

# Transformers for Long Texts

---

Natural Language Processing  
Vladislav Lialin, Text Machine Lab

# Administrative

- BERT / T5 / 🤗 homework
  - Released today, due next Thursday Apr 13 before the class
  - It's actually very important as it teaches you to use modern tooling — 🤗 Transformers and we really hoped we could give it to you earlier. Please spend extra time just playing around with different models and tasks.
- Research paper presentations
  - In two weeks, **on Monday Apr 17**
  - You select a paper from the list and prepare a 10-minute presentation with the following structure:
    - Slide 1: title and your names
    - Slide 2: motivation
    - Slide 3: main contribution of **this specific paper** — not the main method of the subfield, frequently the **specific contribution is subtle**
    - Slide 4: results — numerical evaluations and qualitative comparisons with prior work  
Only the most interesting results, not all of the tables from the paper.
  - Time limit is strict:
    - We will notify you when you have a few minutes left
    - You will be stopped when the 10-minute timer rings
  - One or two follow-up questions from the audience after the presentation

# NLP Research in 2023

# NLP Research in ~~2023~~ 2021

Model: GPT-4

GU

What are the main directions of NLP research in 2023?



As an AI language model, I can't provide real-time information about the state of NLP research in 2023, but I can predict some likely directions based on previous trends and developments up to 2021. Please note that these predictions might not be entirely accurate. Some potential directions for NLP research in 2023 include:



1. Pre-training and fine-tuning improvements: Researchers will likely continue to refine pre-training and fine-tuning techniques to make NLP models more efficient and accurate.
2. Multimodal learning: Combining NLP with other modalities like computer vision and audio processing is a promising direction for creating more versatile AI systems that can understand and generate multimodal content.
3. Multilingual and cross-lingual models: Expanding the capabilities of NLP models to understand and generate text in multiple languages is crucial for creating truly global AI systems.
4. Interpretability and explainability: Understanding the inner workings of large-scale NLP models and making their outputs more interpretable is vital for building trust and ensuring proper usage.

5. Ethical and responsible AI: Addressing the ethical implications of NLP models, such as bias, fairness, and privacy concerns, is essential to ensure responsible development and deployment of these technologies.
6. Low-resource languages and dialects: Developing NLP models for languages with limited available data is essential for increasing the accessibility and inclusiveness of AI technologies.
7. Domain-specific applications: Tailoring NLP models to specific industries and applications, such as healthcare, finance, and legal, will be vital to address unique domain-specific challenges and requirements.
8. Human-AI collaboration: Researching ways to enhance the collaboration between humans and AI systems, including designing better user interfaces, will be crucial for maximizing the benefits of NLP technology.
9. Commonsense reasoning: Developing NLP models that can understand and reason about the world like humans do, including understanding implicit information and common sense, is a challenging but important research direction.
10. Efficient and lightweight models: Reducing the computational requirements of NLP models while maintaining their performance will be essential for deploying AI solutions on a broader range of devices and in more resource-constrained environments.

# NLP Research in 2023 (a very biased opinion)

- Scaling language models
  - More efficient distributed training
  - Better hardware
  - Predictable improvements — scaling laws and hyperparameter studies
  - Alternative ways to scale
- Interpretability and model analysis
  - How do models understand and process texts
  - Can we track down why the model made certain prediction
  - How can we **trust the predictions of these models** and reduce hallucinations
- Model evaluation
  - How can we know the **limits of capabilities** of GPT4? GPT5?
- Prompting and augmented language models
  - How to talk to models, **prompting tricks**, chain of thoughts
  - Language models that use **external tools**, neuro-symbolic methods
  - Seem to move from research to startups and industry

# NLP Research in 2023 (a very biased opinion)

- Alignment
  - ChatGPT
  - How can we ensure that the models are **safe, helpful, and are aligned with human values**
  - Social bias, dangerous information, being kind and empathetic, not killing humanity
- Multimodal models
  - GPT4 **image understanding**
  - **Image generation** DALL-E and Stable Diffusion, Text2Video
  - Robot **control** — generating actions based on a text command and visual/sensor environment
- Classical NLP tasks:
  - Machine translation — low-resource translation to thousands to languages
  - Question answering over long texts
  - Improved search, retrieval and retrieval-based generation
- Efficiency:
  - Efficient inference
  - Efficient fine-tuning

List of papers (we'll add more papers)

