Recent Efficiency Improvements to Transformers

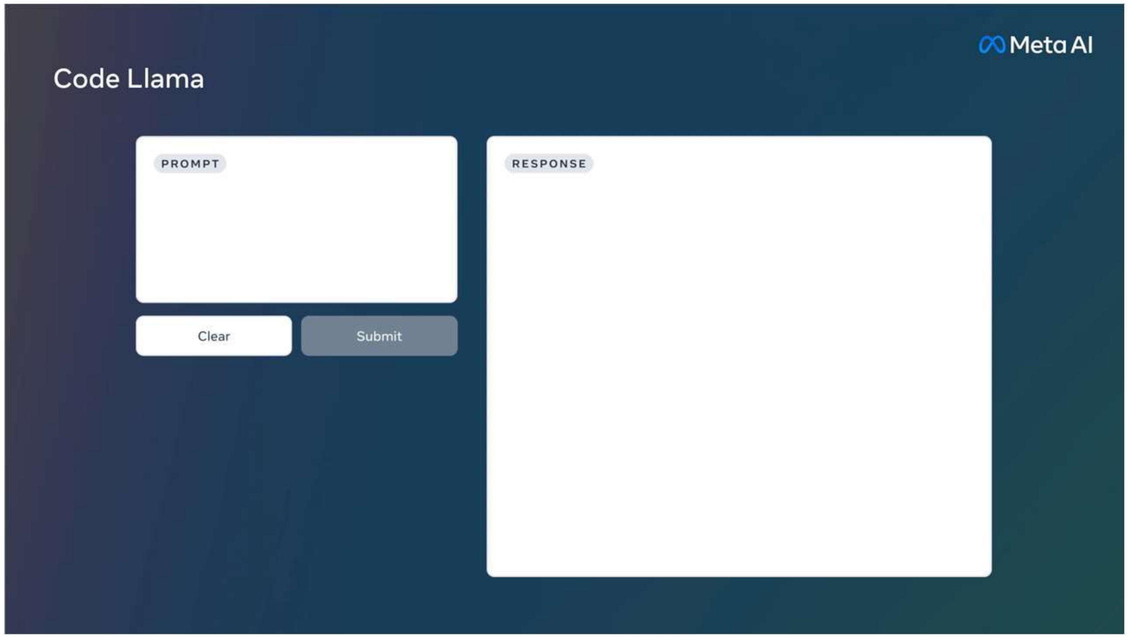
David Woodruff

Carnegie Mellon University / Google

Outline

1. Background on Transformers and time complexity
2. HyperAttention
3. PolySketchFormer
4. Conclusions and Recent Work

AI Revolution



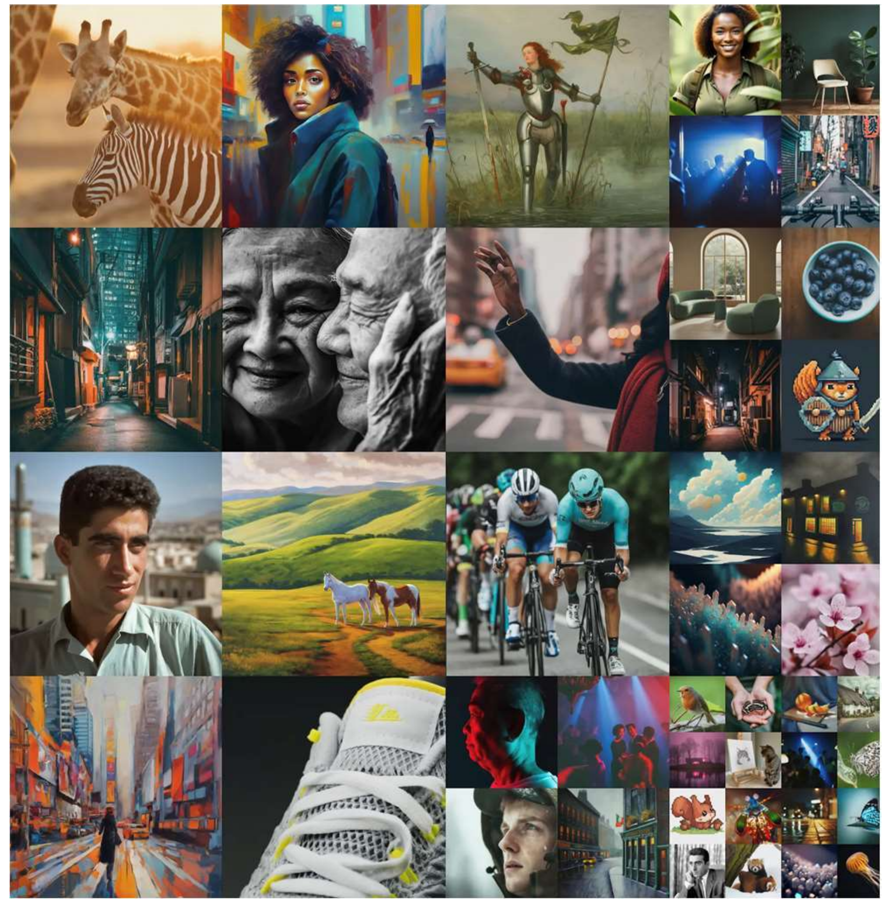
2023 IN REVIEW

THE YEAR A.I. ATE THE INTERNET

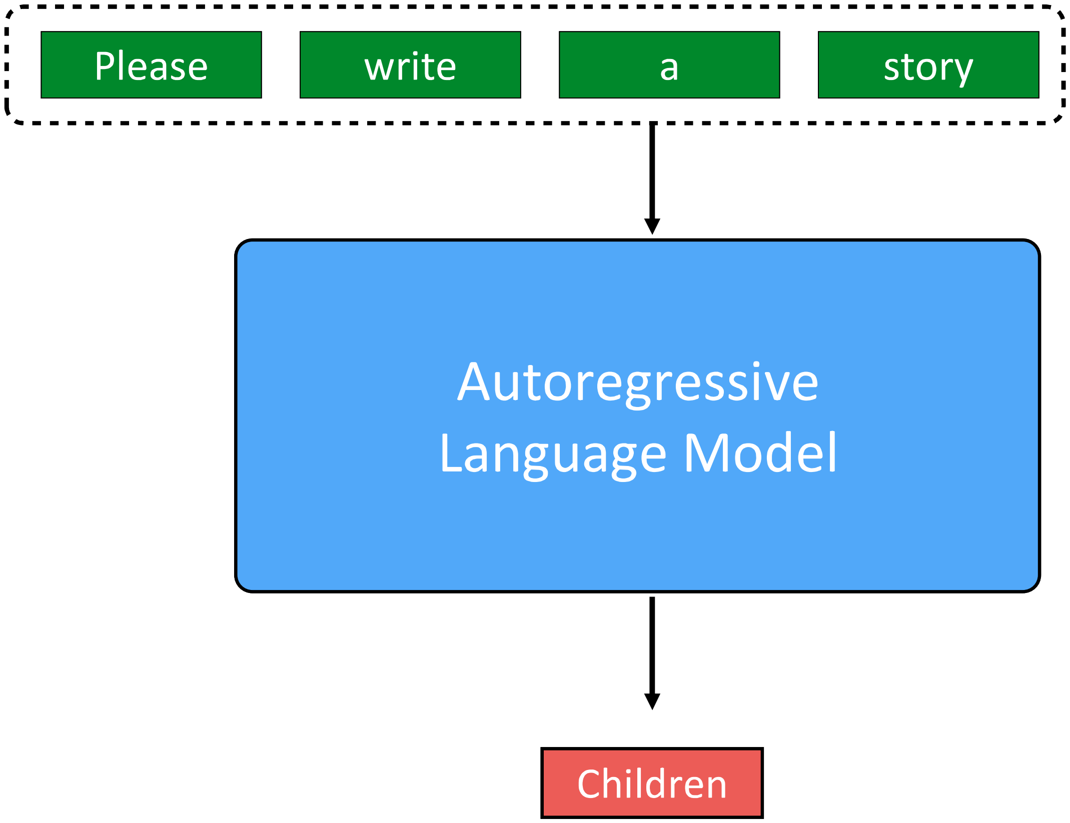
Call 2023 the year many of us learned to communicate, create, cheat, and collaborate with robots.

