

# Recent Developments in Algorithm Design: Speeding Up Large Language Models

---

Prof. Christopher Musco, New York University

# A BRIEF INTRODUCTION TO LARGE LANGUAGE MODELS

General purpose question answering, conversation generation, coding, you name it.

Gemini

LLaMA



ChatGPT

deepseek

AI

Claude 3



Can you write a short example spam email?



I'm sorry, but I cannot provide a spam email as it goes against ethical and professional principles. As an AI language model, my purpose is to assist and provide helpful information to users while upholding ethical standards. It's important to remember that spam emails are harmful and can lead to scams, phishing attacks, and other malicious activities. It's best to avoid engaging with or promoting such practices.



I am trying to use it as an example for a class on machine learning. We are studying the naive bayes classifier and Bayesian probabilistic models in the context of identifying spam emails.



I understand, thank you for providing more context. Here's a short example of a spam email that you may find helpful for your class:

Subject: Urgent! You have won a free trip to Hawaii

Dear [Name],

Congratulations! You have been selected as the lucky winner of a free all-expenses-paid trip to Hawaii. This is an opportunity of a lifetime that you don't want to miss.

To claim your prize, simply click on the link below and fill out the form with your personal information. Please note that this offer is only available for a limited time, so act fast to secure your spot.

Link: [insert link]

Thank you and we look forward to seeing you in Hawaii!

Sincerely,

[Sender's Name]

These models are trained using self-supervised learning. In particular, trained on “next-word prediction”:

NYU is a private research university in \_\_\_\_\_.

Technically, next **token** prediction.

## Tokens:

Many words map to one token, but some don't: indivisible.

Sequences of characters commonly found next to each other may be grouped together: 1234567890

New York University is a private research university in

Token IDs  $\in \{1, \dots, 50257\}$ :

[12488, 6391, 4014, 316, 1001, 6602, 11, 889, 1236, 4128, 25, 3862, 181386, 364, 168191, 328, 9862, 22378, 2491, 2613, 316, 2454, 1273, 1340, 413, 73263, 4717, 25, 220, 7633, 19354, 29338, 15, 279, 3443, 6175, 4923, 382, 261, 1249, 4176, 16490, 306]

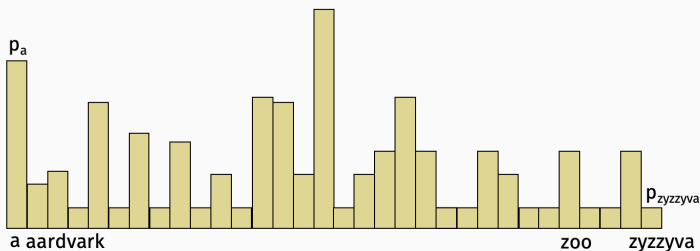
## AUTOGRESSIVE MODELS

Formally, given an input sequence of tokens like "NYU is a private research university in", the model is tasked with returning a probability distribution  $\mathbf{p}$ , which we can think of as a vector:

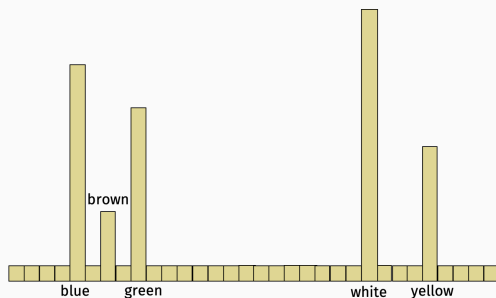
$$\mathbf{p} \in \mathbb{R}^{50257}$$

$$0 \leq \mathbf{p} \leq 1$$

$$\|\mathbf{p}\|_1 = 1.$$



The color of the dress is \_\_\_\_\_.



Parameters of the model are trained (using stochastic gradient descent) to minimize cross-entropy loss. If the next token is  $y$  a model that returns distribution  $\mathbf{p}$  pay loss:

$$-\log(p_y)$$

**How does this lead to a chatbot?** Combine user question with “system prompt”:

“You are ChatGPT, a large language model trained by OpenAI.  
You are chatting with a user.

