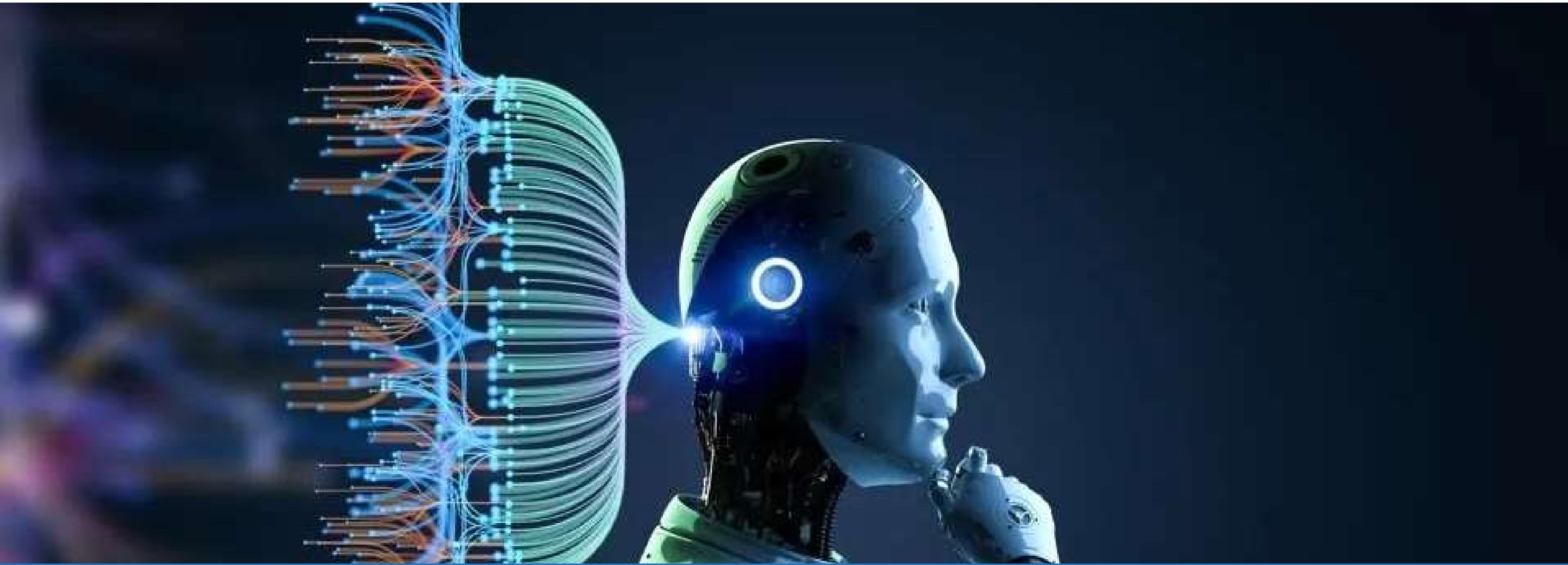
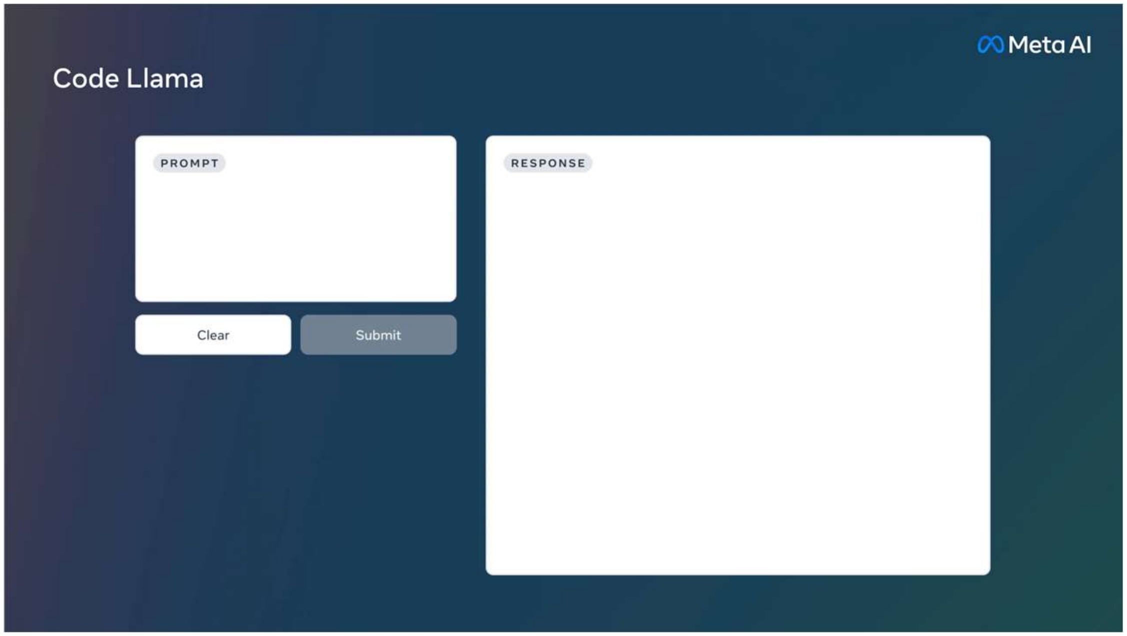


HyperAttention: Long-Context Attention in Near-Linear Time

David Woodruff (CMU / Google)

Joint work with Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, Amir Zandieh (Yale / Google)

AI Revolution



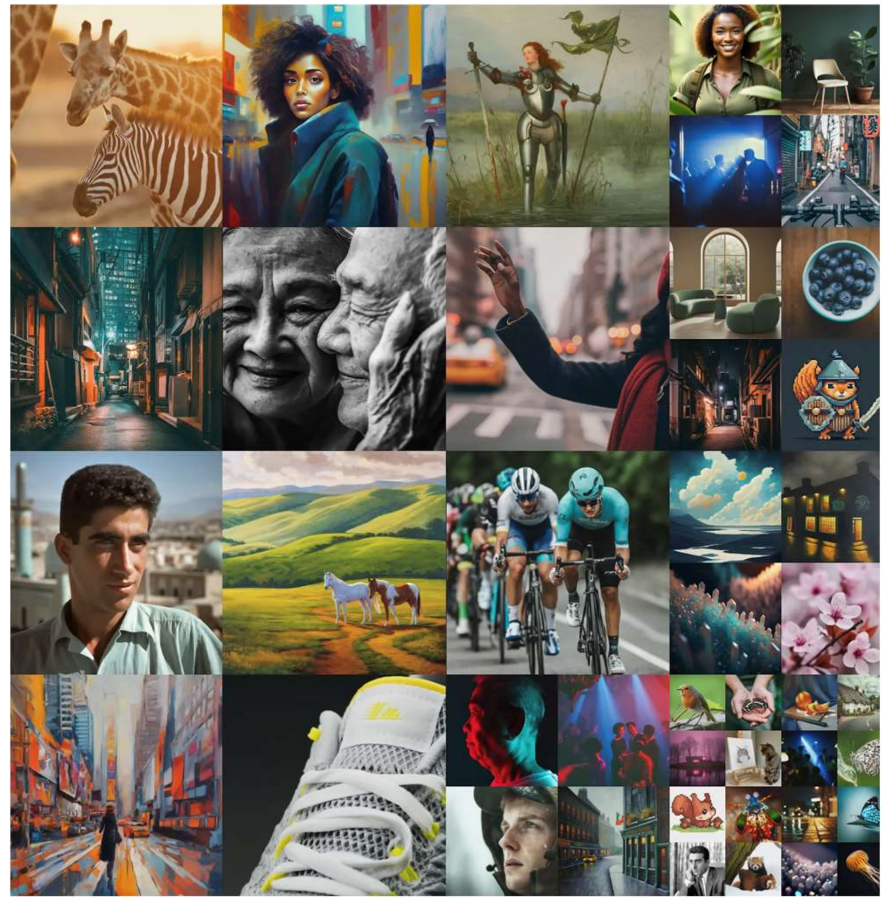
2023 IN REVIEW

THE YEAR A.I. ATE THE INTERNET

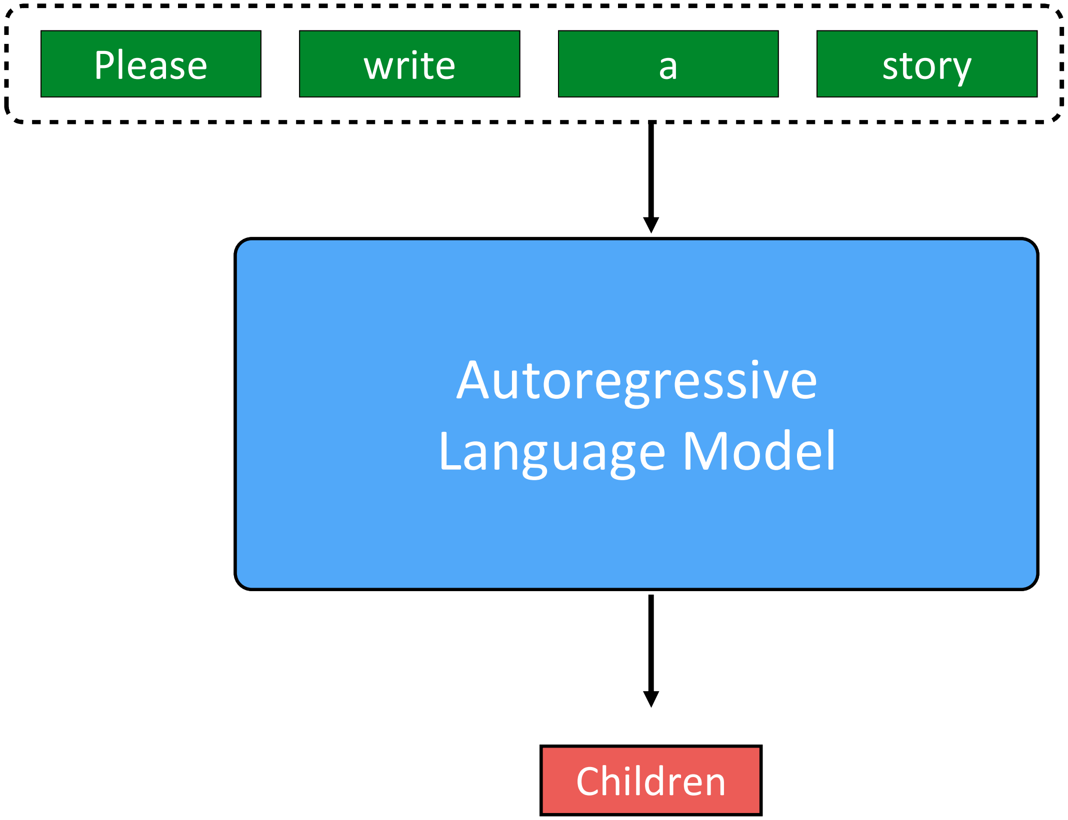
Call 2023 the year many of us learned to communicate, create, cheat, and collaborate with robots.

