How to deal with long sequence in self-attention

어떻게 긴 시퀀스가 셀프 어텐션에 들어갈 수 있을까?

최신 모델 context window 동향

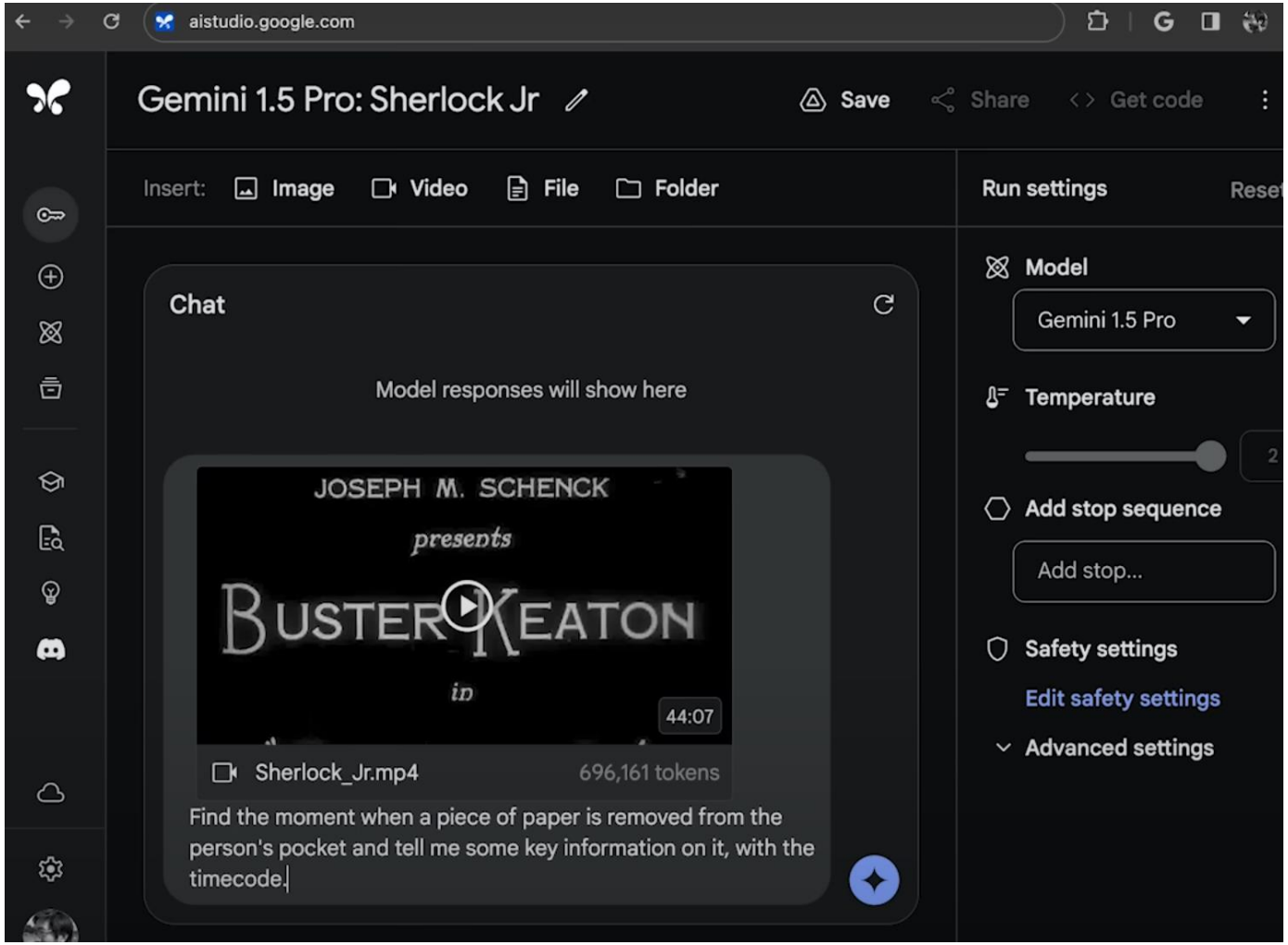
- Llama 3 → Llama 3.1 (8K → 128K)
- Mistral xxx → Mistral NeMo (32K → 128K)
- GPT-4-0613 → GPT-4-1106-preview (8K → 128K)

- Claude 2.1 (200K)
- Gemini 1.5 Pro (2M)
- Command R+ (128K)

- HyperClova X: 4K…

왜 context window가 크면 성능이 좋을까

사진, 비디오까지 고려한다면...?



