Transformer

Efficient Transformers

Learning goals

- Understand the efficiency problems and shortcomings of transformer-based models
- Learn about some strategies to alleviate them

THE $\mathcal{O}(N^2)$ PROBLEM

Quadratic time & memory complexity of Self-Attention

- *Inductive bias of Transformer models:*
  Connect all tokens in a sequence to each other
- **Pro:** Can (theoretically) learn contexts of arbitrary length
- **Con:** Bad scalability limiting (feasible) context size

Resulting Problems:

- Several tasks require models to consume longer sequences
- *Efficiency:* Are there more efficient modifications which achieve similar or even better performance?

EFFICIENT TRANSFORMERS

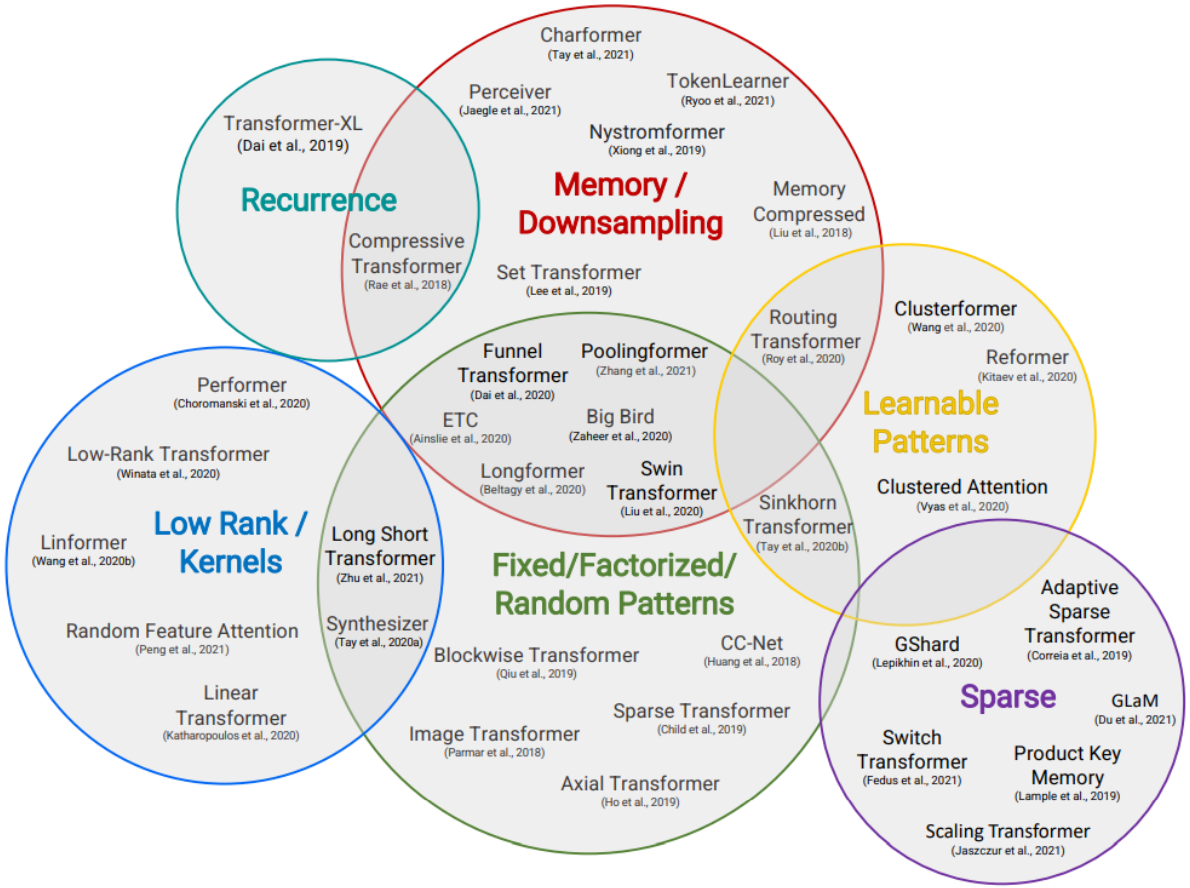
Broad overview on so-called "X-formers" ▸ Tay et al., 2020a