By Sue Halpern

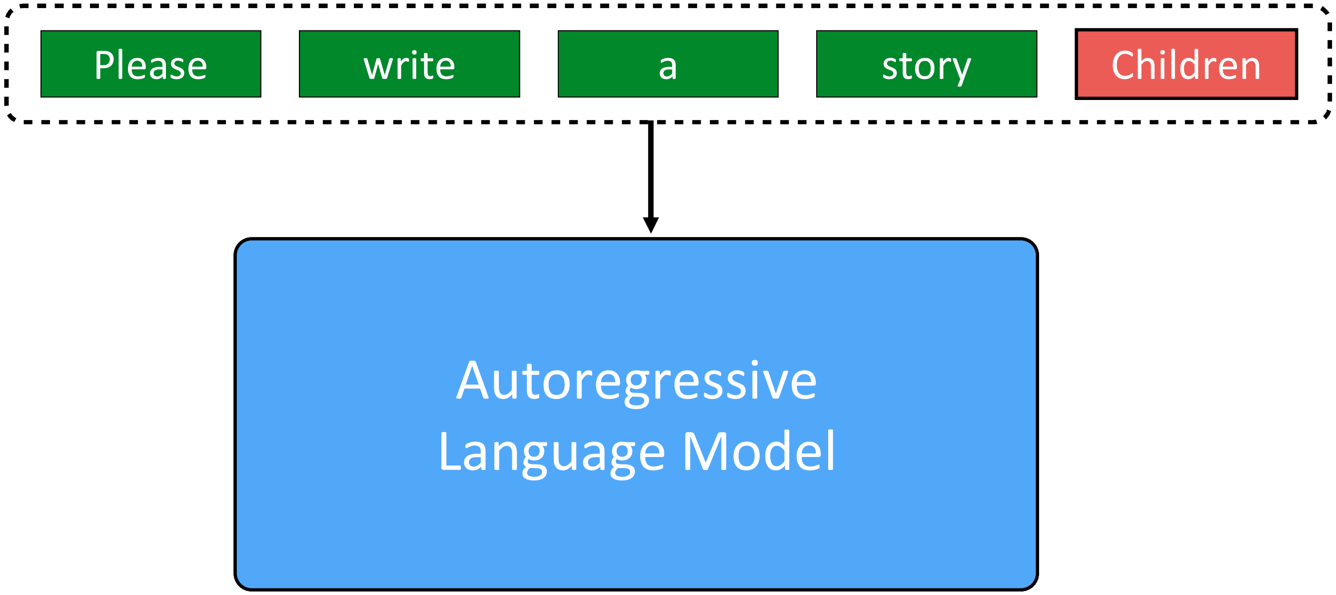
December 8, 2023



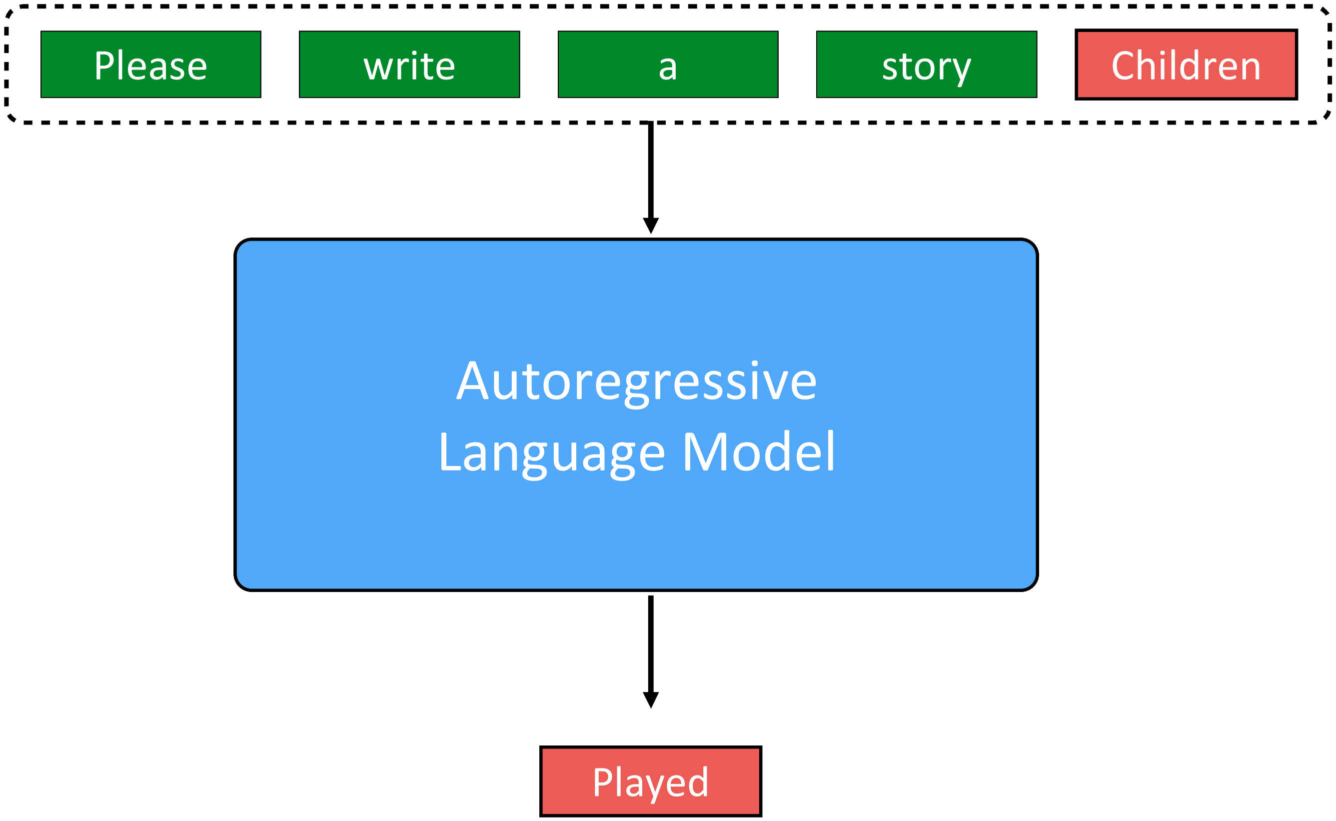
Autoregressive Models



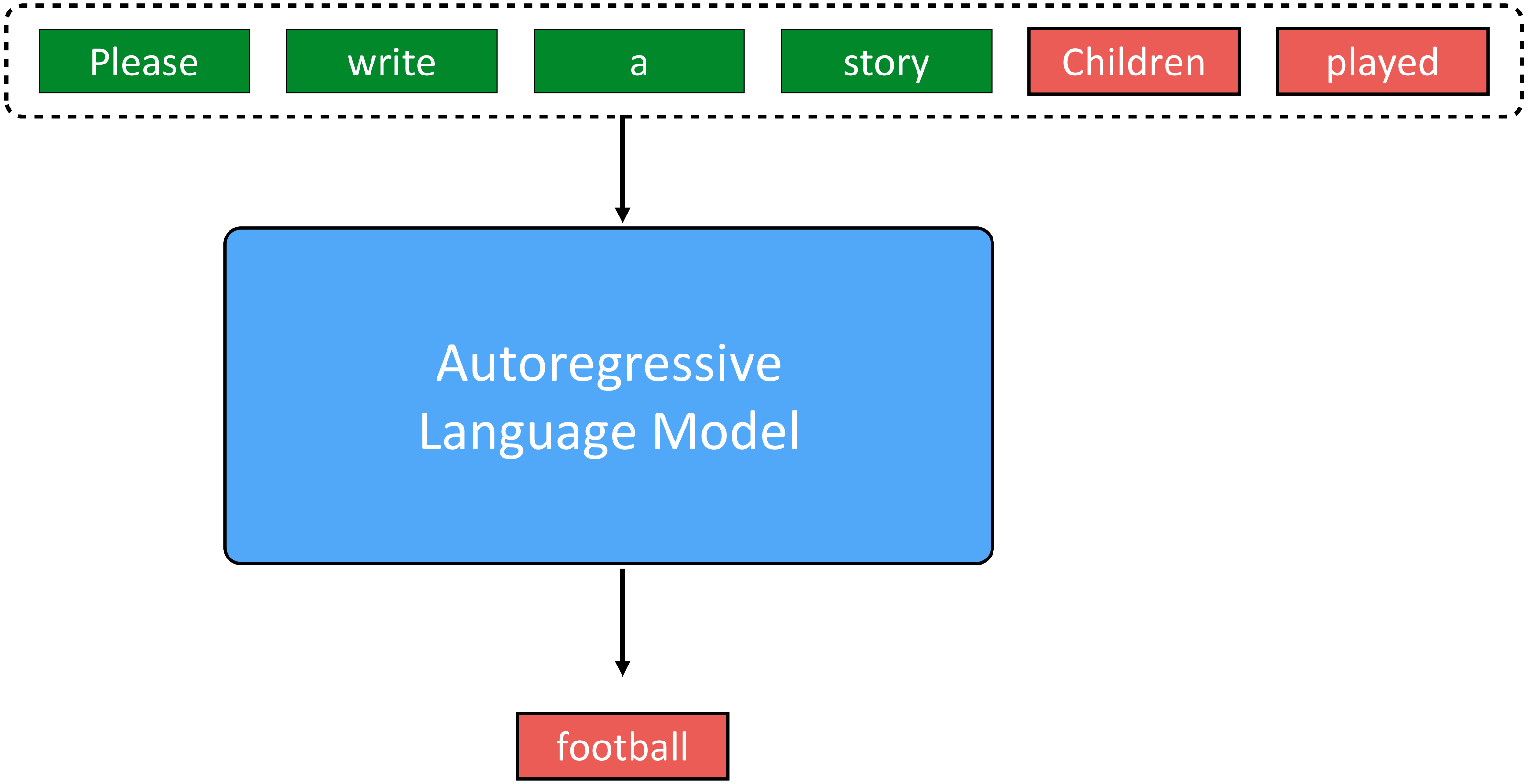
Autoregressive Models



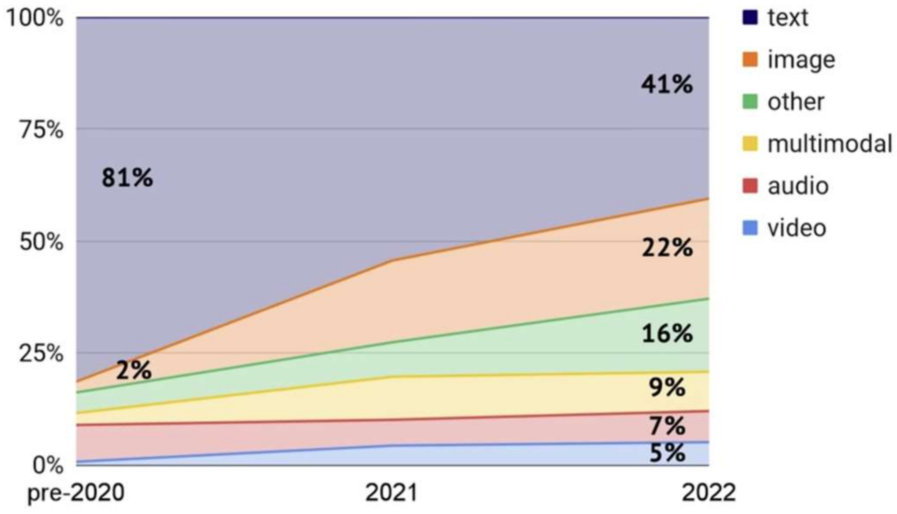
Autoregressive Models



Autoregressive Models

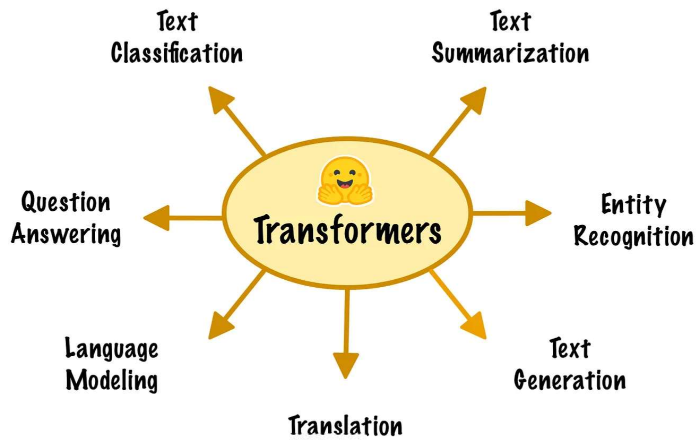


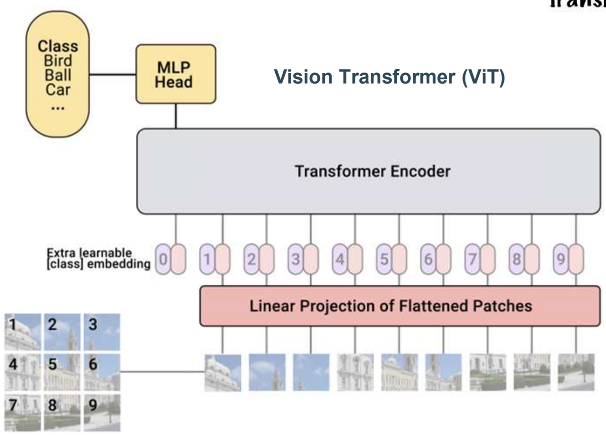
Transformers



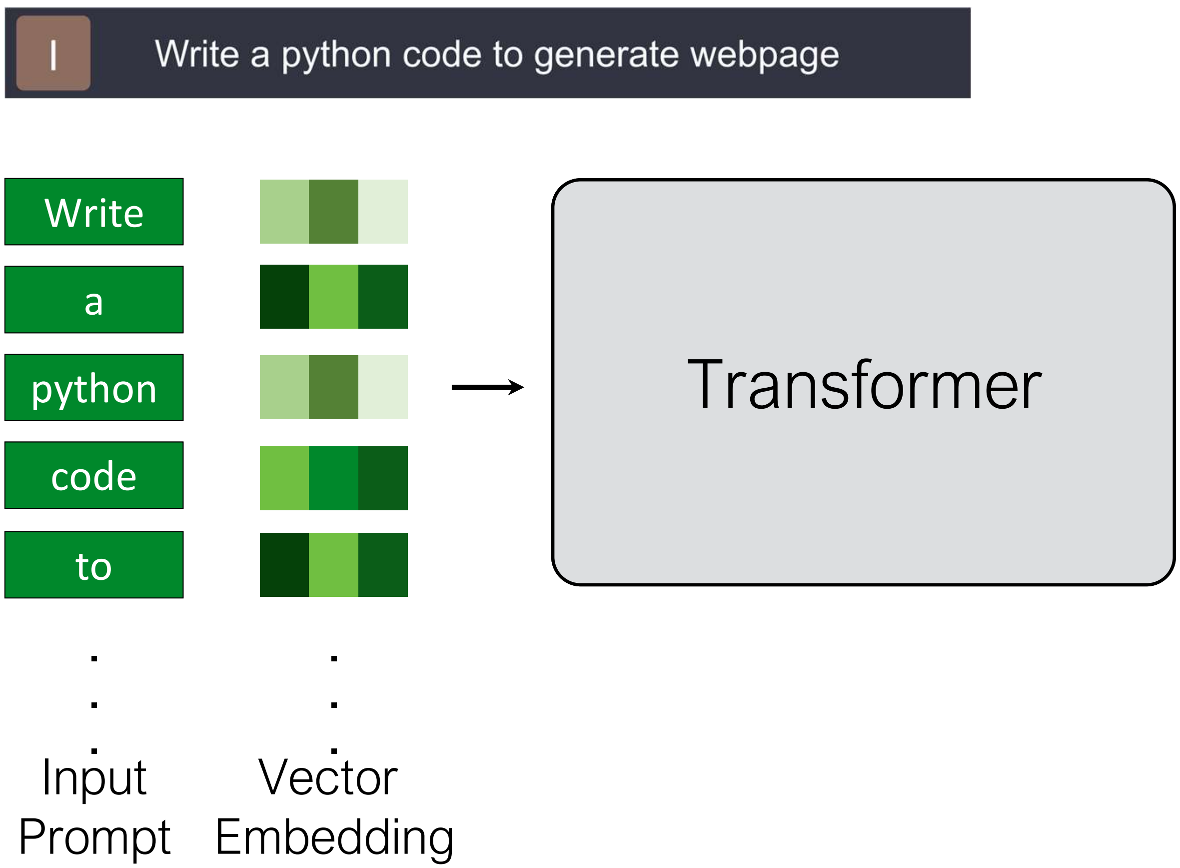
Growth of transformer-related papers

Transformer (Vaswani et al., 17')

