Oblivious Sketching of High-Degree Polynomial Kernels

Thomas Ahle
ITU

Michael Kapralov
EPFL

Jakob Knudsen
Univ. of Copenhagen

Rasmus Pagh
ITU

Ameya Velingker
Google AI

David Woodruff
CMU

Amir Zandieh
EPFL

Oblivious Sketching of High-Degree Polynomial Kernels

Thomas Ahle
ITU

Michael Kapralov
EPFL

Jakob Knudsen
Univ. of Copenhagen

Rasmus Pagh
ITU

Ameya Velingker
Google AI

David Woodruff
CMU

Amir Zandieh
EPFL

IT-UNIVERSITETET I KØBENHAVN

Oblivious Sketching of High-Degree Polynomial Kernels

Thomas Ahle
ITU

Michael Kapralov
EPFL

Jakob Knudsen
Univ. of Copenhagen

Rasmus Pagh
ITU

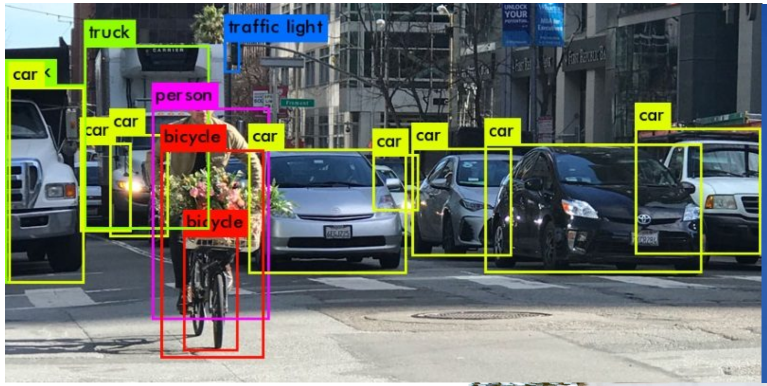
Ameya Velingker
Google AI

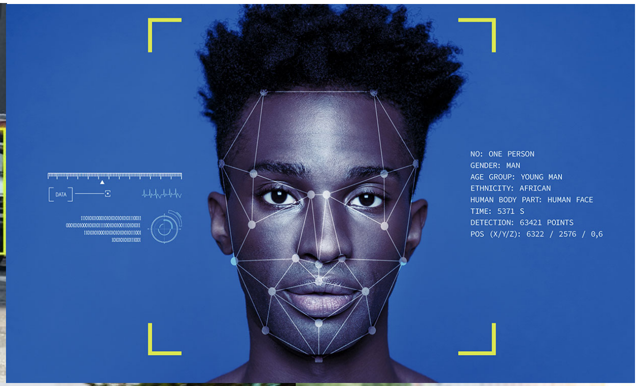
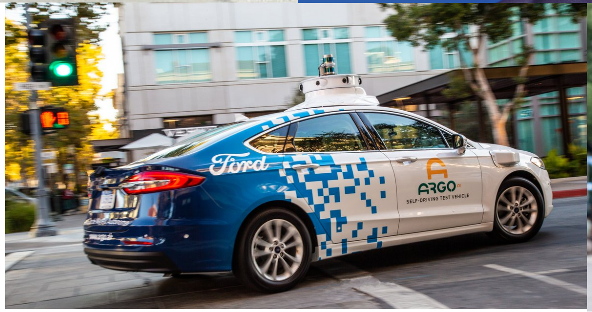
David Woodruff
CMU

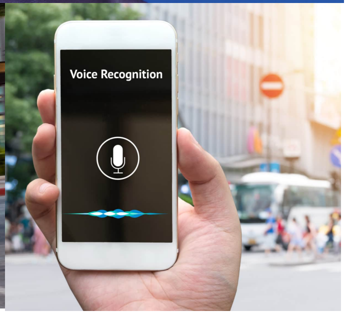
Amir Zandieh
EPFL

Kernel Methods

- Widely used in *kernel-based* learning, statistics, and control
- Classical machine learning tool with real-world applications





Real-World Applications of Kernel Methods

- **Hyperparameter tuning of deep neural networks:** e.g. Google Vizier
- **Multi-Armed Bandit Optimization** [Srinivas, Krause, Kakade, Seeger' 09]
- **Neural Tangent Kernel:** The evolution of a neural network during training can be described by kernel methods [Jacot, Gabriel, Hongler'18]