By Sue Halpern

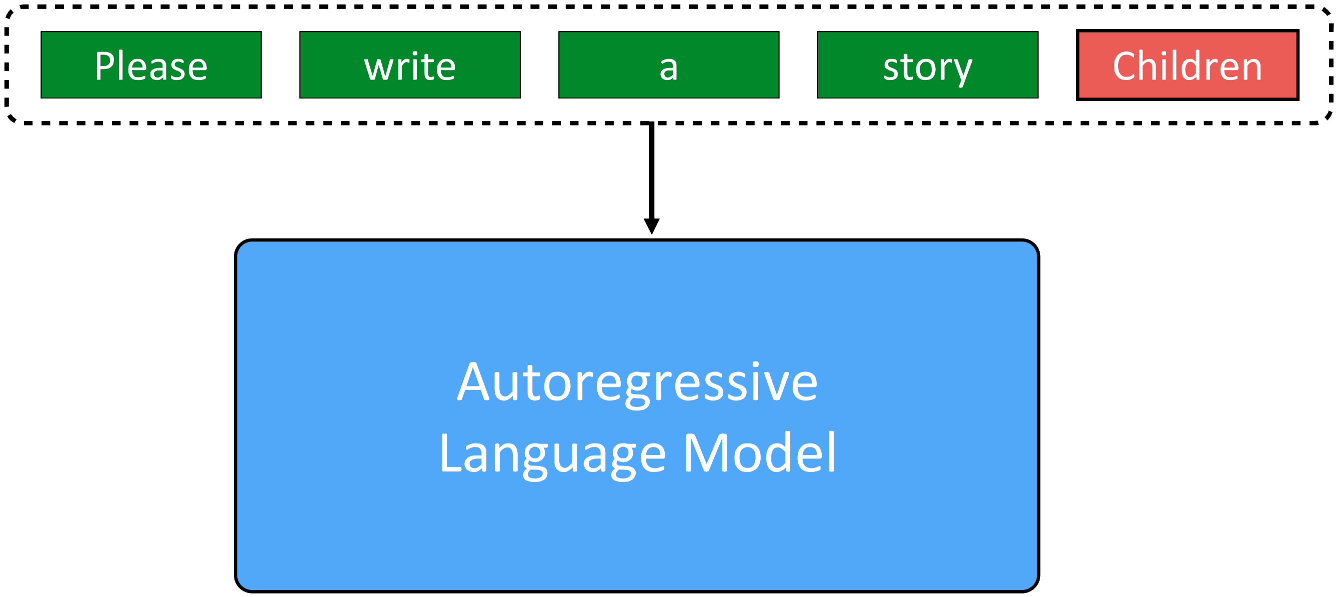
December 8, 2023



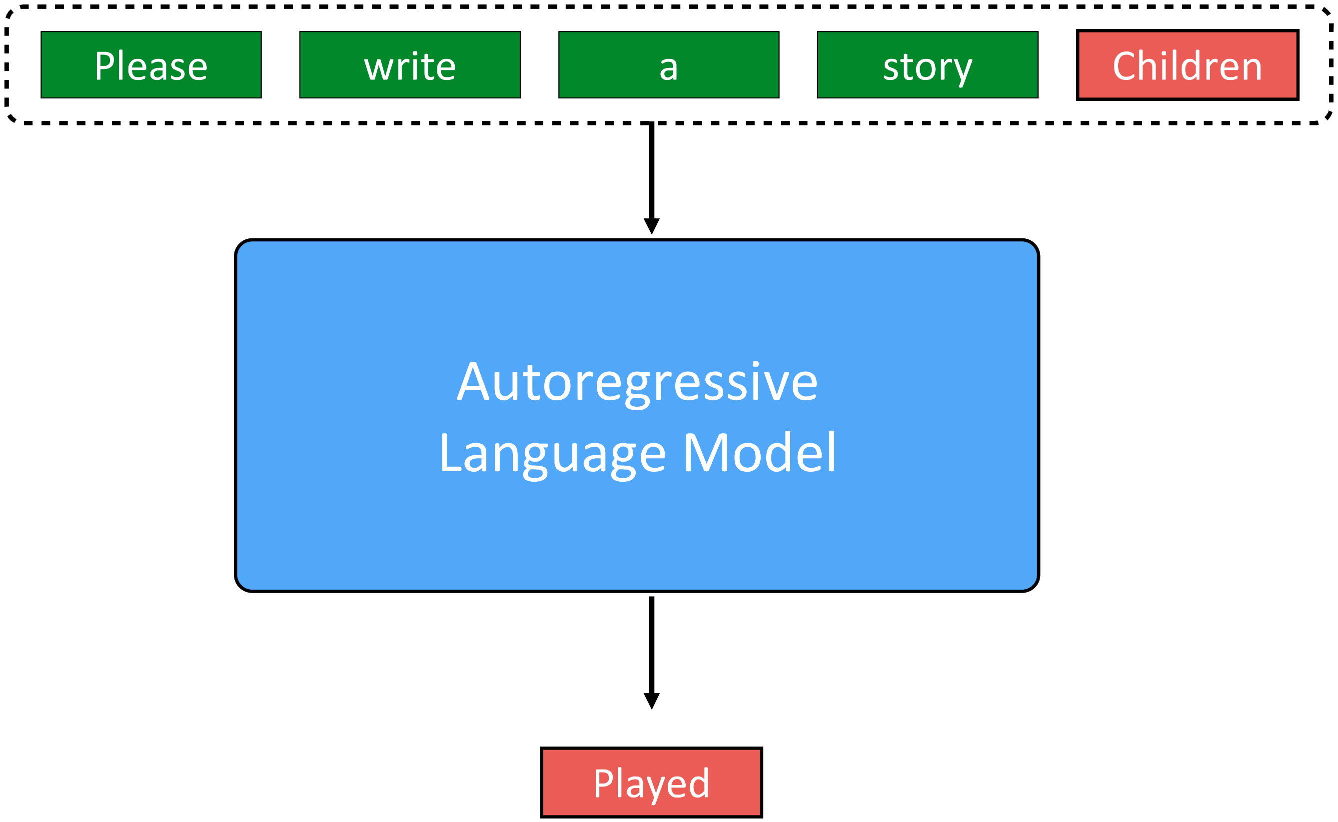
Autoregressive Models



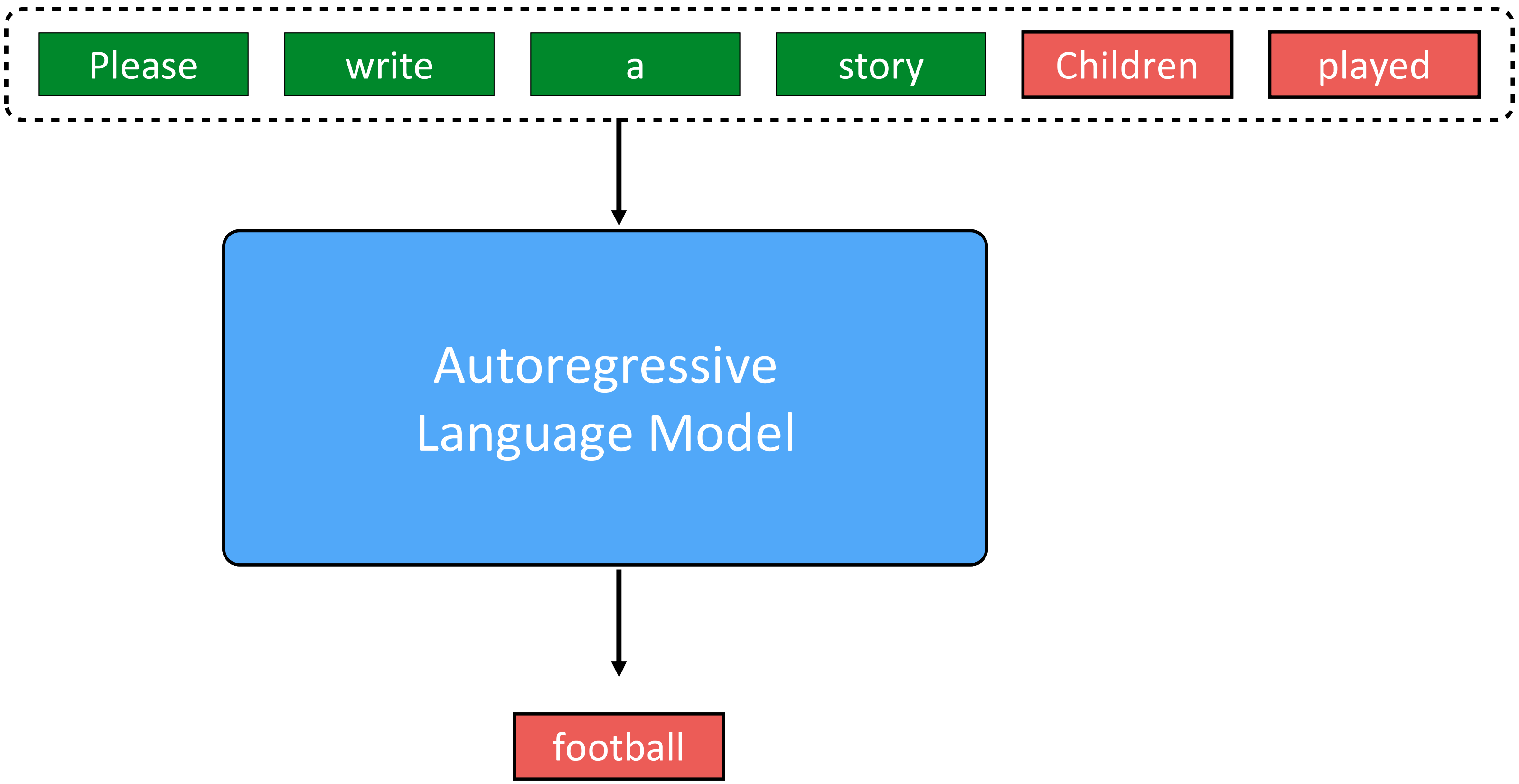
Autoregressive Models



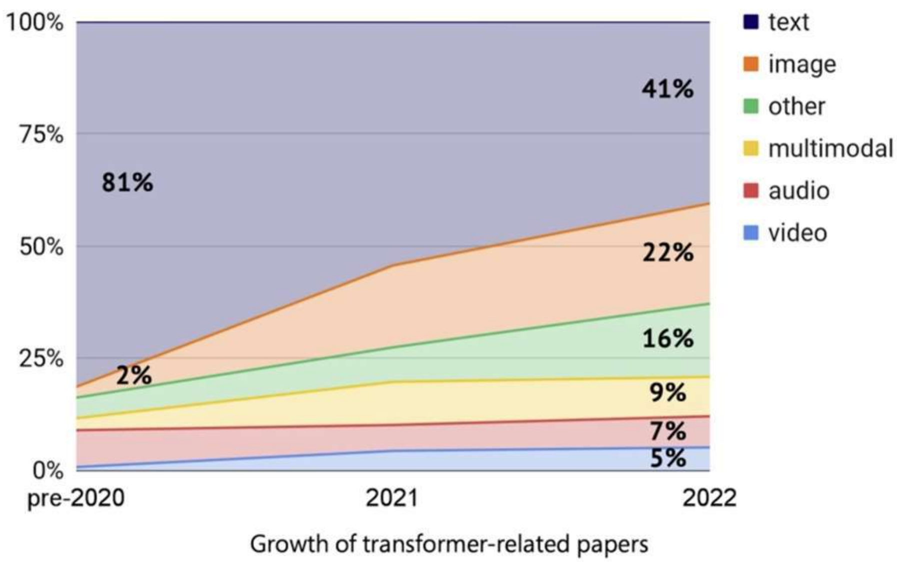
Autoregressive Models



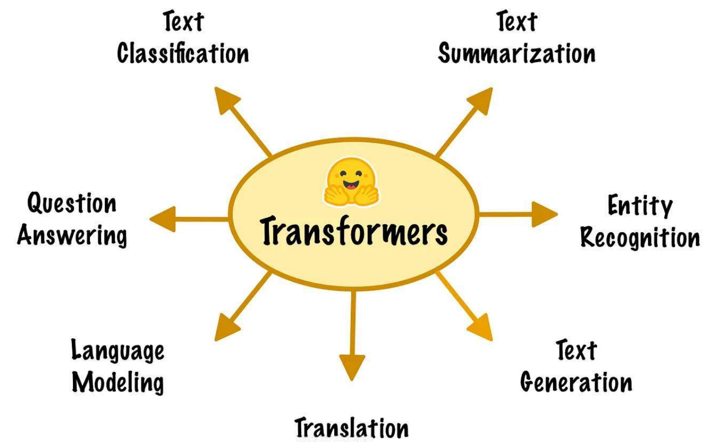
Autoregressive Models

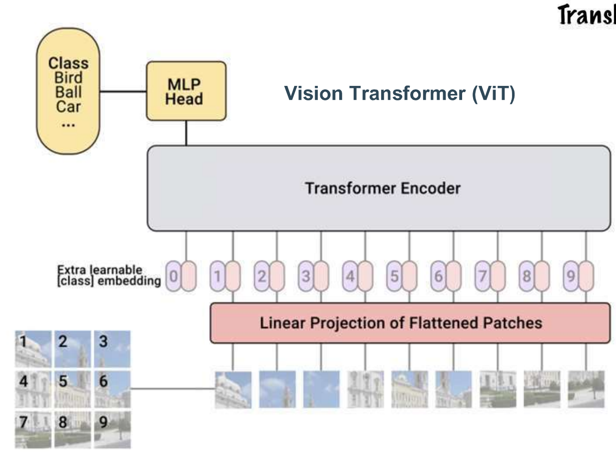


Transformers

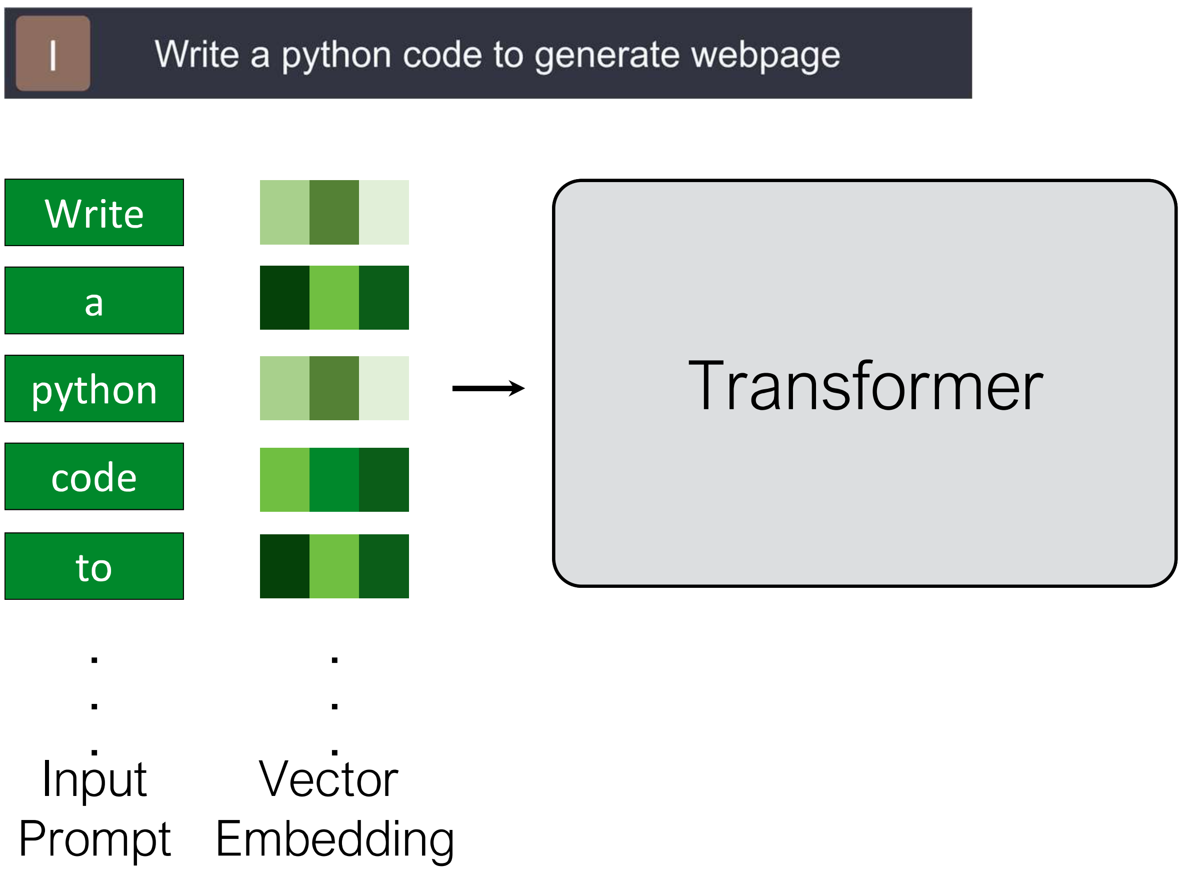


Transformer (Vaswani et al., 17')