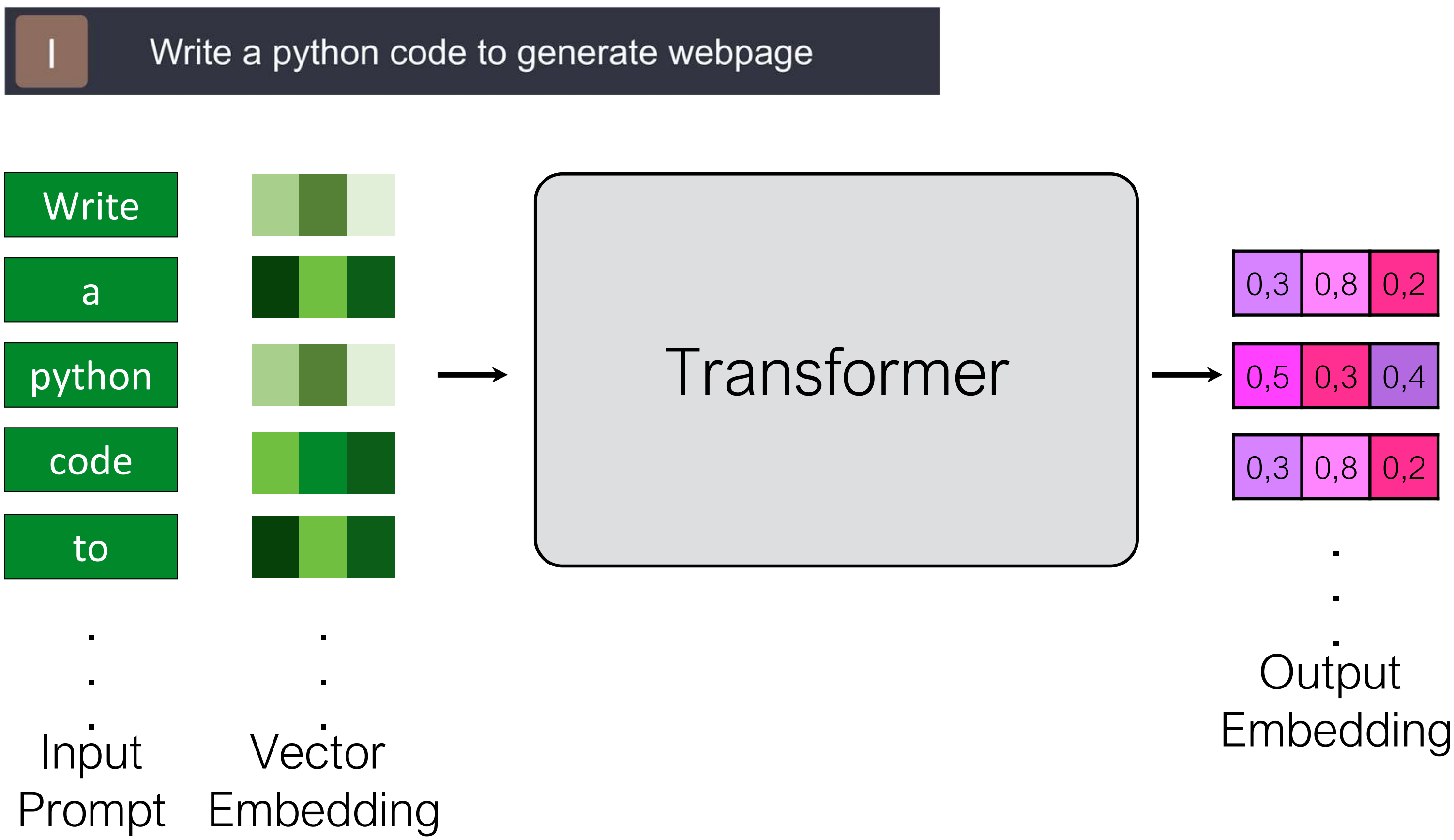




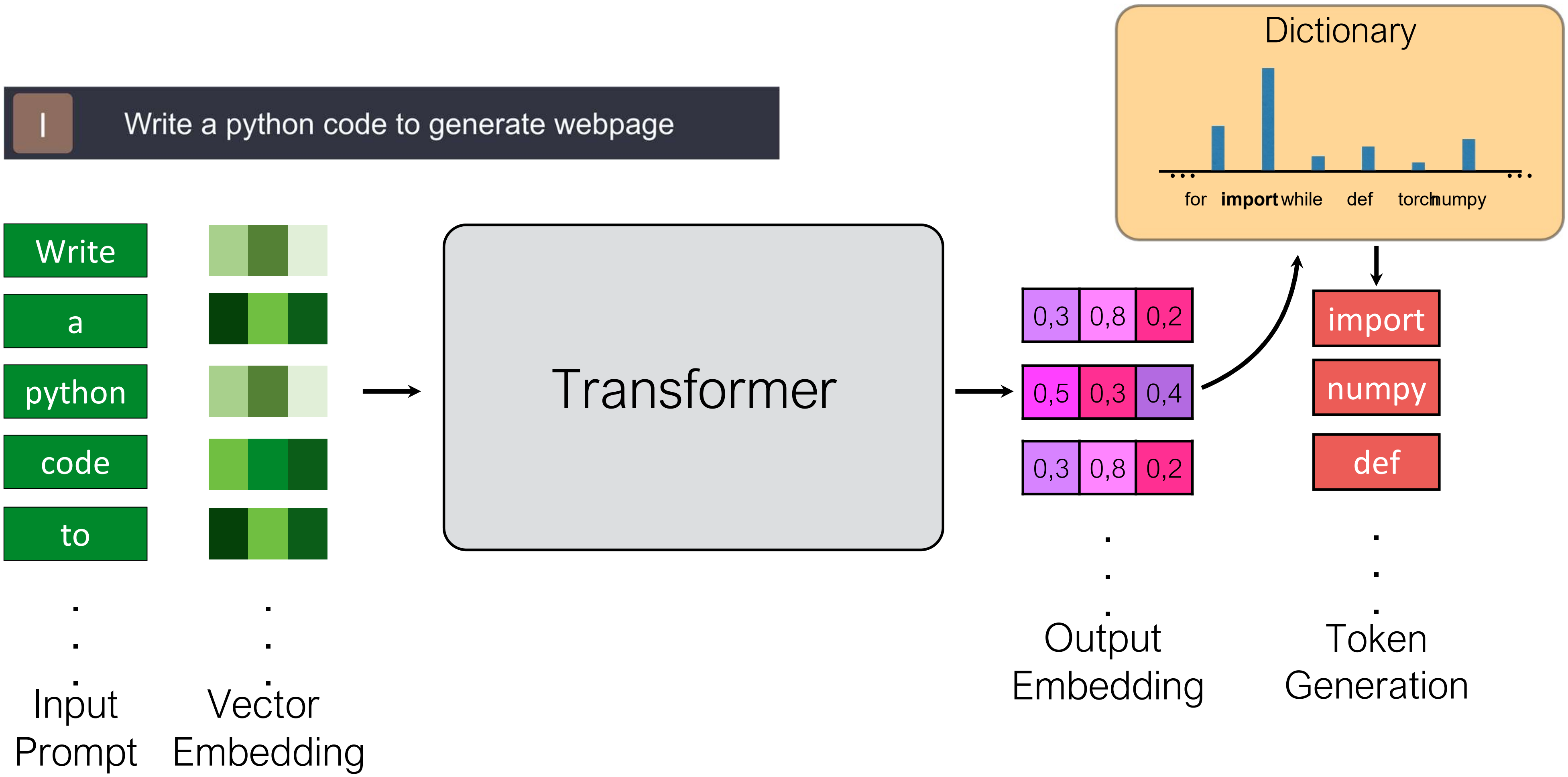
Transformers



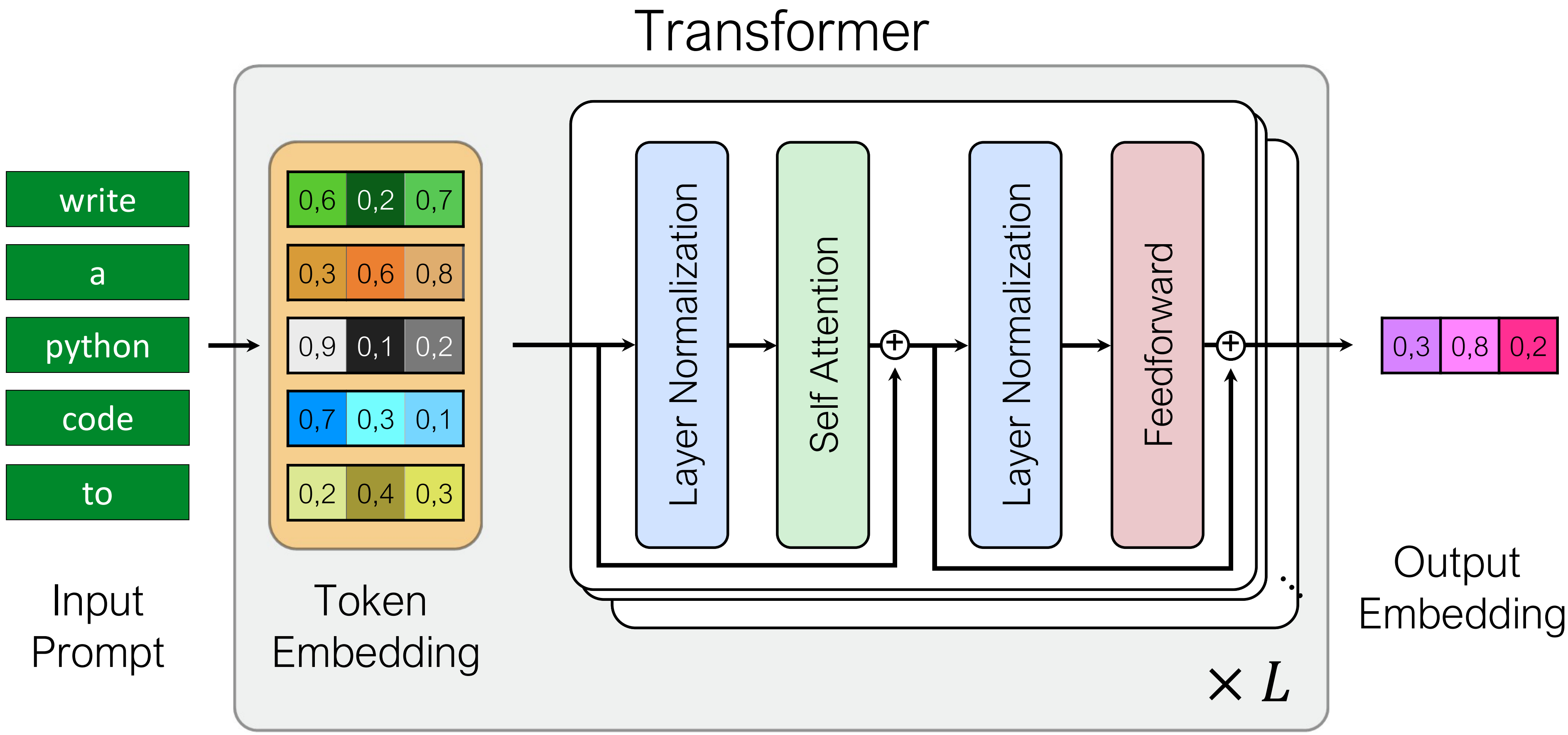
Transformers



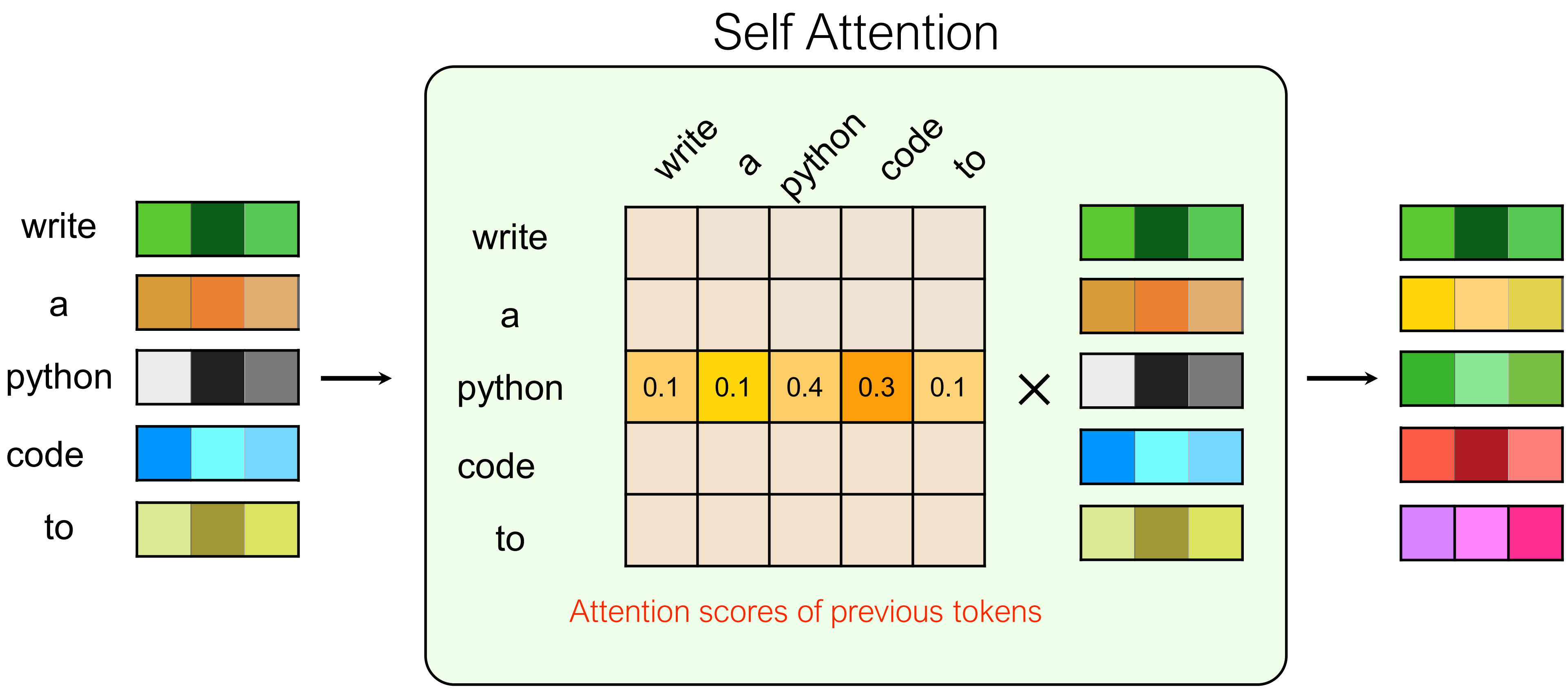
Transformers



Transformers

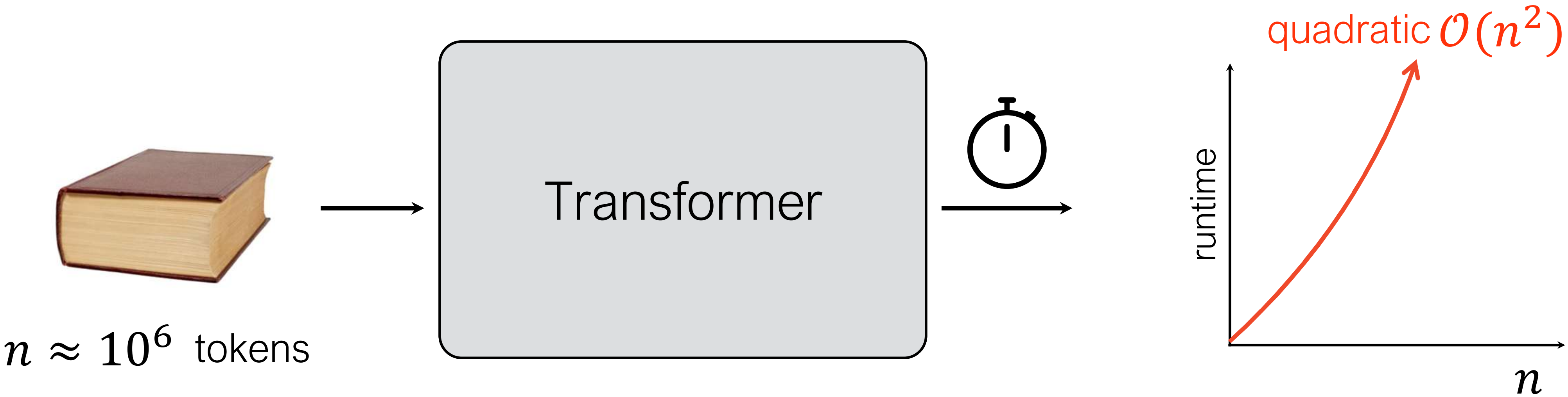


Transformers

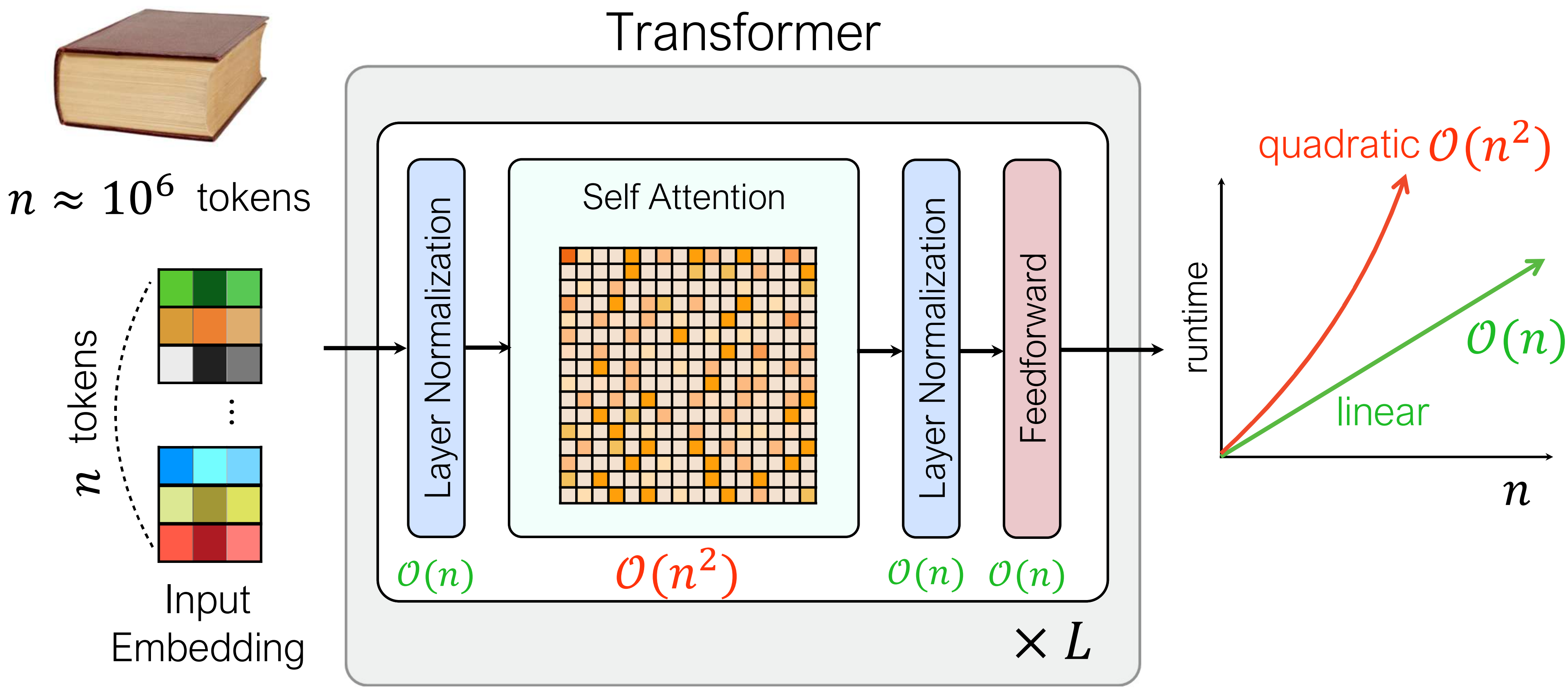


Algorithmic Acceleration

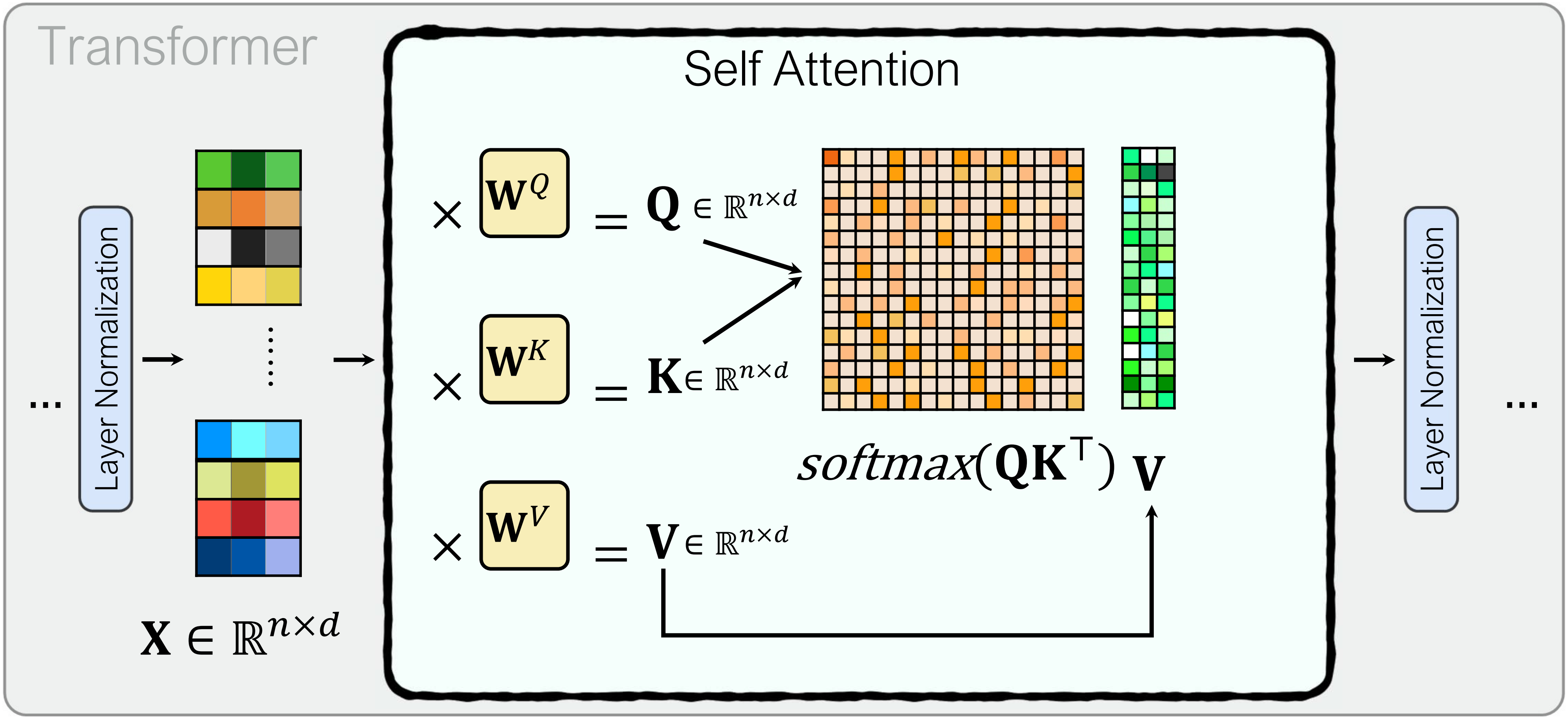
Algorithmic Acceleration



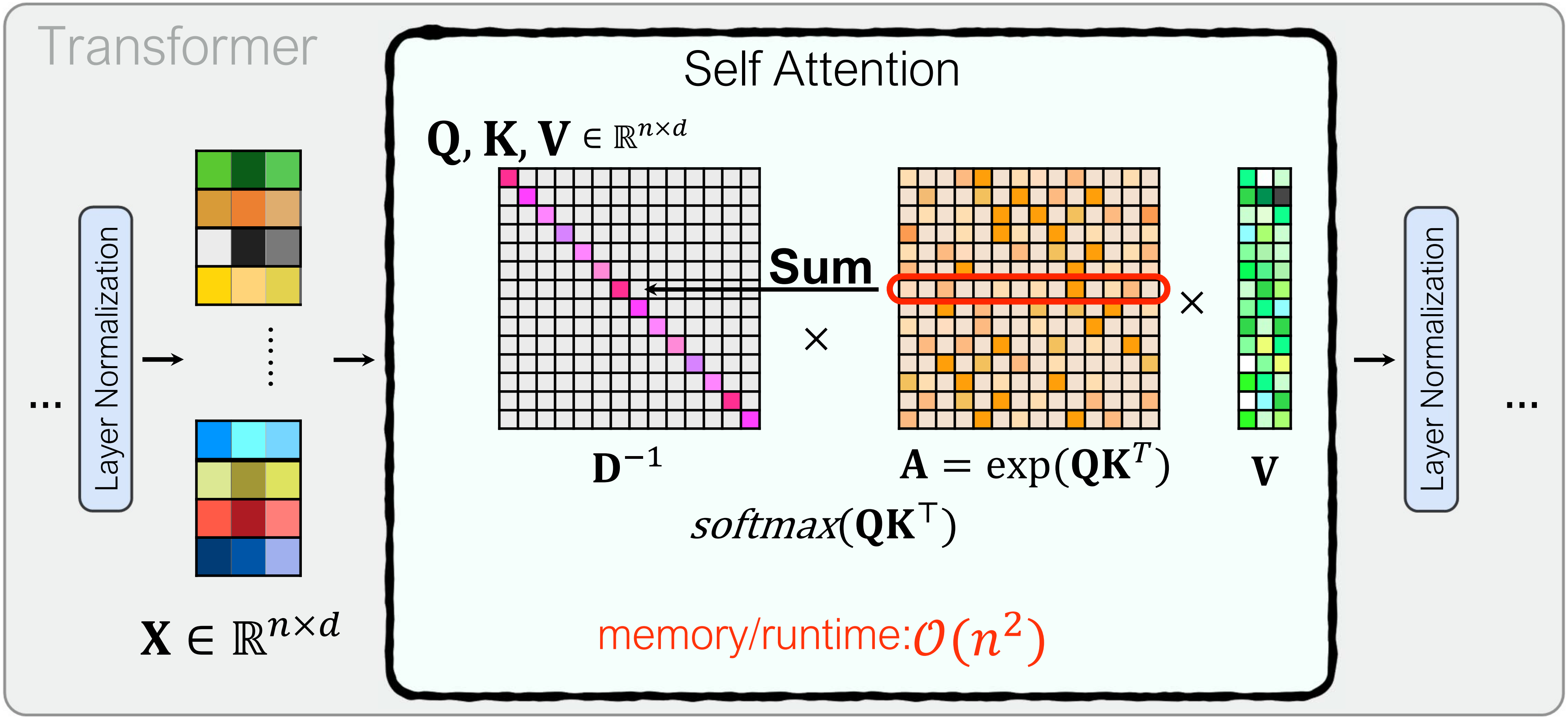
Algorithmic Acceleration



Algorithmic Acceleration



Algorithmic Acceleration

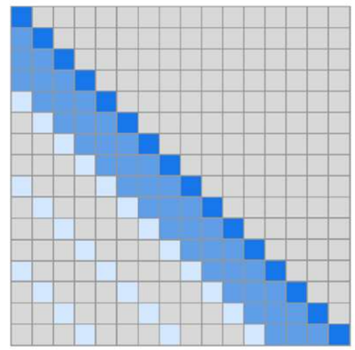