Kernel Methods

- Learn a nonlinear function $f: \mathcal{R}^d \rightarrow \mathcal{R}$ from noisy samples

$$\gamma_i = f(x_i) + \epsilon_i \text{ for } i = 1,2, \ldots n$$

  - ϵ_i are iid Gaussian noise with zero mean and variance λ
- **Kernel Ridge Regression** is a simple and yet powerful solution

Kernel Methods

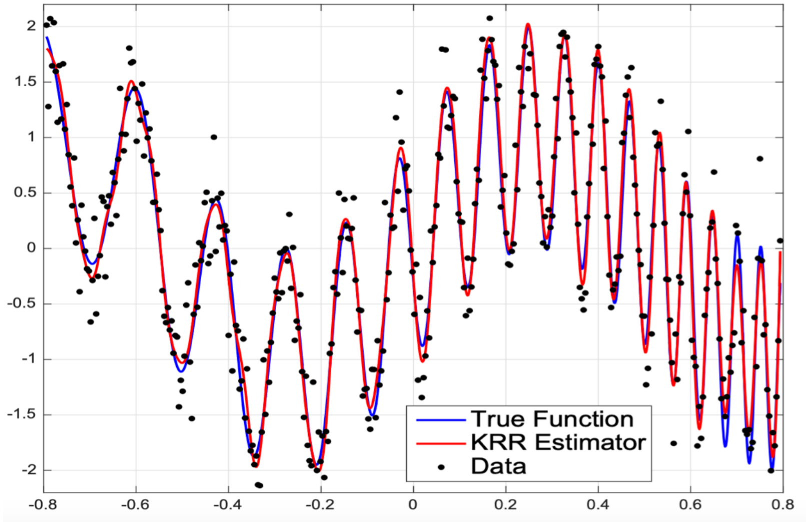
- Learn a nonlinear function $f: \mathcal{R}^d \to \mathcal{R}$ from noisy samples

$$y_i = f(x_i) + \epsilon_i \text{ for } i = 1,2, \ldots n$$

  - ϵ_i are iid Gaussian noise with zero mean and variance λ
- **Kernel Ridge Regression** is a simple and yet powerful solution
  - If $f(\cdot)$ is a GP with covariance $k: \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$, then the optimal estimator is,

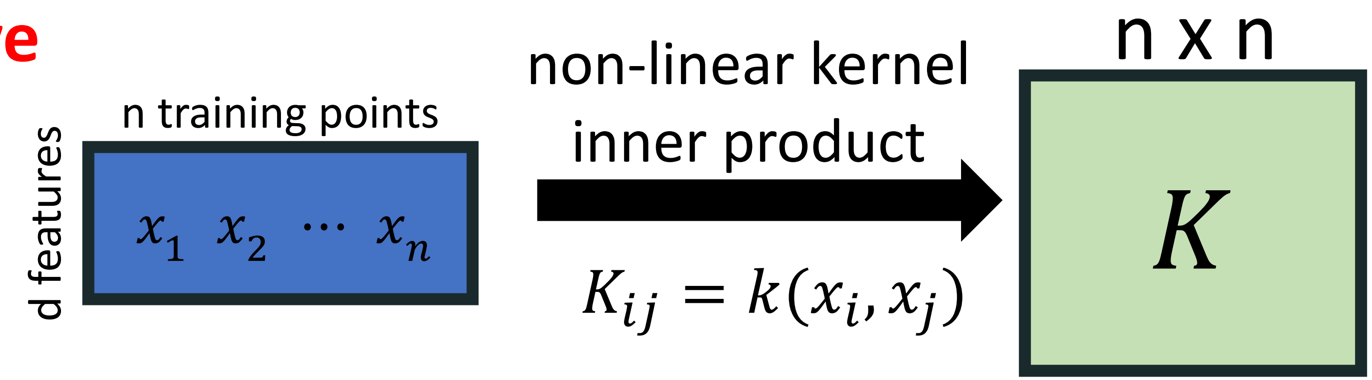
$$\tilde{f}(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$$

$$\alpha = \text{argmin}_{\beta \in \mathbb{R}^n} \|\boldsymbol{K}\beta - y\|_2^2 + \lambda \beta^\top \boldsymbol{K} \beta$$



