Efficient Transformers: A Survey

Yi Tay, Mostafa Dehghani,
Dara Bahri, Donald Metzler

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

Outline

- Introduction
- Efficient Transformers
- Long Range Arena
- Experiments
- Conclusion

Introduction

於 2017 年提出的 Transformer 為深度學習模型帶來的巨大的影響。其核心方法 Self Attention 不論是在 NLP 或是 CV 的相關任務都取得了優秀的表現。

然而在計算過程中，會產生一個 L^2 大小的注意力矩陣。隨著輸入序列增大，耗費的記憶體與計算量也呈平方上升。

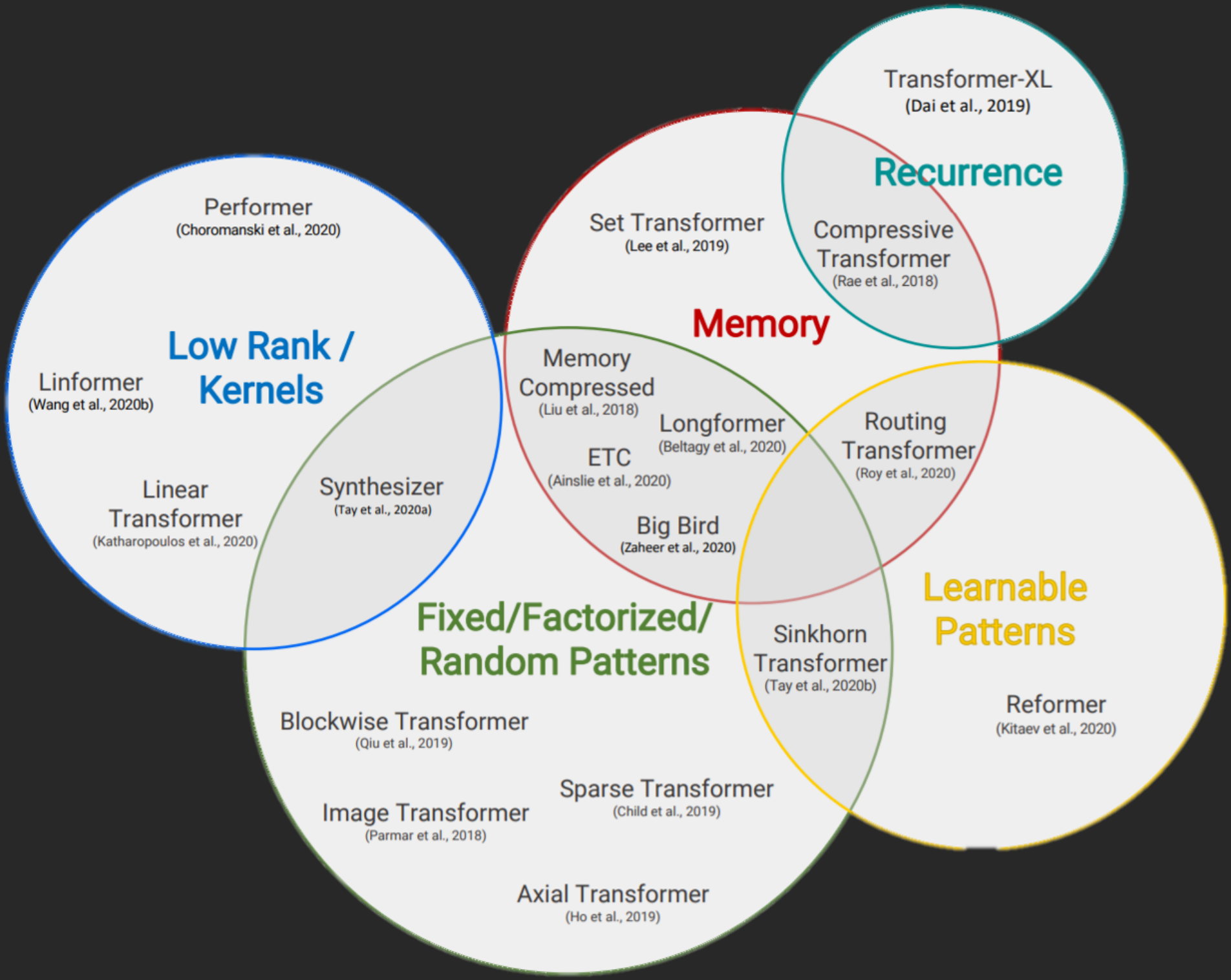
尤其是高記憶體複雜度會嚴重阻礙 Transformer 應用在長序列問題上的可行性。因此近期有許多論文提出各式 X-former，盡量減少需要耗費的記憶體。

Introduction

但目前缺乏一個系統性的基準來評估這些模型在不同任務上的性能優劣。

為此，Tay et al. 提出了 Long Range Arena 這個由複數任務組成的測試基準，用以評估各種 X-former 在文字、影像、推論等等問題上的效果。

本次報告將會簡介各大類型的 Transformer 改善方式與 Long Range Arena 這個評斷指標。


Transformer-XL
(Dai et al., 2019)
Recurrence
Performer
(Choromanski et al., 2020)
Set Transformer
(Lee et al., 2019)
Compressive
Transformer
(Rae et al., 2018)
Memory
Low Rank /
Kernels
Linformer
(Wang et al., 2020b)
Memory
Compressed
(Liu et al., 2018)
Longformer
(Beltagy et al., 2020)
Routing
Transformer
(Roy et al., 2020)
ETC
(Ainslie et al., 2020)
Linear
Transformer
(Katharopoulos et al., 2020)
Synthesizer
(Tay et al., 2020a)
Big Bird
(Zaheer et al., 2020)
Learnable
Patterns
Fixed/Factorized/
Random Patterns
Sinkhorn
Transformer
(Tay et al., 2020b)
Reformer
(Kitaev et al., 2020)
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