44분짜리 영상에서

'어떤 사람의 주머니에서 종이가 떨어지는 순간을 찾아 묘사해라
또 사건이 일어난 영상 시각은?'

이라고 묻고 있음

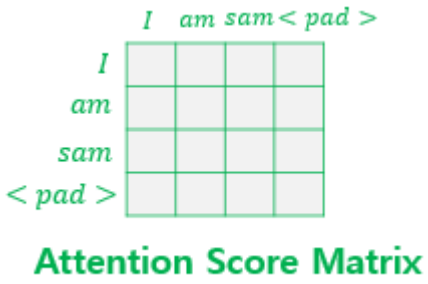
셀프 어텐션...

- $\mathbf{W}_{\mathrm{Q}}$
- $\mathbf{W}_{\mathrm{K}}$
- $\mathbf{W}_{\mathrm{V}}$

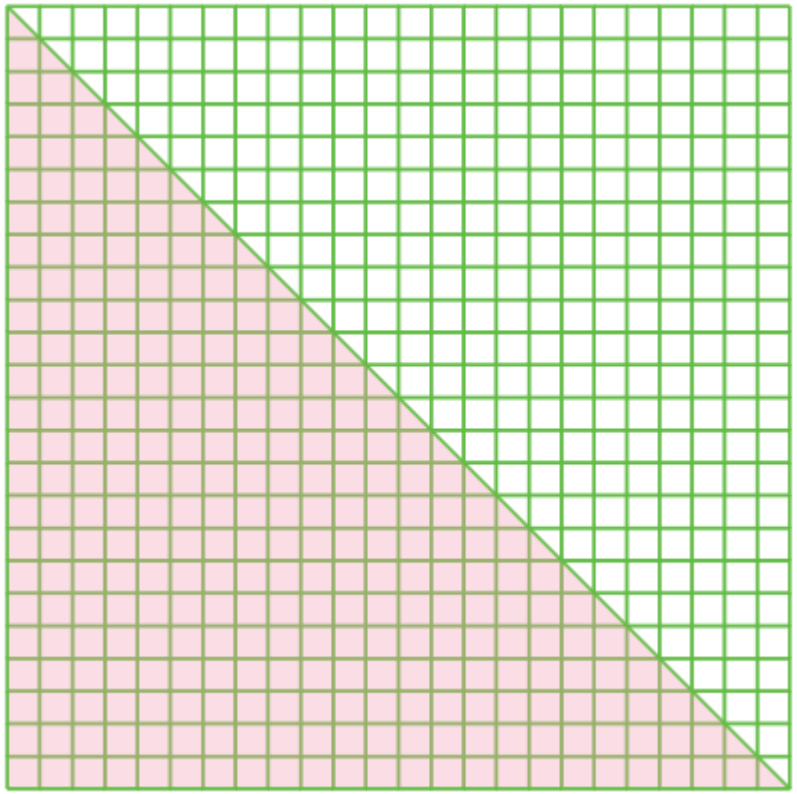
$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_{\mathrm{K}}}})\mathbf{V}$$

긴 시퀀스 처리가 힘든 이유?

Context window: 4K
(그림은 4개지만 4K개라고 생각하고 넘어가자)



Context window: 128K



Attention Score Matrix

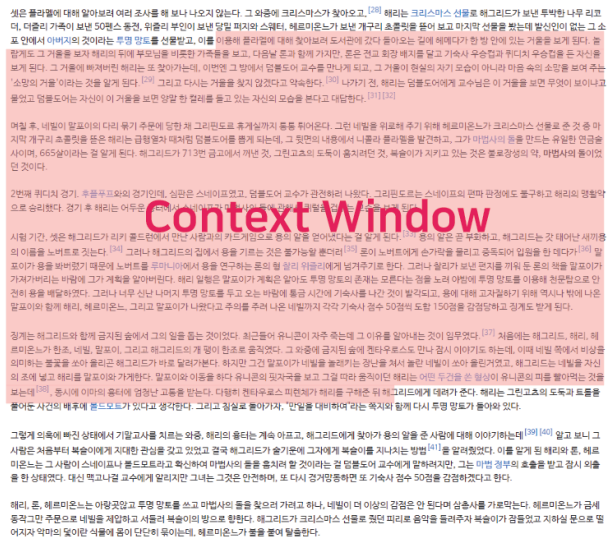
4K * 4K = 16M

128K * 128K = 16.4G

"셀프 어텐션 시 메모리, 연산량은 context window의 제곱에 비례"