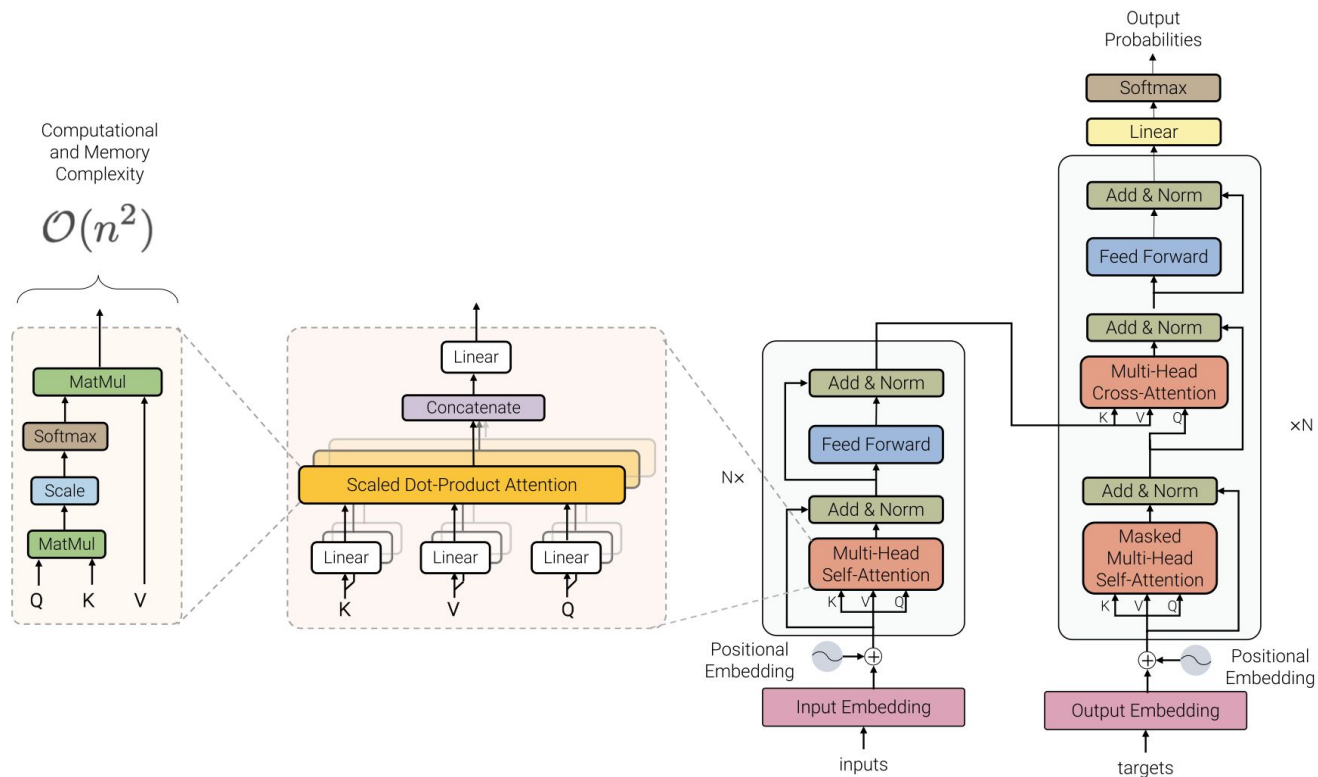
# Long Transformers



# What you should remember after this lecture

- Why long texts are hard
- What do we mean by long texts
- Taxonomy of long transformers
- Simple methods often outperform complex methods

# Transformer: a quick recap



# Taxonomy of long transformers

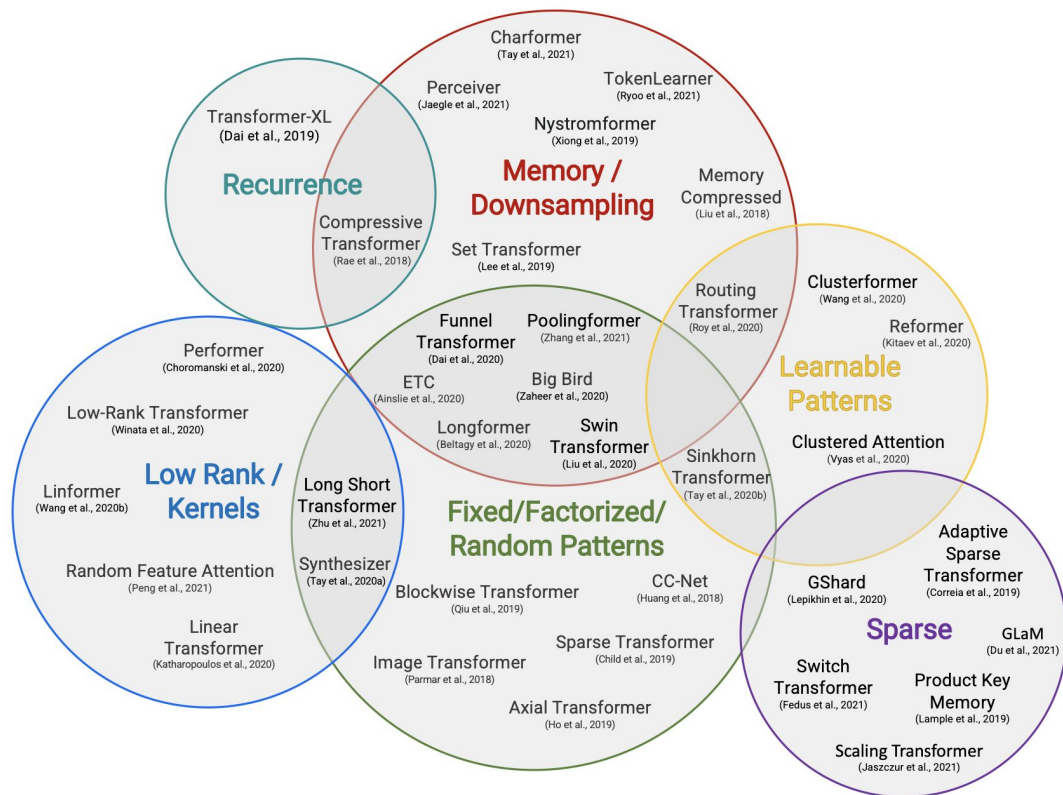
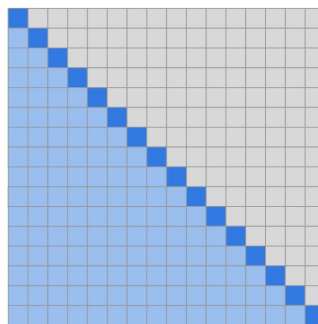
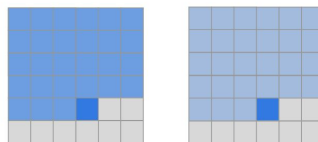