Previous Work

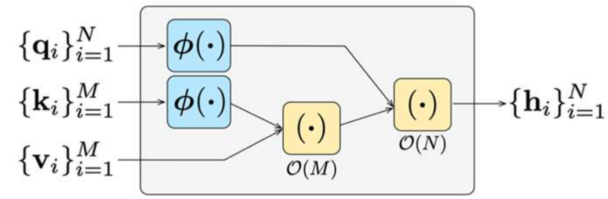
- **Sparse Structure**
  - Local Attention (Parmar et al., 18')
  - Sparse Transformer (Child et al., 19')
  - Longformer (Beltagy et al., 20')
  - Reformer (Kitaev et al., 20')
  - Sinkhorn Attention (Tay et al., 20')

- **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')

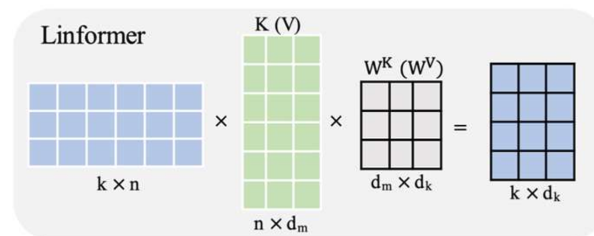
- **Low-rank Approximation**
  - Linformer (Wang et al., 20')
  - Nystromformer (Xiong et al., 21')
  - Nested Attention (Max et al., 21')

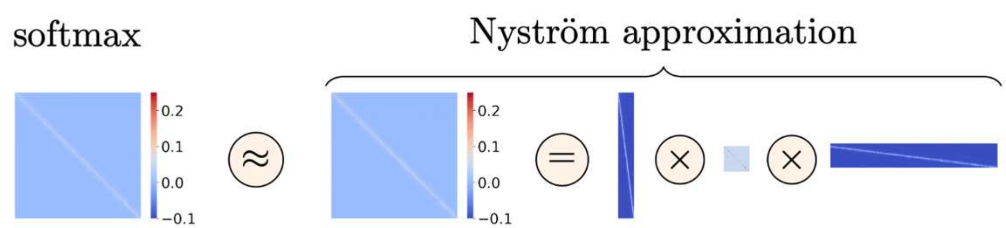


(b) Sparse Transformer (strided)