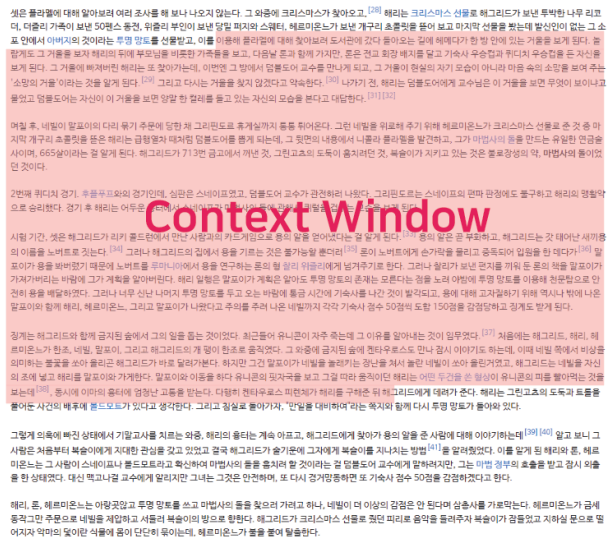
용어 정리: context window란?

"모델이 훈련할 때 사용한 시퀀스 길이"



용어 정리: context window란?

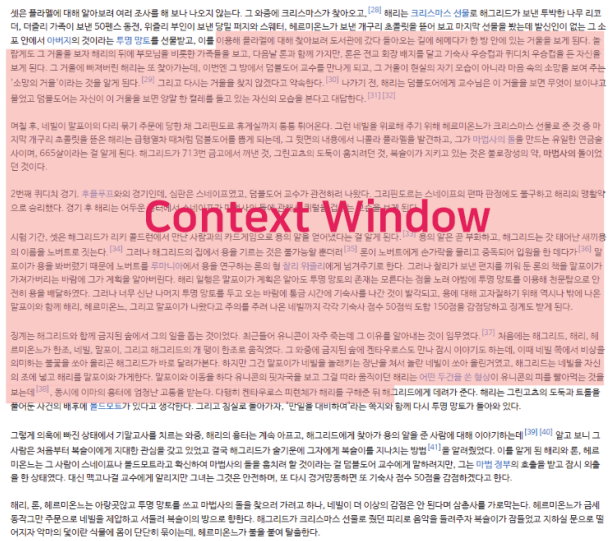
"모델이 훈련할 때 사용한 시퀀스 길이"



Q:
Huggingface에서 Context window가 4096인 Upstage-SOLAR 모델을 다운받았다.
여기다가 10000토큰 문장을 truncation 없이 넣고 훈련이 가능할까?

용어 정리: context window란?

"모델이 훈련할 때 사용한 시퀀스 길이"



Q:
Huggingface에서 Context window가 4096인 Upstage-SOLAR 모델을 다운받았다.
여기다가 10000토큰 문장을 truncation 없이 넣고 훈련이 가능할까?

A: 가능

즉, context window가 '길이 제한' 을 의미하지는 않음

Sliding Window Attention

논문

Longformer: The Long-Document Transformer

Iz Beltagy* **Matthew E. Peters*** **Arman Cohan***
Allen Institute for Artificial Intelligence, Seattle, WA, USA
{beltagy,matthewp,armanc}@allenai.org

Generating Long Sequences with Sparse Transformers

Rewon Child [1] **Scott Gray** [1] **Alec Radford** [1] **Ilya Sutskever** [1]