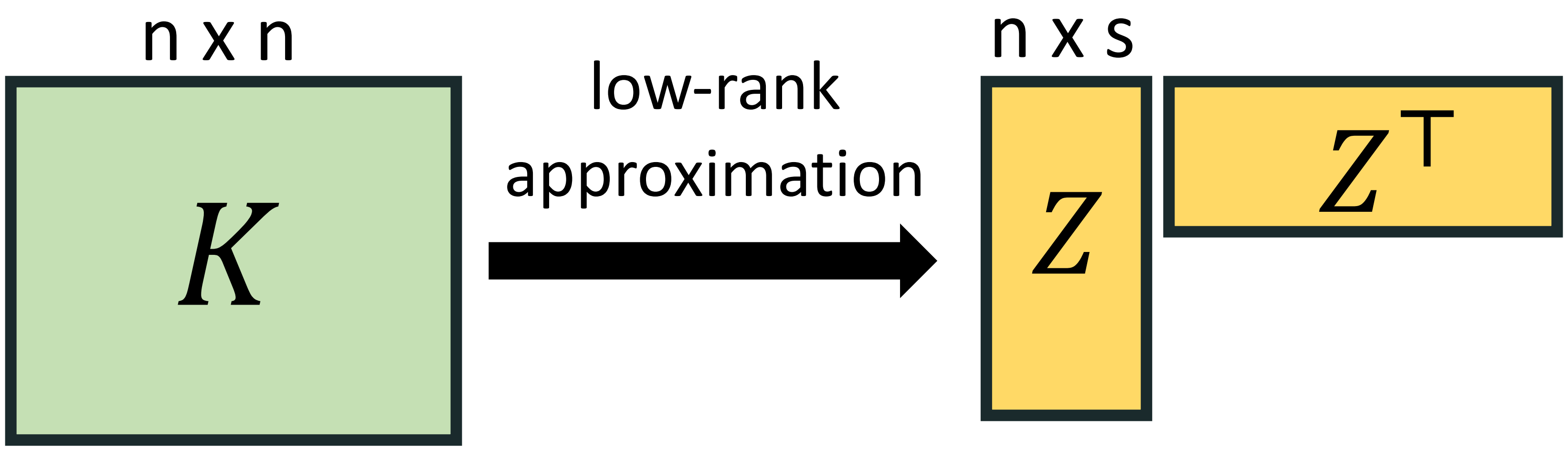
Kernel Method

- **Kernel methods are expensive**



- Computing all kernel entries takes $n \cdot nnz(X) + n^2$ time
- Even writing it down takes n^2 time and memory
- A single iteration of a linear system solver takes n^2 time
- For $n = 100\,000$, K has 10 billion entries. Takes 80 GB of storage

Classical Solution: **Dimensionality Reduction**



- Storing Z uses $O(ns)$ space and computing $ZZ^\top\alpha$ takes $O(ns)$ time.
- Orthogonalization, eigen-decomposition, and pseudo-inversion of $ZZ^\top$ all take just $O(ns^2)$ time.

Efficient Low-Rank Approximation?

- Direct eigen decomposition, or even approximation via Krylov subspace methods are out of question since they at least require fully forming K

Efficient Low-Rank Approximation?

- Direct eigen decomposition, or even approximation via Krylov subspace methods are out of question since they at least require fully forming K

- **Sketching:** a powerful approach to speeding up matrix problems
- Our approach: design a sketching solution for kernel low-rank approximation

Feature Space Mapping

- Any positive definite kernel $k\colon \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$ defines a lifting $\varphi\colon \mathcal{R}^d \to \mathcal{R}^D$ such that for all $x, y \in \mathcal{R}^d$

$$k(x, y) = \varphi(x)^\top \varphi(y)$$

- The kernel computes the inner product between the lifted data points

Feature Space Mapping

- Any positive definite kernel $k: \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$ defines a lifting $\varphi: \mathcal{R}^d \rightarrow \mathcal{R}^D$ such that for all $x, y \in \mathcal{R}^d$

$$k(x, y) = \varphi(x)^\top \varphi(y)$$

- The kernel computes the inner product between the lifted data points

$$K = \boldsymbol{\phi}^\top \boldsymbol{\phi},$$

where $\boldsymbol{\phi}$ is a $D \times n$ matrix whose i^{th} column is the projection of x_i into the feature space $\varphi(x_i)$

Sketching the Feature Space

- **Sketch** the feature space

$$K = \boldsymbol{\phi}^\top \boldsymbol{\phi} \approx \boldsymbol{\phi}^\top \Pi^\top \Pi \boldsymbol{\phi}$$

Sketching the Feature Space

- **Sketch** the feature space

$$K = \boldsymbol{\phi}^\top \boldsymbol{\phi} \approx \boldsymbol{\phi}^\top \Pi^\top \Pi \boldsymbol{\phi}$$

