \{j_k\}_{k=1}^{d}$, whose $k$-th row is $\boldsymbol{Q}_{(j_k)}$;
Compute the matrix $\boldsymbol{B}_J = \mathrm{softmax}\left(\boldsymbol{Q}_J \boldsymbol{K}^T / \sqrt{p}\right)$ ;                // pilot sampling
Based on $\boldsymbol{B}_J$, give the estimated sub-sampling probabilities $\{\hat{p}_i\}_{i=1}^{n}$ as in Equation (5);
With $\{\hat{p}_i\}_{i=1}^{n}$ sample $d$ indices $j'_1, \cdots, j'_d$ without replacement;
Construct the $d$-by-$p$ matrix $\boldsymbol{K}_{J'}$ (resp., $\boldsymbol{V}_{J'}$) according to the indices list $J' := \{j'_k\}_{k=1}^{d}$, whose $k$-th row is $\boldsymbol{K}_{(j'_k)}$ (resp., $\boldsymbol{V}_{(j'_k)}$);
Compute the two matrices $\boldsymbol{A}^{J'} = \exp\left(\boldsymbol{Q}\boldsymbol{K}_{J'}^T / \sqrt{p}\right)$, and $\boldsymbol{R}_{J'} = \boldsymbol{A}^{J'} \boldsymbol{V}_{J'}$ ;                // column sampling
Construct a length $n$ column vector $\boldsymbol{g}$ whose $i$-th element is $(\prod_{k=1}^{d} a_{ij'_k})^{\frac{1}{d}}, \forall i \in [n]$;
Compute the row sum vector $\boldsymbol{d} := \boldsymbol{A}^{J'} \boldsymbol{1}_d + (n-d)\boldsymbol{g}$ ;                // adaptive row normalization
Denote the un-selected part of $\boldsymbol{V}$ as $\boldsymbol{V}_{(J')^C}$, and compute the vector $\boldsymbol{v} = \boldsymbol{V}_{(J')^C}^T \boldsymbol{1}_{n-d}$;
Obtain the intermediate output $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{d}^{-1})(\boldsymbol{R}_{J'} + \boldsymbol{g}\boldsymbol{v}^T)$, where $\boldsymbol{d}^{-1}$ is the element-wise inverse of $\boldsymbol{d}$;
Compute $\boldsymbol{B}_J \boldsymbol{V}$ and assign it to the corresponding rows of $\boldsymbol{R}$ ;                // pilot sampling reutilization
Return the matrix $\boldsymbol{R}$ as the ultimate output of this algorithm;
```

Skeinformer: Classification accuracy on LRA



Models	Text	ListOps	Retrieval	Pathfinder	Image	Average
Standard (Vaswani et al. 2017)	57.69	38.15	80.10	73.59	37.97	57.50
· w/o dropout	59.44	38.17	79.35	72.35	37.58	57.38
V-mean	65.29	28.78	80.49	61.01	34.33	53.98
BigBird (Zaheer et al. 2020)	61.91	38.86	79.73	71.75	35.00	57.45