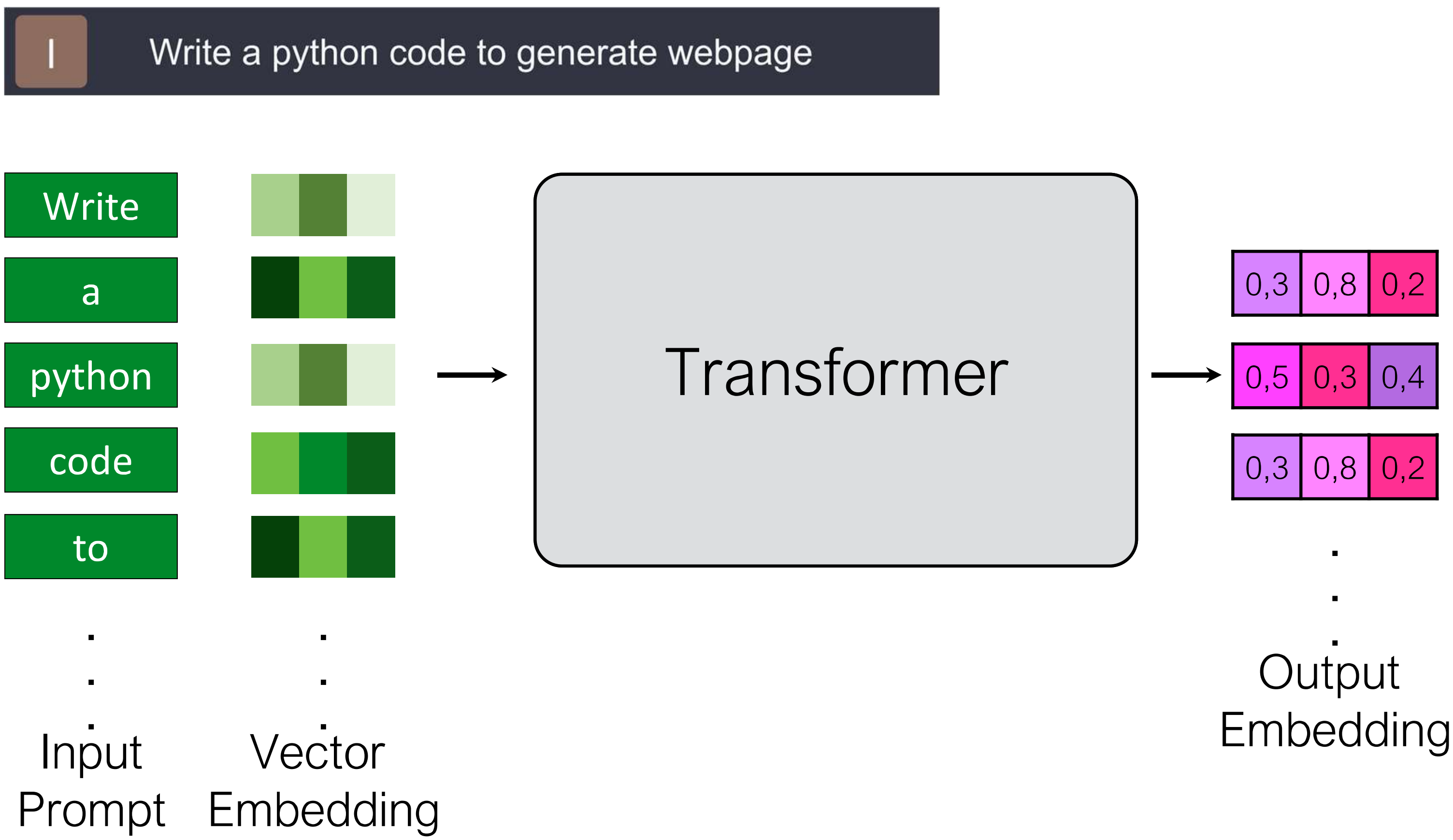




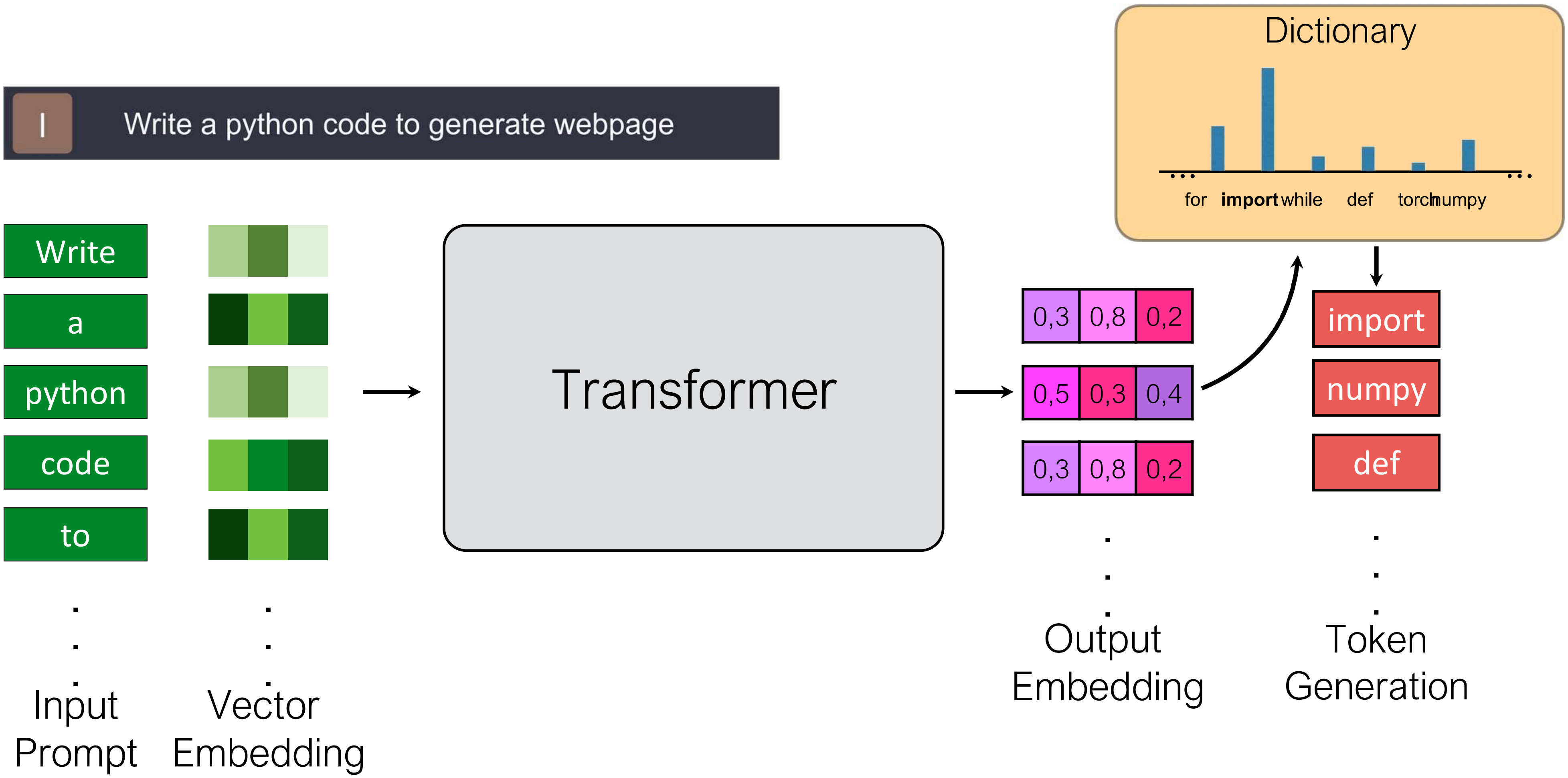
Transformers



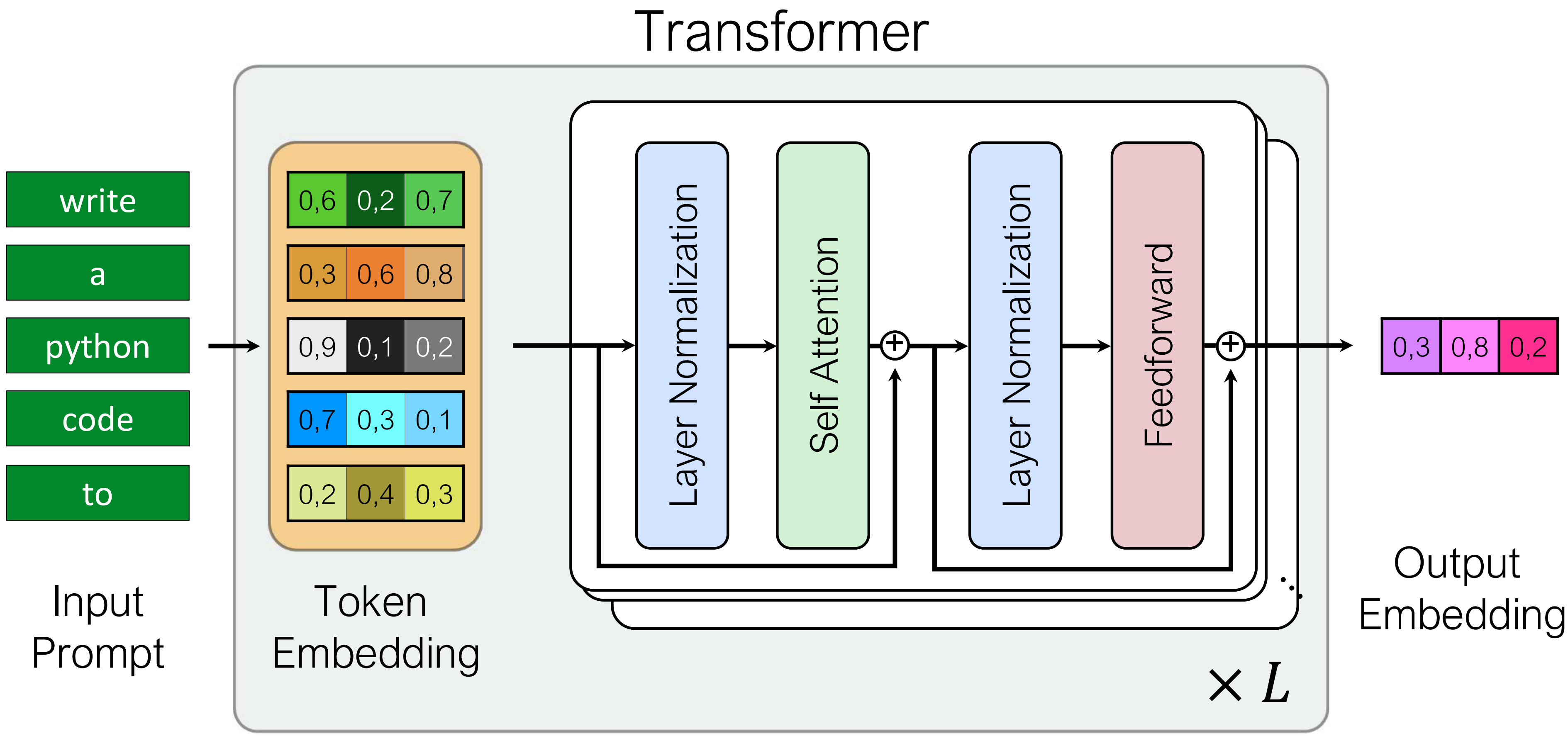
Transformers



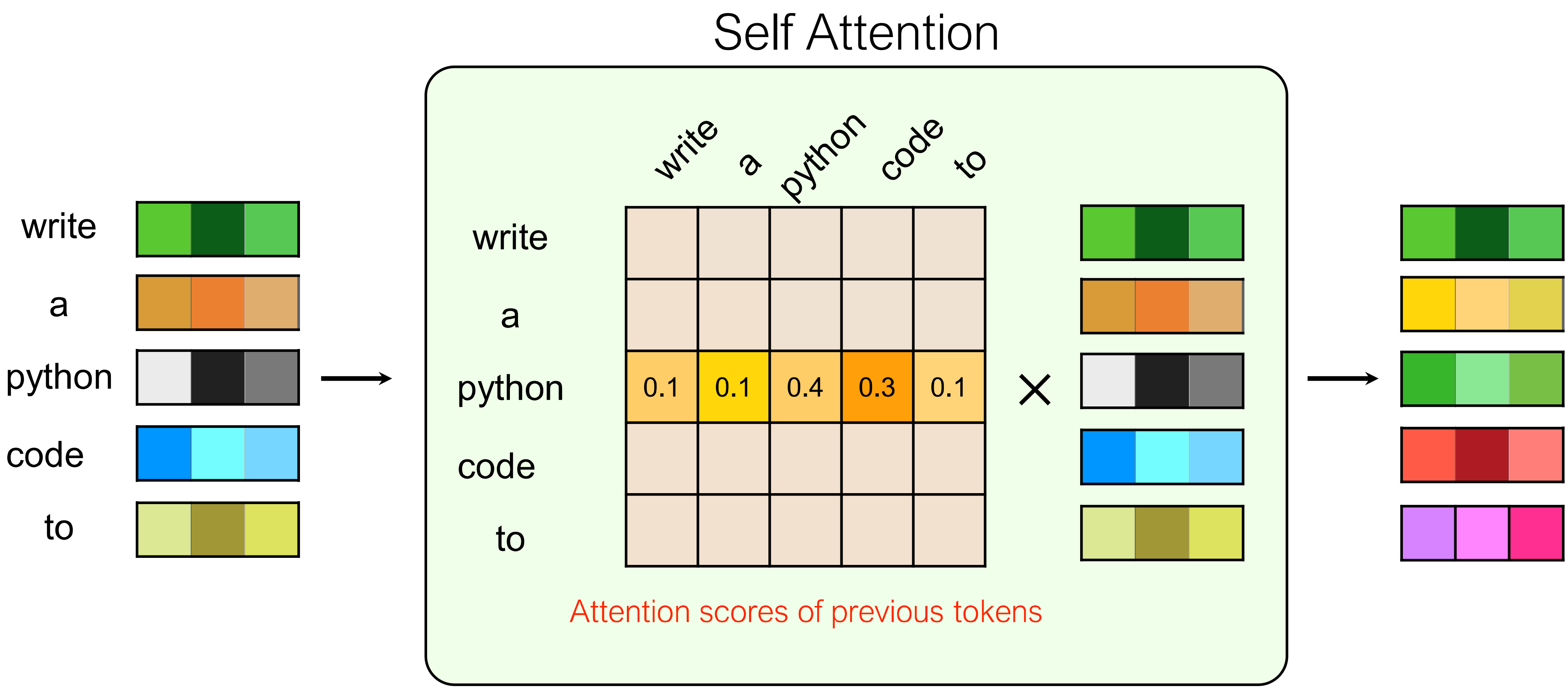
Transformers



Transformers

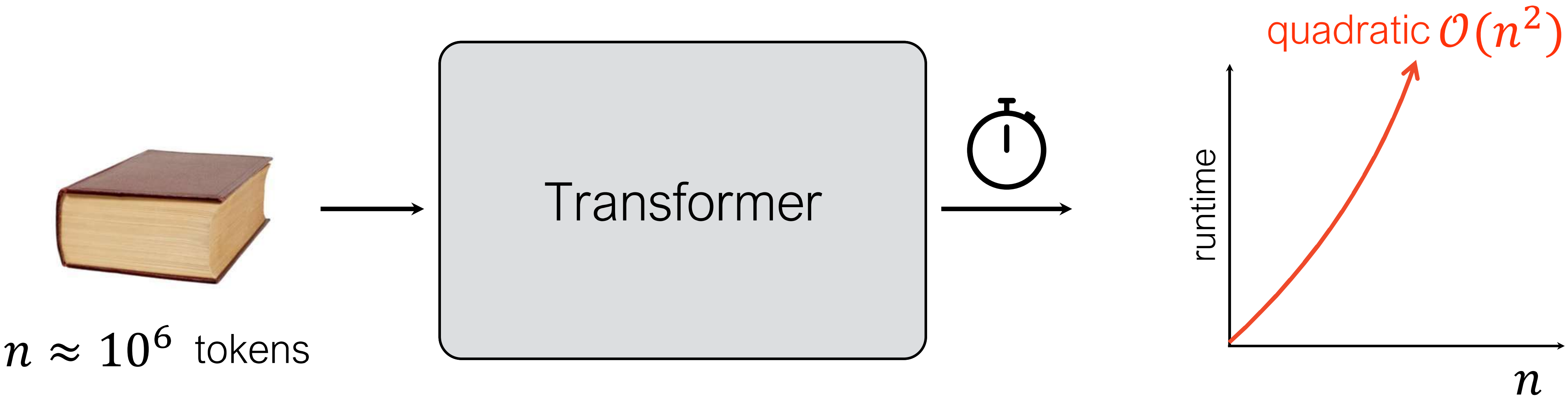


Transformers

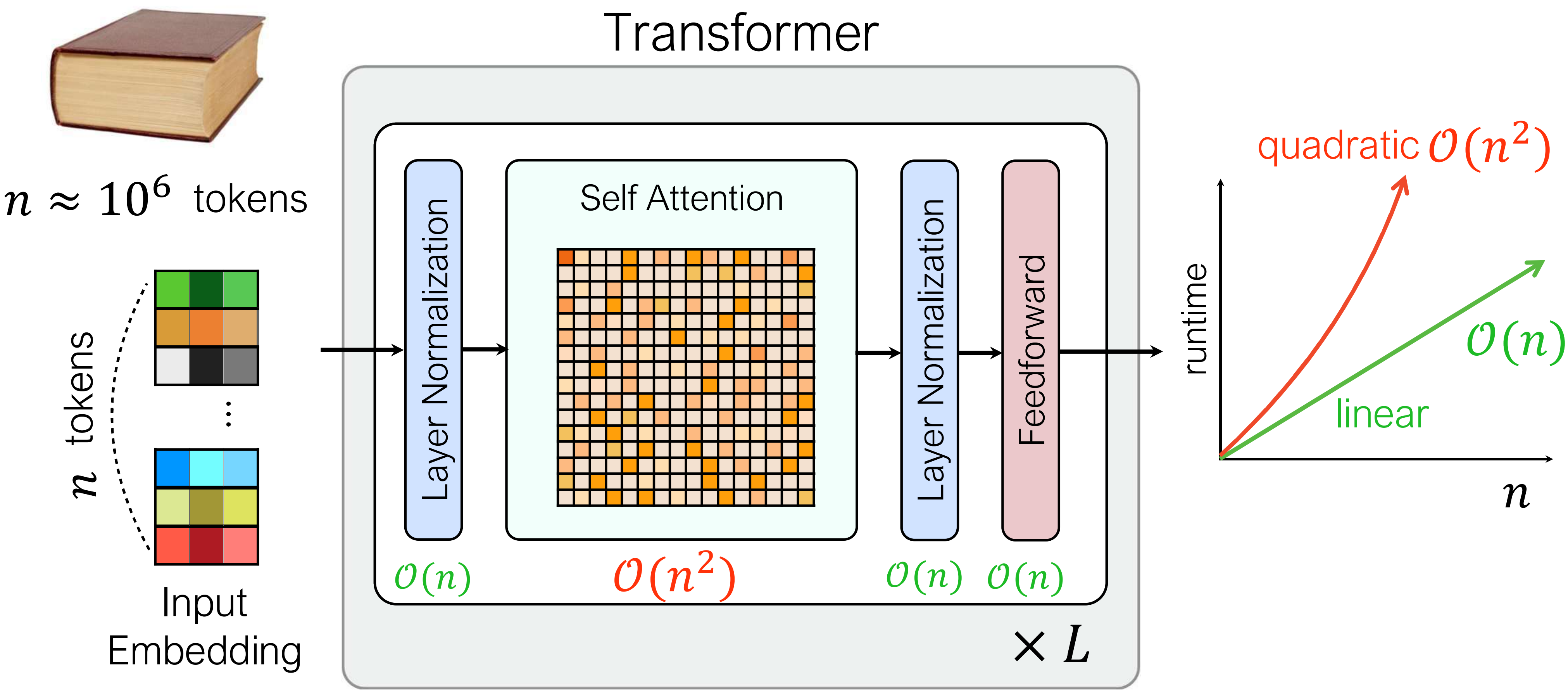


Algorithmic Acceleration

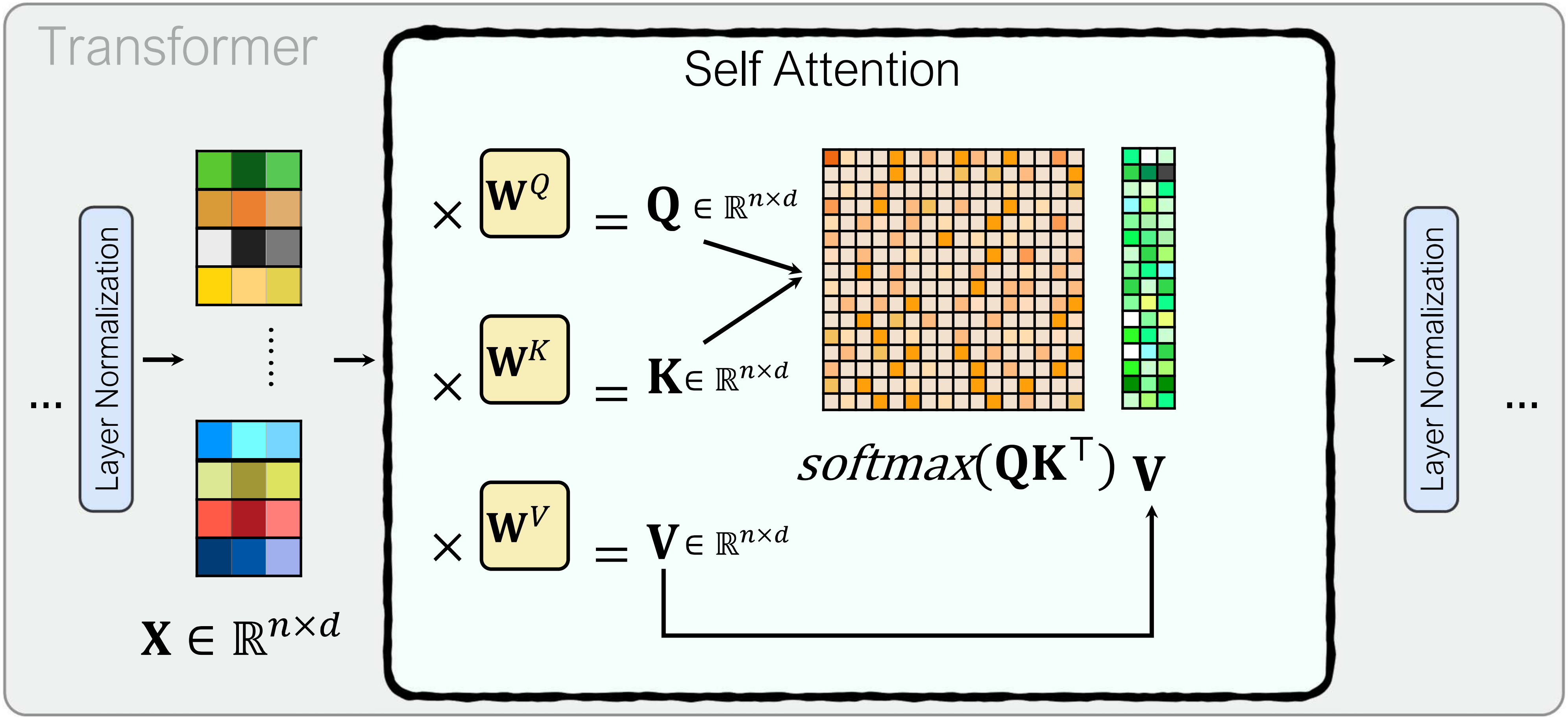
Algorithmic Acceleration



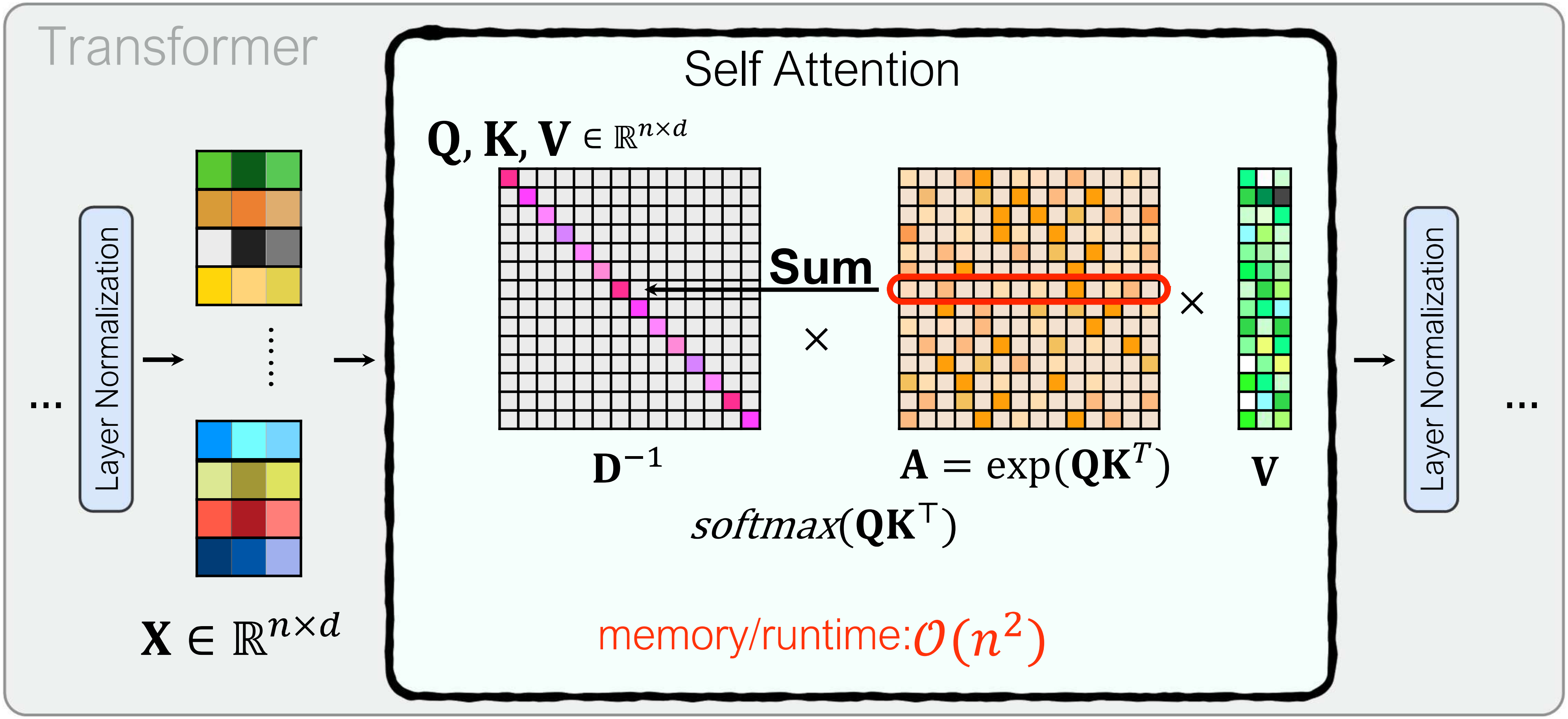
Algorithmic Acceleration



Algorithmic Acceleration

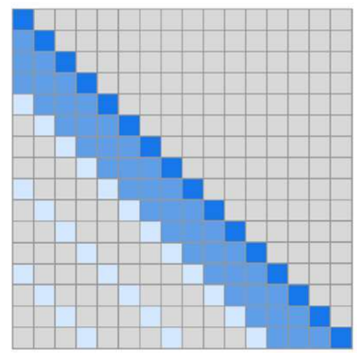


Algorithmic Acceleration

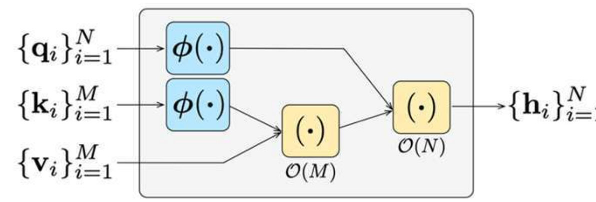


Previous Work

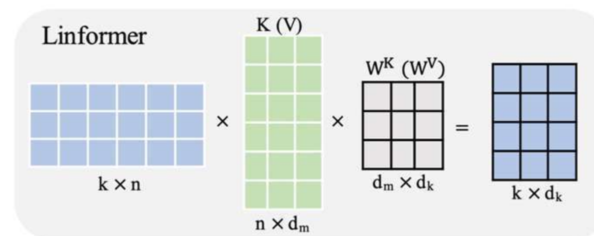
- **Sparse Structure**
  - Local Attention (Parmar et al., 18')
  - Sparse Transformer (Child et al., 19')
  - Longformer (Beltagy et al., 20')
  - Reformer (Kitaev et al., 20')
  - Sinkhorn Attention (Tay et al., 20')
- **Kernel Methods**
  - Lambda network (Bello et al., 21')
  - Performer (Choromanski et al., 21')
  - Random Feature Attention (Peng et al., 21')
  - Randomized Attention (Zheng et al., 22')
- **Low-rank Approximation**
  - Linformer (Wang et al., 20')
  - Nystromformer (Xiong et al., 21')
  - Nested Attention (Max et al., 21')

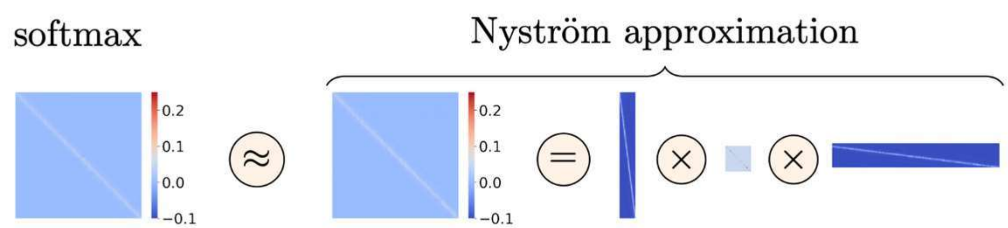


(b) Sparse Transformer (strided)



(b) Random feature attention.





Previous Work

- **Sparse Structure**
  - Local Attention (Parmar et al., 18')
  - Sparse Transformer (Child et al., 19')
  - Longformer (Beltagy et al., 20')
  - Reformer (Kitaev et al., 20')
  - Sinkhorn Attention (Tay et al., 20')

- **Kernel Methods**
  - Lambda network (Bello et al., 21')