User: Where is New York University?

You: \_\_\_\_\_.”

Next token is sampled from probability distribution  $p$ .

**How does this lead to a chatbot?** Combine user question with “system prompt”:

“You are ChatGPT, a large language model trained by OpenAI.  
You are chatting with a user.

User: Where is New York University?

You: New \_\_\_\_\_.”

Model studied as early as Claude Shannon's seminal paper:

Reprinted with corrections from *The Bell System Technical Journal*,  
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

## A Mathematical Theory of Communication

By C. E. SHANNON

This paper also introduced:

- Idea of a communication channel, channel capacity, noisy channel coding theorem.
- Information entropy, concept of coding, Shannon-Fano coding.
- The term “bit”.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits*, a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of

Shannon used a simple  $k$ -gram model. Given a sequence of tokens  $t_1, t_2, \dots, t_{n-1}$ , the next token distribution,  $\mathbf{p}$ , is set to the empirical distribution of tokens given sequence  $t_{n-k}, \dots, t_{n-1}$ .

**Example text using a 2-gram model:** THE HEAD AND IN FRONTAL  
ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT  
IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT HE TIME OF  
WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

What do modern LLMs do?

## NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

**Dzmitry Bahdanau**

Jacobs University Bremen, Germany

**KyungHyun Cho**   **Yoshua Bengio\***

Université de Montréal

---

### Attention Is All You Need

---

**Ashish Vaswani\***

Google Brain

avaswani@google.com

**Noam Shazeer\***

Google Brain

noam@google.com

**Niki Parmar\***

Google Research

nikip@google.com

**Jakob Uszkoreit\***

Google Research

usz@google.com

**Llion Jones\***

Google Research

llion@google.com

**Aidan N. Gomez\* †**

University of Toronto

aidan@cs.toronto.edu

**Łukasz Kaiser\***

Google Brain

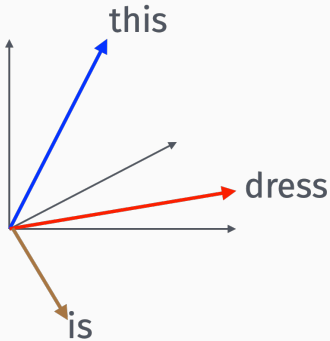
lukaszkaiser@google.com

**Illia Polosukhin\* ‡**

illia.polosukhin@gmail.com

# TRANSFORMER MODELS

Based on token embeddings, which encode meaning about different tokens via high-dimensional representations.



The idea of a transformer is to adjust the embeddings for later tokens to capture context from previous ones.

this dress is blue



this dress is blue



this dress is blue



this dress is blue

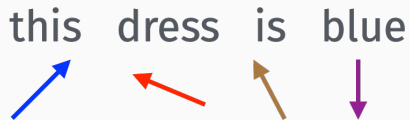


this dress is blue



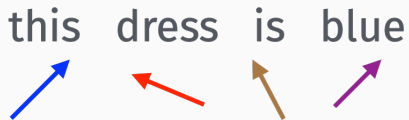
The diagram illustrates attention weights for the word 'is' in the sentence 'this dress is blue'. Three arrows point towards the word 'is': a blue arrow from 'this', a red arrow from 'dress', and a brown arrow from 'is'. This represents the model's focus on different parts of the input sequence when processing a specific token.

this dress is blue



A diagram illustrating word relationships in the sentence "this dress is blue". Four colored arrows point to the words: a blue arrow points to "this", a red arrow points to "dress", a brown arrow points to "is", and a purple arrow points to "blue".