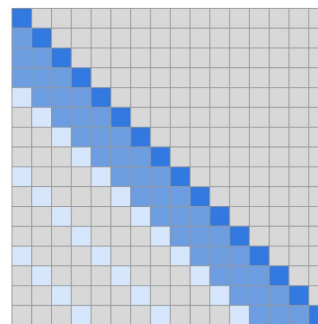
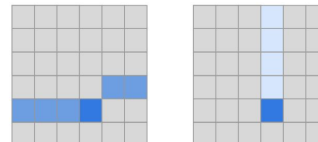
Image from 2020:  
Quite a few  
models are  
missing

# Fixed patterns

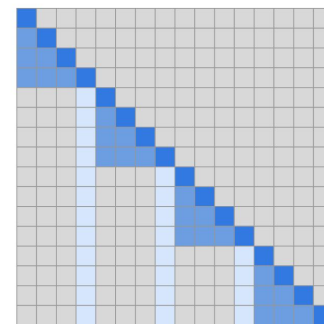
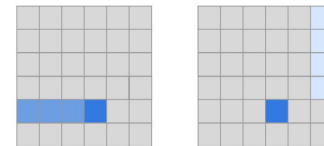
- **Sparse Transformer**  
2019, OpenAI
- Only compute  $q_i k_j$  for some positions using an intuitive human-proposed pattern like these →
- Used in GPT3



(a) Transformer



(b) Sparse Transformer (strided)

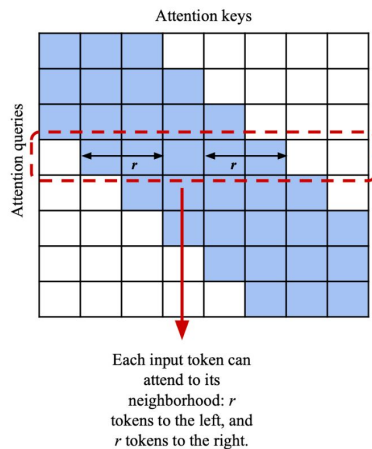


(c) Sparse Transformer (fixed)

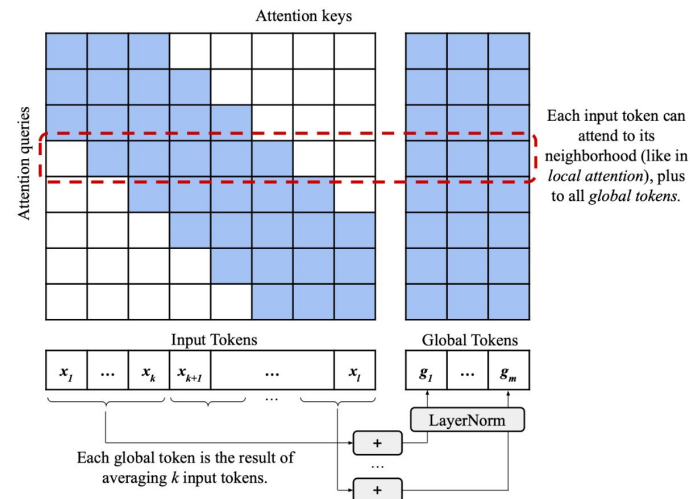
$$O(n \sqrt{n})$$

# Fixed patterns

- **LongT5**  
2022, Google
- Sparse attention + a fixed number of  $G$  special global tokens that can look at the whole sequence



a) LongT5 Local Attention

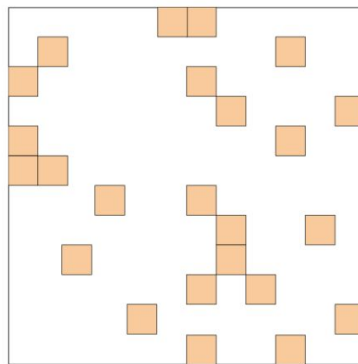


b) LongT5 Transient Global (TGlob) Attention

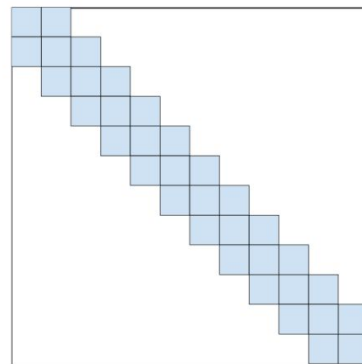
$$O(g n)$$

# Fixed patterns

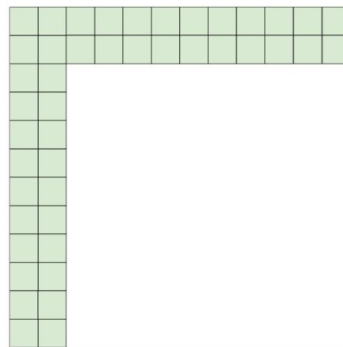
- **BigBird**  
2021, Google
- Sparse attention +  
a fixed number of  $G$   
special global tokens  
that can look at the  
whole sequence