- Challenge: forming the feature matrix $\boldsymbol{\phi}$ explicitly is expensive as the feature space is typically high-dimensional (even infinite-dimensional)

Sketching the Feature Space

- **Sketch** the feature space

$$K = \boldsymbol{\phi}^\top \boldsymbol{\phi} \approx \boldsymbol{\phi}^\top \Pi^\top \Pi \boldsymbol{\phi}$$

- Challenge: forming the feature matrix $\boldsymbol{\phi}$ explicitly is expensive as the feature space is typically high-dimensional (even infinite-dimensional)
- Goal: Design a sketch matrix $\Pi \in \mathcal{R}^{s \times D}$ such that $\Pi \cdot \varphi(x)$ is computable without needing to explicitly form $\varphi(x)$

Kernel Sketching techniques

- The most popular method for kernel sketching is the **Fourier Features Method** of Rahimi & Recht (Test of Time Award winner at NeurIPS'17)
- Works for shift invariant kernels, such as Gaussian kernel

$$\varphi(x)_\xi = e^{-2\pi i \xi^\top x} \text{ for } \xi \in \mathcal{R}^d$$

- Π: Sampling matrix that samples frequencies ξ from the pdf $\hat{k}(\xi)$

Kernel Sketching techniques

- The most popular method for kernel sketching is the **Fourier Features Method** of Rahimi & Recht (Test of Time Award winner at NeurIPS'17)
- Works for shift invariant kernels, such as Gaussian kernel

$$\varphi(x)_\xi = e^{-2\pi i \xi^\top x} \text{ for } \xi \in \mathcal{R}^d$$

- Π: Sampling matrix that samples frequencies ξ from the pdf $\hat{k}(\xi)$
- Avron, Kapralov, Musco, Musco, Velingker, **Z.**' 17: Tight bounds to get spectral approximation guarantee + Modified Fourier Sampling with optimal number of samples for Gaussian kernel in constant dimension

Kernel Sketching techniques

- The most popular method for kernel sketching is the **Fourier Features Method** of Rahimi & Recht (Test of Time Award winner at NeurIPS'17)
- Works for shift invariant kernels, such as Gaussian kernel

$$\varphi(x)_\xi = e^{-2\pi i \xi^\top x} \text{ for } \xi \in \mathcal{R}^d$$

- Π: Sampling matrix that samples frequencies ξ from the pdf $\hat{k}(\xi)$
- Avron, Kapralov, Musco, Musco, Velingker, **Z.**' 17: Tight bounds to get spectral approximation guarantee + Modified Fourier Sampling with optimal number of samples for Gaussian kernel in constant dimension
- Avron, Kapralov, Musco, Musco, Velingker, **Z.**' 19: Optimal sampling strategy for Sinc kernel in dimension 1

Kernel Sketching techniques

- The most popular method for kernel sketching is the **Fourier Features Method** of Rahimi & Recht (Test of Time Award winner at NeurIPS'17)
- Works for shift invariant kernels, such as Gaussian kernel

$$\varphi(x)_\xi = e^{-2\pi i \xi^\top x} \text{ for } \xi \in \mathcal{R}^d$$

- Π: Samp … pdf $\hat{k}(\xi)$
- Avron, Ka … s to get spectral … ampling with optimal number of samples for Gaussian kernel in constant dimension
- Avron, Kapralov, Musco, Musco, Velingker, **Z.**' 19: Optimal sampling strategy for Sinc kernel in dimension 1

Works only for shift invariant kernels and constant dimensional datasets

Polynomial Kernel

- In this work we focus on the important case of **Polynomial Kernel**

$$k(x, y) = (x^\top y)^q$$

- The lifting function for this kernel is $\varphi(x) = x^{\otimes q}$,

where $\varphi(x) \in \mathcal{R}^{d^q}$ is defined as $\varphi(x)_{i_1, i_2, \cdots i_q} = x_{i_1} x_{i_2} \cdots x_{i_q}$ for $i_1, i_2, \cdots i_q \in \{1,2, \cdots d\}$

Polynomial Kernel

- In this work we focus on the important case of **Polynomial Kernel**

$$k(x, y) = (x^\top y)^q$$

- The lifting function for this kernel is $\varphi(x) = x^{\otimes q}$,

where $\varphi(x) \in \mathcal{R}^{d^q}$ is defined as $\varphi(x)_{i_1, i_2, \cdots i_q} = x_{i_1} x_{i_2} \cdots x_{i_q}$ for $i_1, i_2, \cdots i_q \in \{1, 2, \cdots d\}$