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

Algorithmic Acceleration

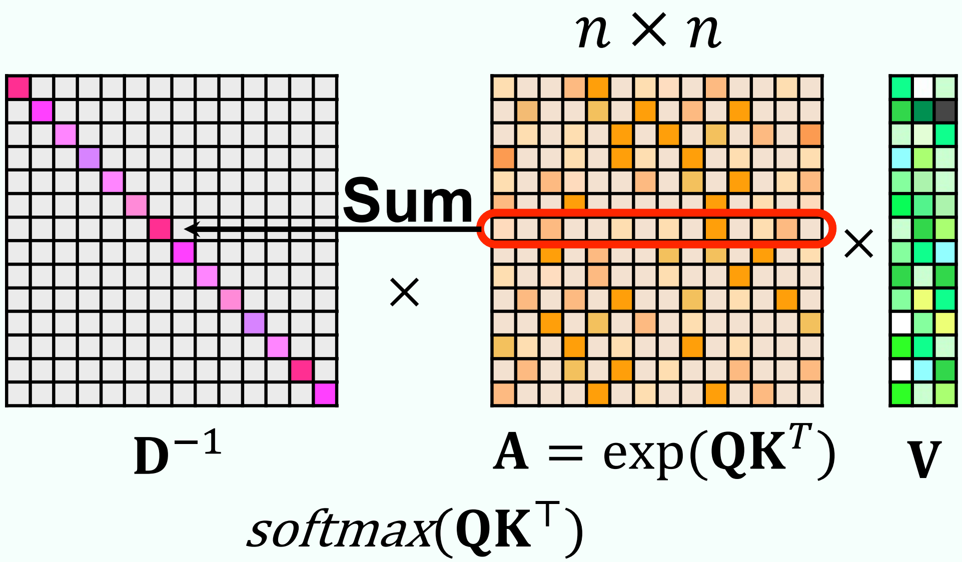
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$

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 $A \cdot V$



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

Algorithmic Acceleration

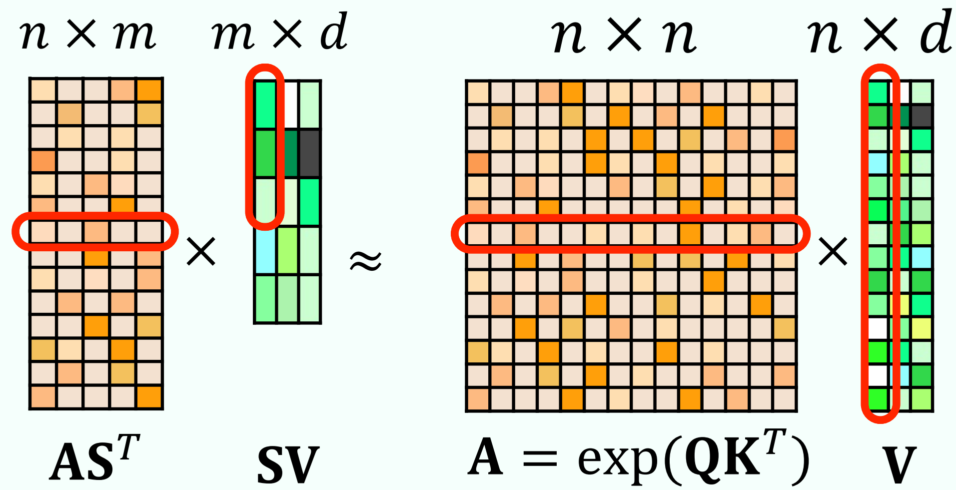
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n\times d}$

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

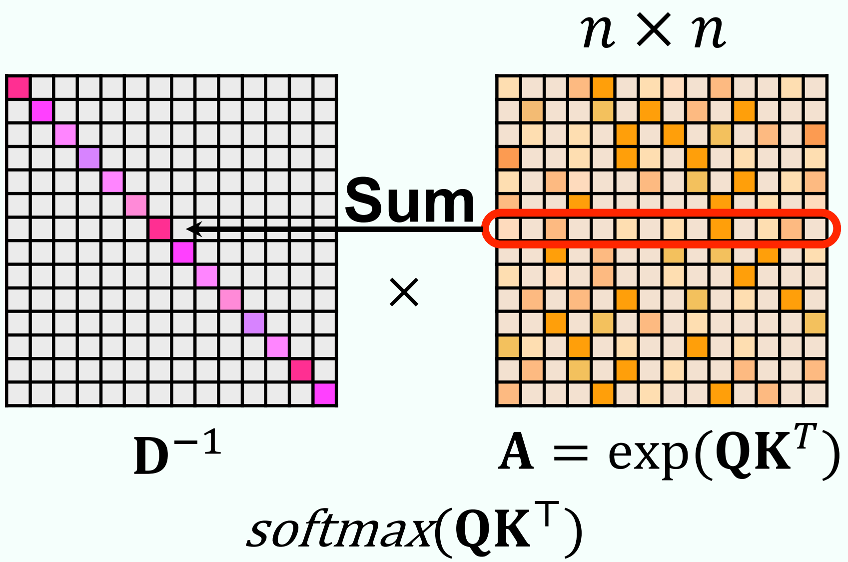
$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$