Performer (Choromanski et al. 2020)	57.67	37.70	75.69	56.50	37.40	52.99
Nystromformer (Xiong et al. 2021)	60.91	37.76	79.87	72.53	31.93	56.60
Reformer (Kitaev, Kaiser, and Levskaya 2020)	62.69	37.94	78.85	69.21	36.42	57.02
Linformer (Wang et al. 2020a)	58.52	37.97	77.40	55.57	37.48	53.39
· w/ unreduced JLT	59.12	37.48	79.39	68.45	35.96	56.08
Informer (Zhou et al. 2020)	61.55	38.43	80.88	59.34	36.55	55.35
· w/ padding mask	60.98	37.26	79.92	62.51	37.19	55.57
Skeinformer	62.47	38.73	80.42	71.51	37.27	**58.08**
· w/ uniform sampling	64.48	30.02	80.57	64.35	36.97	55.28
· w/o row normalization	60.67	37.69	78.67	66.35	37.06	56.09
· w/ simple row normalization	60.26	38.35	78.97	65.41	39.72	56.54
· w/o pilot sampling reutilization	62.39	38.12	79.88	71.53	37.20	57.83

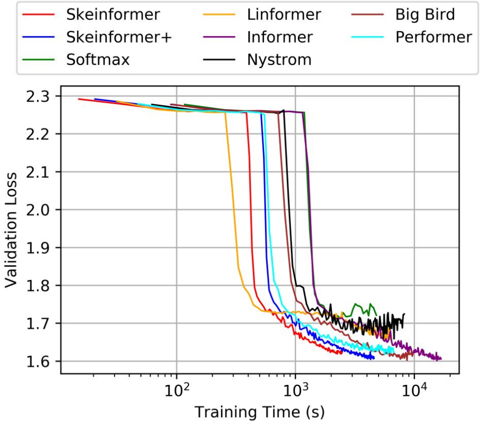
Skeinformer: Time and Space on LRA



Models	Text			ListOps			Retrieval			Pathfinder			Image		
	step	time	accu	step	time	accu	step	time	accu	step	time	accu	step	time	accu
Standard	11	51	8	9	22	4	20	53	4	21	14	4	4	21	4
· w/o dropout	6	39	16	6	20	8	16	42	8	18	12	8	4	15	8
V-mean	8	4	1	10	4	1	27	4	1	16	4	1	7	4	1
BigBird	6	21	2	9	17	2	15	22	2	17	18	2	3	19	1
Performer	30	3	2	10	9	2	13	12	2	9	10	2	5	9	1
Nystromformer	12	12	2	10	12	2	17	13	2	23	20	4	3	10	1
Reformer	7	11	2	14	8	2	26	11	1	24	9	2	5	12	1
Linformer	7	8	2	11	6	2	23	8	1	7	7	2	4	7	1
· w/ unreduced JLT	6	37	16	4	21	8	24	36	16	12	15	4	4	22	2
Informer	8	33	8	10	22	8	25	37	4	9	26	8	7	25	2
· w/ mask	7	26	4	4	22	4	31	36	2	7	16	4	5	23	2
Skeinformer	6	10	2	7	10	2	20	11	1	18	9	2	6	12	1
· w/ US	8	8	1	3	7	1	26	7	1	19	7	1	8	8	1
· w/o RN	6	25	8	6	16	4	14	56	16	13	11	2	5	16	2
· w/ SRN	7	7	1	6	8	1	19	8	1	14	7	1	6	11	1
· w/o PSR	7	7	1	6	7	1	22	9	1	17	7	1	6	10	1

Skeinformer: Experiment conclusions

- Space Efficiency: Skeinformer requires less space and enables larger batch size.
- Time Efficiency: Skeinformer has consistently less time consumption.
- Fast Convergence: Skeinformer efficiently converges to the long-time limit.
- Comparable General Performance: Most O(n) attention acceleration methods have comparable performance with vanilla attention.