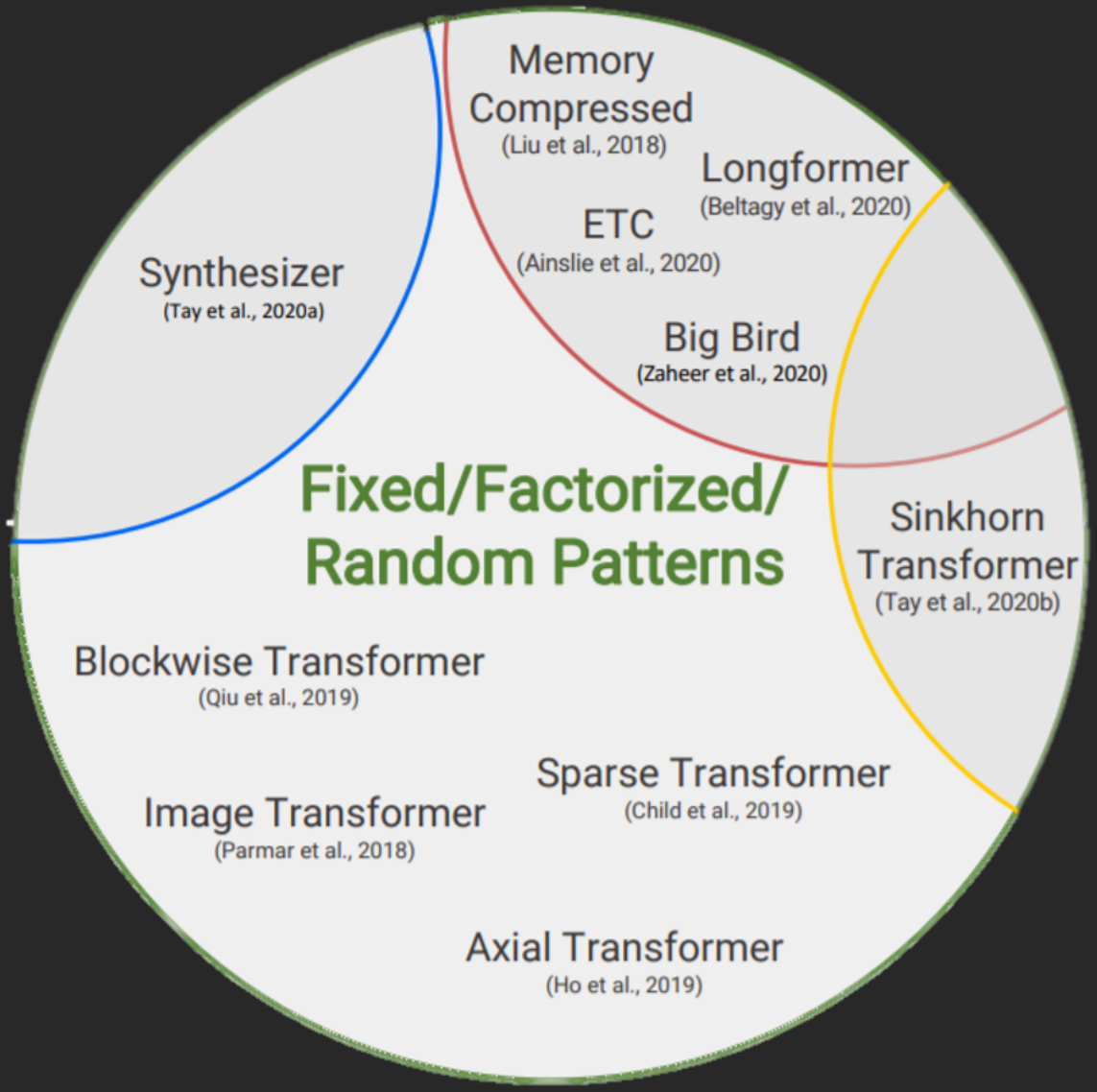
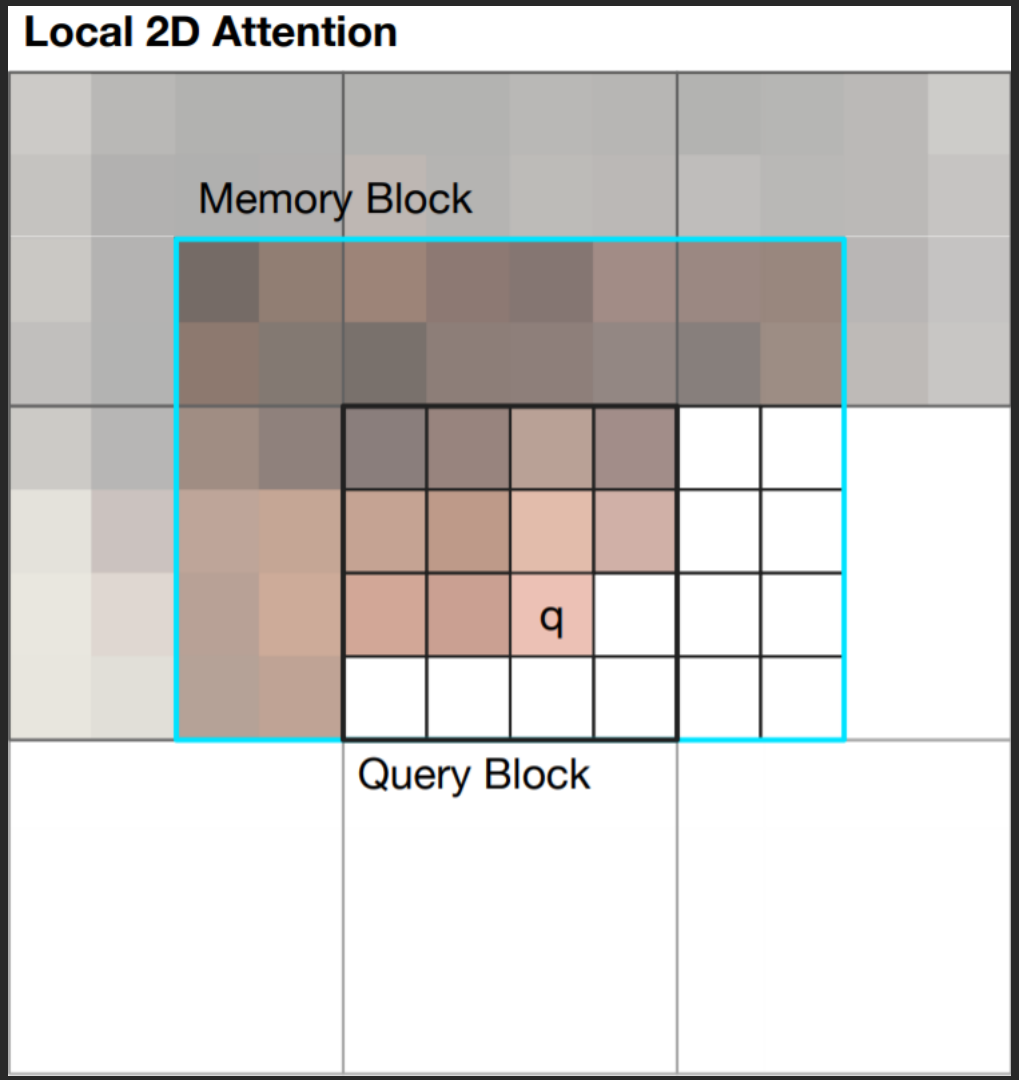
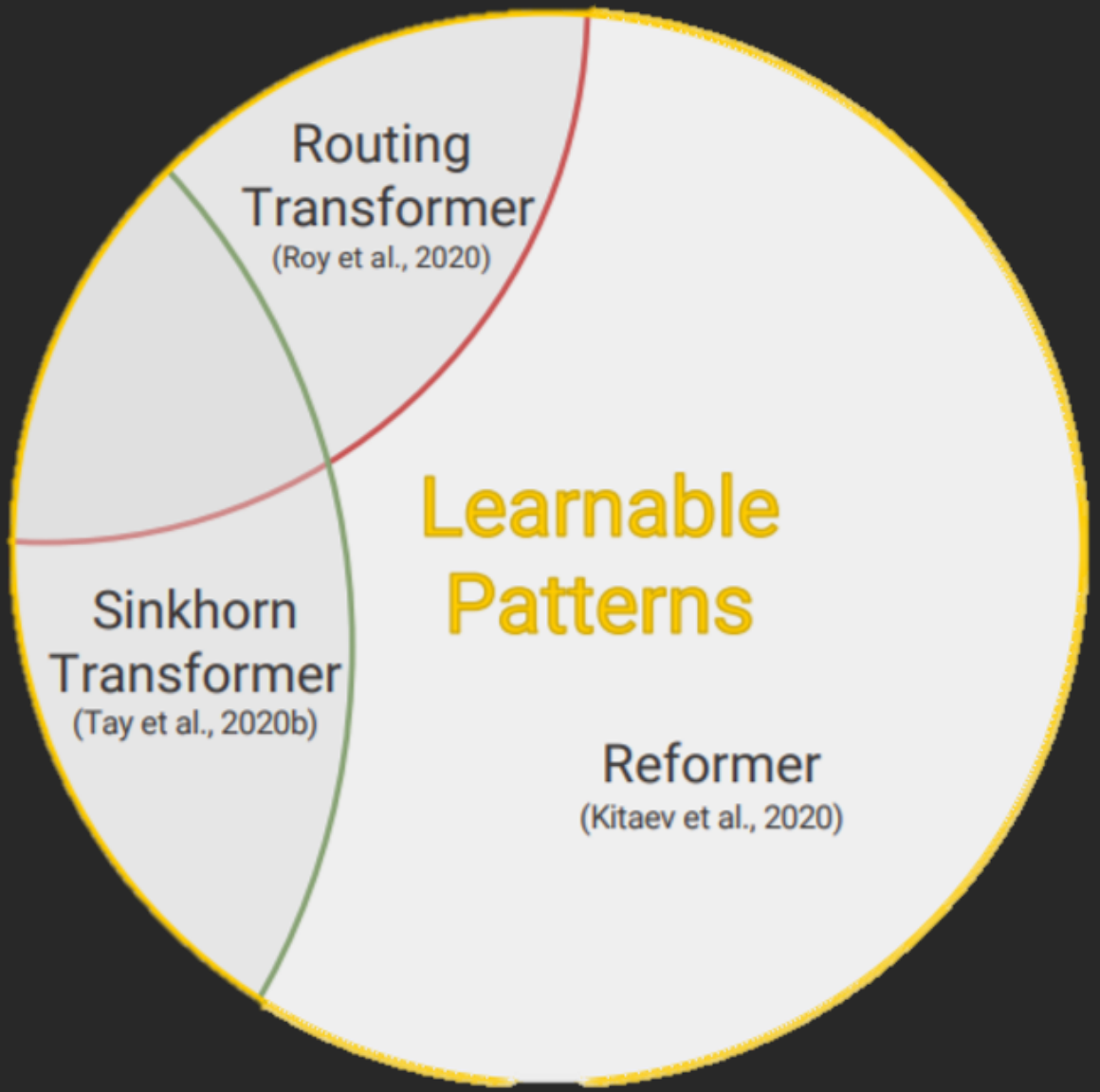