  - Performer (Choromanski et al., 21')
  - Random Feature Attention (Peng et al., 21')
  - Randomized Attention (Zheng et al., 22')

- **Low-rank Approximation**
  - Linformer (Wang et al., 20')
  - Nystromformer (Xiong et al., 21')
  - Nested Attention (Max et al., 21')

(Alman & Song 23') High quality $(1/poly(n))$ entrywise approximation of $Att(Q, K, V)$ requires nearly quadratic time assuming SETH

Previous Works

- **Sparse Structure**
  - Local Attention (Parmar et al., 18')
  - Sparse Transformer (Child et al., 19')
  - Longformer (Beltagy et al., 20')
  - Reformer (Kitaev et al., 20')
  - Sinkhorn Attention (Tay et al., 20')

- **Kernel Methods**
  - Lambda network (Bello et al., 21')
  - Performer (Choromanski et al., 21')
  - Random Feature Attention (Peng et al., 21')
  - Randomized Attention (Zheng et al., 22')

- **Low-rank Approximation**
  - Linformer (Wang et al., 20')
  - Nystromformer (Xiong et al., 21')
  - Nested Attention (Max et al., 21')

No End-to-End approximation (in some works)

- Only approximate matrix $A = \exp(QK^T)$

Would like to:

- Compute $\widetilde{Att} \in \mathbb{R}^{n \times d}$ such that
- $\| \widetilde{Att} - Att(Q, K, V) \|_{op}$ is small

These methods do not support causal masking

(Alman & Song 23') High quality $(1/poly(n))$ entrywise approximation of $Att(Q, K, V)$ is likely impossible in general

HyperAttention

Insu Han (Adobe), Rajesh Jayaram (Google), Amin Karbasi (Yale),

Vahab Mirrokni (Google), David Woodruff (CMU), Amir Zandieh

Algorithmic Acceleration

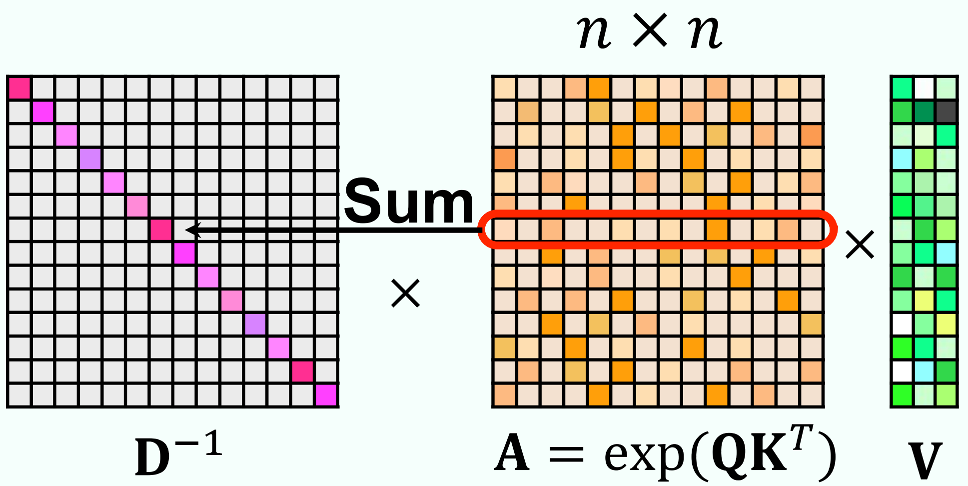
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$