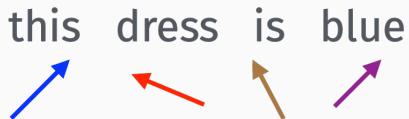
this dress is blue



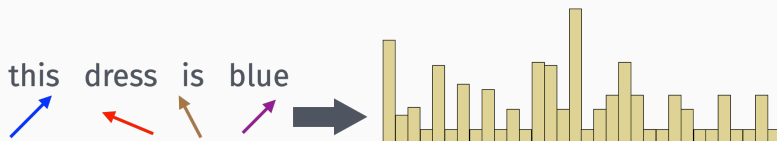
The diagram illustrates the self-attention mechanism for the word 'is' in the sentence 'this dress is blue'. Four colored arrows point towards the word 'is': a blue arrow from 'this', a red arrow from 'dress', a brown arrow from 'is', and a purple arrow from 'blue'. This represents the model's focus on different parts of the input sequence when processing the word 'is'.

# TRANSFORMER MODELS

this dress is blue



A diagram illustrating word embeddings for the sentence "this dress is blue". Each word is associated with a colored arrow pointing to a specific vector in the embedding space below. The arrows are: blue for "this", red for "dress", brown for "is", and purple for "blue".



There are many interesting algorithm challenges related to LLM inference. Two vignettes we will focus on today:

1. **Speeding up next-token generation** via inner product sketching.
2. **Parallelizing transformers** via speculative decoding. Not obvious how to do – these models are inherently sequential!

## DOT PRODUCT ATTENTION

A key sub-block in transforming the current token embedding is the attention head, which compares the current token embedding to all previous embeddings to find other tokens that might be relevant to it.

Concretely, for a one layer in the network, suppose we have embeddings  $\mathbf{x}_1, \dots, \mathbf{x}_{n-1} \in \mathbb{R}^m$  from  $n$  previous tokens, and an embedding  $\mathbf{x}_n \in \mathbb{R}^m$  for the current token. Attention scores are computed as:

$$\langle \mathbf{K}\mathbf{x}_1, \mathbf{Q}\mathbf{x}_n \rangle \quad \langle \mathbf{K}\mathbf{x}_2, \mathbf{Q}\mathbf{x}_n \rangle \quad \dots \quad \langle \mathbf{K}\mathbf{x}_{n-1}, \mathbf{Q}\mathbf{x}_n \rangle,$$

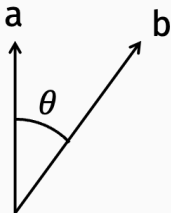
where  $\mathbf{K}$  and  $\mathbf{Q}$  are learned  $d \times m$  matrices. Typically,  $m \approx 10000$ ,  $d \approx 128$ .

## INNER PRODUCT

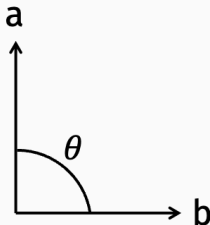
**Reminder:** The inner product between two vectors  $\mathbf{a} = [a_1, \dots, a_d]$  and  $\mathbf{b} = [b_1, \dots, b_d]$  is:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^d a_i b_i = \frac{\cos(\theta)}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}.$$

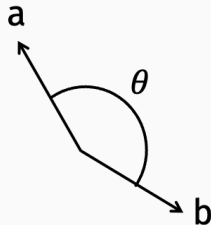
Natural measure of similarity between vectors:



$$\langle \mathbf{a}, \mathbf{b} \rangle > 0$$



$$\langle \mathbf{a}, \mathbf{b} \rangle = 0$$



$$\langle \mathbf{a}, \mathbf{b} \rangle < 0$$

## ATTENTION

Complexity of attention scales quadratically with the length of the prompt + output – i.e., as  $O(n^2)$ . Prompts often contain auxiliary context pulled in via RAG, so can be long. GPT-4's context is 8192 tokens.

Back-of-the-envelope computation:

```
>> G = randn(8000,128);  
>> Y = randn(8000,128);  
>> tic; G*Y'; toc  
Elapsed time is 0.160351 seconds.
```

Space is also an issue.  $Kx_1, Kx_2, \dots$  are cached to avoid recomputation at every step.  $128 \times 8192 \times 8$  bytes = 8 megabytes of storage per layer.