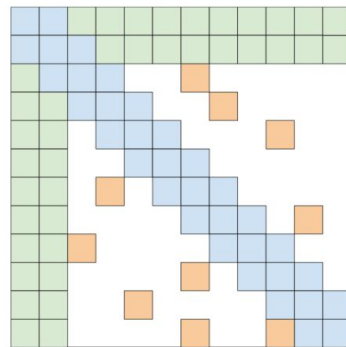
(a) Random attention



(b) Window attention



(c) Global Attention

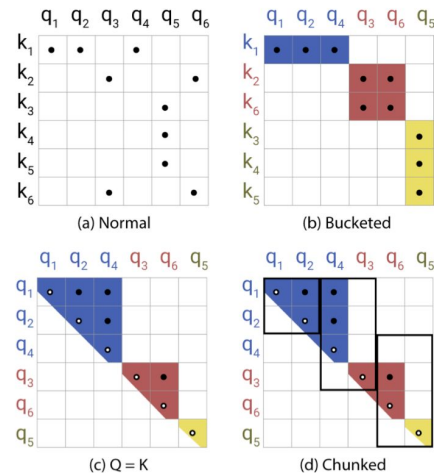
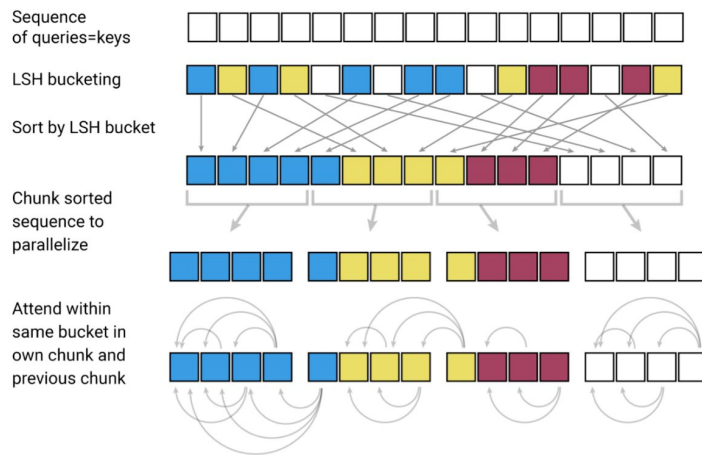


(d) BIGBIRD

$O(n)$

# Learnable patterns and approximate-KNN

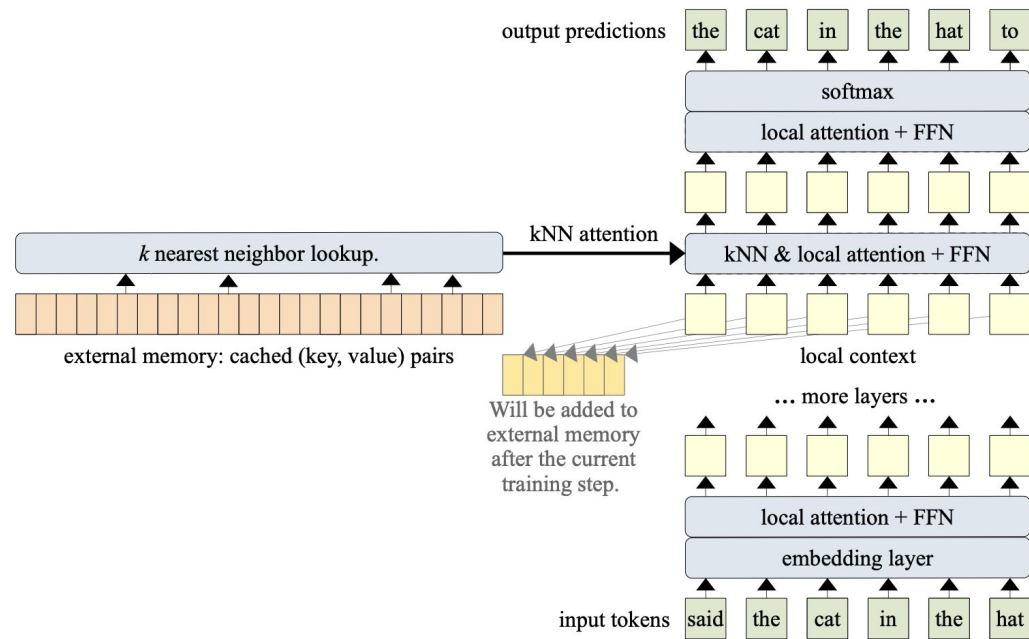
- **Reformer**  
2020, Google
- Hacky approximation through local sensitive hashing (LSH).  
Perform attention only between the tokens that are close in terms of their LHS hash.



$$O(n \log n)$$

# Learnable patterns and approximate-KNN

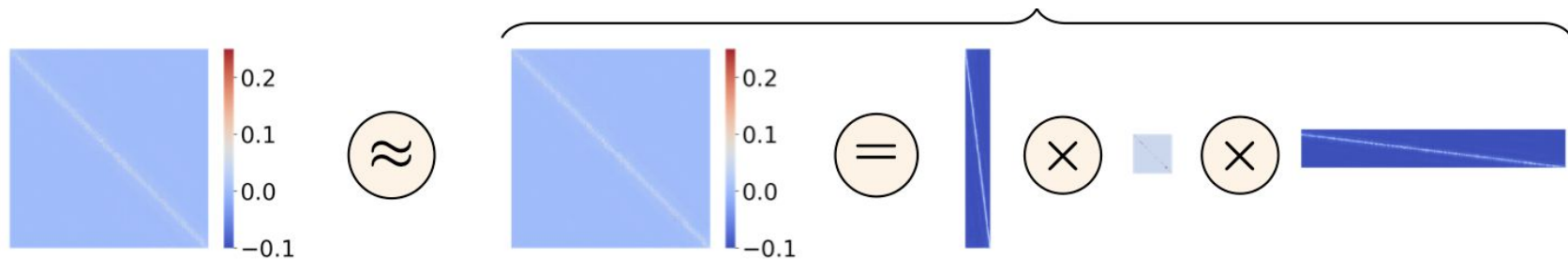
- **Memorizing Transformers**  
2022, Google
1. Independently process chunks of size  $M$  ( $\ll N$ ) and remember the hidden states.
  2. Use approximate KNN to select which of the previously computed hidden states to attend to