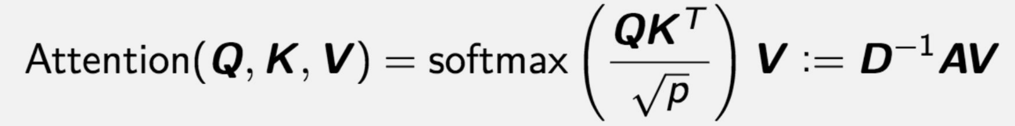
Skyformer: Remodel Self-Attention with Gaussian Kernel and Nystr”om Method

Yifan Chen*, Qi Zeng*, Heng Ji, Yun Yang
NeurIPS 2021

Skyformer: Background

Transformers are expensive to train

- Quadratic Complexity
  - Quadratic time and space complexity in the self-attention mechanism
  - Transformers cannot support long sequence processing and large batch size



$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{p}}\right)\boldsymbol{V} := \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{V}$$

- Training Instability
  - Small perturbations in parameter updates tend to be amplified, resulting in significant disturbances in the model output
  - Training process is sensitive to hyper-parameters

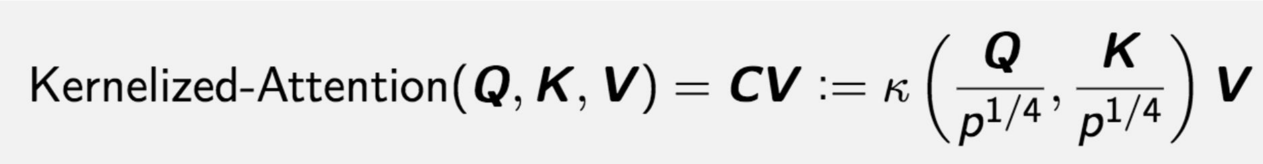
Skyformer: Background (cont’)

Kernel methods may be the answer to both challenges (quadratic complexity and training instability)

- Connections between Self-attention and Gaussian Kernels:
    - The un-normalized attention score matrix can be formed via basic matrix operations on an empirical Gaussian Kernel matrix
    - The form of Gaussian kernels has the natural interpretation of assigning “attention” to different tokens