Image Transformer

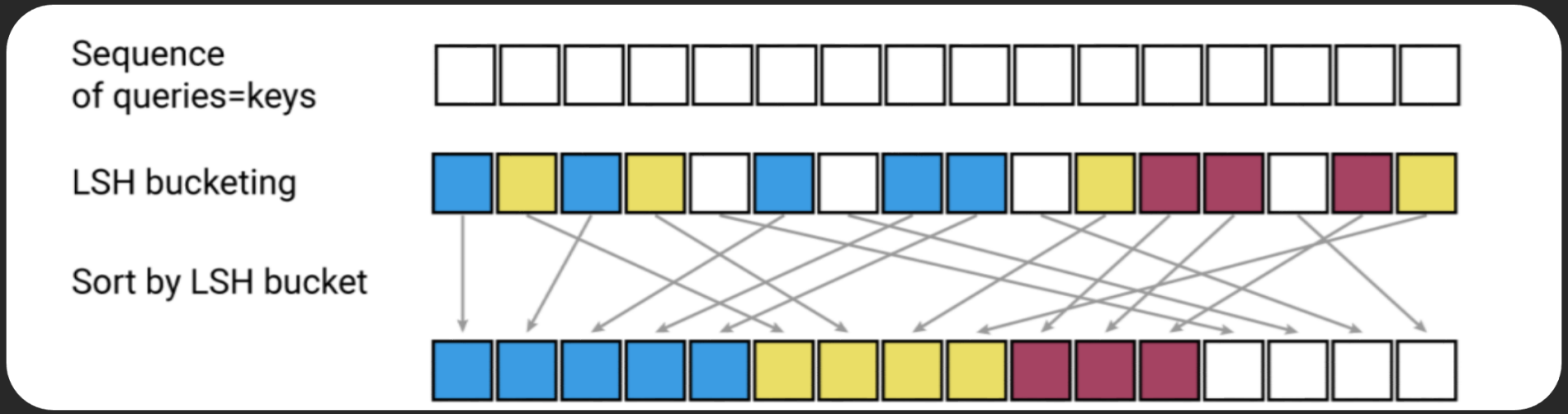


Fixed Patterns 利用只計算部分區域間的注意力權重，來降低計算量。

Learnable Patterns



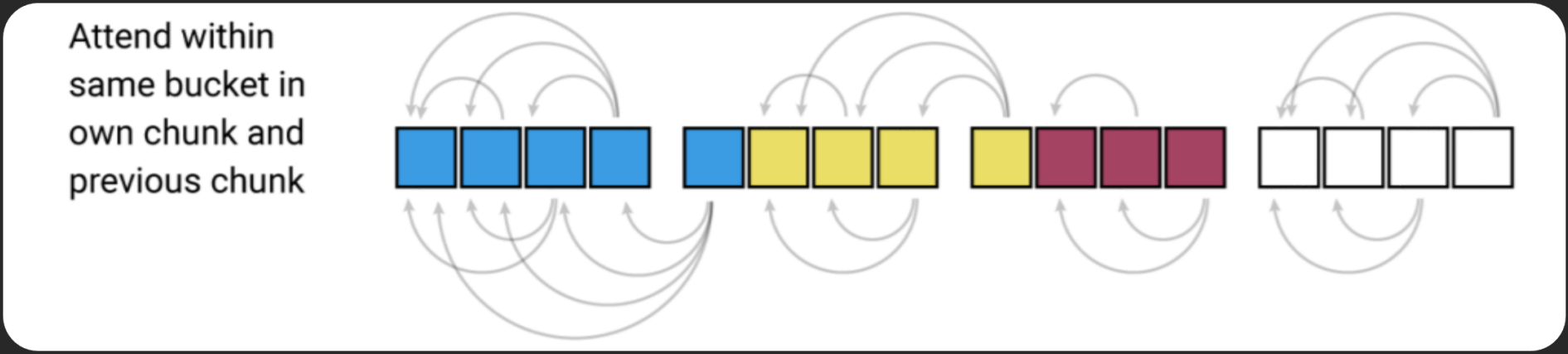
Sort



依據每個 key、query 的相似度，將其分組

- 在 Reformer 中使用 Locality Sensitive Hash 分組
- Routing Transformer 則是用 k-means 分組

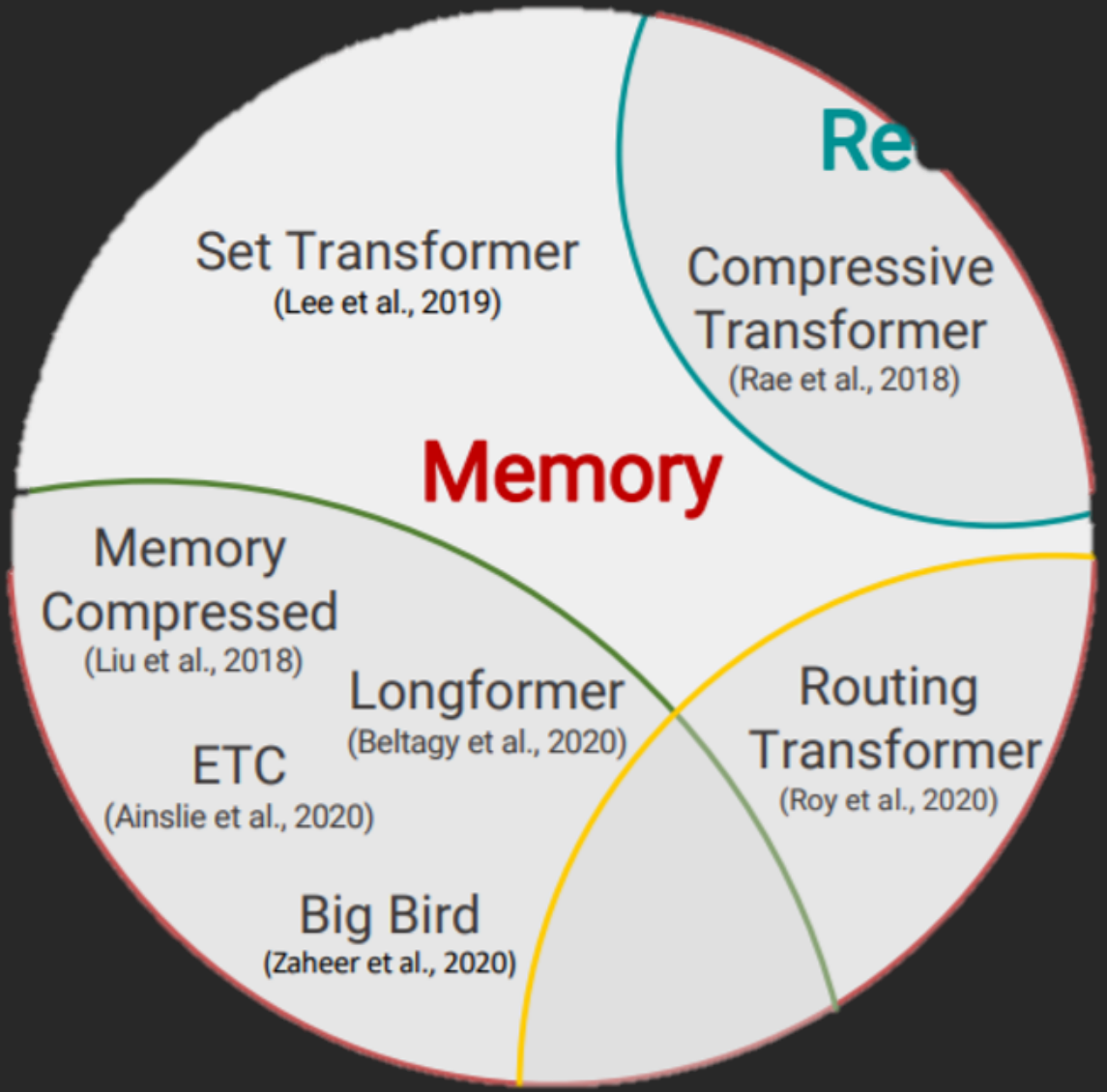
Attention



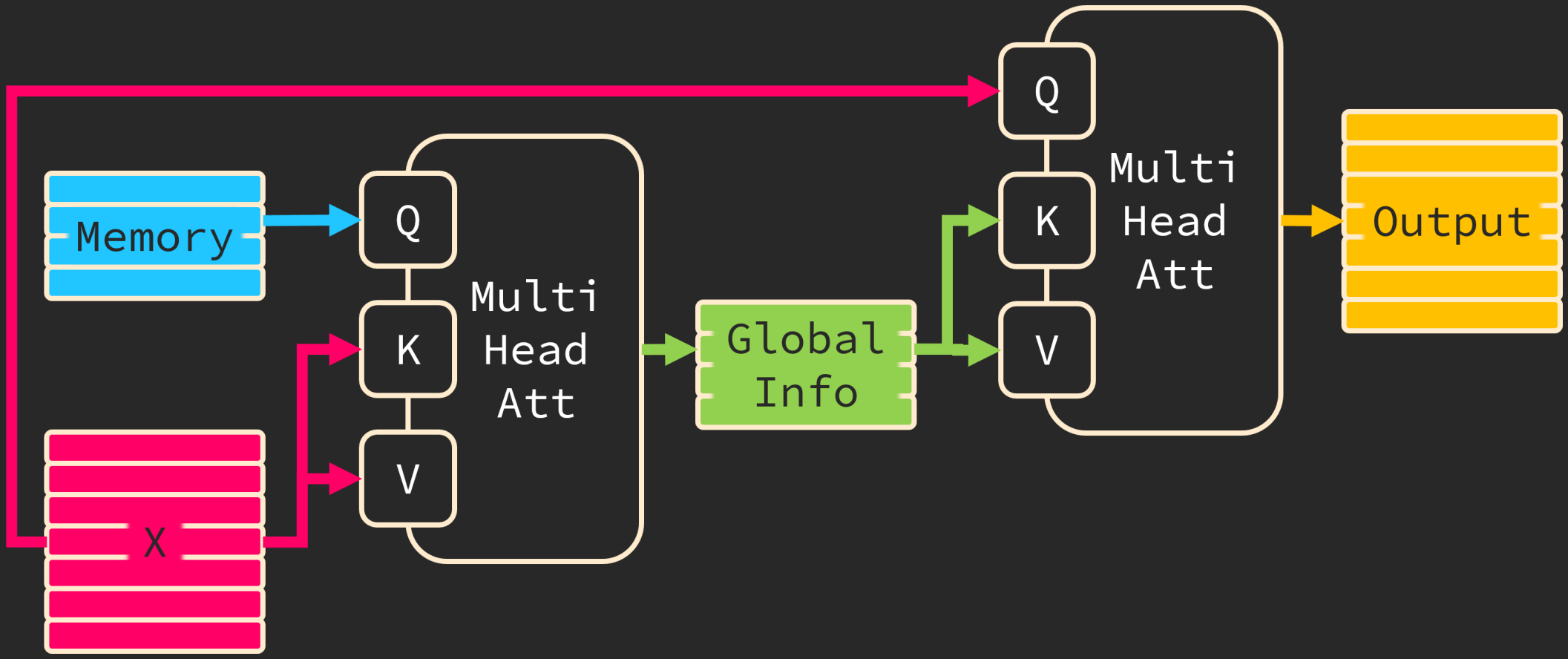
進行組內的 Self Attention

不同組代表相似度低，沒有互相匹配的必要

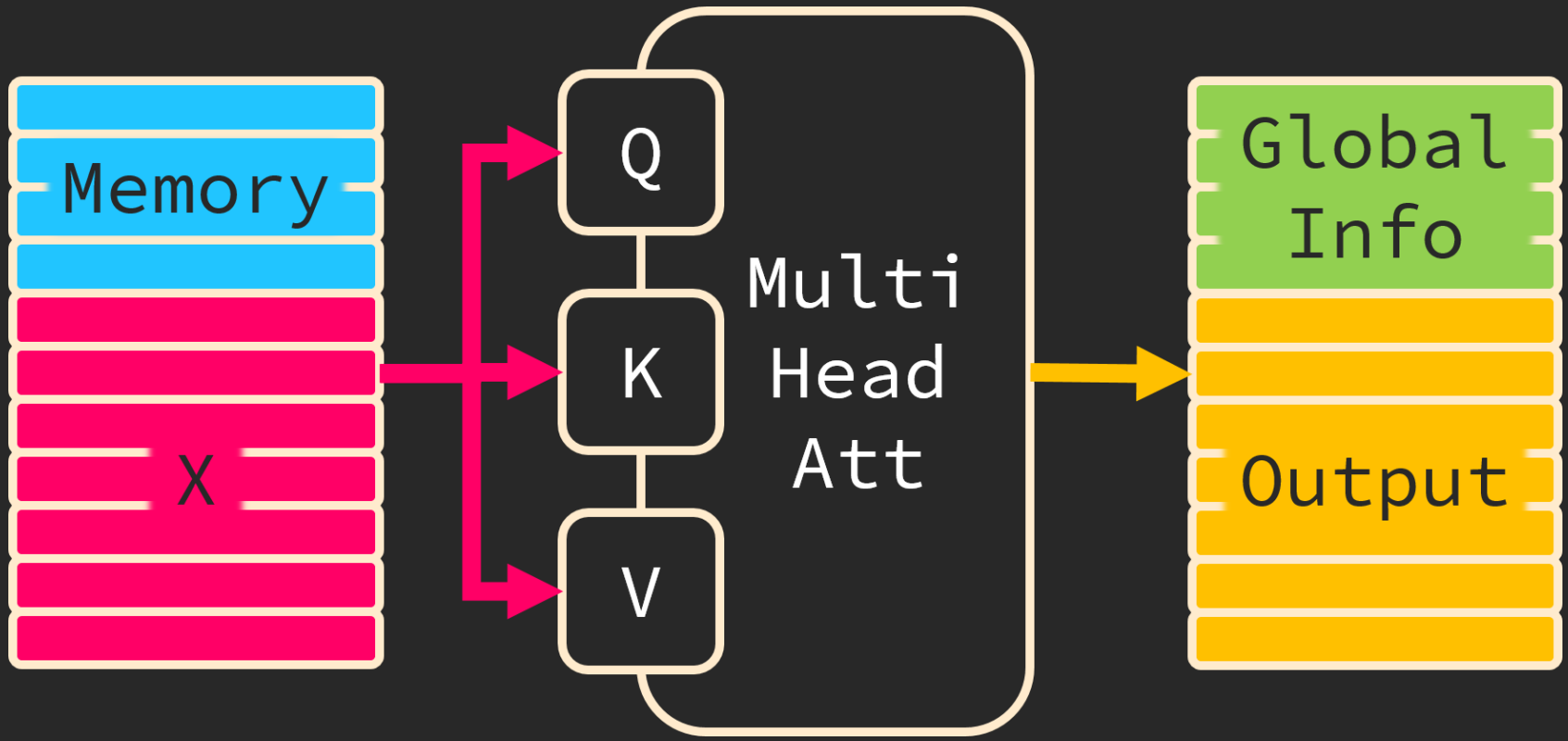
Memory


Set Transformer
(Lee et al., 2019)
Re
Compressive
Transformer
(Rae et al., 2018)
Memory
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)