Self Attention

1. Approximate

$$D_{i,i} = \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$$

2. Approximate matrix product $A \cdot V$



$softmax(\mathbf{QK}^\top)$

memory/runtime: $\mathcal{O}(n^2)$

Algorithmic Acceleration

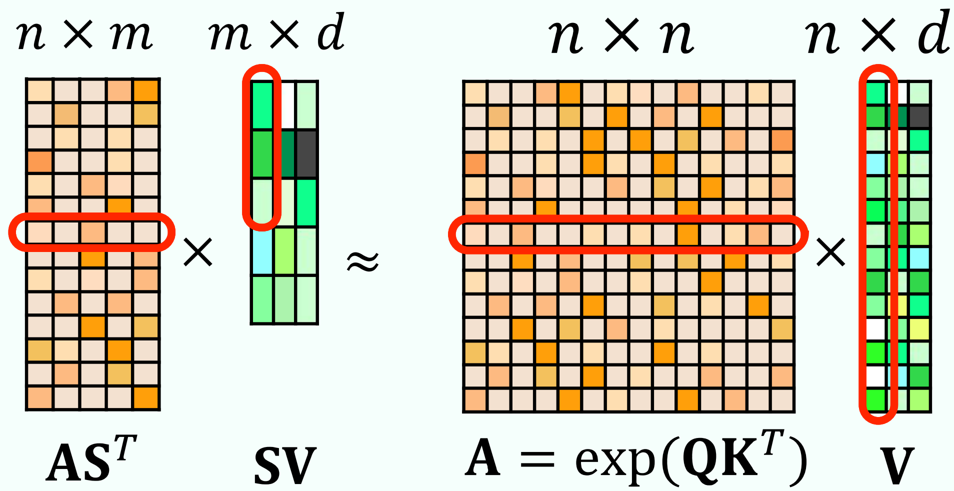
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$

Self Attention

1. Approximate

$$D_{i,i} = \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$$

2. Compute a row sampling sketch S $\in \mathbb{R}^{m \times n}$ where row i is sampled with probability $\propto \| v_i \|_2^2$ $\rightarrow m$ $\approx srank(softmax(QK^\top)) \cdot d$



Algorithmic Acceleration

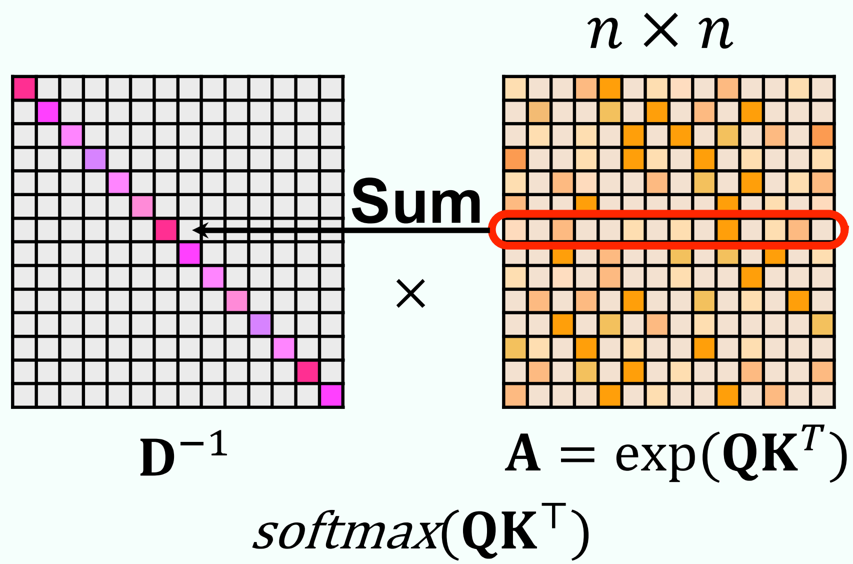
Self Attention

$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$

1. Approximate

$$\tilde{D}_{i,i} \approx \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$$

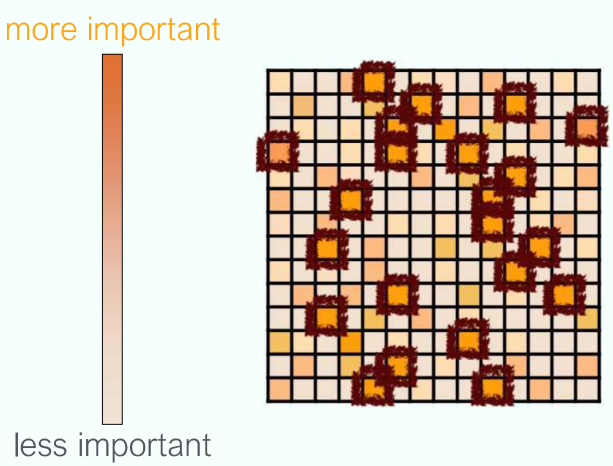
2. Compute a row sampling sketch $S \in \mathbb{R}^{m \times n}$ where row i is sampled with probability $\propto \| v_i \|_2^2 \rightarrow m \approx srank(softmax(QK^\top)) \cdot d$



Algorithmic Acceleration

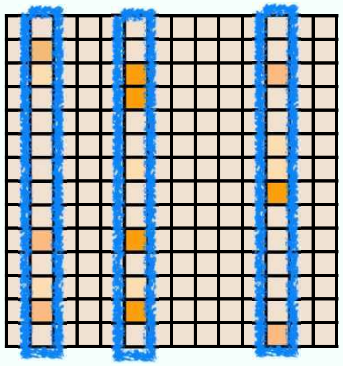
Approximate $D_{i,i} = \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$

Find 'Heavy' elements of $\mathbf{A} = \exp(\mathbf{QK}^T)$



$+$

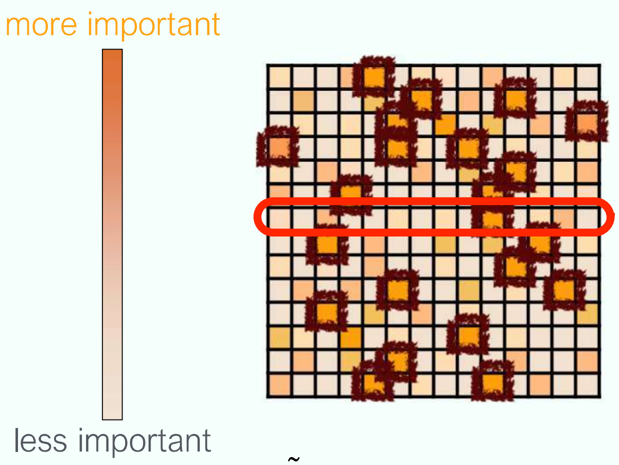
Estimate 'Light' elements of $\mathbf{A}$ via uniform column sampling



Algorithmic Acceleration

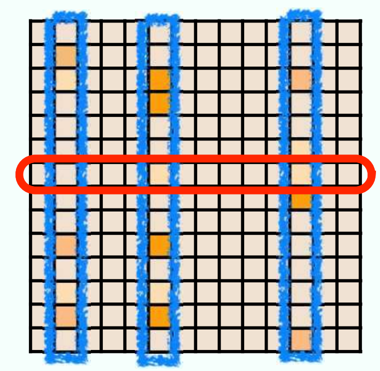
Approximate $D_{i,i} = \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$

Find ‘Heavy’ elements of $\mathbf{A} = \exp(\mathbf{Q}\mathbf{K}^T)$



$+$

Estimate ‘Light’ elements of $\mathbf{A}$ via uniform column sampling



$\tilde{D}_{i,i}$ =contribution of heavy elements + contribution of light elements

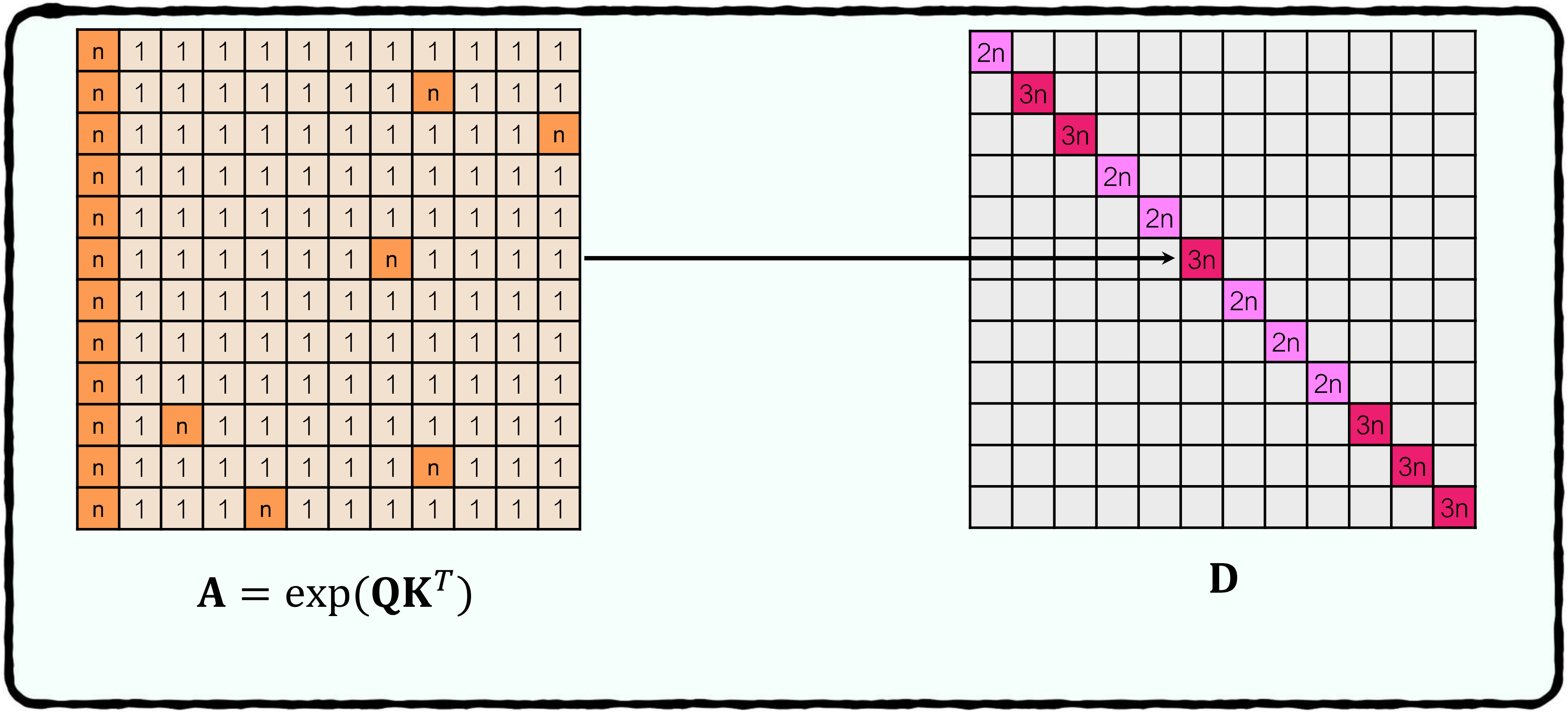
Algorithmic Acceleration

Theorem (informal). If the maximum squared column norm in $\mathit{softmax}(\mathbf{QK}^\top)$ is $\frac{1}{n^{1-o(1)}}$ and the ratio of max and min row sums in $A = \mathit{exp}(\mathbf{QK}^\top)$ after removing heavy elements is $n^{o(1)}$,then $\tilde{Att}$ can be computed in $O(dn^{1+o(1)})$ time with:

$$\| \mathit{softmax}(\mathbf{QK}^\top)\mathbf{V} - \tilde{Att} \|_{op} \leq \varepsilon \| \mathit{softmax}(\mathbf{QK}^\top) \|_{op} \| \mathbf{V} \|_{op}$$

- Column norm bound non-trivial – allows for entries as large as $\frac{1}{n^{\frac{1}{2}-o(1)}}$ in $\mathrm{softmax}(\mathrm{QK}^\top)$
- Estimating the contribution of light elements is non-trivial
- Tested assumption of squared column norms in first attention layer of T2T-ViT on ImageNet
- For chatglm2-6b-32k and LongBeach, only the lexicographically first few columns had large norm

Algorithmic Acceleration



Algorithmic Acceleration

$$
\begin{bmatrix}
1/2 & & & & & & & & & & & \\
1/3 & & & & & & & & 1/3 & & & \\
1/3 & & & & & & & & & & & 1/3 \\
1/2 & & & & & & & & & & & \\
1/2 & & & & & & & & & & & \\
1/3 & & & & & & & 1/3 & & & & \\
1/2 & & & & & & & & & & & \\
1/2 & & & & & & & & & & & \\
1/2 & & & & & & & & & & & \\
1/3 & & 1/3 & & & & & & & & & \\
1/3 & & & & & & & & 1/3 & & & \\
1/3 & & & & 1/3 & & & & & & &
\end{bmatrix}
\times
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
=
\begin{bmatrix} 1/2 \\ 1/3 \\ 1/3 \\ 1/2 \\ 1/2 \\ 1/3 \\ 1/2 \\ 1/2 \\ 1/2 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}
$$

$softmax(\mathbf{QK}^\top) = \mathbf{D}^{-1}\mathbf{A}$ $\quad\quad$ $\mathbf{V}$

- $\| softmax(\mathbf{QK}^\top) \|_{op}^{2} \approx n$
- $\| \mathbf{V} \|_{op}^{2} = 1$

$$\| softmax(\mathbf{QK}^\top)\mathbf{V} - \widetilde{Attn} \|_{op}^{2} \leq n/10$$

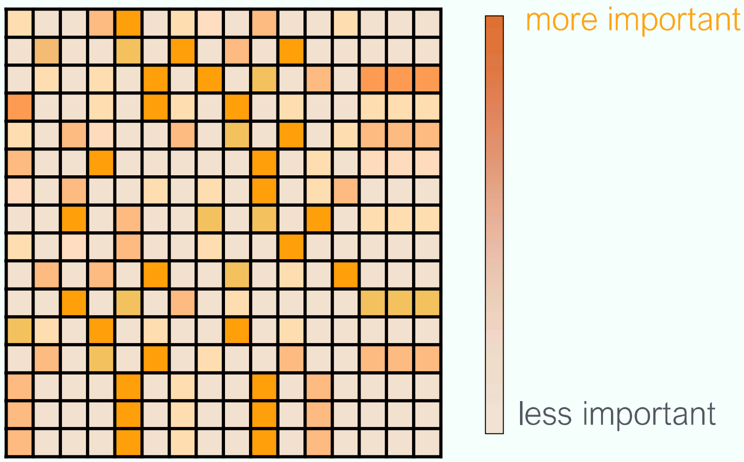
Hardness: $n^{2-o(1)}$

Bounded column norms in $softmax(\mathbf{QK}^\top)$ avoids this hard instance!