- Goal: design a linear sketch $\Pi \in \mathcal{R}^{s \times d^q}$ such that $\Pi x^{\otimes q}$ is efficiently computable without needing to form $x^{\otimes q}$ explicitly

Key Properties of Sketch

- **Approximate Matrix Product:** for every matrices $A, B \in \mathcal{R}^{d^q \times n}$ whp
$$\|A^\top \Pi^\top \Pi B - A^\top B\|_F \leq \epsilon \|A\|_F \|B\|_F$$
- **Oblivious Subspace Embedding:** for every $\lambda > 0$ and every matrix $A \in \mathcal{R}^{d^q \times n}$ whp
$$\frac{A^\top A + \lambda I}{1 + \epsilon} \preccurlyeq A^\top \Pi^\top \Pi A + \lambda I \preccurlyeq \frac{A^\top A + \lambda I}{1 - \epsilon}$$
  - Want: target dimension at most **statistical dimension** $\boldsymbol{tr}(A^\top A(A^\top A + \lambda I)^{-1})$

Prior Work: TensorSketch

- Originally introduced by Pagh and Pham [KDD 2013], [TOCT 2013]

Prior Work: TensorSketch

- Originally introduced by Pagh and Pham [KDD 2013], [TOCT 2013]
- Avron, Nguyen, and Woodruff [NeurIPS 2014] proved:

1. Satisfies **Approximate Matrix Product** with probability 9/10 if target dimension $s = \Omega\left(\frac{3^q}{\epsilon^2}\right)$
2. Satisfies **Oblivious Subspace Embedding** with probability 9/10 if target dimension $s = \Omega\left(\frac{3^q}{\epsilon^2} \cdot s_\lambda^2\right)$
3. Time to sketch the tensor $x^{\otimes q}$ is $\tilde{O}\big(\mathrm{q}s + \mathrm{q} \cdot nnz(x)\big)$

Statistical Dimension $s_\lambda := \boldsymbol{tr}(K(K + \lambda I)^{-1})$

Prior Work: TensorSketch

- Originally introduced by Pagh and Pham [KDD 2013], [TOCT 2013]
- Avron, Nguyen, and Woodruff [Neu

1. Satisfies **Approximate Matrix Pr** target dimension $s = \Omega\left(\frac{\mathbf{3^q}}{\epsilon^2}\right)$
2. Satisfies **Oblivious Subspace Em** if target dimension $s = \Omega\left(\frac{\mathbf{3^q}}{\epsilon^2} \cdot s_\lambda^2\right)$
3. Time to sketch the tensor $x^{\otimes q}$ is $\tilde{O}\left(\mathrm{q}s + \mathrm{q} \cdot nnz(x)\right)$

Main contribution:
improve the exponential dependence on **q** to polynomial

Statistical Dimension $s_\lambda := \boldsymbol{tr}\left(K(K + \lambda I)^{-1}\right)$

Prior Work: TensorSketch

- Originally introduced by Pagh and Pham [KDD 2013], [TOCT 2013]
- Avron, Nguyen, and Woodruff [NeurIPS 2014] proved:

1. Satisfies **Approximate Matrix Pro**… …rget dimension $s = \Omega\left(\frac{3^q}{\epsilon^2}\right)$
2. Satisfies **Oblivious Subspace Embedding** with probability 9/10 if target dimension $s = \Omega\left(\frac{3^q}{\epsilon^2} \cdot \boldsymbol{s_\lambda^2}\right)$
3. Time to sketch the tensor $x^{\otimes q}$ is $\tilde{O}\left(\mathrm{q}s + \mathrm{q} \cdot nnz(x)\right)$

Contribution 2: improve the quadratic dependence on s_λ to linear

Statistical Dimension $s_\lambda := \boldsymbol{tr}\left(K(K+\lambda I)^{-1}\right)$

Prior Work: TensorSketch

- Originally introduced by Pagh and Pham [KDD 2013], [TOCT 2013]
- Avron, Nguyen, and Woodruff [NeurIPS 2014] proved:

1. Satisf **uct** with probability **9/10** if
   targe
2. Satisf **dding** with probability **9/10** if
   target dimension $s = \Omega\left(\frac{\ }{\epsilon^2} \cdot s_\lambda^2\right)$
3. Time to sketch the tensor $x^{\otimes q}$ is $\tilde{O}(\mathrm{q}s + \mathrm{q} \cdot nnz(x))$