Self Attention

1. Approximate

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

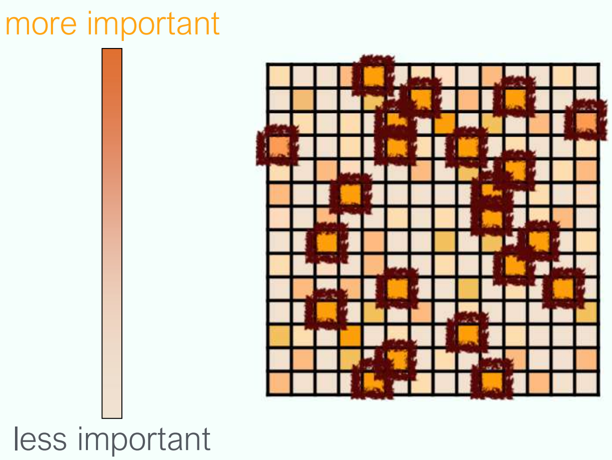
2. Compute a row sampling sketch $S \in \mathbb{R}^{m \times n}$ where row i is sampled with probability $\propto \| v_i \|_2^2 \rightarrow m \approx srank(softmax(QK^\top)) \cdot d$



Algorithmic Acceleration

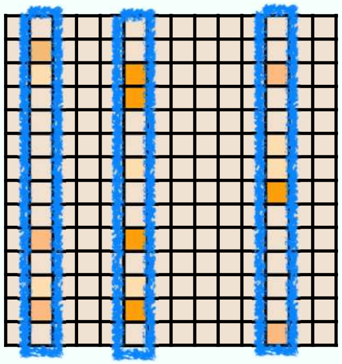
Approximate $D_{i,i} = \sum_{j \in [n]} A_{i,j} = \sum_{j \in [n]} \exp(\langle q_i, k_j \rangle)$

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



$+$

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

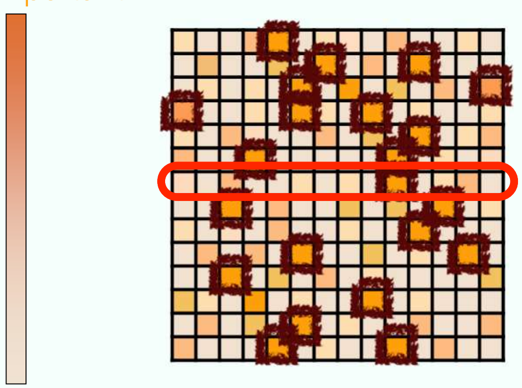


Algorithmic Acceleration

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

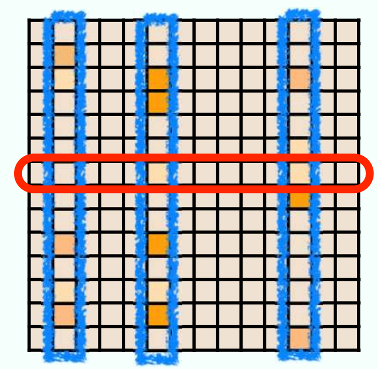
more important



less important

$+$

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



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

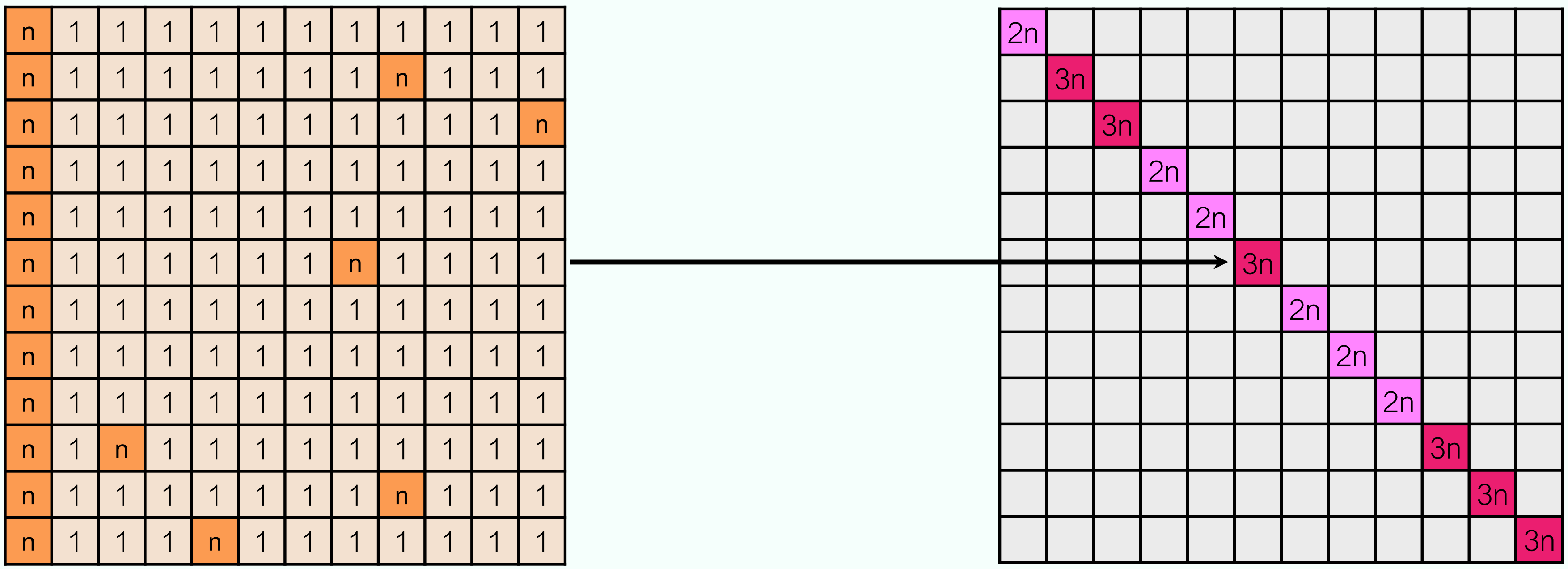
Algorithmic Acceleration

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 $\widetilde{Att}$ can be computed in $O(dn^{1+o(1)})$ time with:

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

- 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

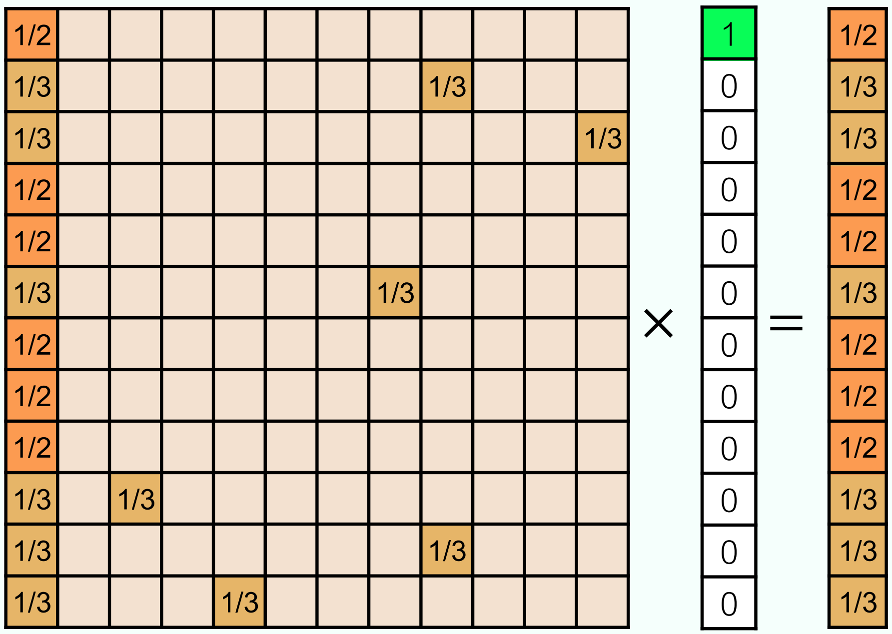
Algorithmic Acceleration



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

$\mathbf{D}$

Algorithmic Acceleration



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

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

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

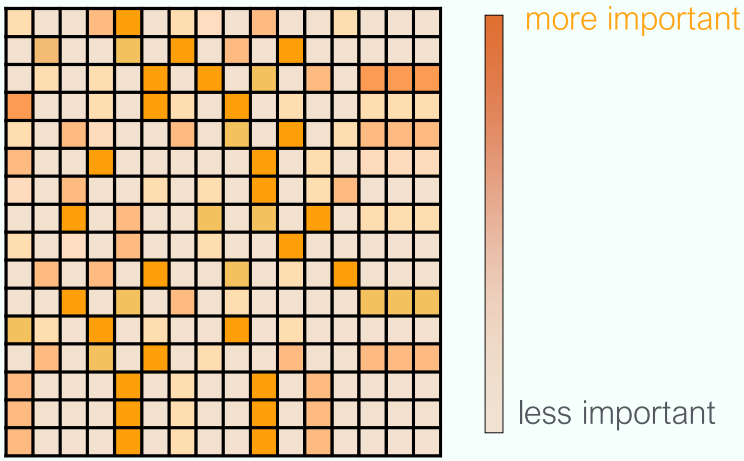
Hardness: $n^{2-o(1)}$

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

Algorithmic Acceleration

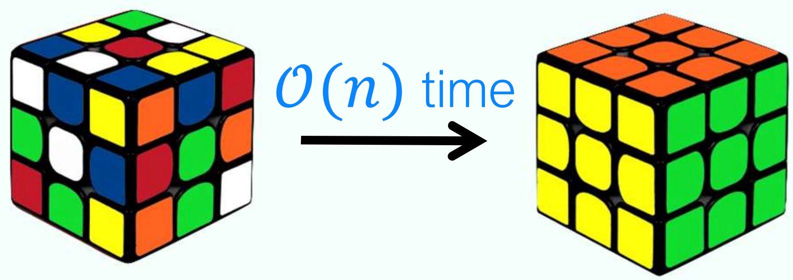
Finding Heavy contributions in practice

$n \times n$



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

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

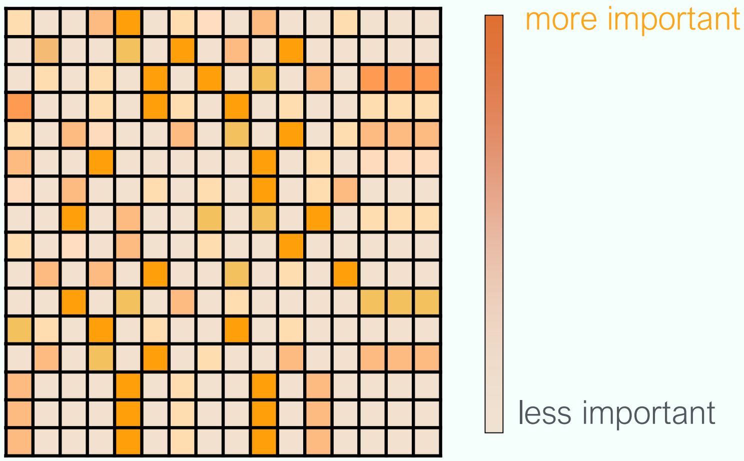


A GPU-friendly algorithm to compute heavy entries and minimize I/O

Algorithmic Acceleration

Finding Heavy contributions in practice

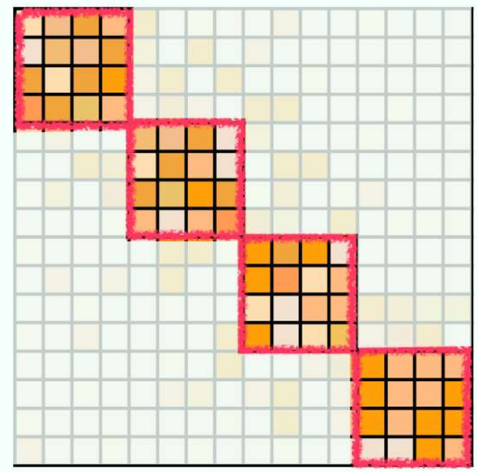
$n \times n$



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

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

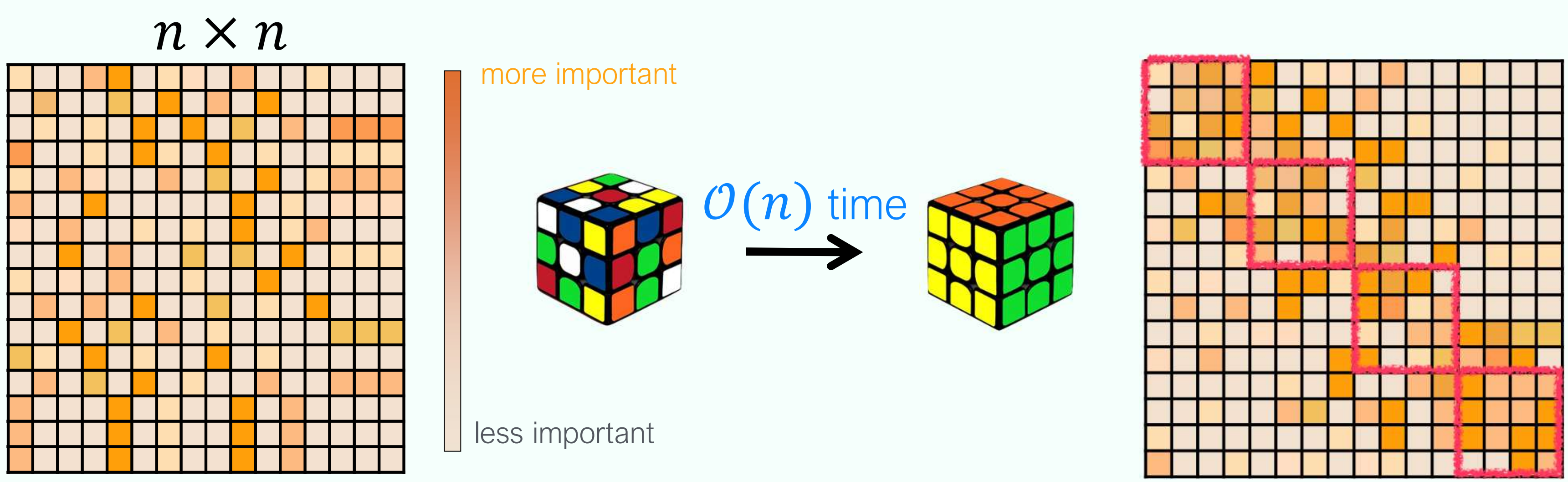
$\approx$



A Permutation algorithm that gathers heavy entries around the diagonal

Algorithmic Acceleration

Self Attention



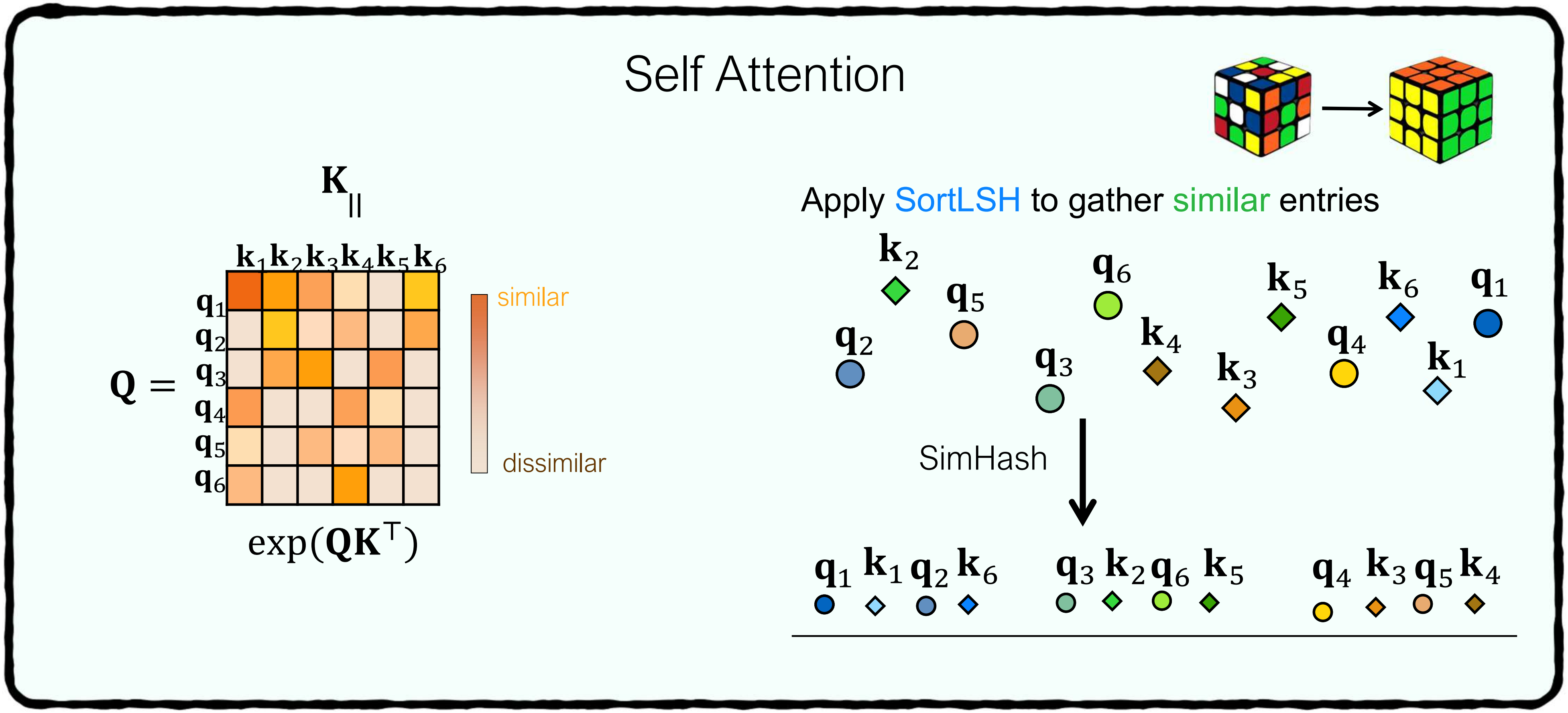
$\mathbf{A} = \exp(\mathbf{QK}^T)$

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

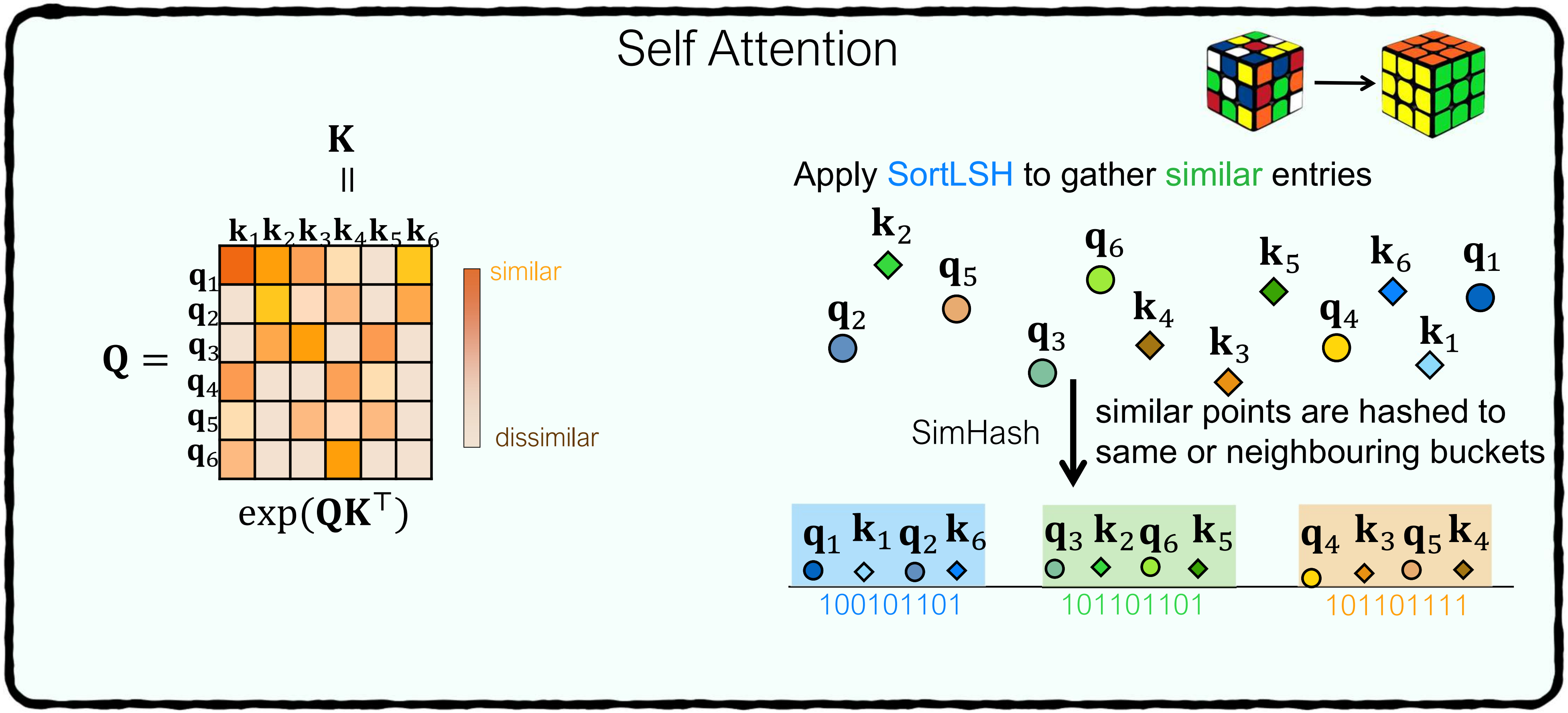
How can Heavy entries be gathered?

⇒ SortLSH (Locality Sensitive Hashing)