Algorithmic Acceleration

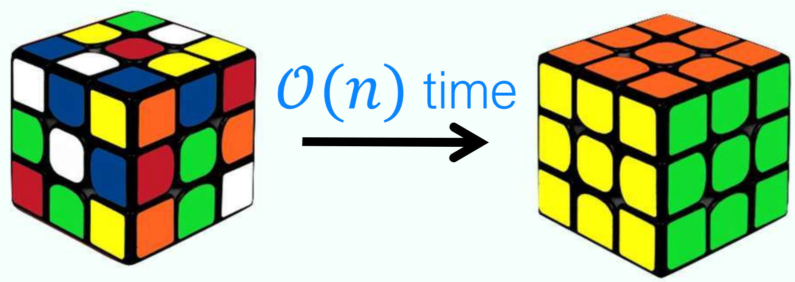
Finding Heavy contributions in practice

$n \times n$



$\mathbf{A} = \exp(\mathbf{QK}^T)$

memory/runtime: $\mathcal{O}(n^2)$

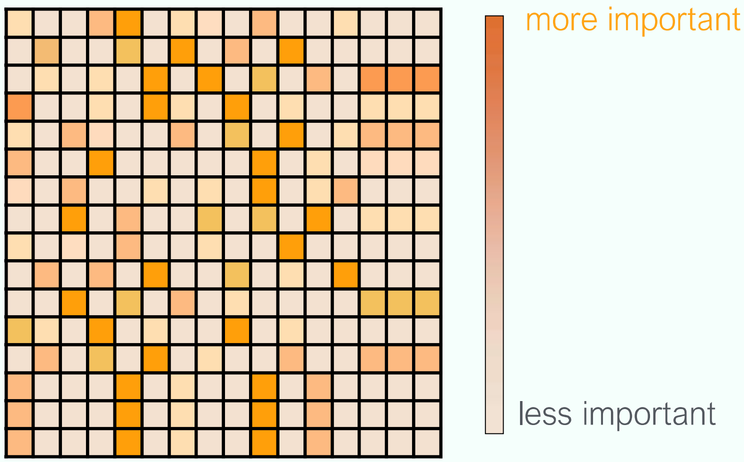


A GPU-friendly algorithm to compute heavy entries and minimize I/O

Algorithmic Acceleration

Finding Heavy contributions in practice

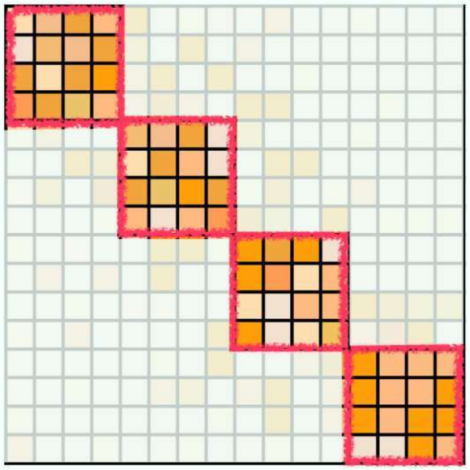
$n \times n$



$\mathbf{A} = \exp(\mathbf{Q}\mathbf{K}^T)$

memory/runtime: $\mathcal{O}(n^2)$

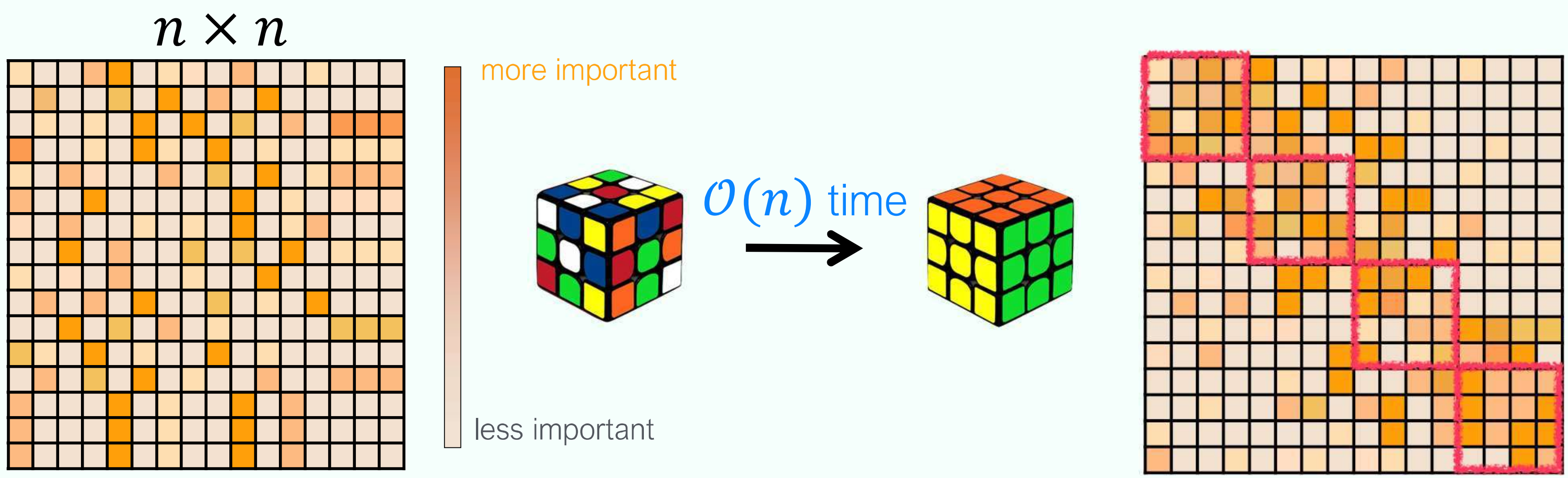
$\approx$



A Permutation algorithm that gathers heavy entries around the diagonal

Algorithmic Acceleration

Self Attention



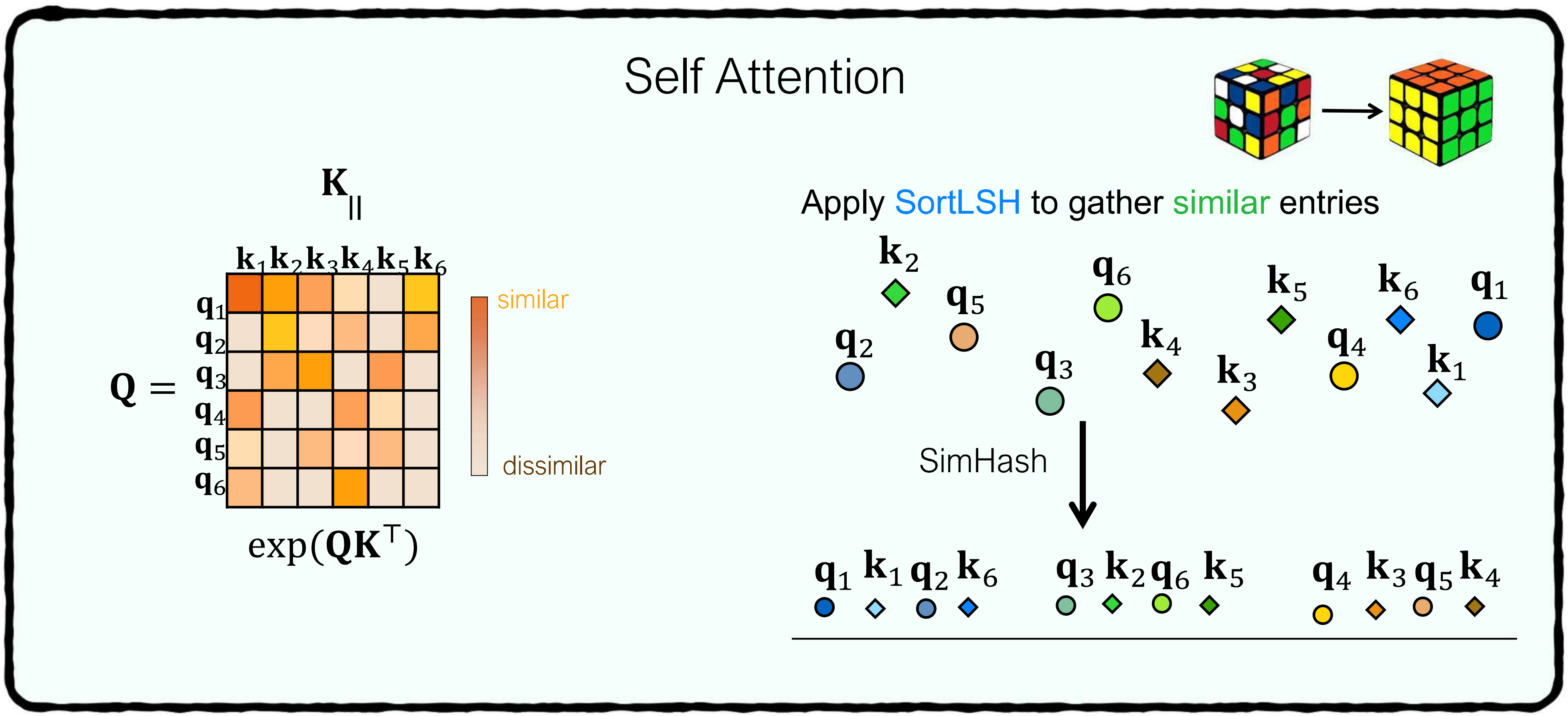
$\mathbf{A} = \exp(\mathbf{Q}\mathbf{K}^T)$

memory/runtime: $\mathcal{O}(n^2)$

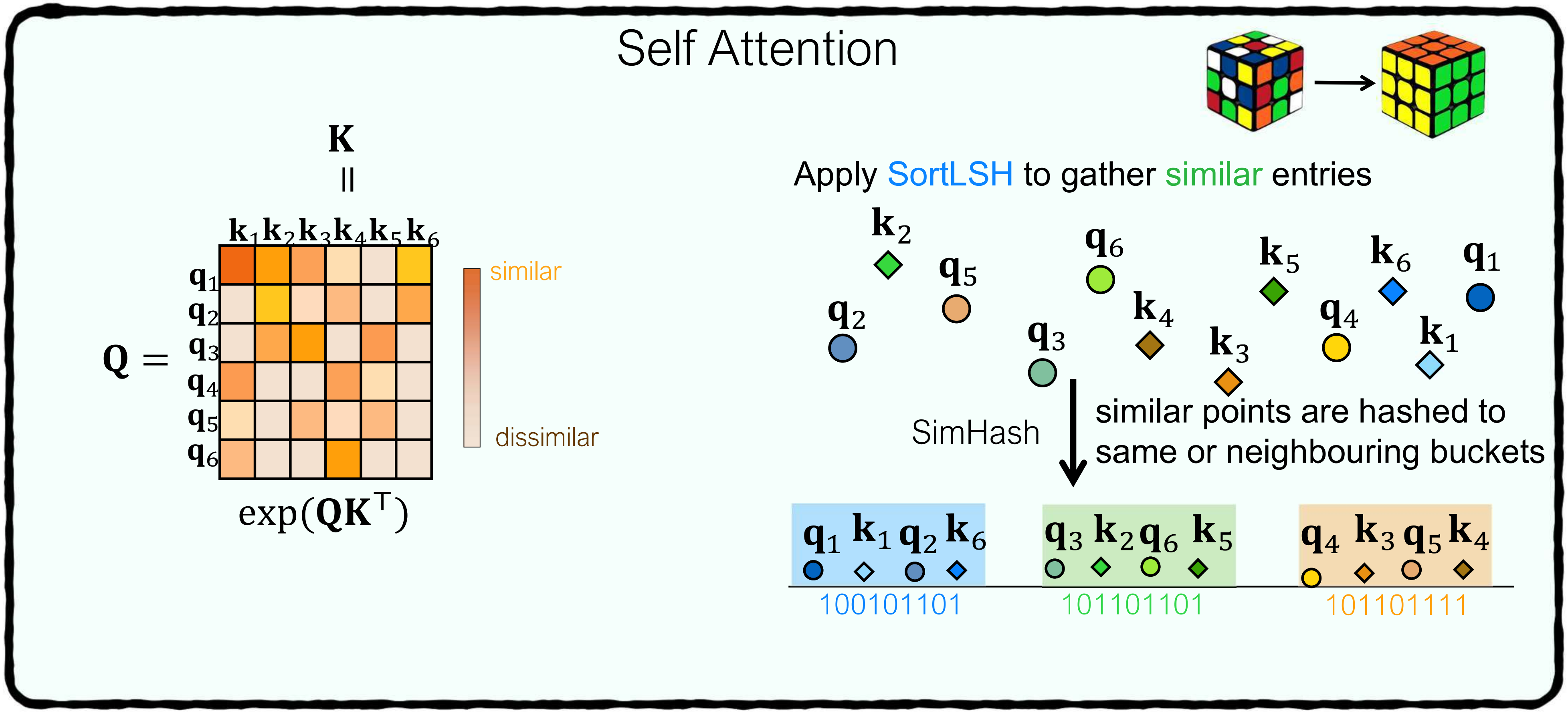
How can Heavy entries be gathered?

$\Rightarrow$ SortLSH (Locality Sensitive Hashing)