Contribution 3: improve the success probability to $1 - \frac{1}{\mathrm{poly}(n)}$

Statistical Dimension $s_\lambda := \boldsymbol{tr}(K(K + \lambda I)^{-1})$

Main Results

- **Theorem 1:** there exists a distribution on linear sketches $\Pi \in \mathcal{R}^{s \times d^q}$ such that:

1. If target dimension $s = \Omega\left(\frac{q}{\epsilon^2}\right)$ then Π has the **Approximate Matrix Product** property with probability 9/10
2. If target dimension $s = \Omega\left(\frac{q}{\epsilon^2} \cdot s_\lambda^2\right)$ then Π is an **Oblivious Subspace Embedding** with probability 9/10
3. For any $x \in \mathcal{R}^d$, $\Pi \cdot x^{\otimes q}$ is computable in time $\tilde{O}(\mathrm{q}s + \mathrm{q} \cdot nnz(x))$

Main Results

- **Theorem 2:** there exists a distribution on linear sketches $\Pi \in \mathcal{R}^{s \times d^q}$ such that:

1. If target dimension $s = \tilde{\Omega}\left(\frac{q^4}{\epsilon^2}\right)$ then Π has the **Approximate Matrix Product** property with **high probability**
2. If target dimension $s = \tilde{\Omega}\left(\frac{q^4}{\epsilon^2} \cdot s_\lambda\right)$ then Π is an **Oblivious Subspace Embedding** with **high probability**
3. For any vector $x \in \mathcal{R}^d$, the product $\Pi \cdot x^{\otimes q}$ is computable in time $\tilde{O}\left(qs + q^5 \epsilon^{-2} nnz(x)\right)$

Review: TensorSketch

$$\Pi x^q = \mathcal{F}^{-1}\left[(\mathcal{F}C_1 x) \circ (\mathcal{F}C_2 x) \circ \cdots \circ \left(\mathcal{F}C_q x\right)\right]$$

- $\mathcal{F}$ is the Fourier transform matrix and $C_1, C_2, \cdots C_q \in \mathcal{R}^{s \times d}$ are independent copies of CountSketch

Review: TensorSketch

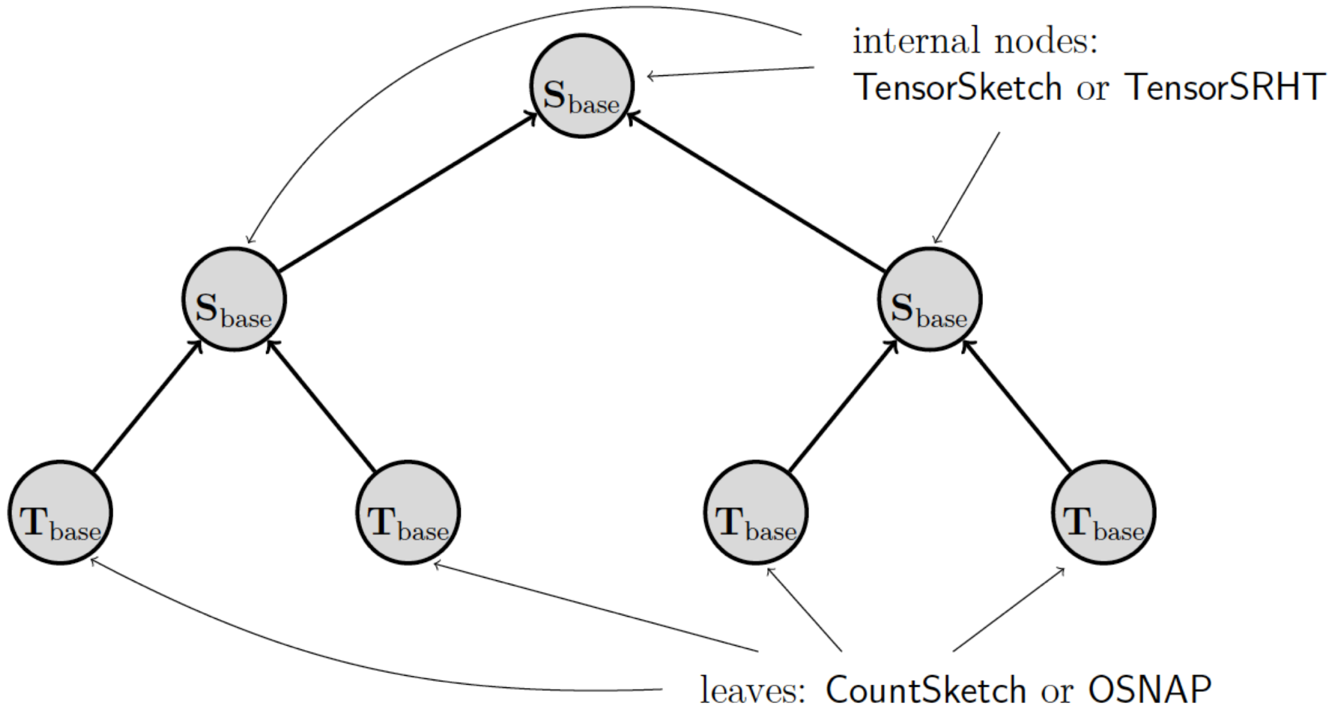
$$\Pi x^q = \mathcal{F}^{-1}\left[(\mathcal{F}C_1 x) \circ (\mathcal{F}C_2 x) \circ \cdots \circ \left(\mathcal{F}C_q x\right)\right]$$