모델

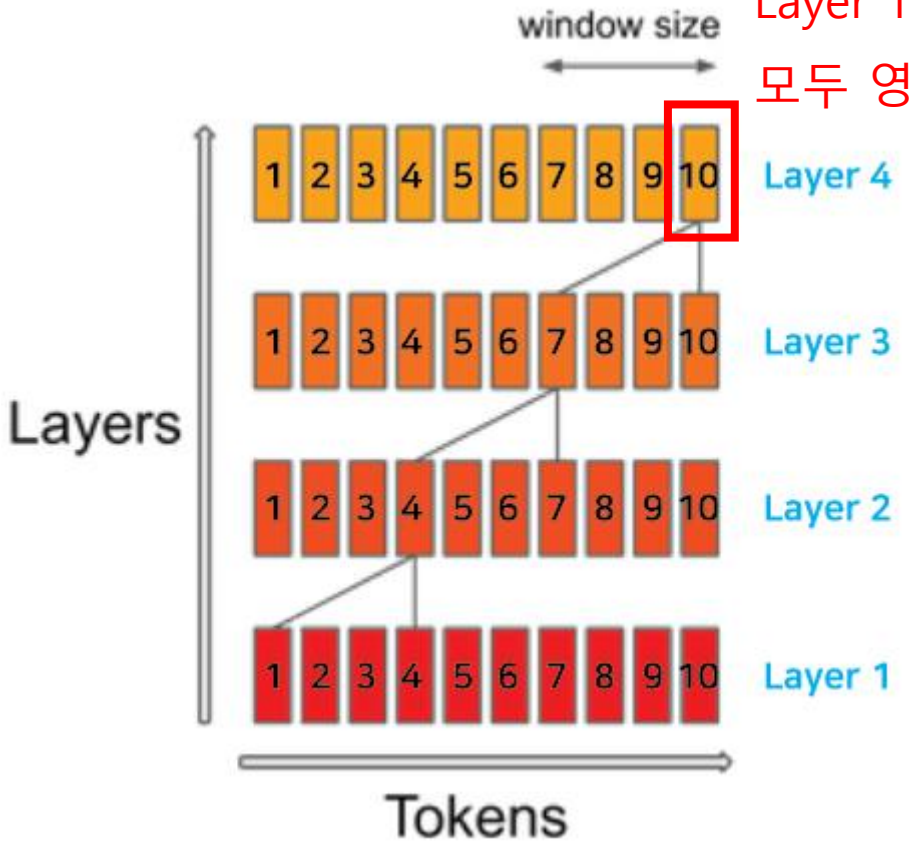
Mistral 초기 모델에만 들어가 있고,
요즘 모델에는 사용되지 않음!

Sliding Window Attention

Attend할 수 있는 토큰 개수를

[window_size]개로 제한!

(예시 window_size=4)