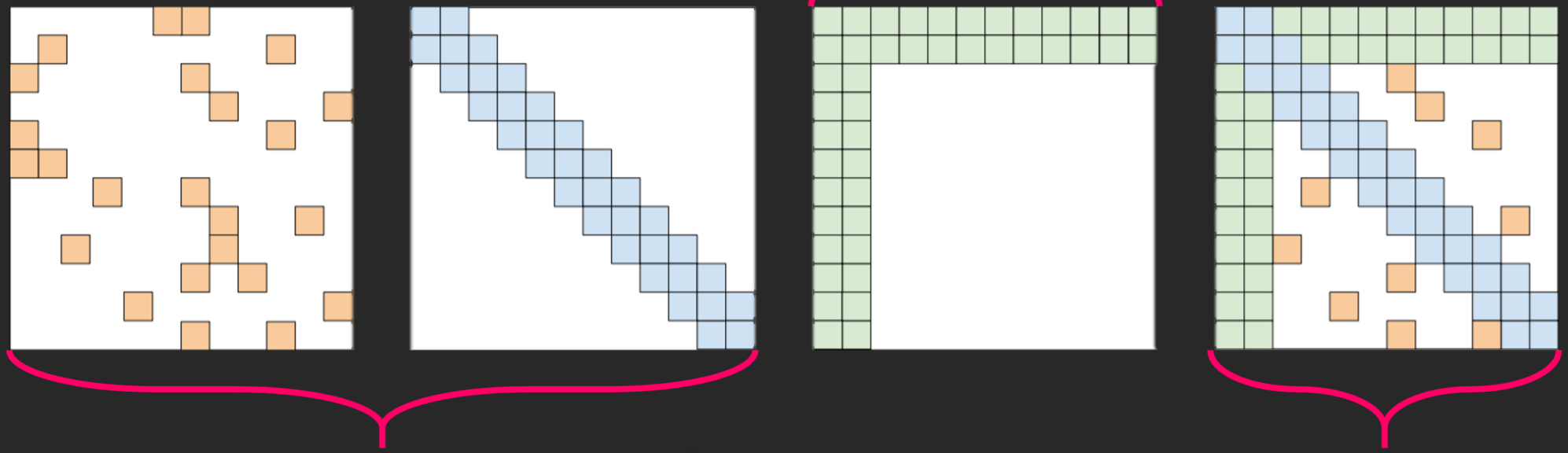
Set Transformer



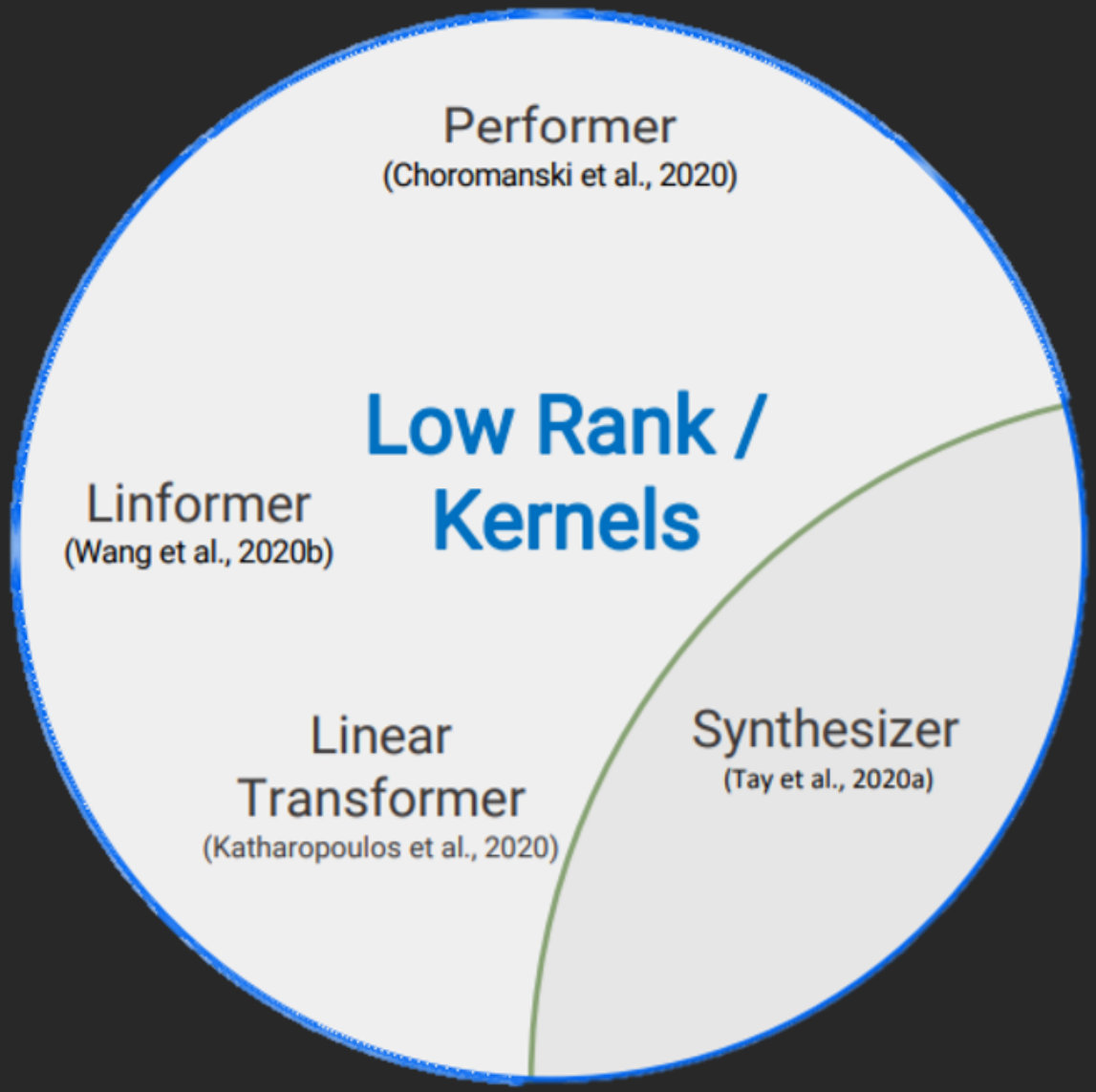
Global Attention



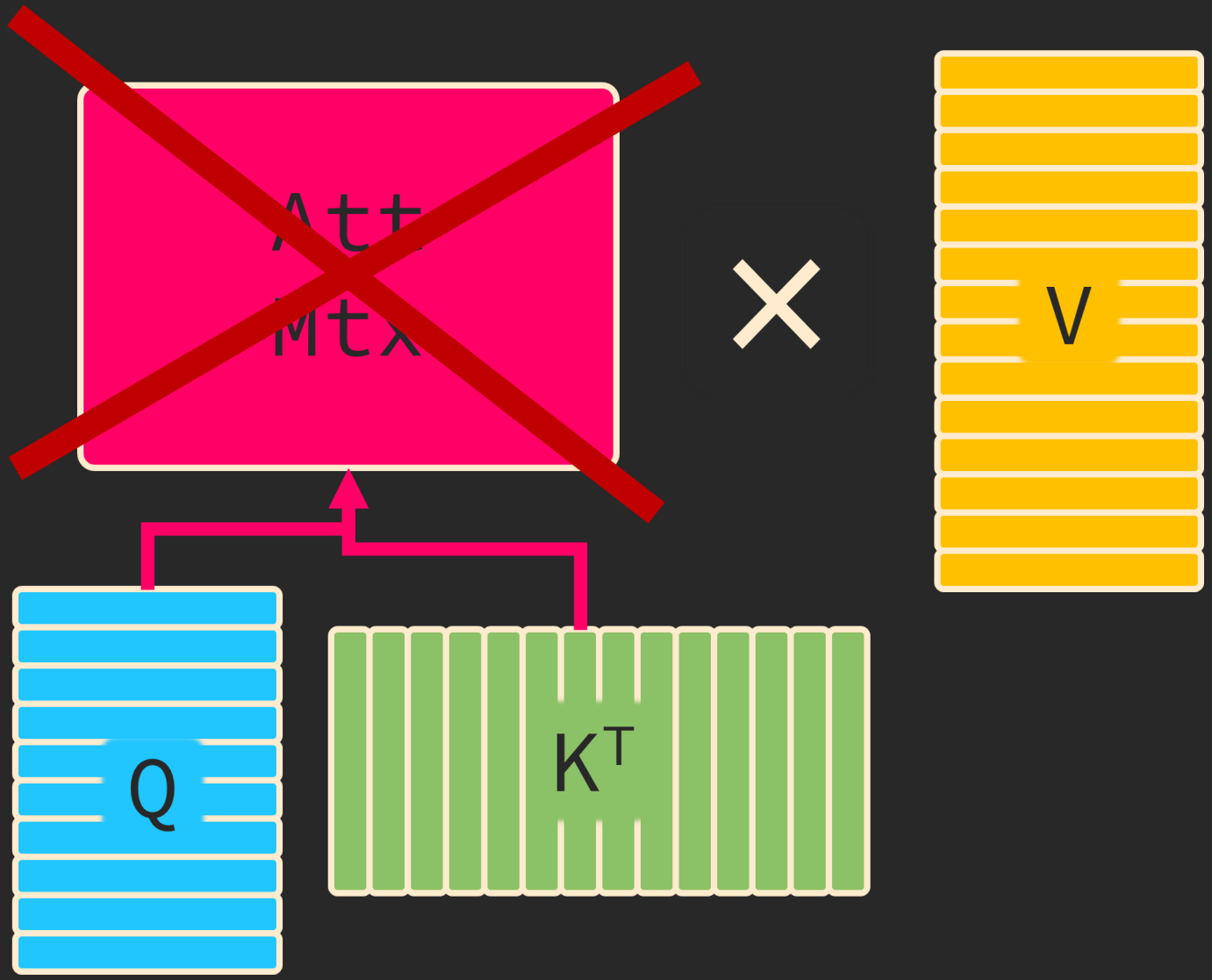
Big Bird