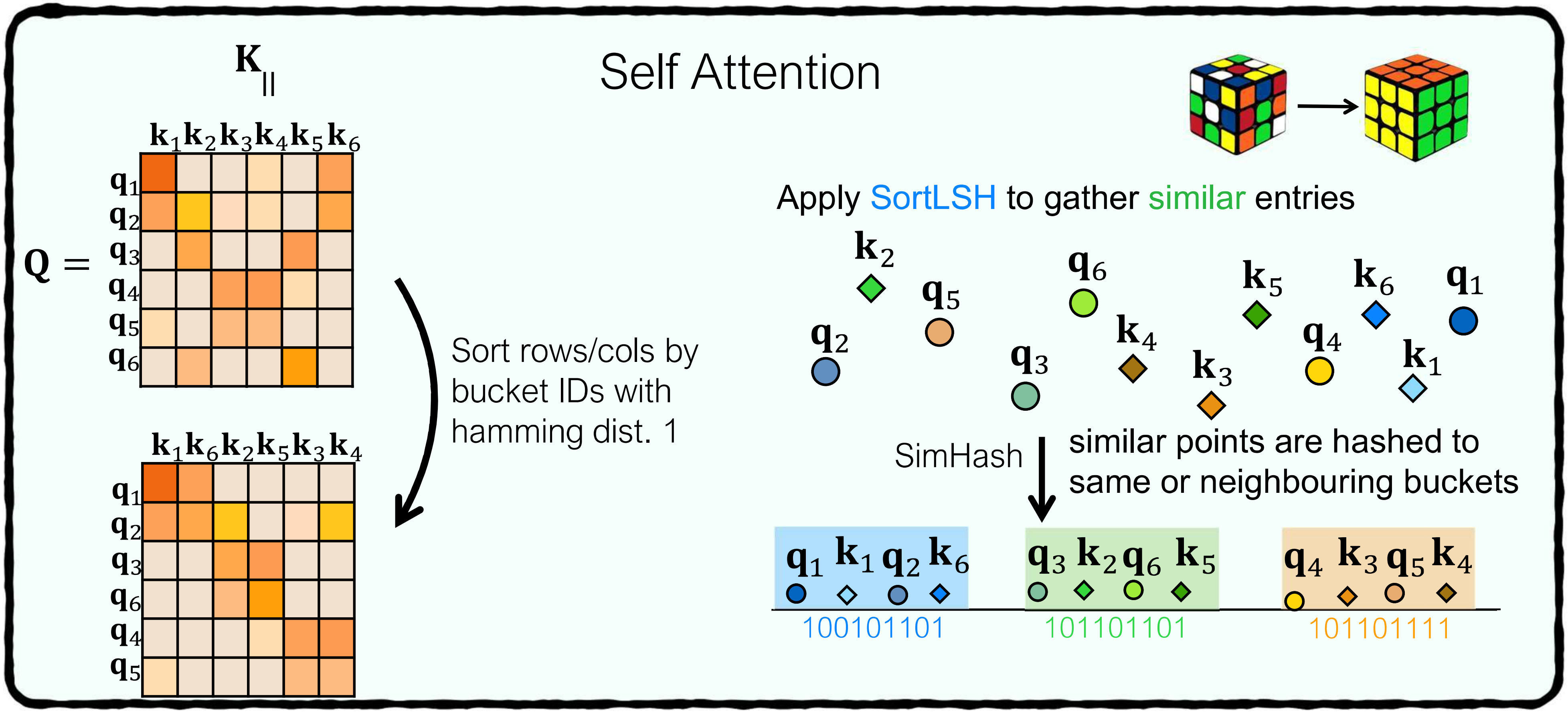
Algorithmic Acceleration



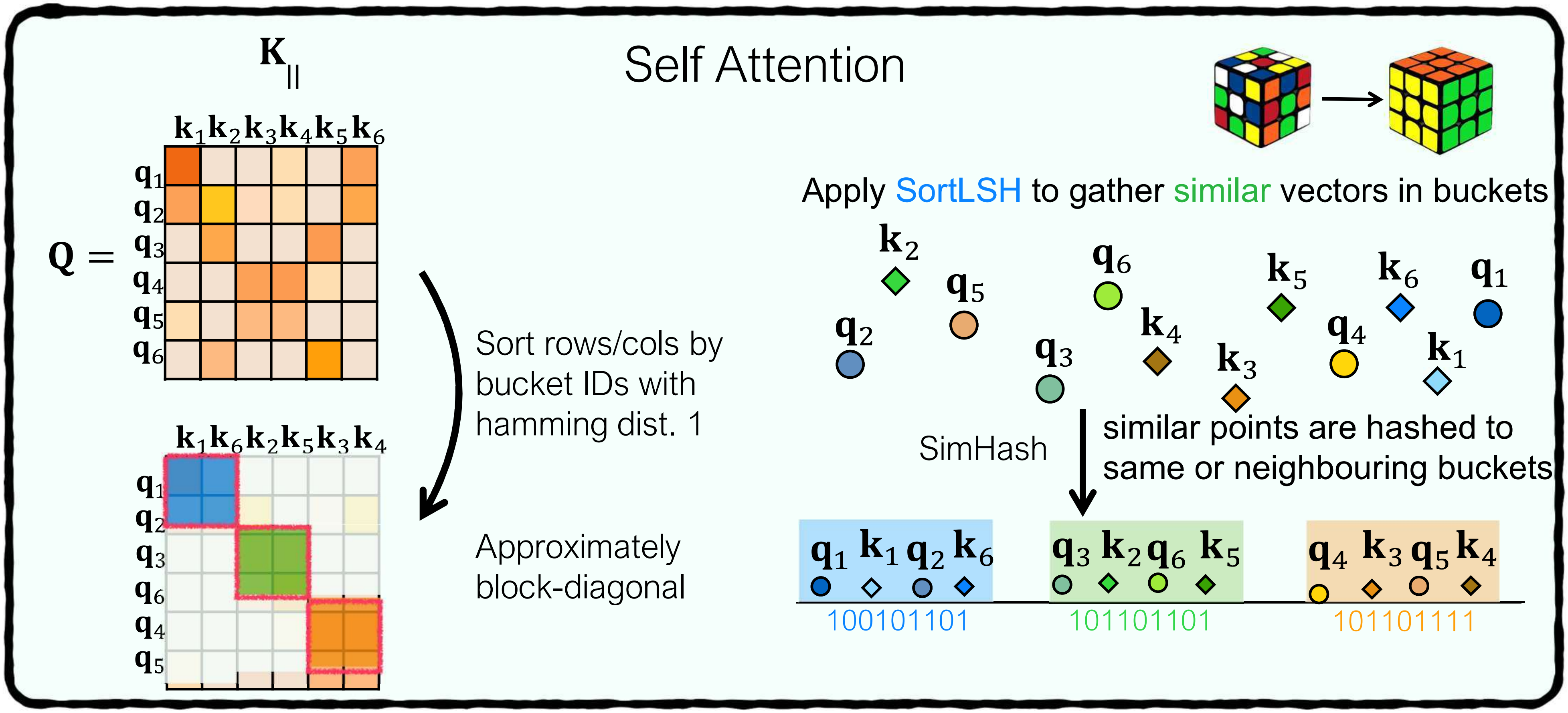
Algorithmic Acceleration



Algorithmic Acceleration

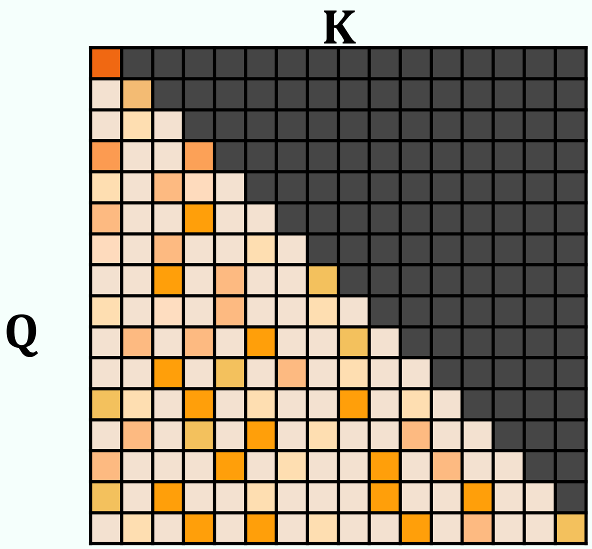


Algorithmic Acceleration



Algorithmic Acceleration

Causal Masking

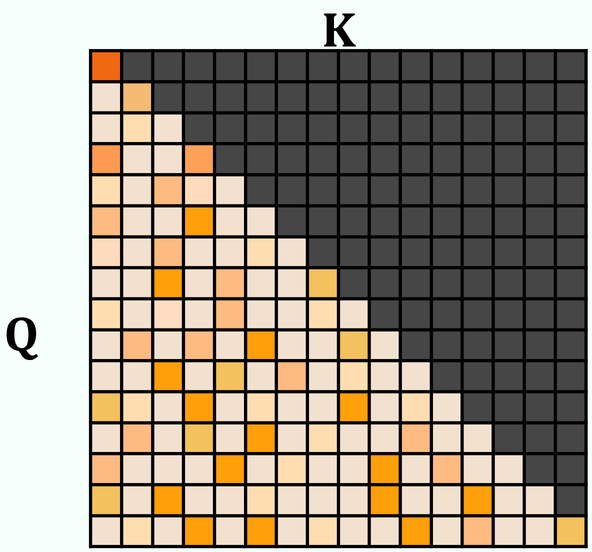


$\exp(\mathbf{QK}^{\top}) \odot \mathbf{M}_{causal}$

Output embeddings only depend on input embeddings in the past

Algorithmic Acceleration

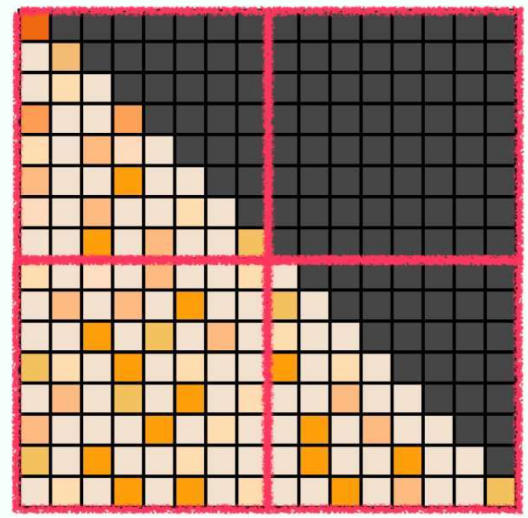
Causal Masking



$$\exp(\mathbf{QK}^\top) \odot \mathbf{M}_{causal}$$

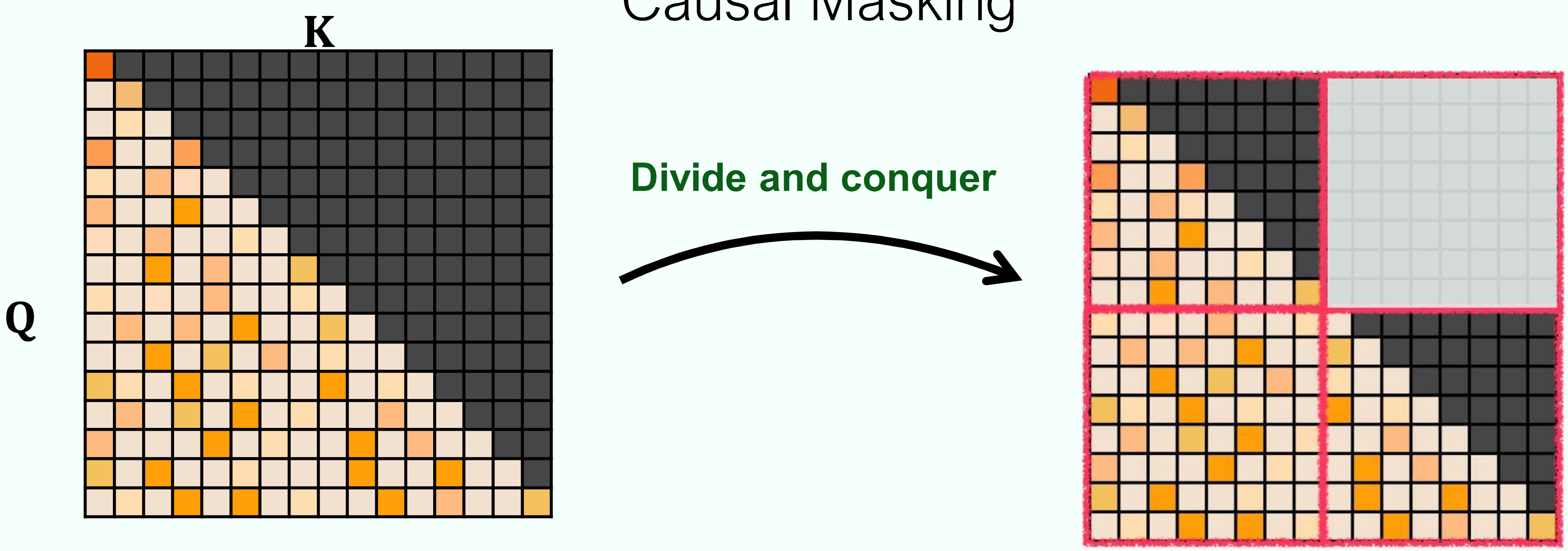
Output embeddings only depend on input embeddings in the past

Divide and conquer



Algorithmic Acceleration

Causal Masking



$$\exp(\mathbf{Q}\mathbf{K}^\top) \odot \mathbf{M}_{causal}$$

Output embeddings only depend on input embeddings in the past

Algorithmic Acceleration

ChatGLM2
6B

+

HyperAttention: Long-context Attention in Near-Linear Time

Insu Han
Yale University
insu.han@yale.edu

Rajesh Jayaram
Google Research
rkjayaram@google.com

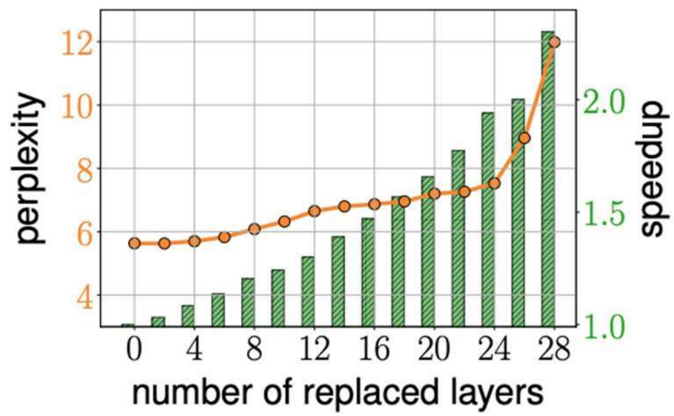
Amin Karbasi
Yale University, Google Research
amin.karbasi@yale.edu

Vahab Mirrokni
Google Research
mirrokni@google.com

David P. Woodruff
CMU, Google Research
dwoodruf@cs.cmu.edu

Amir Zandieh
Independent Researcher
amir.zed512@gmail.com

Dialog:
Marisol: it's so sweet he had been waiting
Jackie: we don't know yet when we'll get married but you are all invited ofc
Carlita: PLEASE don't pick June, I'll be in Canada then
Eunica: I hate weddings but I'll make an exception
Marisol: can't wait!