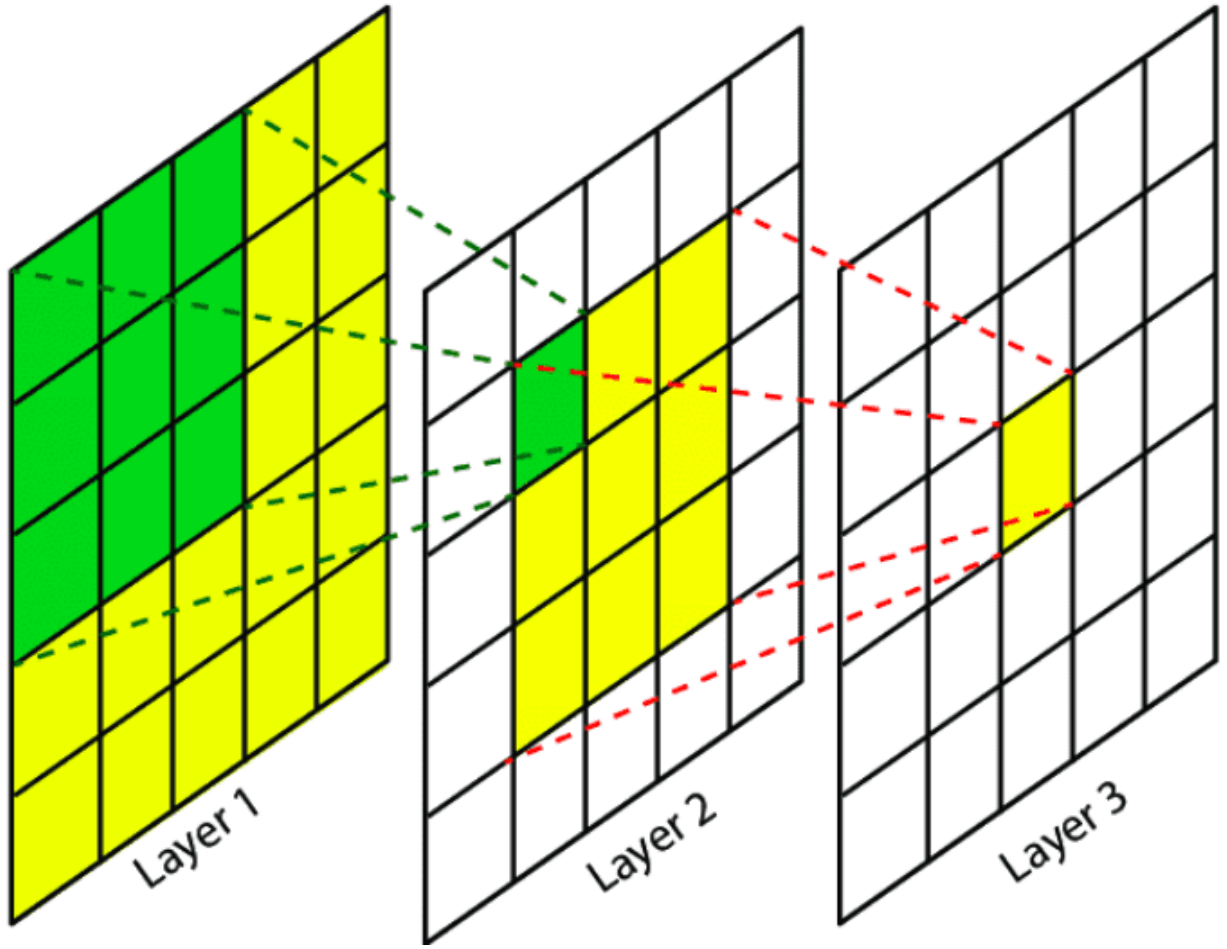
결국 Layer 4의 10번 토큰은

Layer 1의 1~10번 토큰에

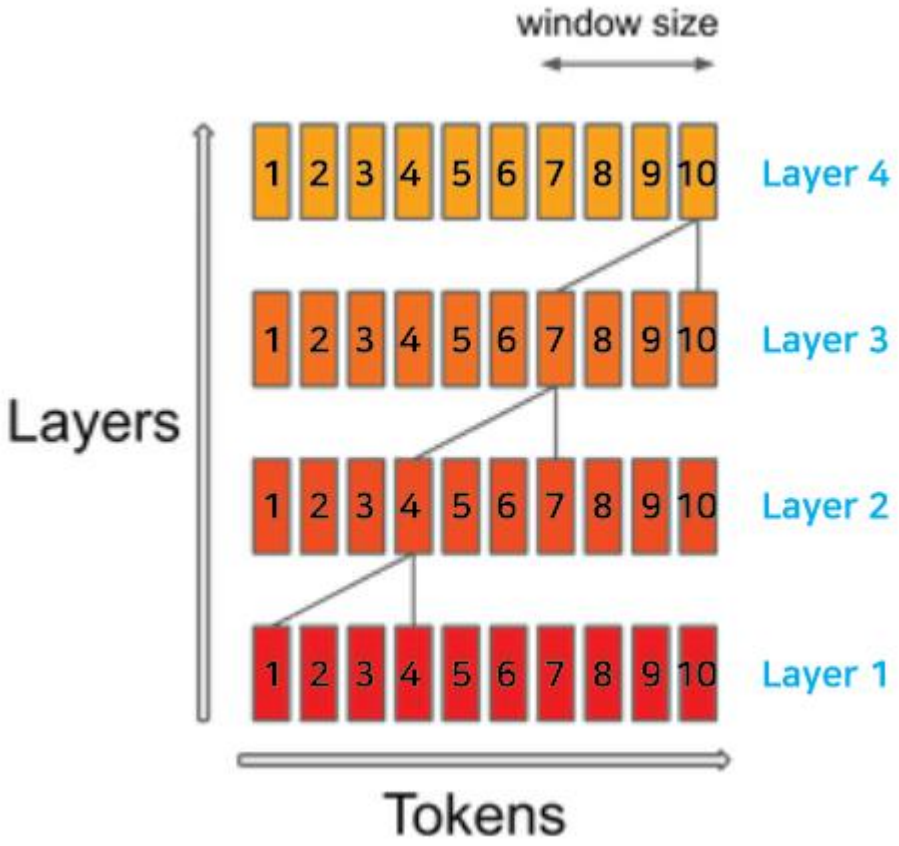
모두 영향을 받게 됨

Sliding Window Attention

CNN의 Receptive Field
개념과 비슷하다.