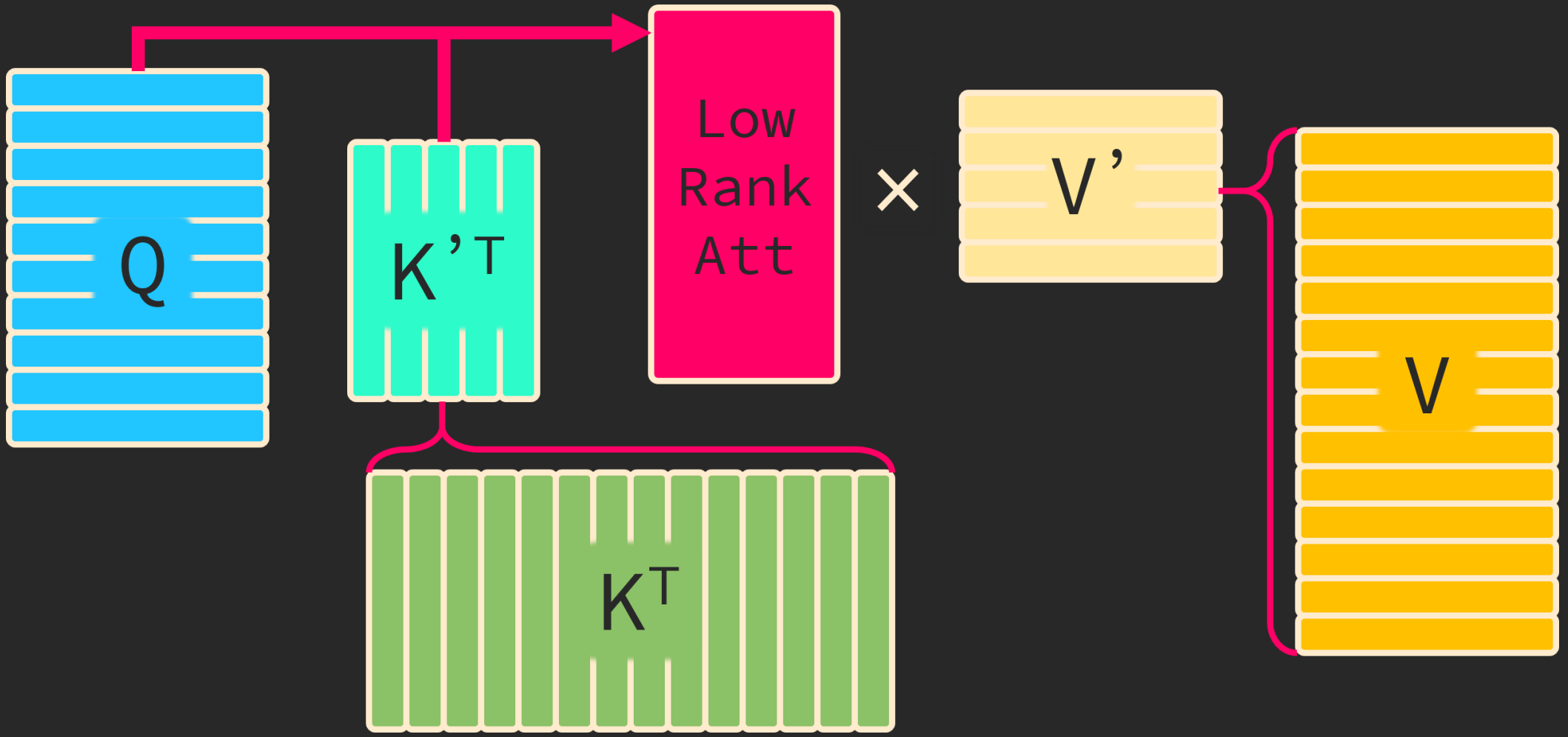
Low Rank & Kernel



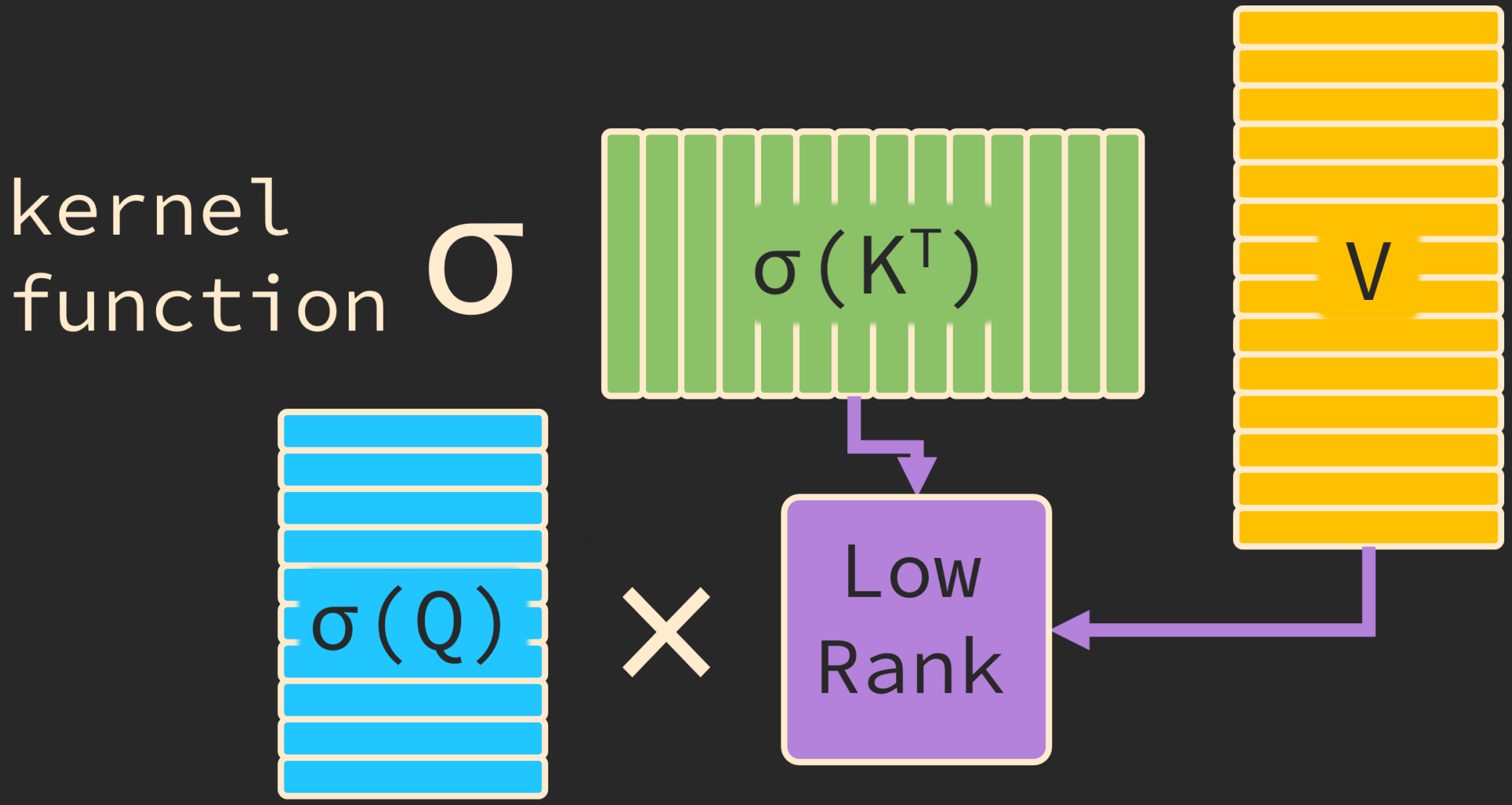
No Attention Matrix



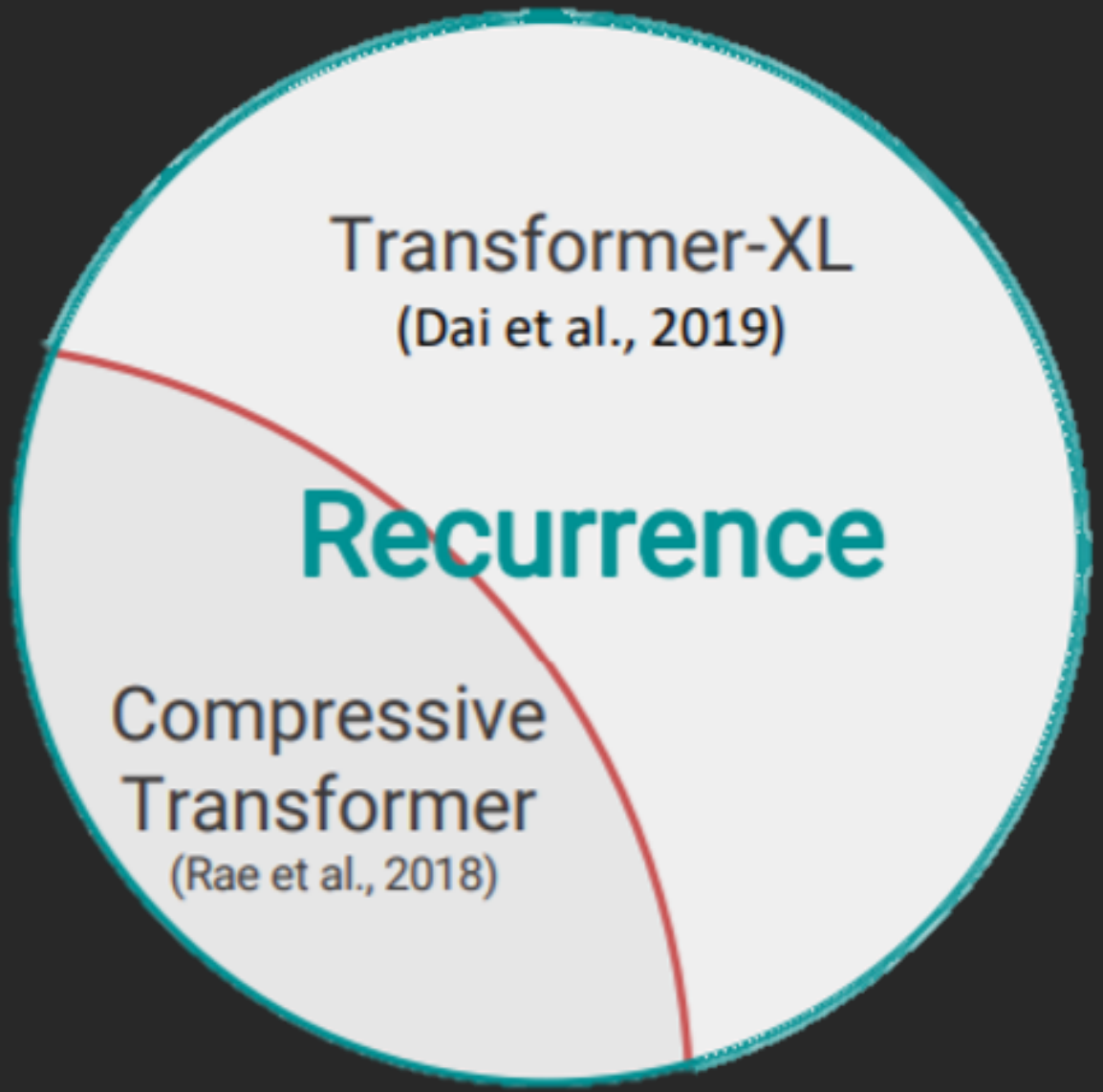
Linformer



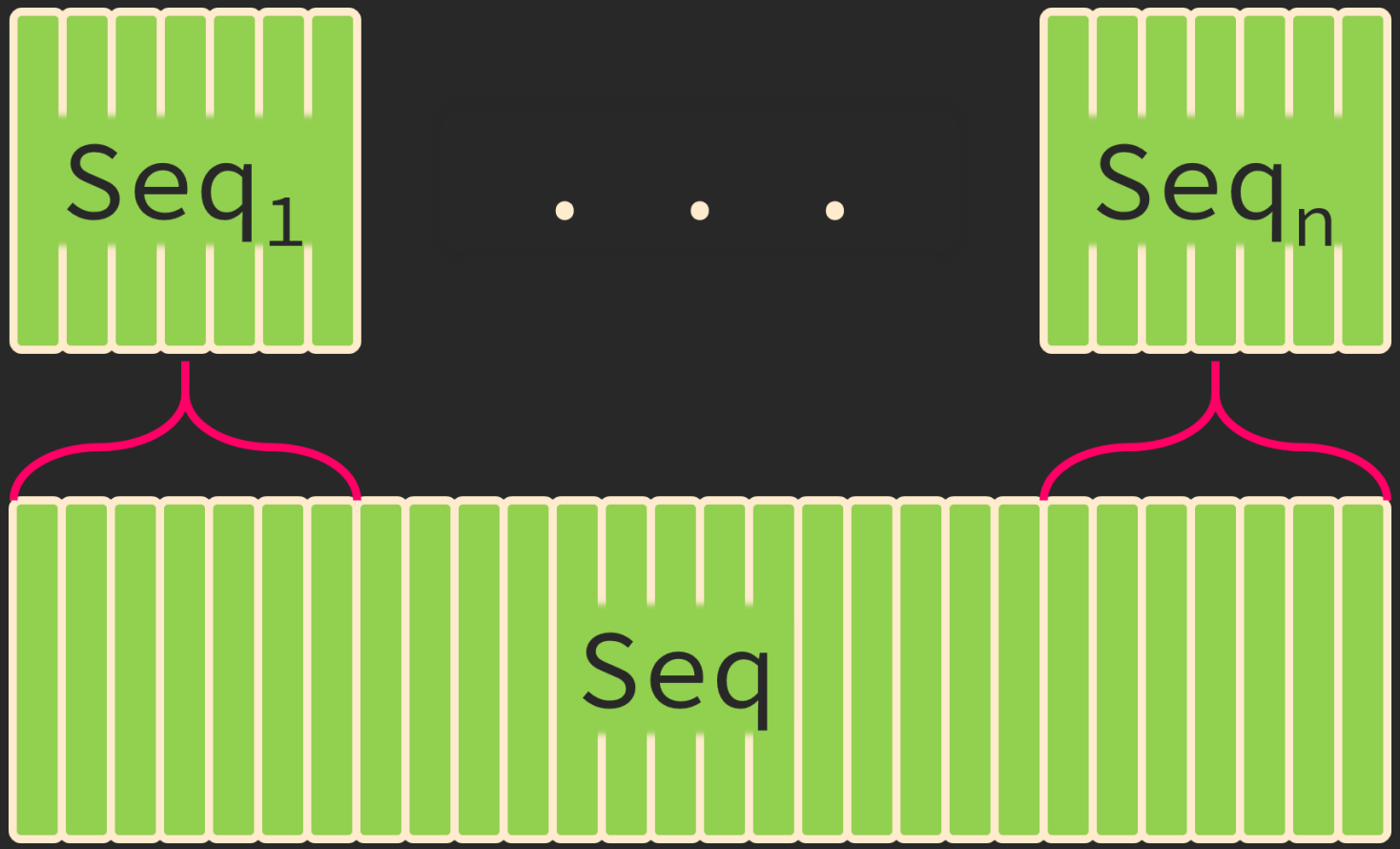