LongBench datasets with n = 32768

PolySketchFormer

Praneeth Kacham, Vahab Mirrokni, Peilin Zhong (Google Research)

Generalizations of Softmax Attention

- Let $sim(q,k) \geq 0$ be an arbitrary function that measures similarity between the query q and key k
- Attention mechanism w.r.t sim is

$$o_j = \sum_{i \leq j} \frac{sim(q_j, k_i)}{\sum_{i' \leq j} sim(q_j, k_{i'})} v_i$$

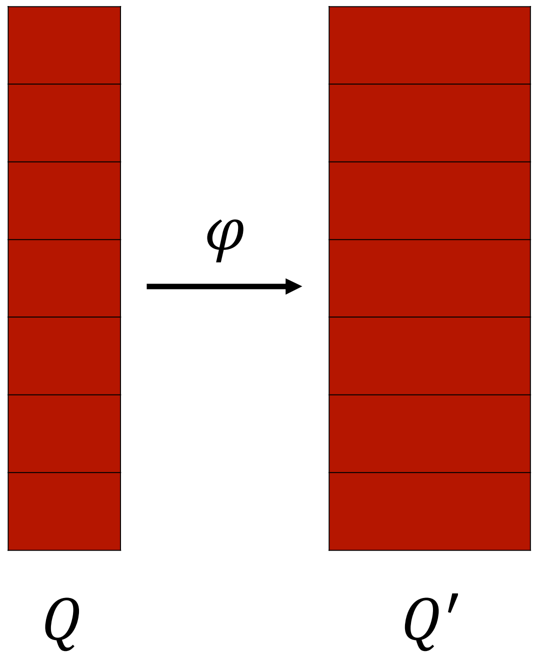
- Softmax: $sim(q,k) \doteq \exp(\langle q, k \rangle)$

Kernel View of Attention

- Suppose φ is such that $sim(q,k) = \langle \varphi(q), \varphi(k) \rangle$
- If $Q' = \varphi(Q)$ and $K' = \varphi(K)$, output is

$$D^{-1} \cdot LT(Q' \cdot (K')^{\top}) \cdot V$$

- Here LT is the lower triangular part for the causal setting
- Why write this way?
  - Linear time algorithm for computing $LT(A \cdot B^{\top}) \cdot C$
  - Runtime depends on output dimension of $\varphi(\cdot)$
- What about φ for softmax?
  - No finite dimensional feature maps



Previous Work

- Performer (Choromanski et al.,) uses a finite-dimensional map φ to approximate exponential
  - Vectors with larger norms require φ with larger dimension
- Other works consider arbitrary φ instead of first defining $sim(\cdot,\cdot)$
  - $\varphi(x) \doteq elu(x) + 1$ (Katharopoulos et al. '20), $\varphi(x) \doteq relu(x)$
  - Model quality is worse compared to softmax
- Is softmax necessary? Do other functions with similar properties work?
- Consider $sim(q,k) = \langle q,k \rangle^p$ where $p \geq 2$ is an even integer
  - Always ≥ 0
  - Increases as $\langle q,k \rangle$ goes up

Feature map for Polynomials

- A finite dimensional φ such that $\langle \varphi(q), \varphi(k) \rangle = \langle q, k \rangle^p$?
  - $\varphi: x \mapsto x^{\otimes p}$
  - If $x \in \mathbb{R}^h$, then $x^{\otimes p} \in \mathbb{R}^{h^p}$
  - $(x^{\otimes p})_{(i_1, i_2, \ldots, i_p)} = x_{i_1} \cdot x_{i_2} \cdot \cdots \cdot x_{i_p}$

$$\langle q^{\otimes p}, k^{\otimes p} \rangle = \langle q, k \rangle^p$$

Linear Attention using Polynomials

- Given $Q, K, V \in \mathbb{R}^{n \times h}$
  - Compute $Q^{\otimes p}$ and $K^{\otimes p}$
  - $LT(Q^{\otimes p} \cdot (K^{\otimes p})^{\top}) \cdot V$ in $O(nh^{p+1})$ time
- Typically, h = 64, 128, 256
  - Too expensive even for $p = 4$
- Use sketching to approximate!

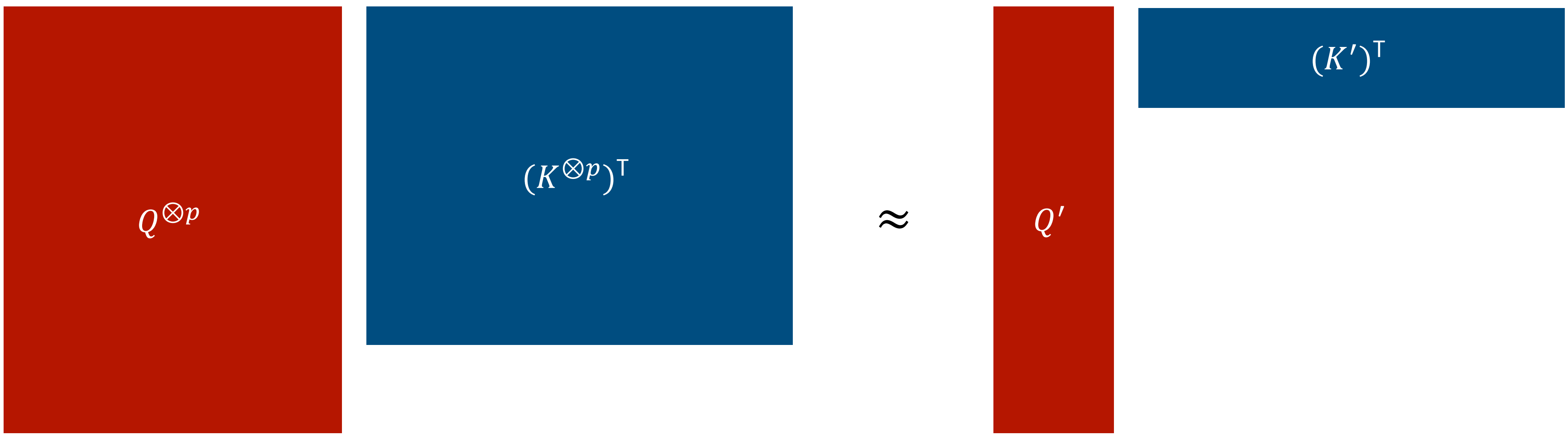
Sketching for Approximate Matrix Multiplication

- Want to compute

$$LT(Q^{\otimes p} \cdot (K^{\otimes p})^{\mathsf{T}}) \cdot V$$

- $Q^{\otimes p}$ and $K^{\otimes p}$ can have a large number of columns
- Can we compute matrices Q' and K' such that $Q^{\otimes p} \cdot (K^{\otimes p})^{\mathsf{T}} \approx Q' \cdot (K')^{\mathsf{T}}$?
  - Ahle et al. '20 give a fast sketch called TensorSketch
  - Can approximate using $LT(Q' \cdot (K')^{\mathsf{T}}) \cdot V$

Sketching for Approximate Matrix Multiplication



Matrix Sketching

- Never have to compute the matrices $Q^{\otimes p}, K^{\otimes p}$ and just use Q' and K'
- Can simply compute $LT(Q' \cdot (K')^\top) \cdot V$ in linear time
- Does this work?
  - Model training fails to converge
- Non-negativity
  - $Q' \cdot (K')^\top$ can have negative entries, whereas entries of $Q^{\otimes p} \cdot (K^{\otimes p})^\top$ are ≥ 0

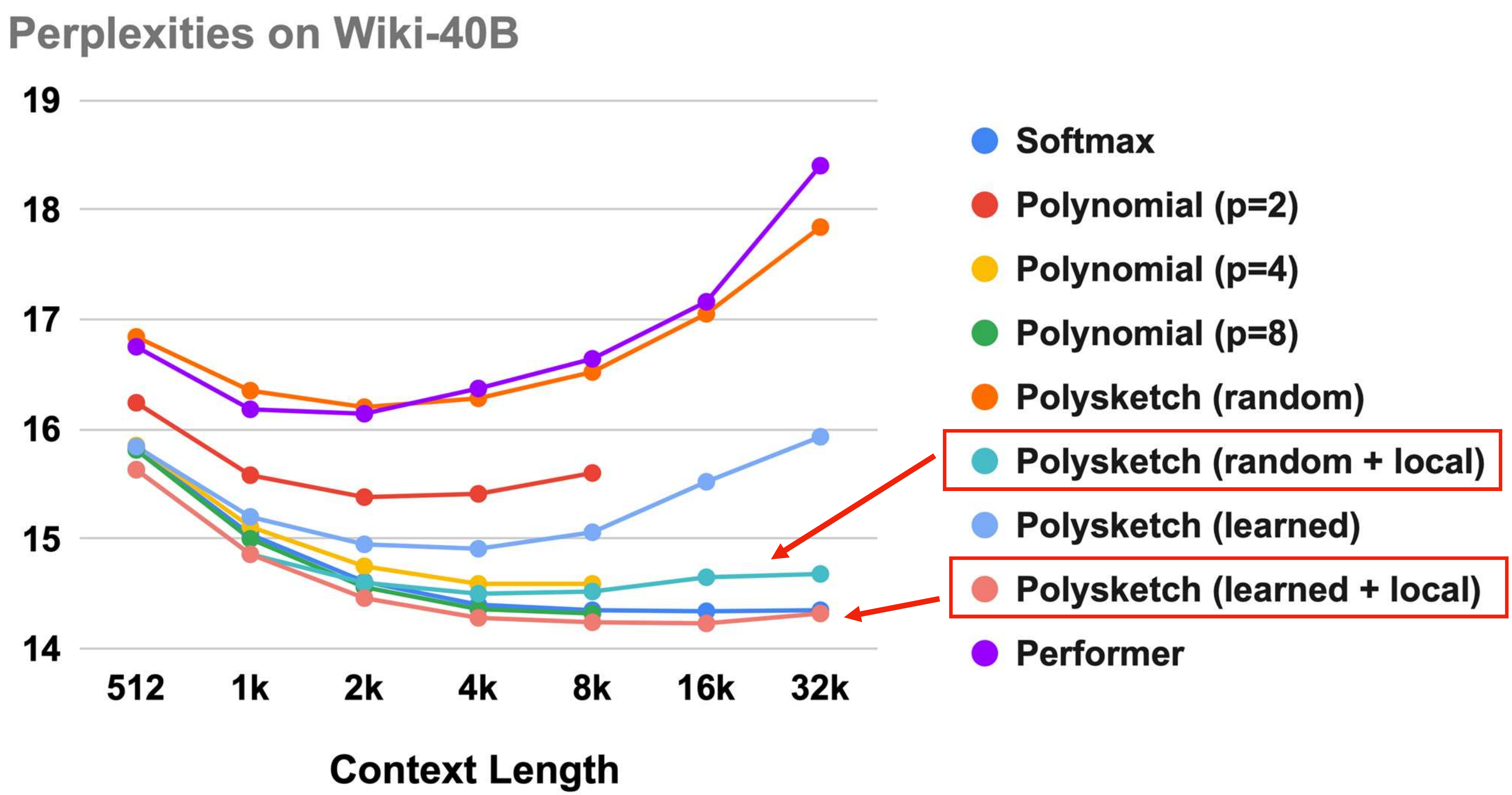
Solving Issue of Negative Entries

- Consider $Q'' = (Q')^{\otimes 2}$ and $K'' = (K')^{\otimes 2}$
  - Q', K' are sketches for degree $p/2$
- All entries of $Q'' \cdot (K'')^{\mathsf{T}}$ are non-negative! They are of the form $\langle q', k' \rangle^2 \geq 0$
- Show that if Q' and K' have an approximate matrix product property for degree $p/2$, then Q'' and K'' have a similar guarantee for degree p
  - $\| \, Q'' \cdot (K'')^{\mathsf{T}} - Q^{\otimes p} (K^{\otimes p})^{\mathsf{T}} \, \|_F$ is small
- Compute $LT(Q'' \cdot (K'')^{\mathsf{T}}) \cdot V$
- The model converges!

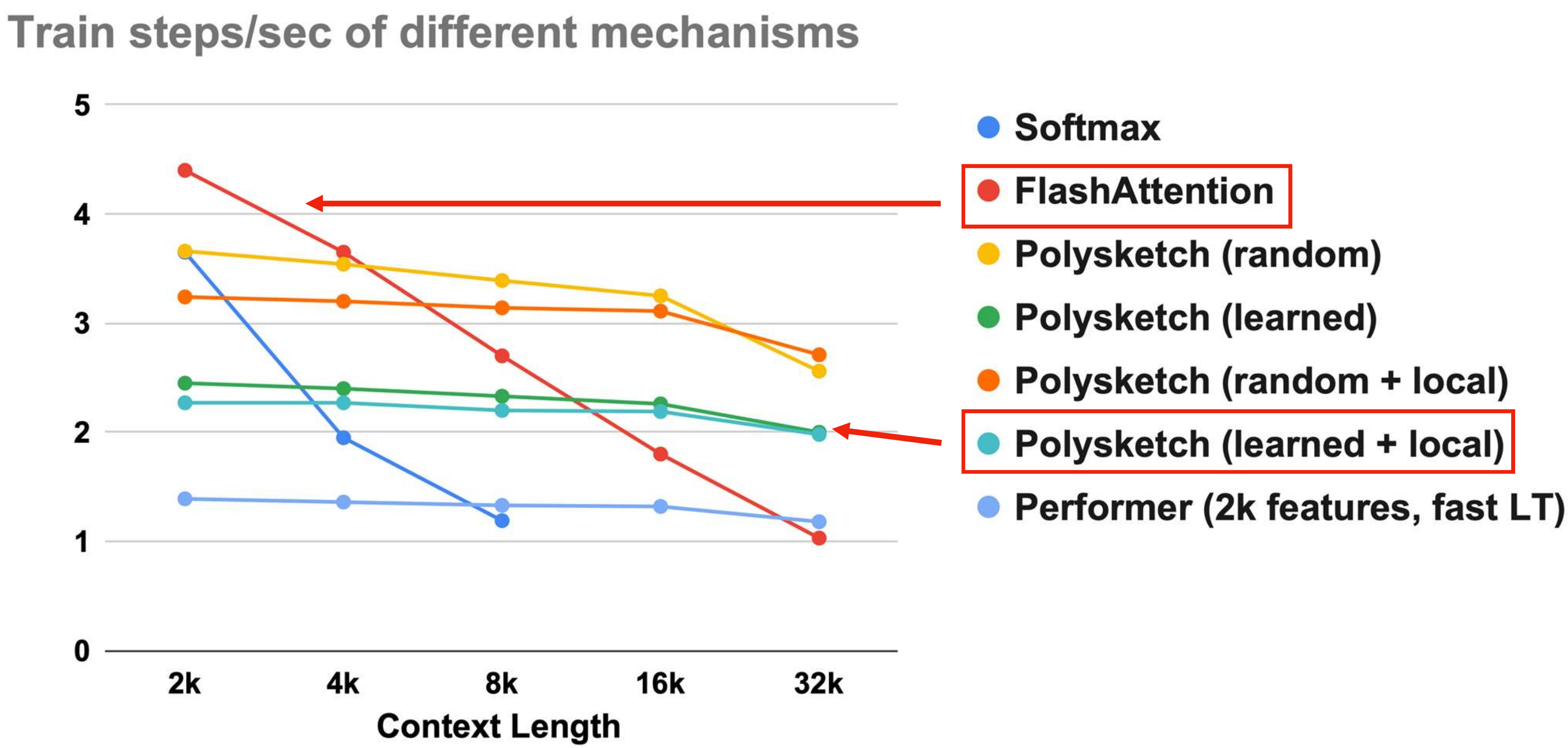
Other Optimizations

- TensorSketch is a random sketch – instead, treat the sketch as learnable parameters
- When computing $LT(A \cdot B^{\top}) \cdot C$, use block multiplication and cumulative sums
- Compute diagonal blocks exactly as such blocks are sensitive to approximation

Model Perplexities



Training Latencies



Conclusions and Future Work

- In practice,
  - FlashAttention is an optimized implementation of softmax attention and used heavily
  - HyperAttention on pretrained models may reduce quality too much. Fine-tuning increases quality but depends on the hash bucket sizes – still being tested!
  - Expect PolySketchFormer to do worse than softmax on very long contexts, but still may be useful for shorter contexts
- [Kannan, Bhattacharyya, Kacham, W] use tools from randomized linear algebra to show for a class of sym functions, there is a small subset of keys so that any heavy attention score involves a key from that subset. Still being tested!
- Open questions
  - Can we achieve linear time for Hyperattention with weaker assumptions?
  - Can we design a non-negative tensorsketch for PolySketchFormer without squaring the embedding dimension?
  - Are there other natural assumptions that allow us to break the quadratic time worst case hardness?