Sliding Window Attention 문제



- Global Attention (모든 토큰을 전부 보는 방식)에 비해 성능은...?
- 메모리의 효율성은...?

논문 1: Longformer: The long-document transformer

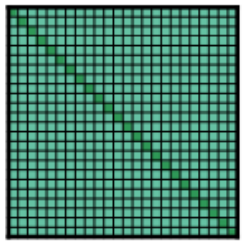
Longformer: The Long-Document Transformer

Iz Beltagy* **Matthew E. Peters*** **Arman Cohan***
Allen Institute for Artificial Intelligence, Seattle, WA, USA
{beltagy,matthewp,armanc}@allenai.org

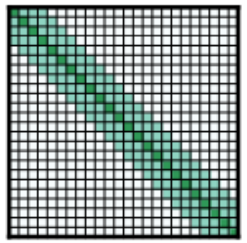
Sliding window attention의 아이디어를 LM에서 구체화한 논문
사실 아이디어는 두 번째 논문이 다 하긴 했지만…

그래도 Language model을 타겟으로 구체화해서 그런지, 인용 수가 많음
(3700+ 회)

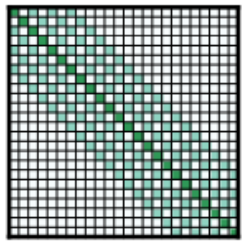
논문 1: Longformer: The long-document transformer



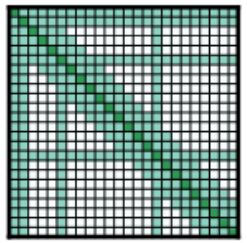
(a) Full n^2 attention



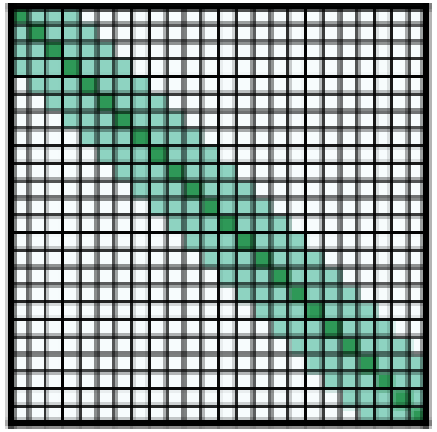
(b) Sliding window attention



(c) Dilated sliding window



(d) Global+sliding window



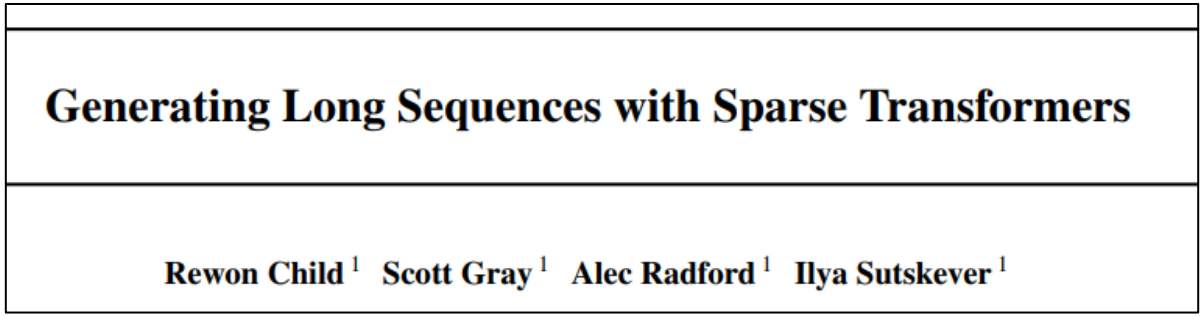
(b) Sliding window attention

논문 1: Longformer: The long-document transformer

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)