Linear Transformer
Performer



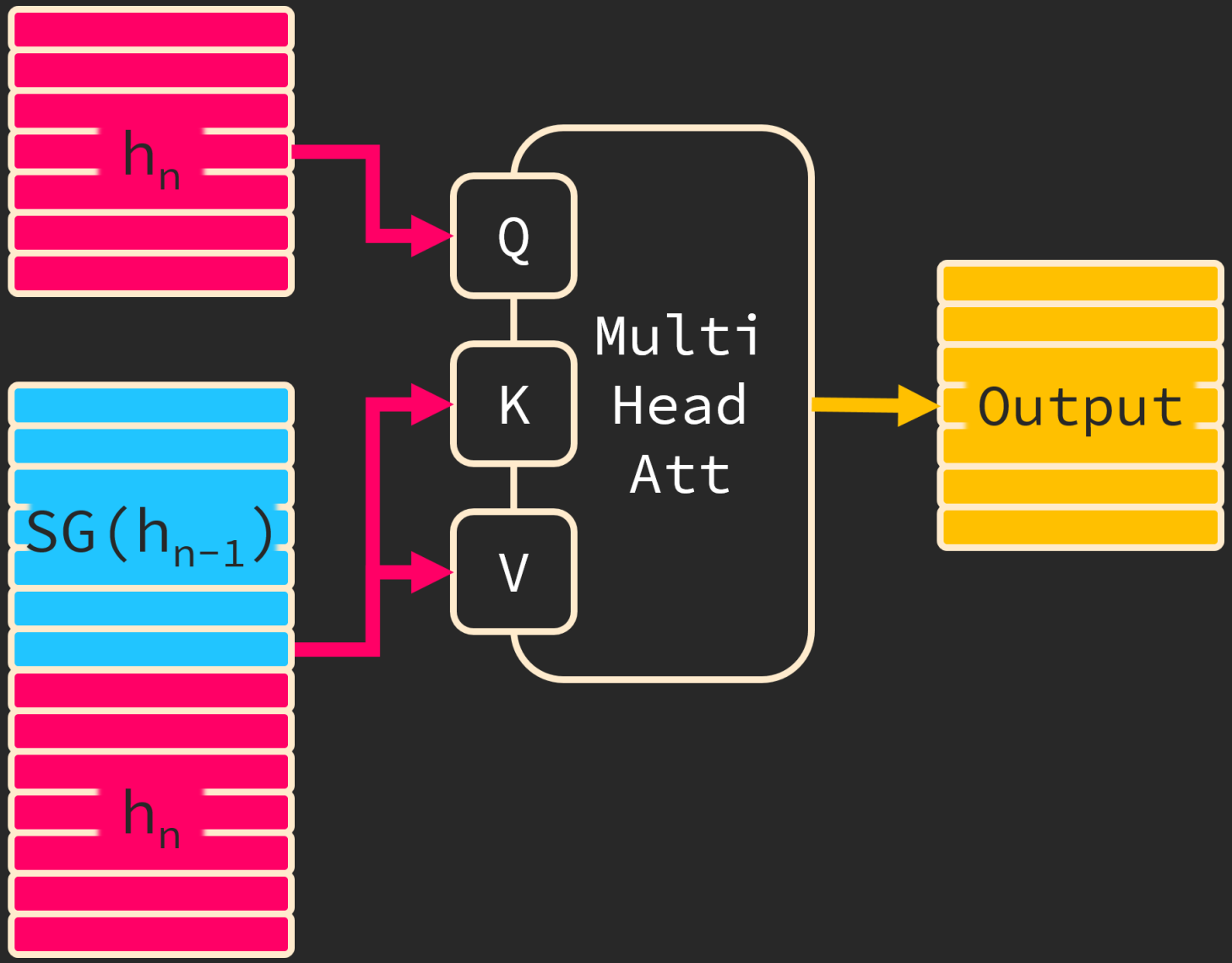
Recurrence


Transformer-XL
(Dai et al., 2019)
Recurrence
Compressive
Transformer
(Rae et al., 2018)