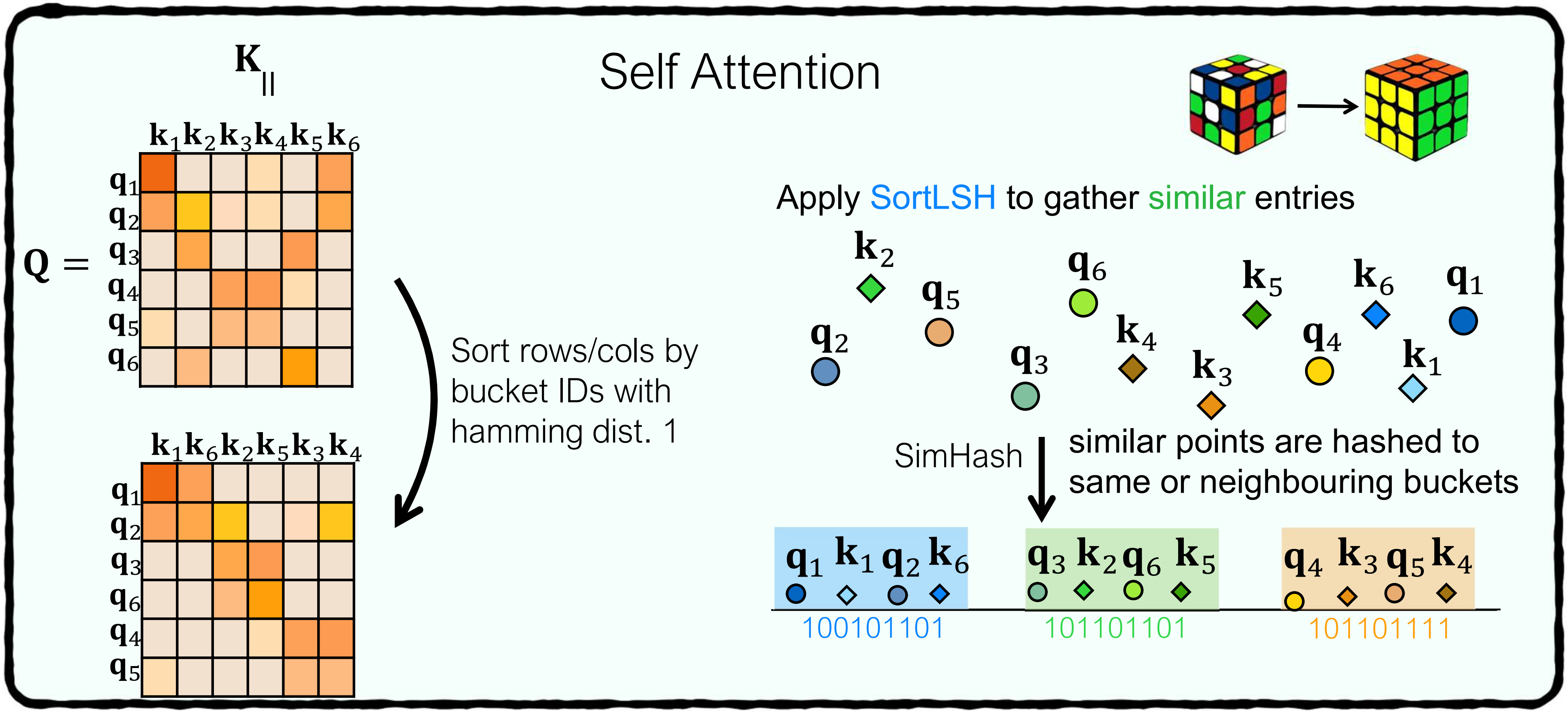
Algorithmic Acceleration



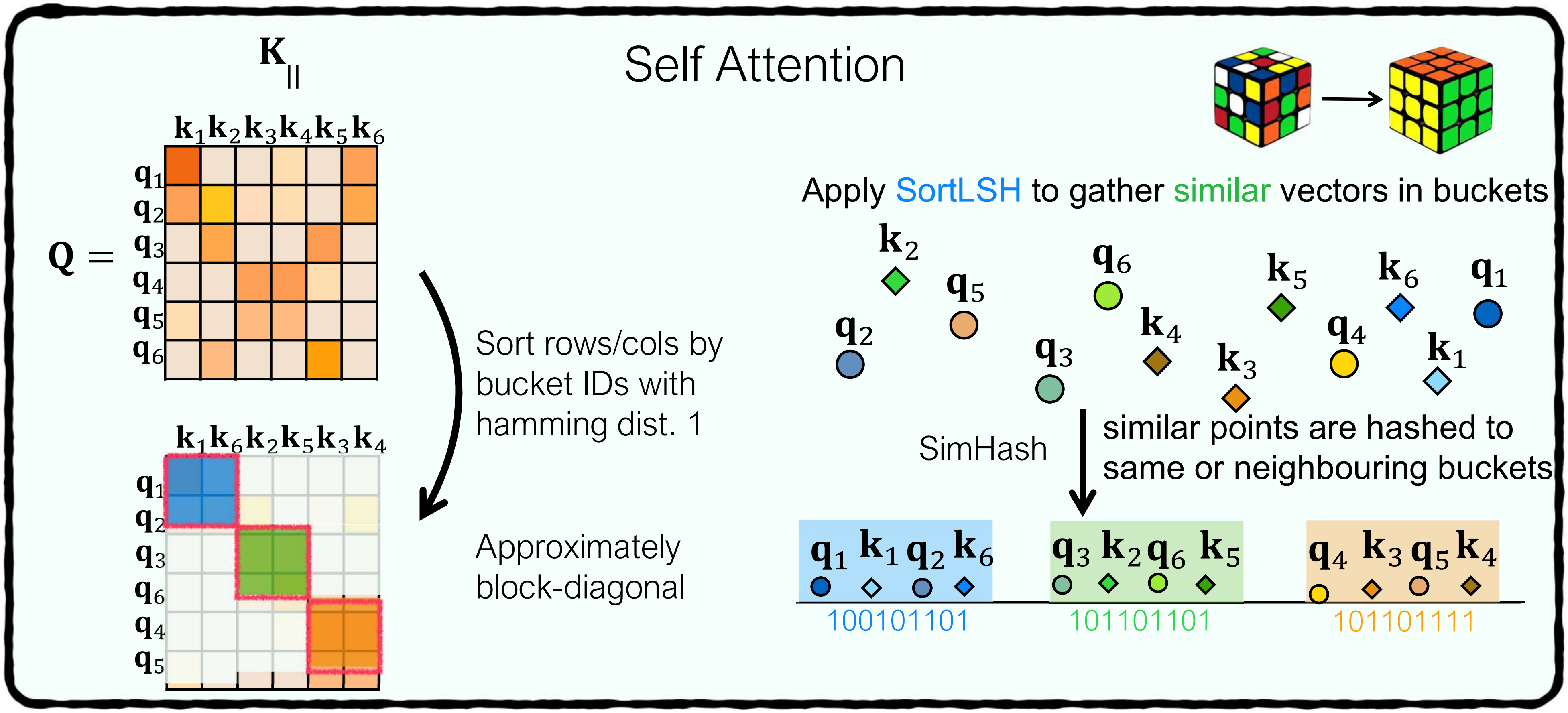
Algorithmic Acceleration



Algorithmic Acceleration

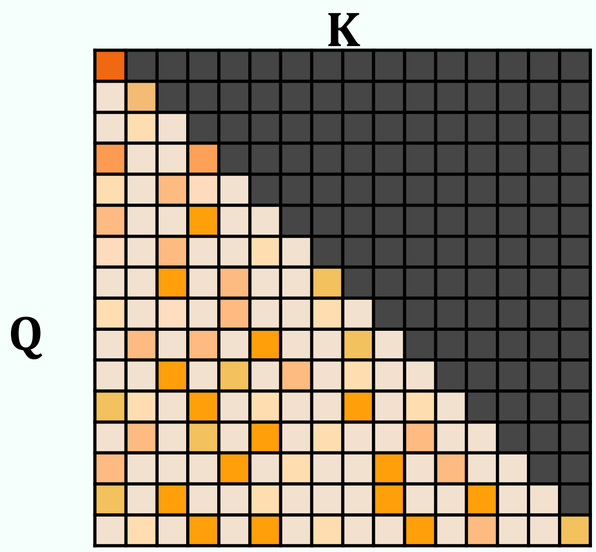


Algorithmic Acceleration



Algorithmic Acceleration

Causal Masking

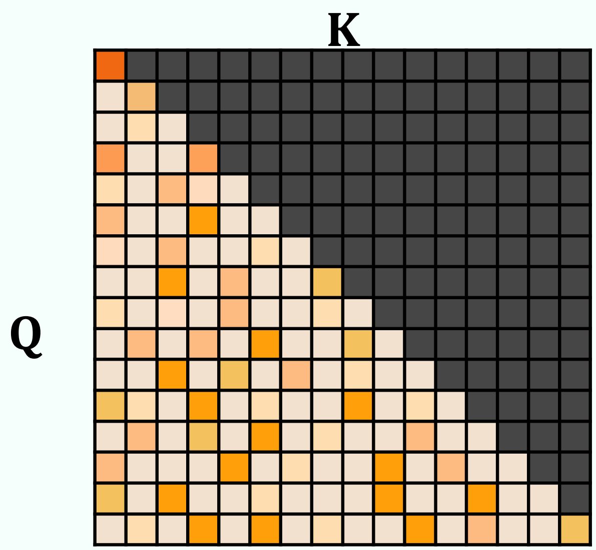


$$\exp(\mathbf{Q}\mathbf{K}^\top) \odot \mathbf{M}_{causal}$$

Output embeddings only depend on input embeddings in the past

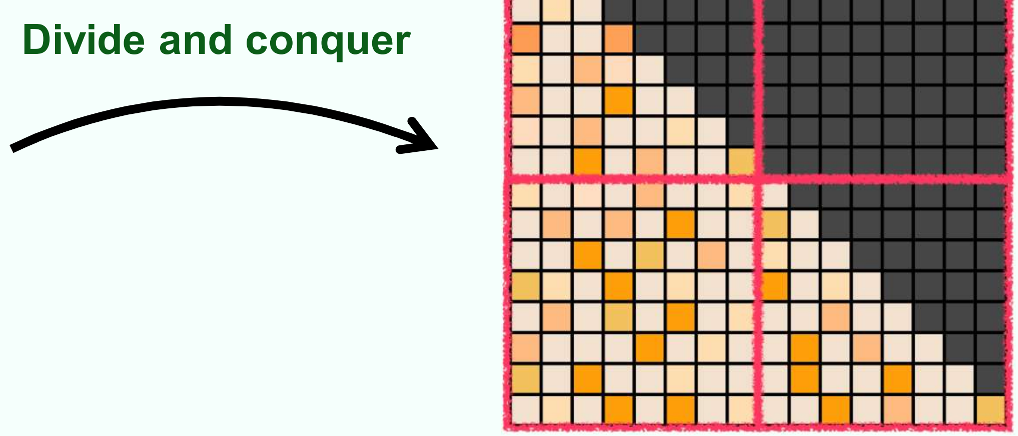
Algorithmic Acceleration

Causal Masking



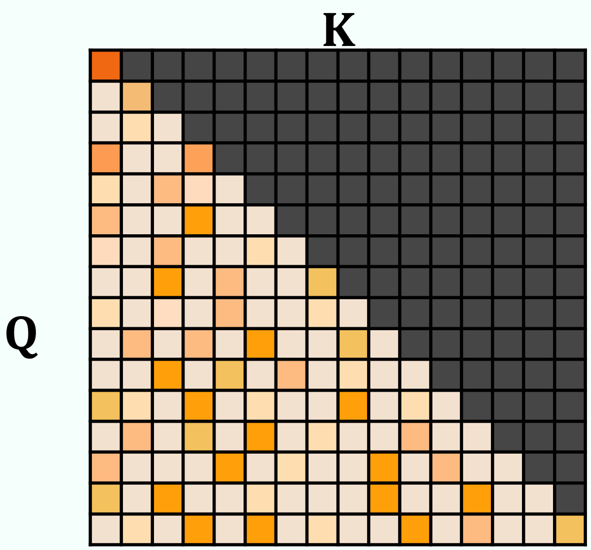
$$\exp(\mathbf{QK}^\top) \odot \mathbf{M}_{causal}$$

Output embeddings only depend on input embeddings in the past



Algorithmic Acceleration

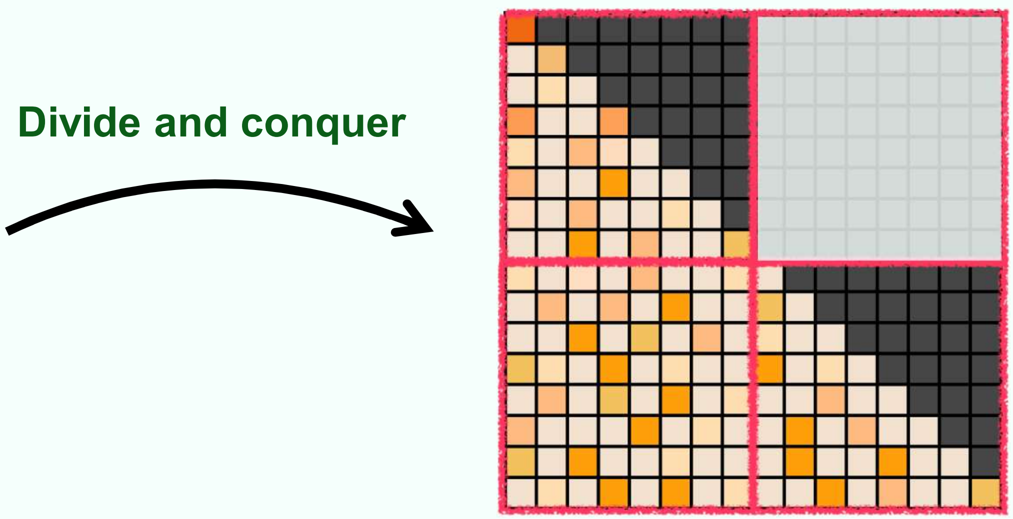
Causal Masking



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

Output embeddings only depend on input embeddings in the past

Divide and conquer



Algorithmic Acceleration

+

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

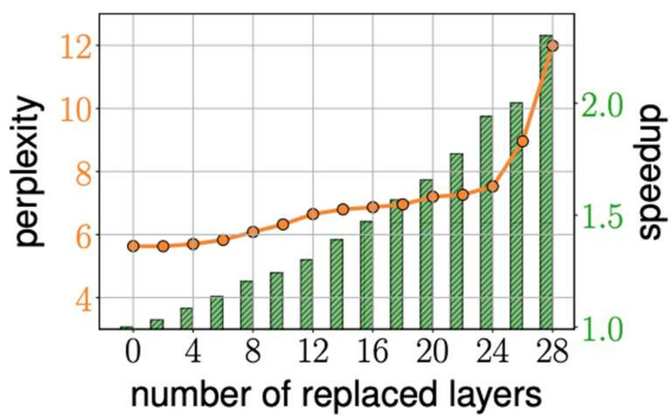
Vahab Mirrokni
Google Research
mirrokni@google.com

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