$O(n \log n)$



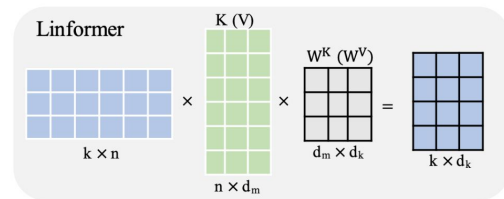
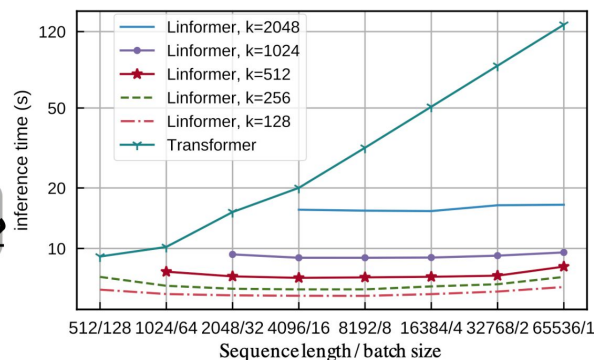
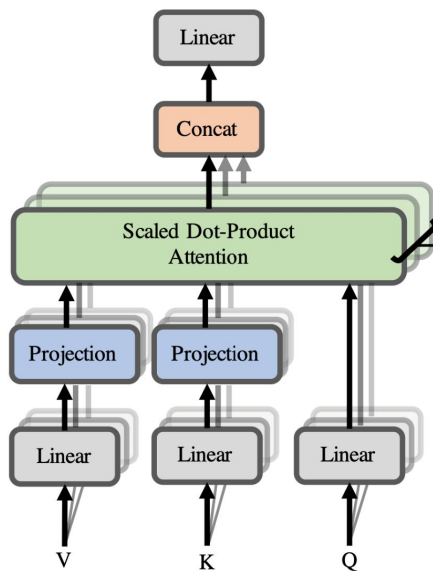
# Low-rank attention approximation



- Attention patterns are usually simple — e.g. most of the elements are diagonal. In other words, they are low-rank.
- A large low-rank matrix can be represented as a matmul of multiple small matrices

# Low-rank attention approximation

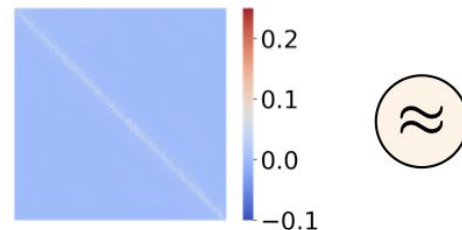
- **Linformer**  
2020, Meta AI
- Linformer compresses keys and values over sequence dimension into a smaller length via a simple linear projection.
- Keys:  $[\text{seq}, h] \rightarrow [k, h]$   
Values:  $[\text{seq}, h] \rightarrow [k, h]$



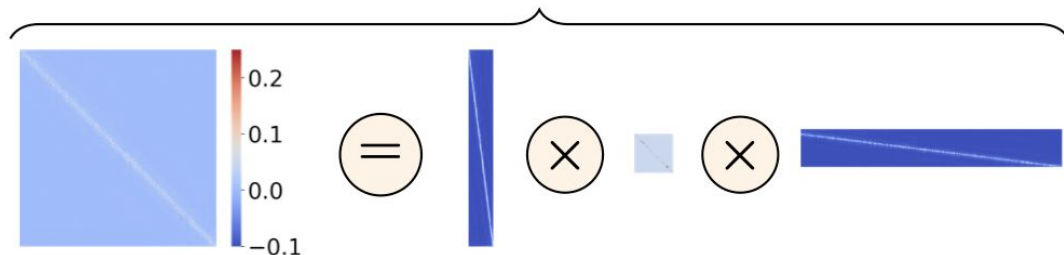
$O(kn)$

# Low-rank attention approximation

- **Nyströmformer**  
2021, University of Wisconsin-Madison
- Use Nyström method to approximate attention matrix