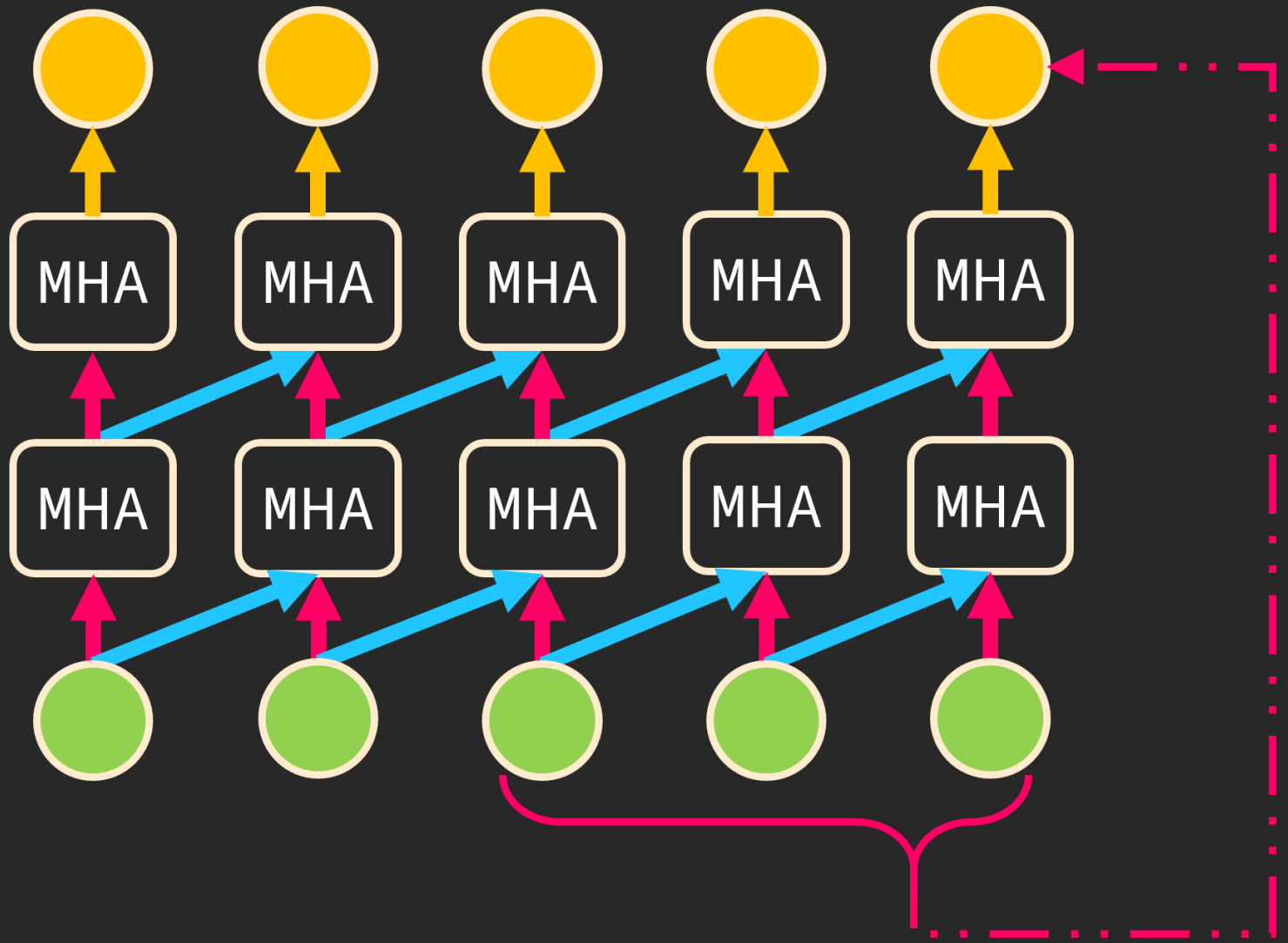
Transformer XL



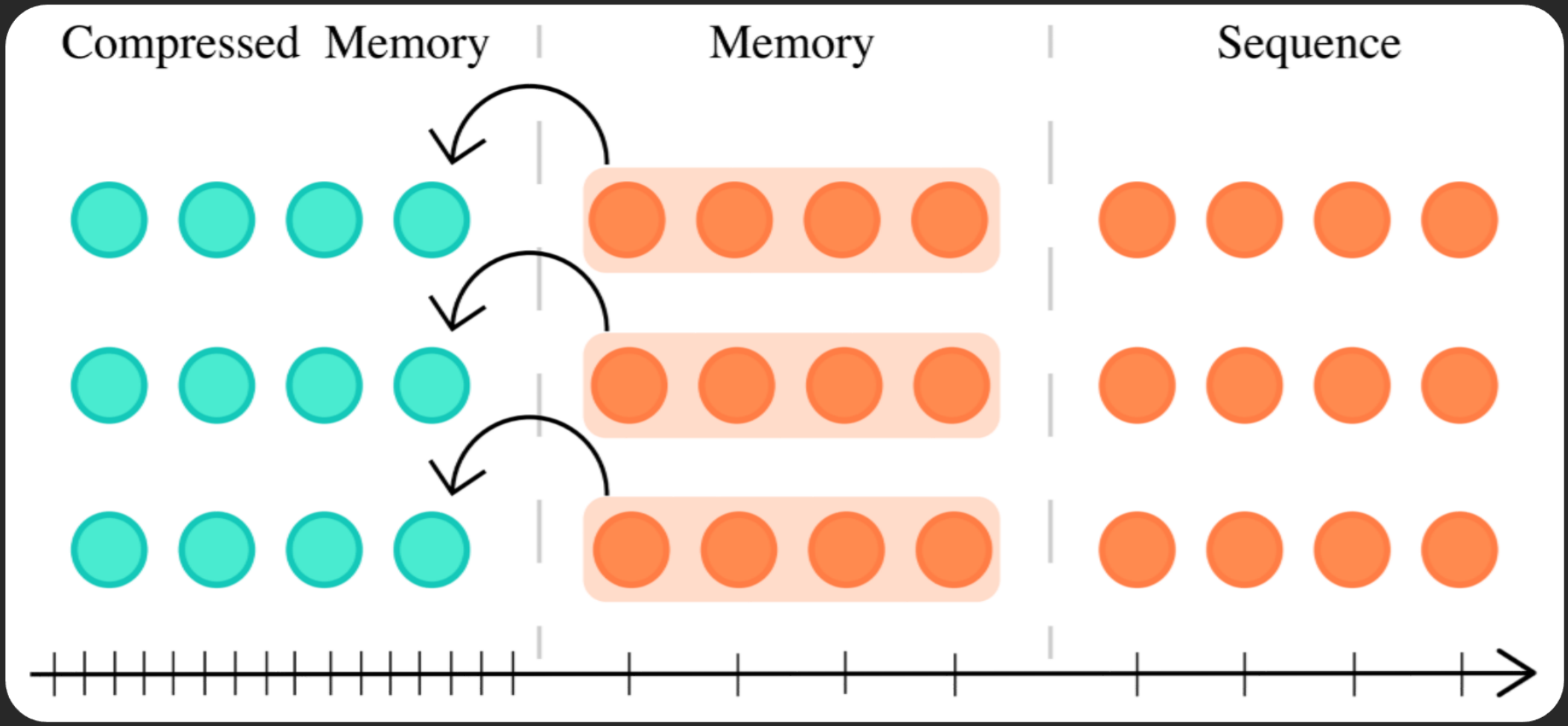
Transformer XL



Transformer XL



Compressive Transformer



Shortcomings

- Fixed Patterns
  - 固定的關注區塊很容易出現資訊缺失的狀況。
- Learnable Patterns
  - 在分組的過程中會產生 overhead，導致計算速度無法提升。
- Memory
  - 因壓縮上下文資訊，無法直接應用於自回歸任務。
  - 且壓縮亦會造成資訊缺損。

Shortcomings

- Low Rank & Kernel
  - 難以在自回歸任務實施平行化訓練。
  - Performer query 與 key 經過轉換後需要更大的 depth 才能維持正確率。
  - Linformer 將 keys&values 的長度 L 壓縮至 k，但為了保持正確率，k 大小還是與 L 有關。

Task

1. Long ListOpt
2. Byte-Level Text Classification
3. Byte-Level Document Retrieval
4. Image Classification on Sequences of Pixels
5. Pathfinder
6. ~~Pathfinder X~~

Long ListOpt

- Input
  [MAX 4 3
  [MIN 2 3]
  1 0
  [MEDIAN 1 5 8 9 2]]
- Output：5
- Max Seq Len：2K

由 max、min、medium、sum_mod 這幾種運算組成的前序表達式作為輸入，其運算答案作為輸出。

Byte-Level Text Classification

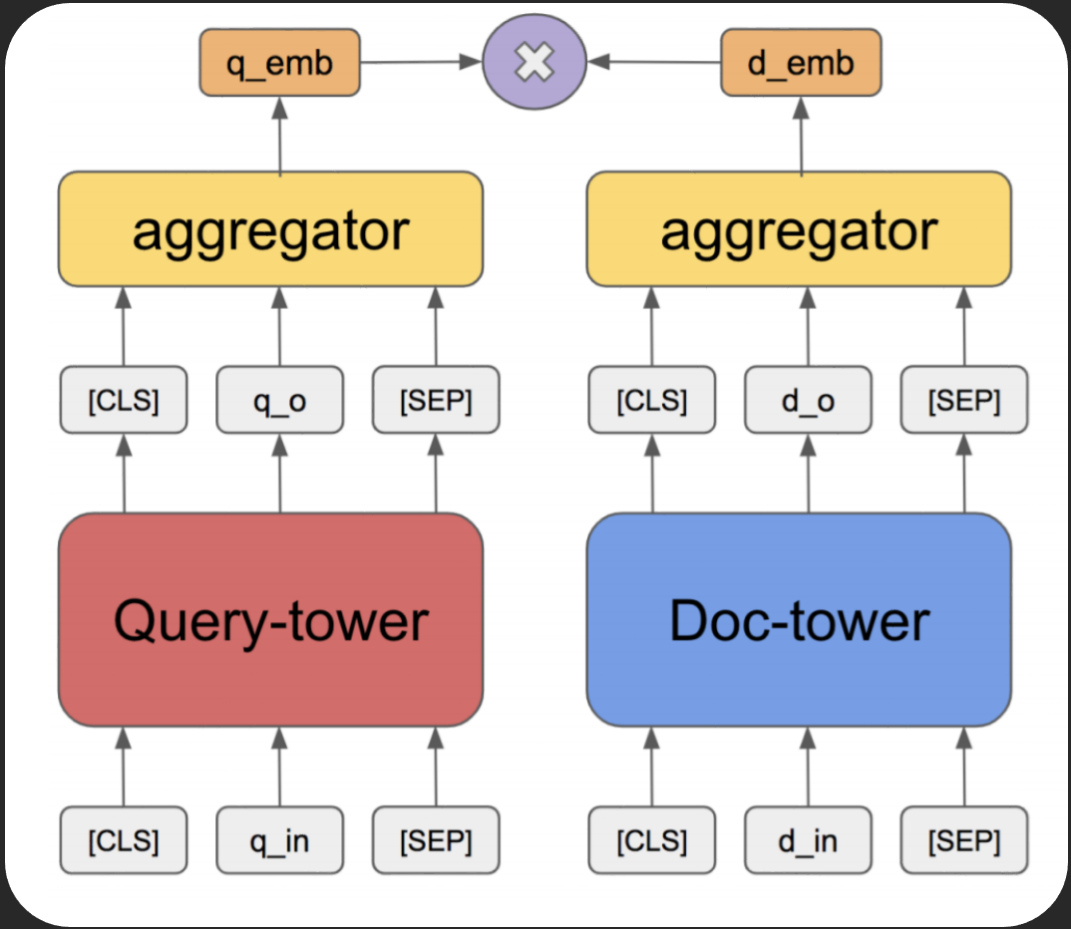
- Dataset：IMDb reviews
- Binary Sentiment Classification
- Max Seq Len：4K

Byte-Level Document Retrieval Task

- Dataset：ACL Anthology Network
- Two Tower Model, Not Cross-Attention
- Similarity Score
- Max Seq Len：4K & 4K

Byte-Level Document Retrieval

Two Tower



Cross Attention

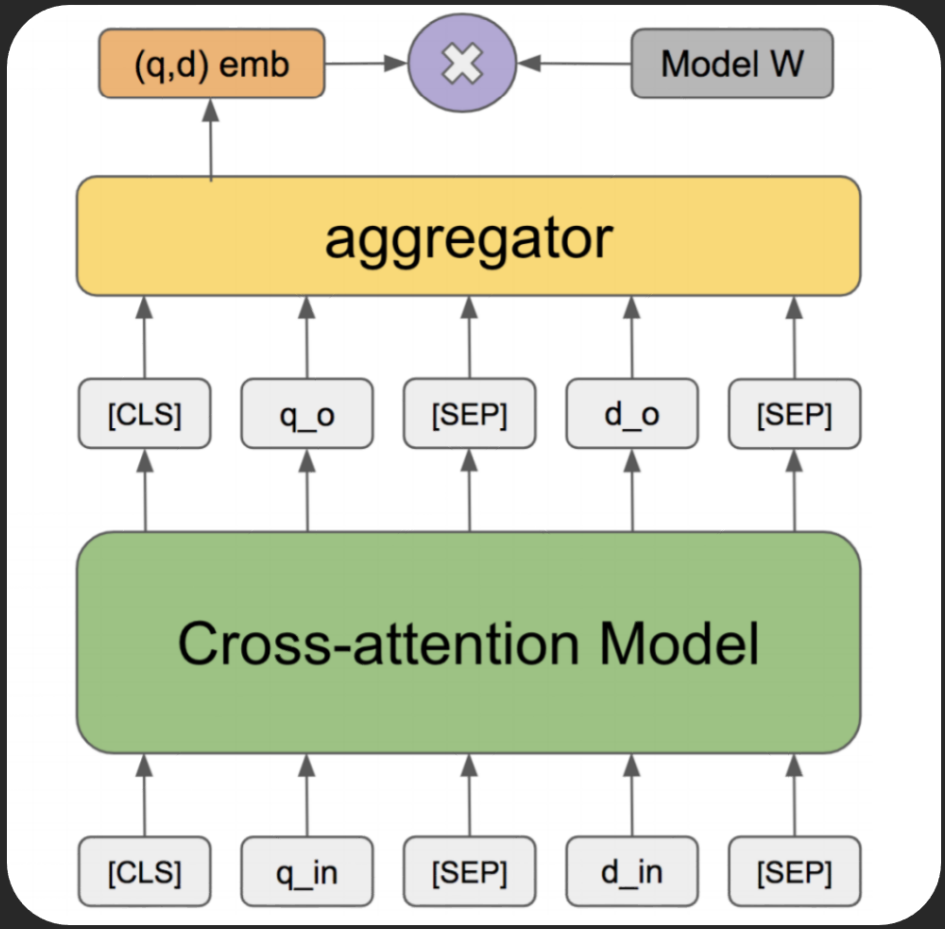
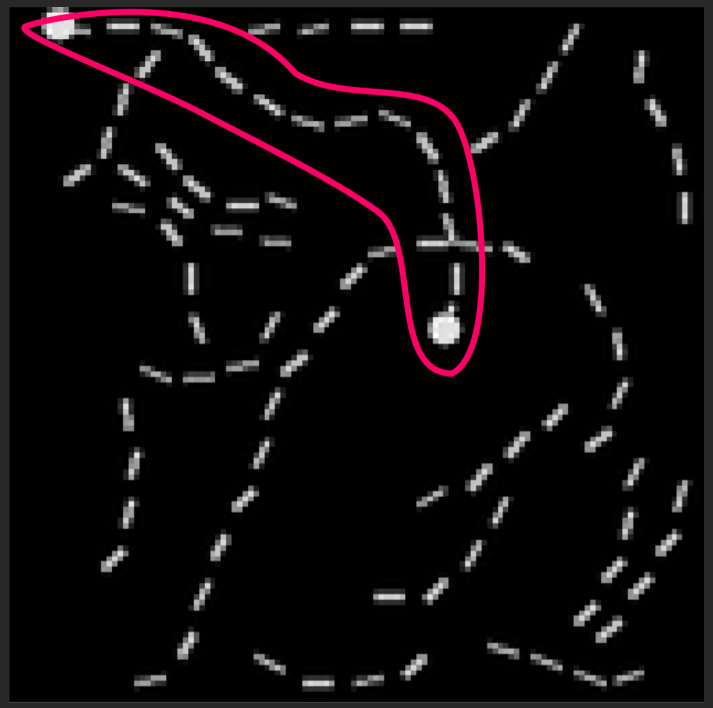