- $\mathcal{F}$ is the Fourier transform matrix and $C_1, C_2, \cdots C_q \in \mathcal{R}^{s \times d}$ are independent copies of CountSketch
- The second moment of this estimator for $x = \{1\}^d$

$$\mathbb{E}\left[\left\|\Pi x^{\otimes q}\right\|_2^4\right] \geq \frac{\mathbf{3^q}}{2s^2}\left\|x^{\otimes q}\right\|_2^4$$

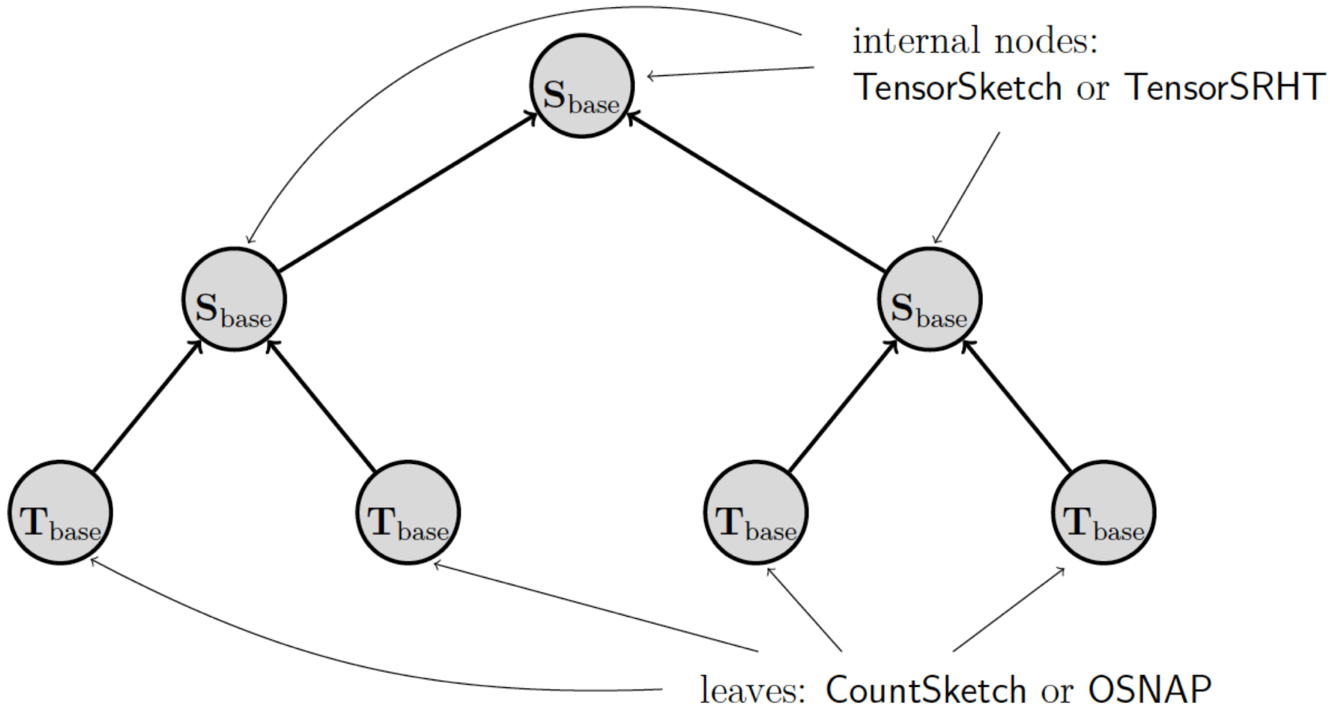
Our Sketch Construction

- Every node is an independent instance of some base sketch
- Leaves: sketch the input vector
- Internal nodes: sketch the tensor product of their children