    - Gaussian kernels automatically perform the normalization similar to how softmax does

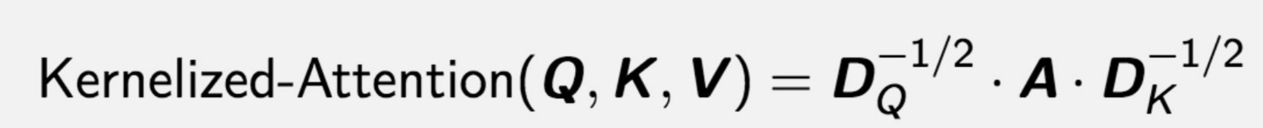
Skyformer: Method - Kernelized Attention

- Kernelized Attention replaces the softmax structure with a Gaussian kernel



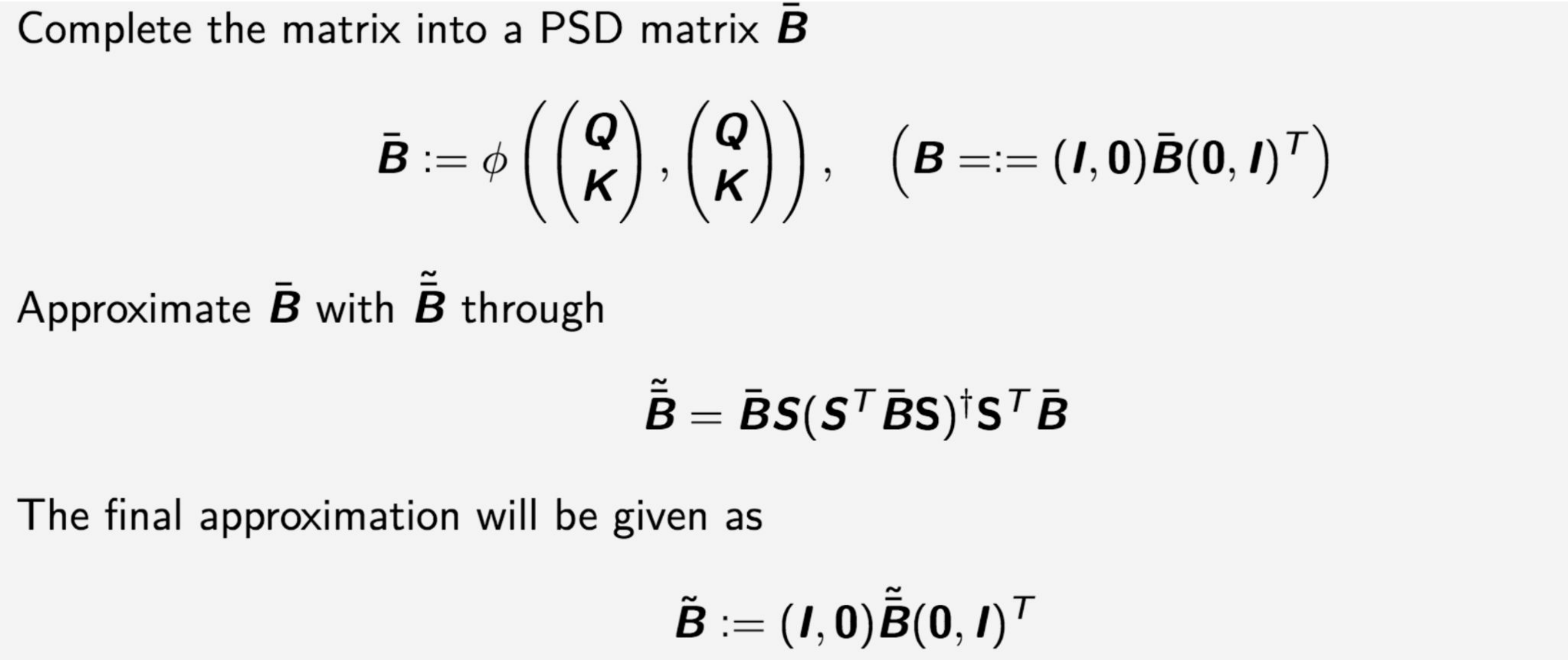
$$\text{Kernelized-Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{C}\boldsymbol{V} := \kappa\left(\frac{\boldsymbol{Q}}{p^{1/4}}, \frac{\boldsymbol{K}}{p^{1/4}}\right)\boldsymbol{V}$$

- It can be rewritten in terms of the un-normalized attention score matrix as



$$\text{Kernelized-Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{D}_Q^{-1/2} \cdot \boldsymbol{A} \cdot \boldsymbol{D}_K^{-1/2}$$

Skyformer: a modified Nyström method



Complete the matrix into a PSD matrix $\bar{\boldsymbol{B}}$

$$\bar{\boldsymbol{B}} := \phi\left(\begin{pmatrix}\boldsymbol{Q}\\ \boldsymbol{K}\end{pmatrix}, \begin{pmatrix}\boldsymbol{Q}\\ \boldsymbol{K}\end{pmatrix}\right), \quad \left(\boldsymbol{B} =:= (\boldsymbol{I}, \mathbf{0})\bar{\boldsymbol{B}}(\mathbf{0}, \boldsymbol{I})^T\right)$$

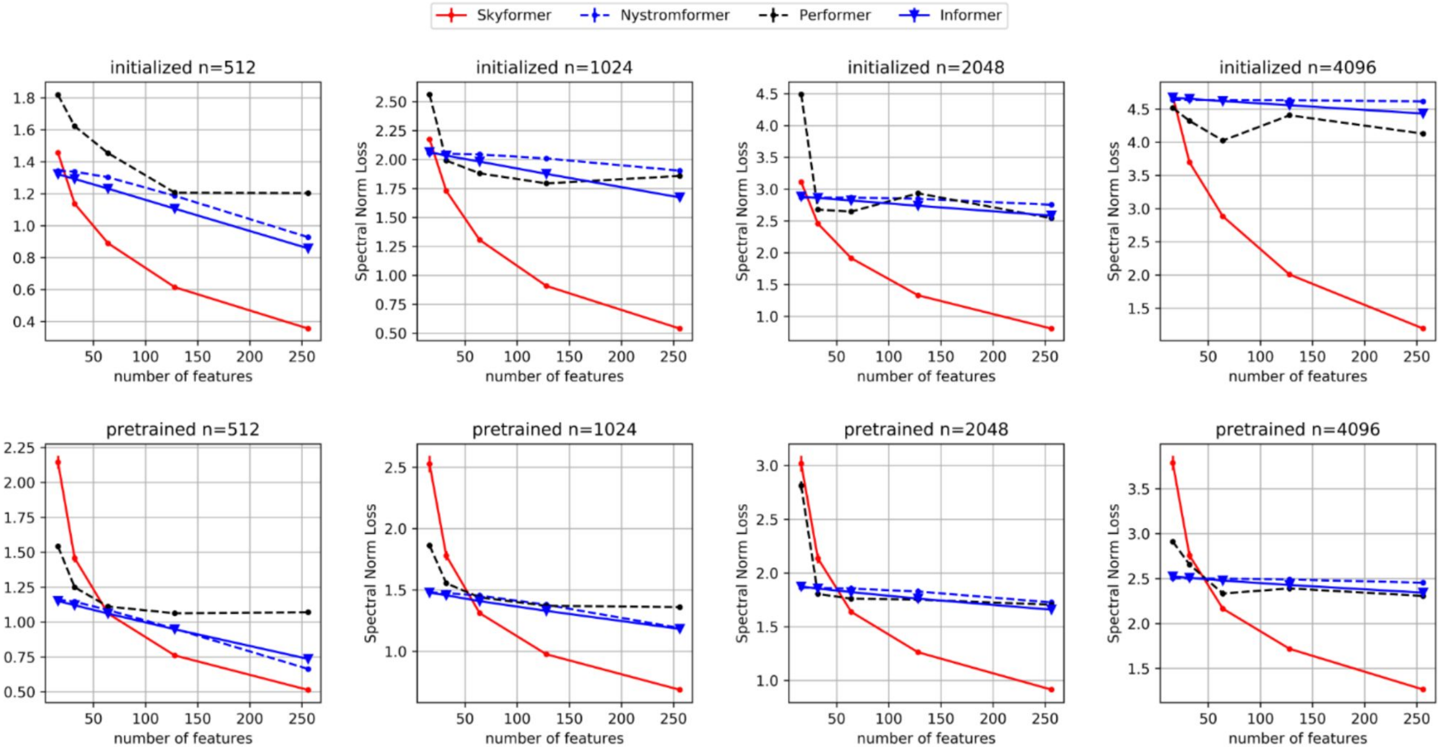