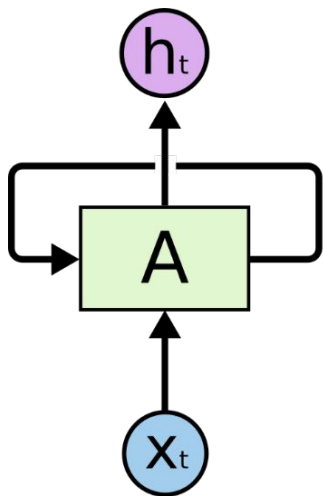


## Nyström approximation

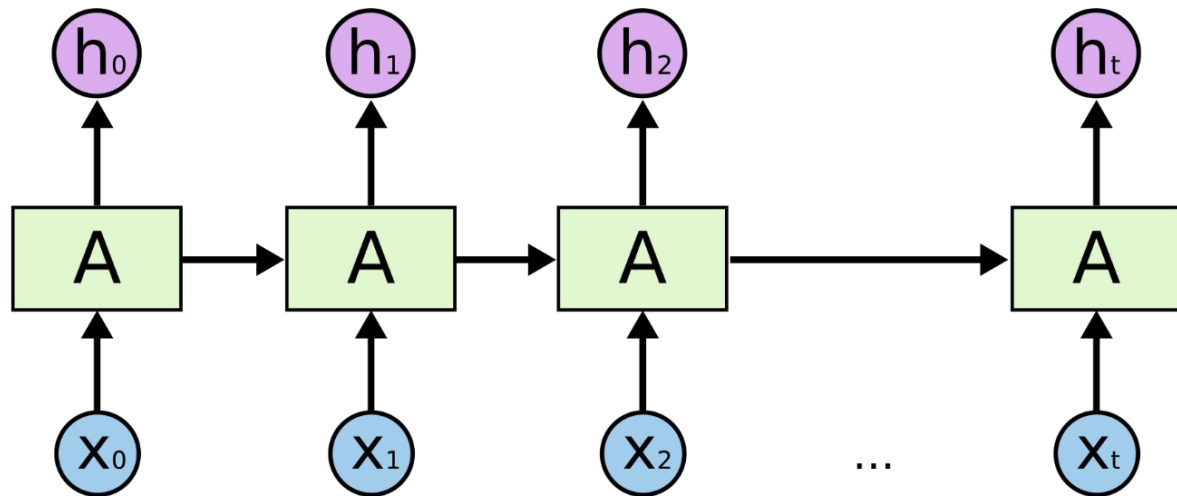


$O(n)$

# Recurrence



=

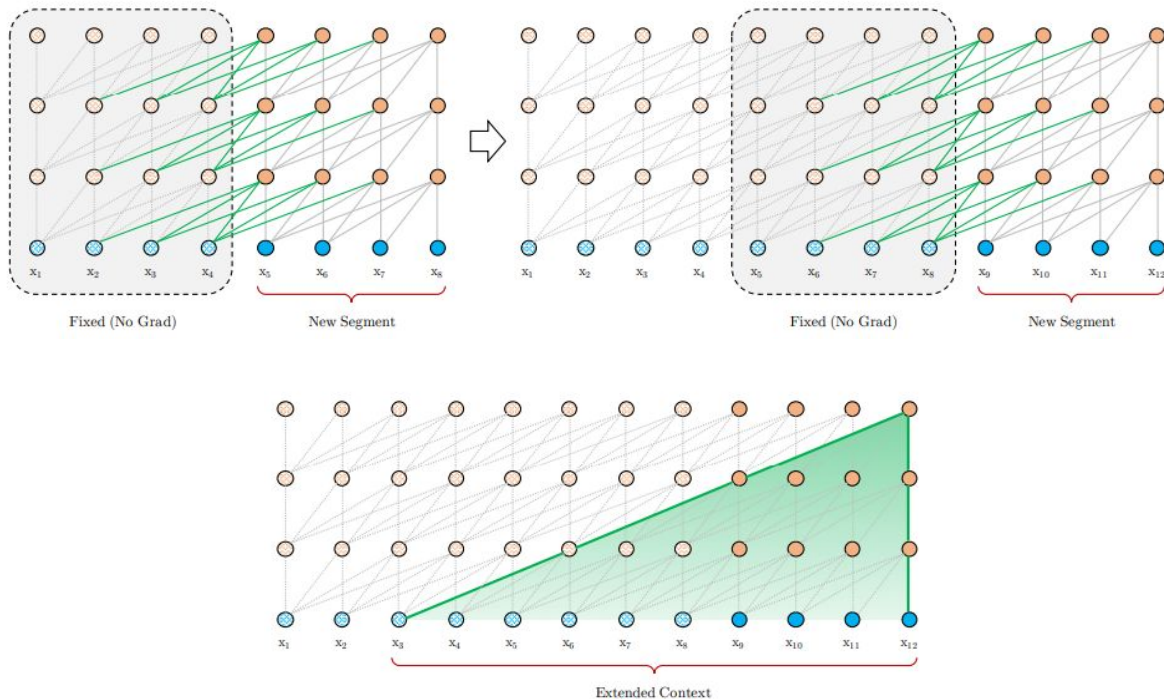


$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

# Recurrence

- **Transformer XL**  
2019, CMU
- Compute hidden states in chunks of size  $M$  ( $\ll N$ ) independently and cache them. Allow the final chunk to look at the hidden states of all chunks.

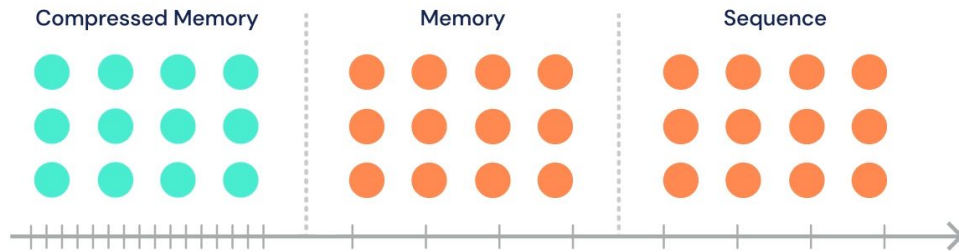


$O(mn)$

<https://arxiv.org/abs/1901.02860>

# Recurrence

- **Compressive Transformer**  
2019, DeepMind
- Learn a function to compress current context into memory with as least loss as possible



$O(mn)$

<https://arxiv.org/abs/1911.05507>

# Recurrence

- **Block-Recurrent Transformer**  
2022, Google
- Process text chunk-by-chunk and attend to both current text and the **recurrent state** of the previous chunk, produce the recurrent state of the current chunk

