Sparse Attention for Long Documents

Group meeting update

Your Name

Your Lab, Your University

November 13, 2026

Outline

1. Motivation
2. Method
3. Results

Why sparse attention?

- Full attention costs $\mathcal{O}(n^2)$ time and memory in the sequence length n.
- Most attention weight lands on a few keys per query.

Why sparse attention?

- Full attention costs $\mathcal{O}(n^2)$ time and memory in the sequence length n.
- Most attention weight lands on a few keys per query.
- Idea: keep only the top-k keys for each query.

Why sparse attention?

- Full attention costs $\mathcal{O}(n^2)$ time and memory in the sequence length n.
- Most attention weight lands on a few keys per query.
- Idea: keep only the top-k keys for each query.

Goal: same accuracy at a fraction of the cost.

Top-k routing

For a query q and keys K, keep the k largest scores:

$$\mathrm{Attn}(q, K, V) = \sum_{j \in \mathrm{top}_k(qK^\top)} \frac{e^{qk_j^\top}}{\sum_{l \in \mathrm{top}_k} e^{qk_l^\top}} v_j$$

Cost

$\mathcal{O}(nk)$ instead of $\mathcal{O}(n^2)$, with $k \ll n$.

Two columns: setup and intuition

Setup

- 4 benchmark tasks
- Sequences up to 16k tokens
- $k \in \{16, 32, 64\}$

Intuition

Observation

Attention is already sparse after the first few layers.

Results

Method	Accuracy (%)	Speed-up
Full attention	81.2	1.0×
Top-k, $k = 32$	80.9	3.4×
Top-k, $k = 16$	79.8	5.1×

Top-k with $k = 32$ keeps accuracy within 0.3 points.

Takeaways

1. Top-k routing cuts attention cost by 3–5$\times$.
2. Accuracy drops less than one point at $k = 32$.
3. Next: learn k per layer.

Questions?