Approximate $\bar{\boldsymbol{B}}$ with $\tilde{\bar{\boldsymbol{B}}}$ through

$$\tilde{\bar{\boldsymbol{B}}} = \bar{\boldsymbol{B}}\boldsymbol{S}(\boldsymbol{S}^T\bar{\boldsymbol{B}}\mathbf{S})^\dagger\mathbf{S}^T\bar{\boldsymbol{B}}$$

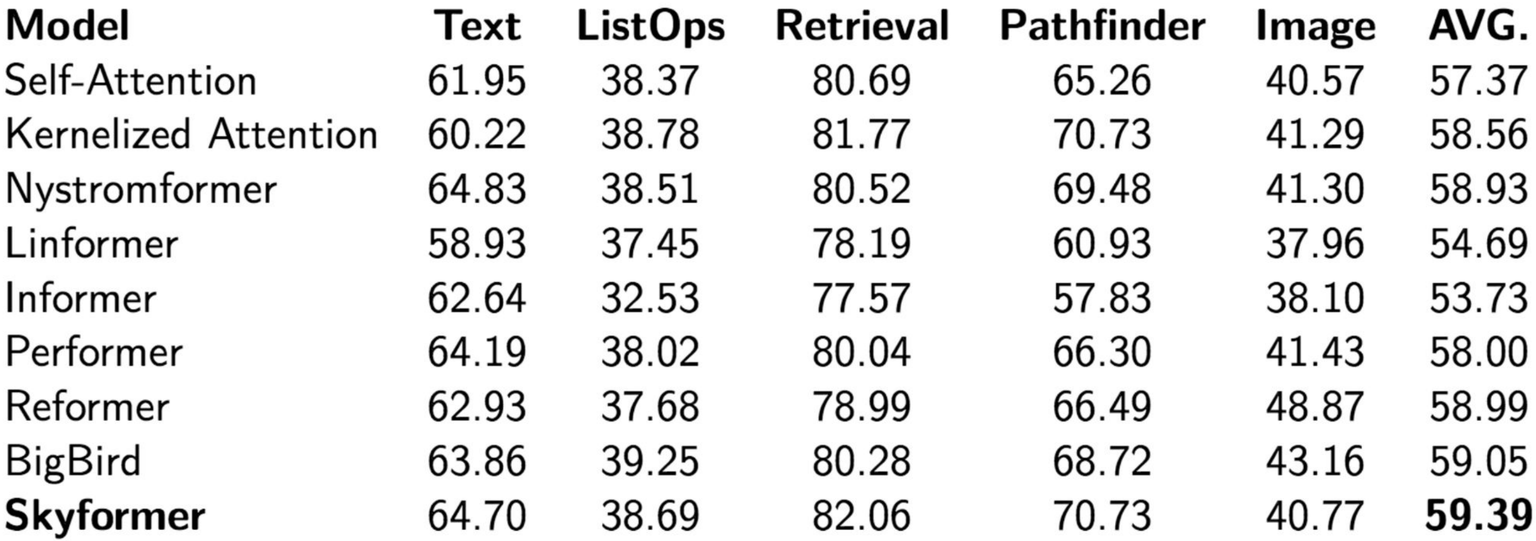
The final approximation will be given as

$$\tilde{\boldsymbol{B}} := (\boldsymbol{I}, \mathbf{0})\tilde{\bar{\boldsymbol{B}}}(\mathbf{0}, \boldsymbol{I})^T$$

Skyformer: Approximation evaluation results



Skyformer: Classification accuracy on LRA



Model	**Text**	**ListOps**	**Retrieval**	**Pathfinder**	**Image**	**AVG.**
Self-Attention	61.95	38.37	80.69	65.26	40.57	57.37
Kernelized Attention	60.22	38.78	81.77	70.73	41.29	58.56
Nystromformer	64.83	38.51	80.52	69.48	41.30	58.93
Linformer	58.93	37.45	78.19	60.93	37.96	54.69
Informer	62.64	32.53	77.57	57.83	38.10	53.73
Performer	64.19	38.02	80.04	66.30	41.43	58.00
Reformer	62.93	37.68	78.99	66.49	48.87	58.99
BigBird	63.86	39.25	80.28	68.72	43.16	59.05
Skyformer	64.70	38.69	82.06	70.73	40.77	**59.39**

Skyformer: Conclusion

- We revisit the intrinsic connection between self-attention and kernel methods, and explore a new kernel-based structure to stabilize the training of Transformers
- We approximate the Kernelized Attention via low dimensional randomized sketches by adapting the Nystrom method to a non-PSD matrix
- We conduct extensive experiments showing that Skyformer achieves comparable performance to the original self-attention with fewer computational costs

Resources

Resource: Project Topic

- Knowledge-enhanced Abstractive Summarization
- Possible extensions
  - Adaptive (plug-in) layers for Knowledge Infusion
  - A long transformer specializing in handling masks
  - Concept Representation learning
  - IE-assisted consistency checking
  - Evaluation benchmark

Resources: Datasets and Tools

- Long summarization datasets
    - GovReport https://drive.google.com/drive/folders/128KyqPTwZ0Si9RV_IX-md2dcHeRTUHkr
    - ArXiv https://drive.google.com/file/d/1b3rmCSIoh6VhD4HKWjI4HOW-cSwcwbeC/view?usp=sharing
    - PubMed https://drive.google.com/file/d/1lvsqvsFi3W-pE1SqNZI0s8NR9rC1tsja/view?usp=sharing
- BART for quick experiment setup
    - https://github.com/pytorch/fairseq/blob/main/examples/bart/README.summarization.md
    - https://huggingface.co/docs/transformers/model_doc/bart

Thanks!