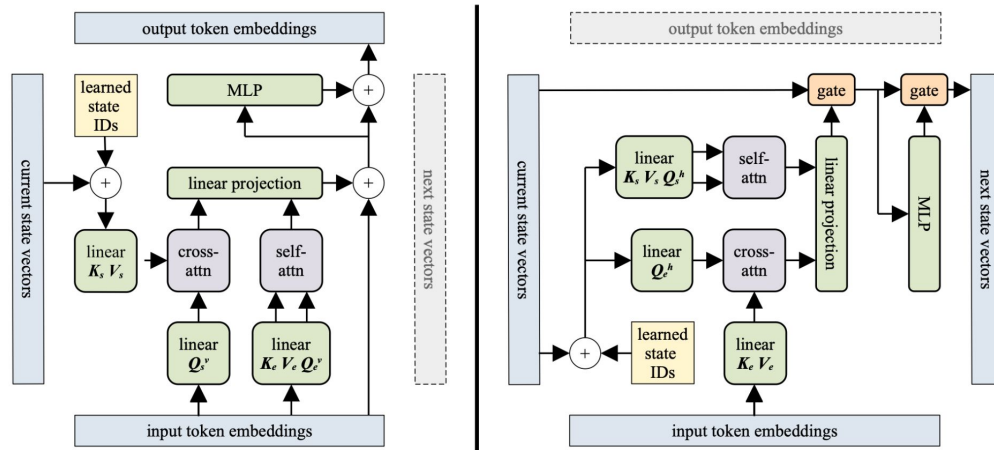


Figure 1: Illustration of our recurrent cell. The left side depicts the vertical direction (layers stacked in the usual way) and the right side depicts the horizontal direction (recurrence). Notice that the horizontal direction merely rotates a conventional transformer layer by  $90^\circ$ , and replaces the residual connections with gates.

$O(n)$

Exact attention with sub-quadratic memory

Does attention really need quadratic time?

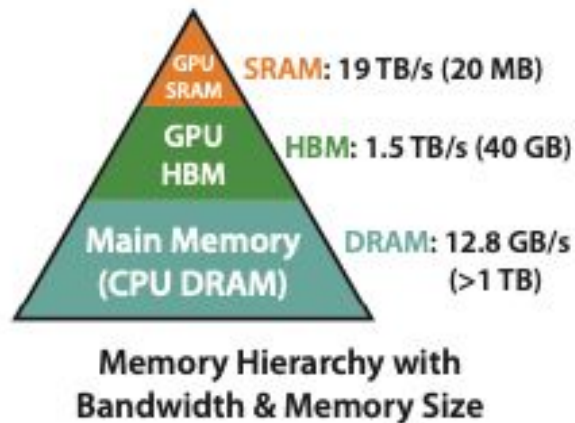


Exact attention with sub-quadratic memory

Does attention really need quadratic time?  
What about memory?

# Exact attention with sub-quadratic memory

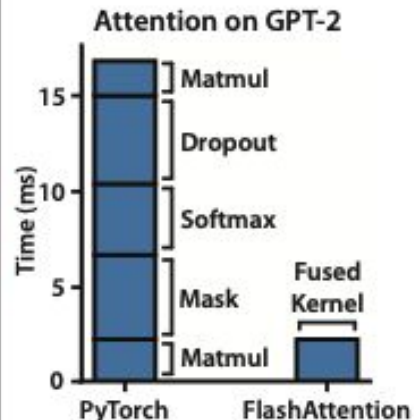
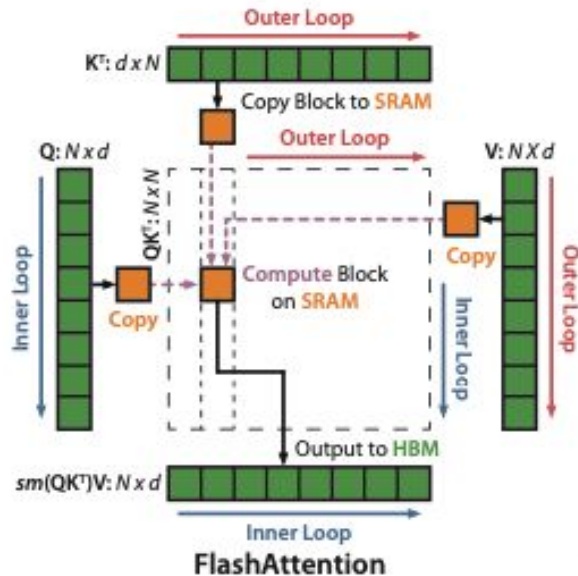
- **FLASH attention**  
2022, Stanford
- CUDA-cores specific  
Uses an efficient  
tradeoff between  
memory bandwidth  
and parallelization



$O(n)$

# Exact attention with sub-quadratic memory

- **FLASH attention**  
2022, Stanford
- CUDA-cores specific  
Uses an efficient tradeoff between memory bandwidth and parallelization



$O(n)$

# How to evaluate long-range methods?

Dataset	Example Count			Input Length			
	Train	Validation	Test	Average	Median	Max	90th percentile
CNN / Daily Mail	287,113	13,368	11,490	982.39	894	5268	1659
arXiv	203,037	6,436	6,440	10,720.18	8,519	378,825	20,170
PubMed	119,924	6,633	6,658	4,747.97	3,883	452,915	8,883
BigPatent	1,207,222	67,068	67,072	6,537.32	5,236	294,004	11,328
MediaSum	443,596	10,000	10,000	2,302.02	1,748	125,974	4,128
Multi-News	44,972	5,622	5,622	2,593.81	1,902.5	683,544	4,853

Table 1: Statistics for the summarization datasets. Input length measured in tokens using a SentencePiece Model.