- Efficient & fast Transformer-based models
  $\rightarrow$ Reduce complexity from $\mathcal{O}(n^2)$ to (up to) $\mathcal{O}(n)$
- Claim on-par (or even) superior performance
- Different techniques used:
  - Fixed/Factorized/Random Patterns
  - Learnable Patterns (extension of the above)
  - Low-Rank approximations or Kernels
  - Recurrence (see e.g. ▸ Dai et al., 2019)
  - Memory modules

Side note:

- Most Benchmark data sets not explicitly designed for evaluating long-range abilities of the models.
- Recently proposed: *Longe Range Arena: A benchmark for efficient transformers* ▸ Tay et al., 2020b

EFFICIENT TRANSFORMERS



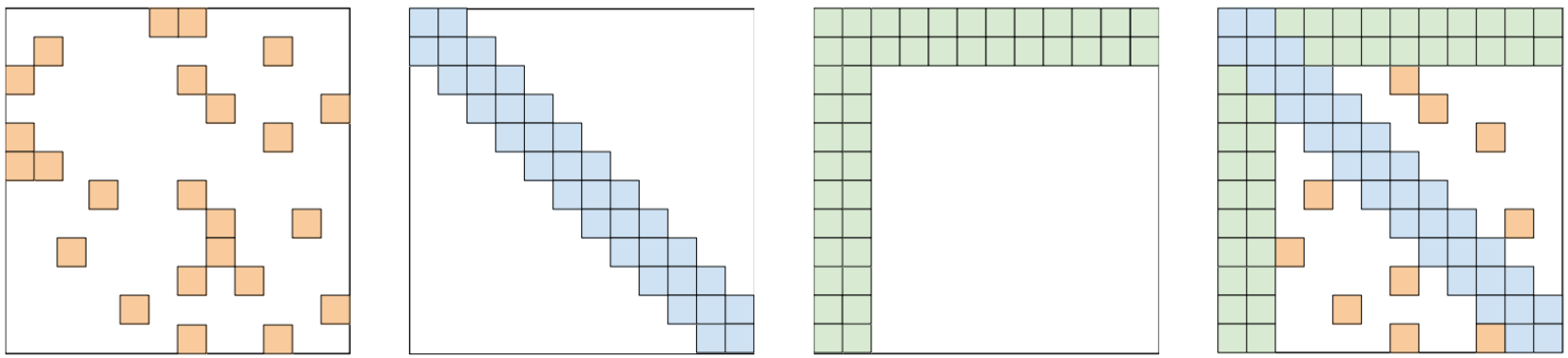
▸ Tay et al., 2020a

EFFICIENT TRANSFORMERS - EXAMPLE

Reasoning:

- Making every token attend to every other token might be unnecessary
- Introduce sparsity in the commonly dense attention matrix

Example:



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Figure 1: Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with $r = 2$, (b) sliding window attention with $w = 3$ (c) global attention with $g = 2$. (d) the combined BIGBIRD model.

▸ Source: Zaheer et al., 2020

EFFICIENT TRANSFORMERS - TRADEOFF

- *Quality:* How well can we approximate the results we would get when applying full attention?

- *Speed:* Can we reduce the computational complexity from quadratic to (nearly) linear?