Image Classification

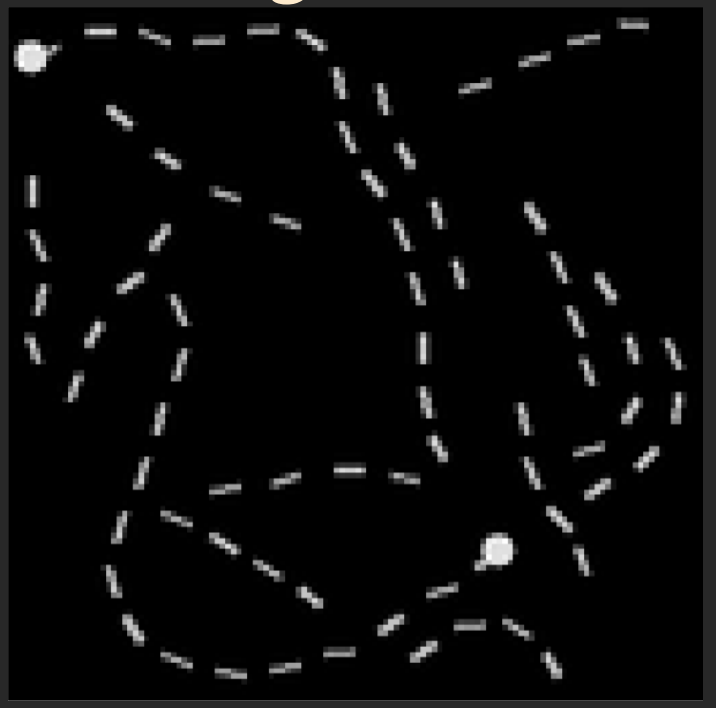
- Dataset：CIFAR-10
- Gray Scale to embedding index
- Vocabulary size of 256(=Gray Scale)
- Max Seq Len：1024 (32x32)

Pathfinder

Positive

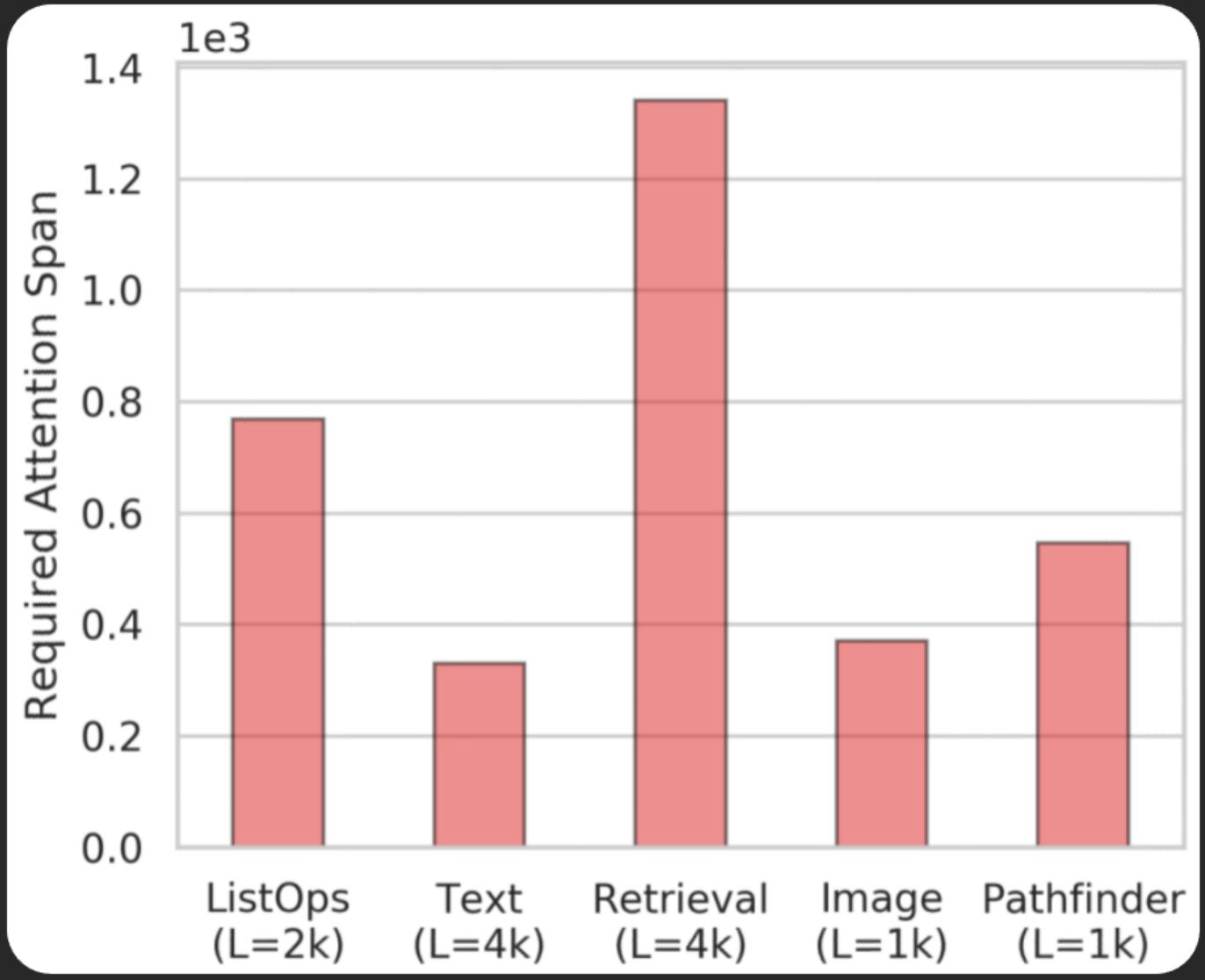


Negative



Max Seq Len：1024 (32x32)

Required Attention Span



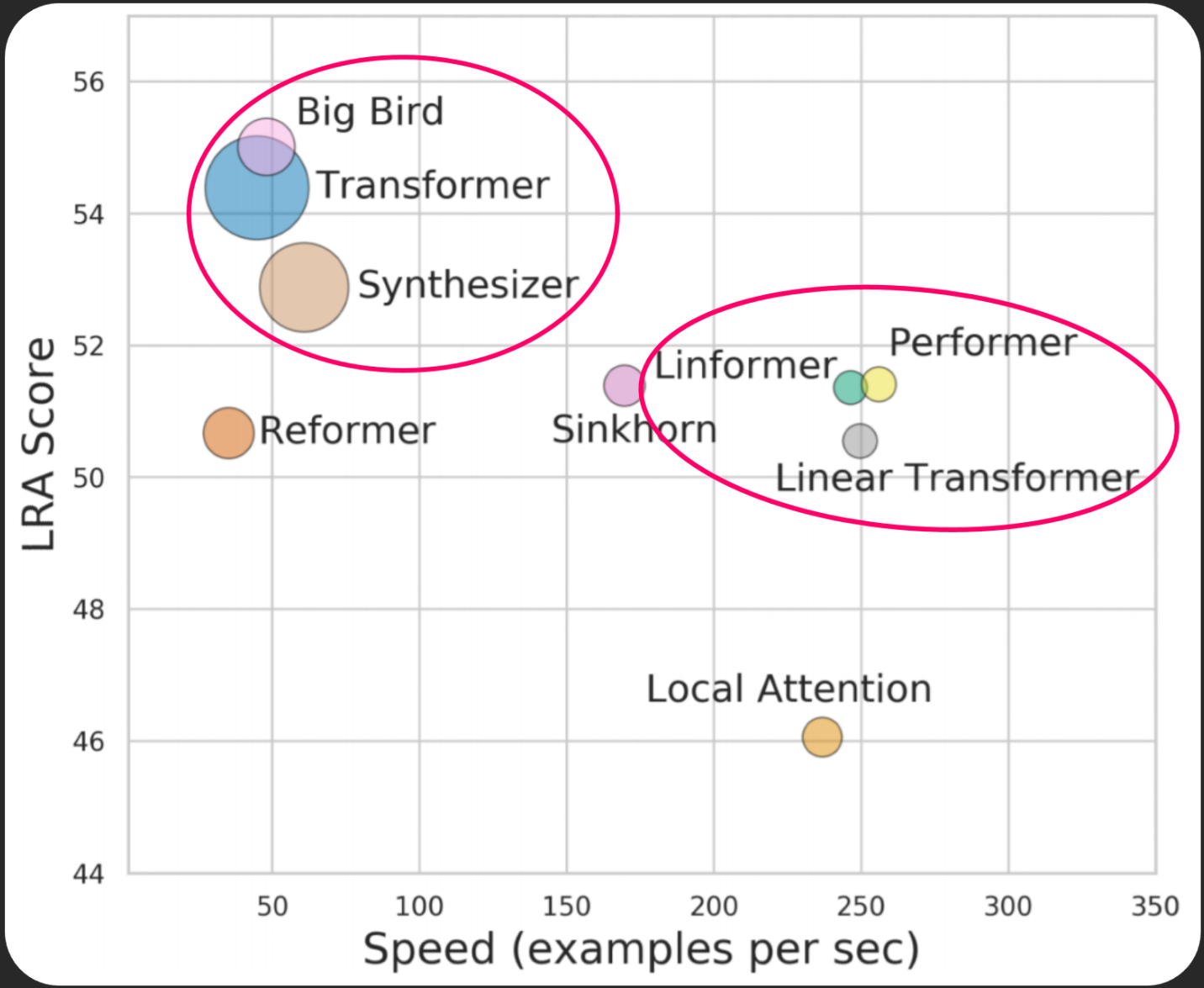
Task Score

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)

Power

Model	Steps per second 1K	2K	3K	4K	Peak Memory Usage (GB) 1K	2K	3K	4K
Transformer	8.1	4.9	2.3	1.4	0.85	2.65	5.51	9.48
Local Attention	9.2 (1.1x)	8.4 (1.7x)	7.4 (3.2x)	7.4 (5.3x)	0.42	0.76	1.06	1.37
Linformer	9.3 (1.2x)	9.1 (1.9x)	8.5 (3.7x)	7.7 (5.5x)	**0.37**	**0.55**	0.99	**0.99**
Reformer	4.4 (0.5x)	2.2 (0.4x)	1.5 (0.7x)	1.1 (0.8x)	0.48	0.99	1.53	2.28
Sinkhorn Trans	9.1 (1.1x)	7.9 (1.6x)	6.6 (2.9x)	5.3 (3.8x)	0.47	0.83	1.13	1.48
Synthesizer	8.7 (1.1x)	5.7 (1.2x)	6.6 (2.9x)	1.9 (1.4x)	0.65	1.98	4.09	6.99
BigBird	7.4 (0.9x)	3.9 (0.8x)	2.7 (1.2x)	1.5 (1.1x)	0.77	1.49	2.18	2.88
Linear Trans.	9.1 (1.1x)	9.3 (1.9x)	8.6 (3.7x)	7.8 (5.6x)	**0.37**	0.57	**0.80**	1.03
Performer	**9.5** (1.2x)	**9.4** (1.9x)	**8.7** (3.8x)	**8.0** (5.7x)	**0.37**	0.59	0.82	1.06

Performance



Conclusion

- 沒有銀彈。
- 結合 Local Attention 與 Global Attention 的方式，可以在降低記憶體複雜度的情況維持正確率。
- 基於 Low Rank 的方法，在「長序列任務」上具有優異的速度與低記憶體複雜度，但卻不適合用於自回歸任務上。
- 長序列自回歸任務適合使用基於 Recurrence 的方法。

Problem

為何生成任務都要用自回歸模型？