# How to evaluate long-range methods?

Approach	arXiv		
	R-1	R-2	R-L
DANCER PEGASUS	45.01	17.6	40.56
BigBird-PEGASUS (large)	46.63	19.02	41.77
HAT-BART	46.68	19.07	42.17
LED (large)	46.63	19.62	41.83
PRIMER	47.6	20.8	42.6
LongT5 (large - 16k input)	48.28	21.63	44.11
LongT5 (xl - 16k input)	<b>48.35</b>	<b>21.92</b>	<b>44.27</b>
Approach	PubMed		
	R-1	R-2	R-L
DANCER PEGASUS	46.34	19.97	42.42
BigBird-PEGASUS (large)	46.32	20.65	42.33
HAT-BART	48.36	21.43	37.00
LongT5 (large - 16k input)	49.98	24.69	46.46
LongT5 (xl - 16k input)	<b>50.23</b>	<b>24.76</b>	<b>46.67</b>

Approach	MultiNews		
	R-1	R-2	R-L
TG-MultiSum	47.10	17.55	20.73
PRIMER	<b>49.9</b>	<b>21.1</b>	<b>25.9</b>
LongT5 (large - 8k input)	47.18	18.44	24.18
LongT5 (xl - 8k input)	48.17	19.43	24.94
Approach	MediaSum		
	R-1	R-2	R-L
BART (large)	35.09	18.05	31.44
LongT5 (large - 4k input)	35.54	19.04	32.20
LongT5 (xl - 4k input)	<b>36.15</b>	<b>19.66</b>	<b>32.80</b>
Approach	CNN / Daily Mail		
	R-1	R-2	R-L
HAT-BART	<b>44.48</b>	21.31	<b>41.52</b>
LongT5 (large - 4k input)	42.49	20.51	40.18
LongT5 (xl - 4k input)	43.94	<b>21.40</b>	41.28

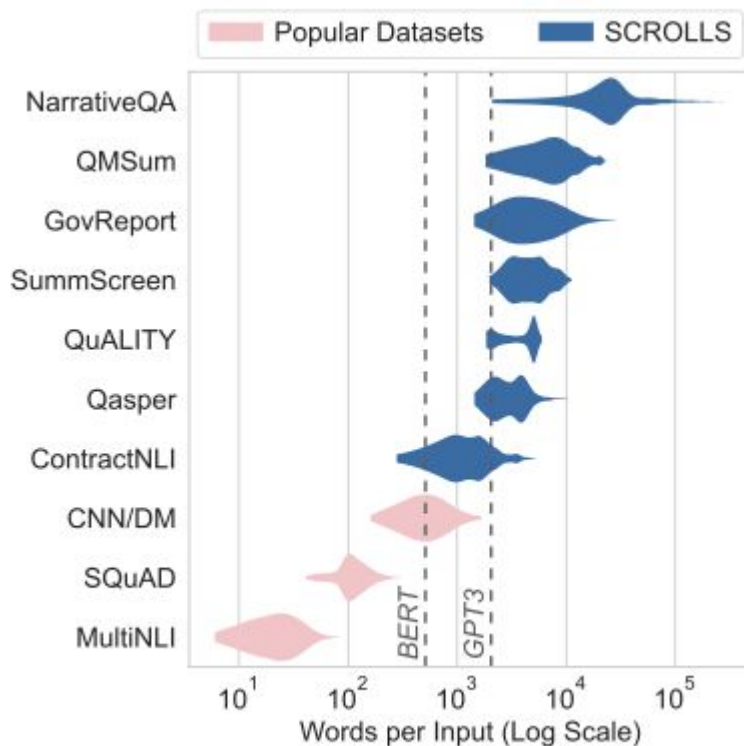
# How to evaluate long-range methods?



# How to evaluate long-range methods? SCROLLS

Dataset	Paper	Description	Download
GovReport	<a href="#">Huang et al., 2021</a>	Summarization of long reports from the Congressional Research Service and the U.S. Government Accountability Office.	<a href="#">jsonl.zip</a>
SummScreenFD	<a href="#">Chen et al., 2021</a>	Summarizing episodes of TV shows from their scripts.	<a href="#">jsonl.zip</a>
QMSum	<a href="#">Zhong et al., 2021</a>	Query-based summarization over meeting transcripts.	<a href="#">jsonl.zip</a>
NarrativeQA	<a href="#">Kočiský et al., 2018</a>	Question answering about entire books and movie scripts.	<a href="#">jsonl.zip</a>
Qasper	<a href="#">Dasigi et al., 2021</a>	Question answering over research papers.	<a href="#">jsonl.zip</a>
QuALITY	<a href="#">Pang et al., 2021</a>	Multiple-choice questions over long articles and stories.	<a href="#">jsonl.zip</a>
Contract NLI	<a href="#">Koreeda and Manning, 2021</a>	Natural language inference over non-disclosure agreements.	<a href="#">jsonl.zip</a>

# How to evaluate long-range methods? SCROLLS





# Simple Local Attentions Remain Competitive for Long-Context Tasks

Model	ListOps	Text	Matching	Avg Acc	GFlops
<i>Learnable attention pattern</i>					
Sinkhorn	37.6	63.8	80.4	60.6	0.289
LSH	37.9	62.5	80.5	60.3	0.273
<i>Low-rank/kernel-based approximation</i>					
Linformer	37.7	61.9	78.4	59.3	0.271
Nystrom	37.9	66.1	81.0	61.7	0.256
Performer	37.1	66.1	79.8	61.0	0.205
<i>Hybrid attention</i>					
Long-Short	37.7	65.7	81.6	61.7	0.199
<i>Fixed attention pattern</i>					
Local Window	37.4	65.7	81.6	61.6	0.153
Blockwise LW	37.4	65.6	81.3	61.4	0.146

## Extra materials:

- [\[2009.06732\] Efficient Transformers: A Survey](#)