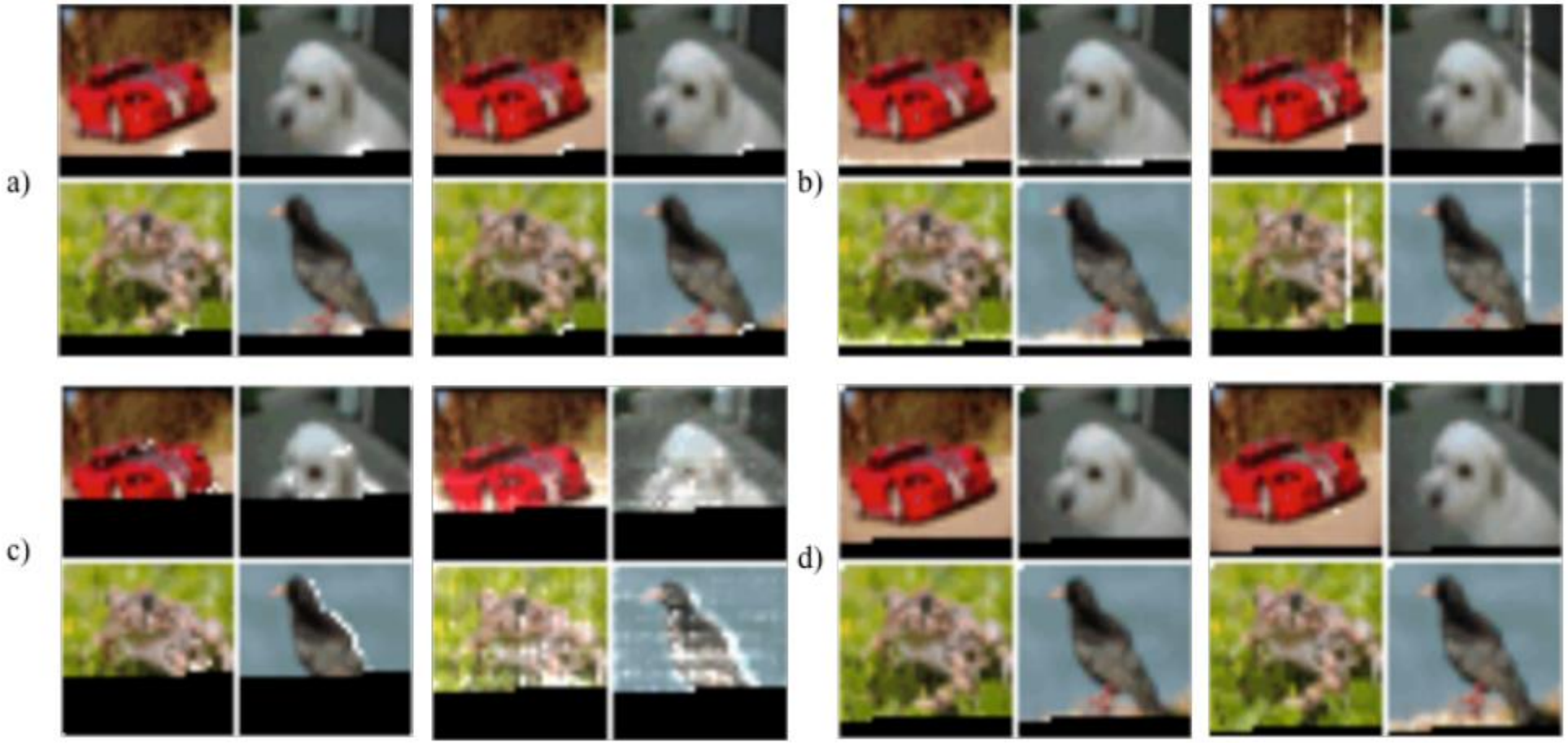
논문 2: Generating Long Sequences with Sparse Transformers