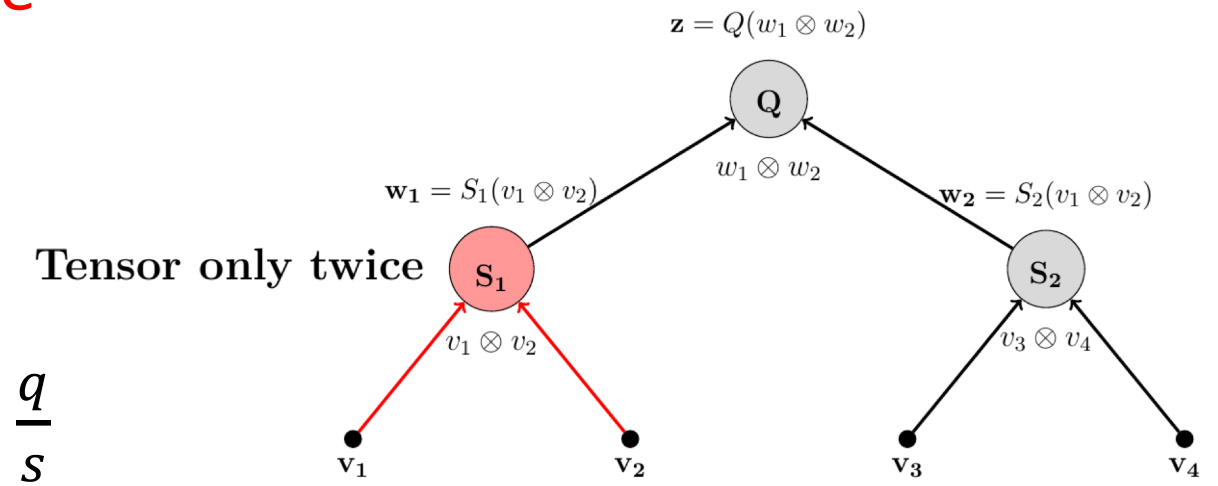
Our Sketch Construction

- Every leaf is a sketch that runs in input sparsity time
- Internal nodes support fast application time



Our Sketch Construction

- Intermediate nodes tensor only twice
- Loss in internal nodes is only $\frac{3^2}{s}$
- Number of such nodes is $O(q)$
- Hence $\frac{Var(\|\Pi x\|_2^2)}{\|x\|_2^4} \approx \left(1 + \frac{1}{s}\right)^q - 1 \approx \frac{q}{s}$
- In particular, there is no exponential dependence on q



s: target dimension of intermediate sketches

OSE for **Gaussian Kernel**

- The polynomial dependence of our sketch on the degree q leads to significant improvements on sketching the **Gaussian kernel** in high-d

OSE for **Gaussian Kernel**

- The polynomial dependence of our sketch on the degree q leads to significant improvements on sketching the **Gaussian kernel** in high-d

Prior work

- **Fast multipole method** of Greengard and Rokhlin: suffers from curse of dimensionality $(\log n)^d$
- **Fourier features** method of Rahimi & Recht: significantly suboptimal runtime of $\frac{n}{\lambda} \cdot nnz(X)$
- **Modified Fourier features** of Avron, Kapralov, Musco, Musco, Velingker, **Z**' 17: Optimal for constant dimensions d but does not apply to high dimensional data

OSE for **Gaussian Kernel**

- **Theorem 3:** for any dataset $x_1, x_2, \cdots x_n \in \mathbb{R}^d$ such that $\|x_i\|_2^2 \leq r$ if $K \in \mathbb{R}^{n\times n}$ is the Gaussian kernel matrix defined as $K_{i,j} = e^{-\|x_i - x_j\|_2^2}$ there exists an algorithm that computes $Z \in \mathbb{R}^{n\times s}$ such that:

1. If target dimension $s = \tilde{\Omega}\left(\frac{r^5}{\epsilon^2} \cdot s_\lambda\right)$ then $\mathrm{ZZ}^\top$ is an **Oblivious Subspace Embedding** for kernel K with **high probability**
2. The runtime to compute Z is $\tilde{O}\left(r^6\epsilon^{-2}ns_\lambda + r^6\epsilon^{-2}nnz(X)\right)$