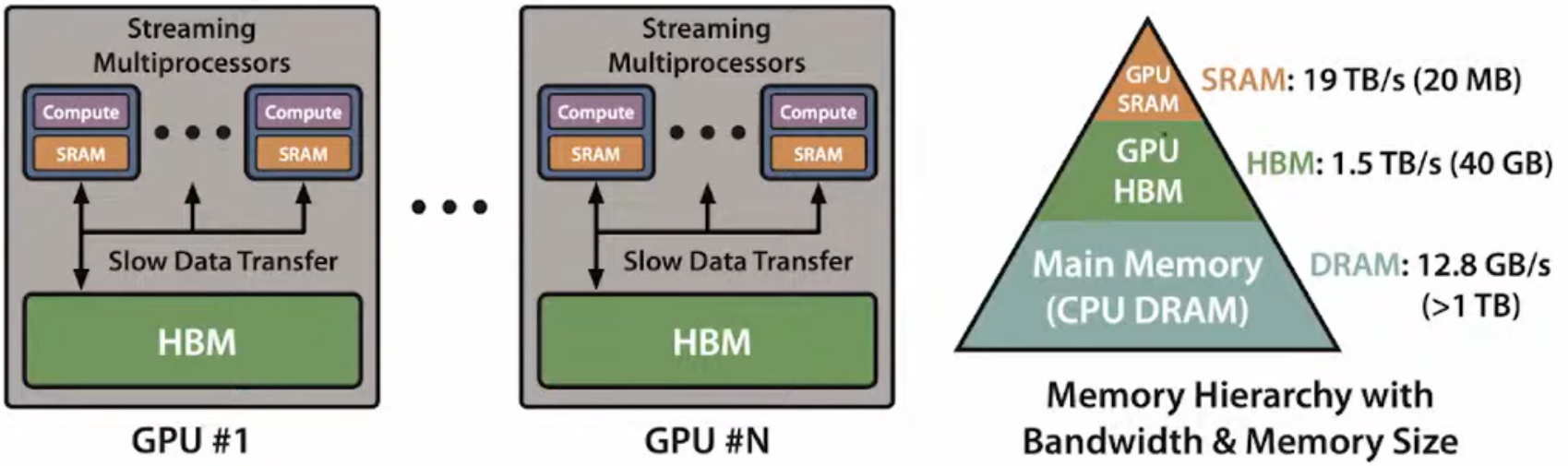
A PRIMER ON FLASH ATTENTION

Compute-bound vs. Memory-bound operations:

- Wall-clock time not (necessarily) a function of FLOPs
- Most of these “Efficient Transformers“ focus on FLOP reduction

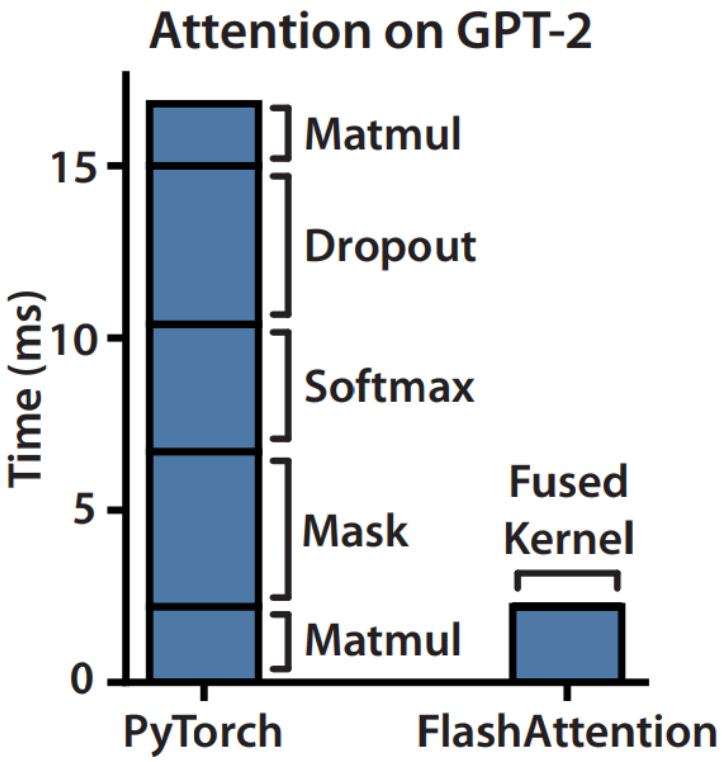
  $\rightarrow$ This ignores existing memory overheads
- *Compute-bound operations:* ($\approx$ FLOPs)
  - Matrix Multiplication
- *Memory-bound operations:* ($\approx$ data transfer)
  - Attention

A PRIMER ON FLASH ATTENTION



▸ Source: Talk by Tri Dao, 2023

A PRIMER ON FLASH ATTENTION



▸ Source: Dao, 2022

- *Compute-bound operations:*
  - Matmul
- *Memory-bound operations:*
  - Mask
  - Dropout
  - Softmax

$\rightarrow$ Reducing memory-bound operations most effective for reducing wall-clock time

TBD