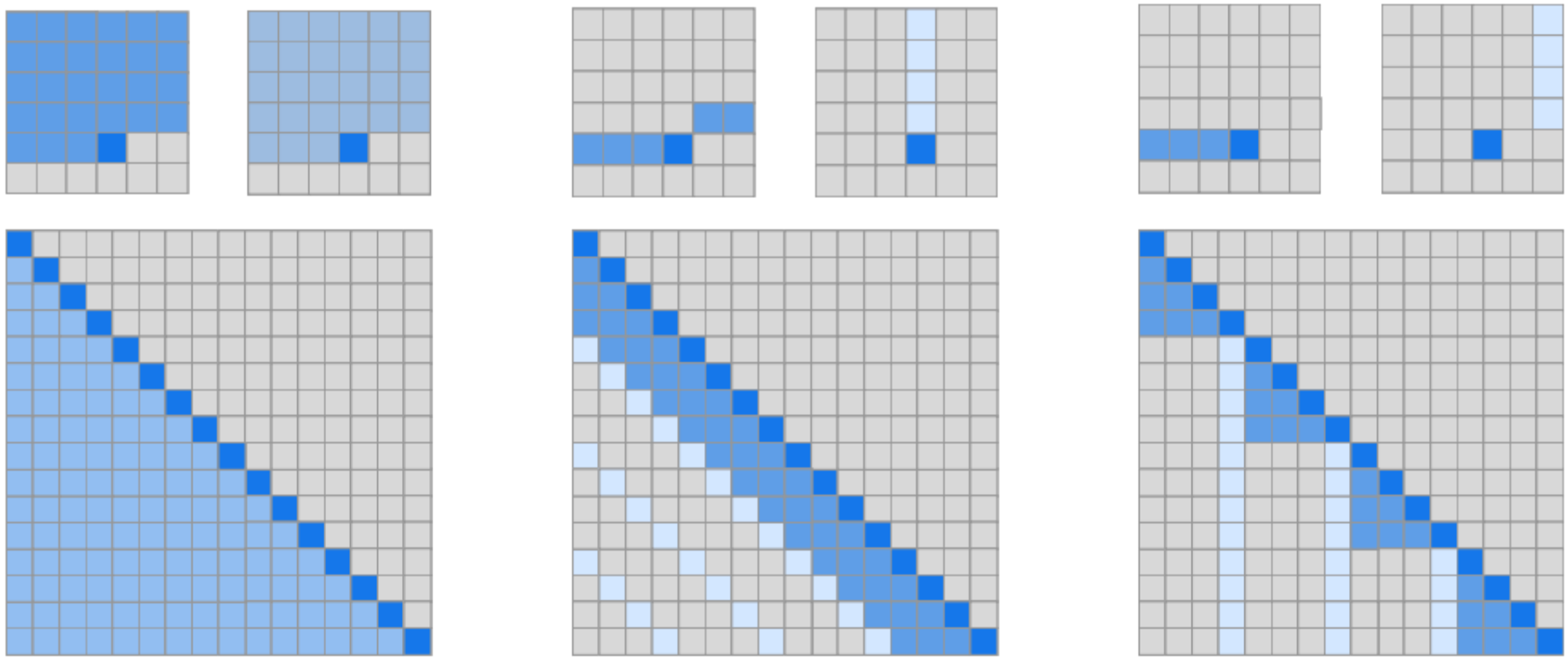
Sliding window attention의 아이디어를 일반적으로 실험한 논문
논문이 매우 짧고 간결하다. 장황한 말 대신 수식과 예제로 설명한다.
이해하는데 시간이 좀 걸린다. 성의 없게 그린 그림을 이해해야 한다.

얘는 OpenAI 논문이고, 비전 모델까지 모두 아우르는 연구라서 그런지 인용 수가 많다.
(1680+ 회)

논문 2: Generating Long Sequences with Sparse Transformers



논문 2: Generating Long Sequences with Sparse Transformers



(a) Transformer

(b) Sparse Transformer (strided)

(c) Sparse Transformer (fixed)

그러나... SWA 없이 힘으로 밀어붙이는 추세

- Mistral NeMo: 128K, No sliding window
- Command R+: 128K, No sliding window
- Llama 3.1: 128K, No sliding window

대신 H100 80G가 10000장 이상 들었다는 것!

Long sequence 처리와 관련있는 기술들

- RoPE (Rotary Position Embedding)
- GQA (Grouped Query Attention)
  - KV-cache가 왜 필요하고, 어떨 때 쓰이는지 이해하면 좋음

- Ring Attention
  - Gemini-1.5가 어떻게 1.5M context window를 달성할 수 있었는가?
  - SWA를 제외하면 가장 유력한 방법... 정확하게는 알 수 없음.