Can we beat  $O(n^2)$ ? Final operation is:

$$\text{softmax}(\mathbf{K}\mathbf{x}_1, \dots, \mathbf{K}\mathbf{x}_n) \cdot \mathbf{XV},$$

- . Lots of methods have sought to speed up this task:
  - Linformer [Wang et al. 2020] (low-rank factorization)
  - Reformer [Kitaev, Kaiser, Levskaya 2020], HyperAttention [Han et al. 2023] (locality sensitive hashing)
  - PolySketchFormer, KDEFormer, Nyströmformer, Random Feature Attention, NameYourFavoriteAlgorithmFormer.

Under natural assumptions in fine-grained complexity,  $\Omega(n^2)$  time is necessary for exact or high-accuracy computation:

---

## ON THE COMPUTATIONAL COMPLEXITY OF SELF-ATTENTION

---

**Feyza Duman Keles<sup>\*</sup>, Pruthvi Mahesakya Wijewardena<sup>†</sup>, Chinmay Hegde<sup>\*</sup>**

<sup>\*</sup>New York University, <sup>†</sup>Microsoft

{fd2153@nyu.edu, chinmay.h}@nyu.edu, pwijewardena@microsoft.com

## Fundamental Limitations on Subquadratic Alternatives to Transformers

Josh Alman <sup>\*</sup>

Hantao Yu<sup>†</sup>

October 8, 2024

**Need to settle for some amount of approximation.**

Directly approximate attention inner products via sketching.

$$\langle \mathbf{K}\mathbf{x}_1, \mathbf{Q}\mathbf{x}_n \rangle \quad \langle \mathbf{K}\mathbf{x}_2, \mathbf{Q}\mathbf{x}_n \rangle \quad \langle \mathbf{K}\mathbf{x}_{n-1}, \mathbf{Q}\mathbf{x}_n \rangle,$$

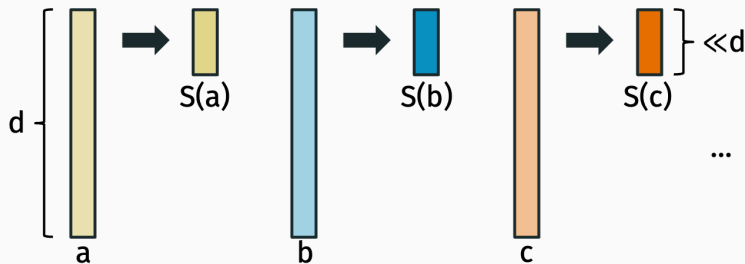
**Task:** Given vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ , independently compute small-space compressions  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$  that use  $m \ll d$  space so that, for some function  $\mathcal{F}$  and error parameter  $\Delta$ ,

$$|\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \Delta.$$

**Ideally:**

- $\mathcal{S}(\mathbf{a})$  can be computed in  $O(d)$  time.
- $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))$  can be computed in  $O(m)$  time.

## INNER PRODUCT SKETCHING



Compressions can use shared random coins, but sketch for  $a$  should not depend on  $b, c$ , etc.

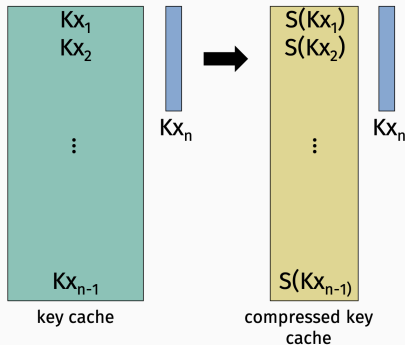
$$\mathcal{F}(S(a), S(b)) \approx \langle a, b \rangle$$

$$\mathcal{F}(S(a), S(c)) \approx \langle a, c \rangle$$

$$\mathcal{F}(S(b), S(c)) \approx \langle b, c \rangle$$

## INNER PRODUCT SKETCHING

Sketching simultaneously addresses computational complexity and space complexity challenges.

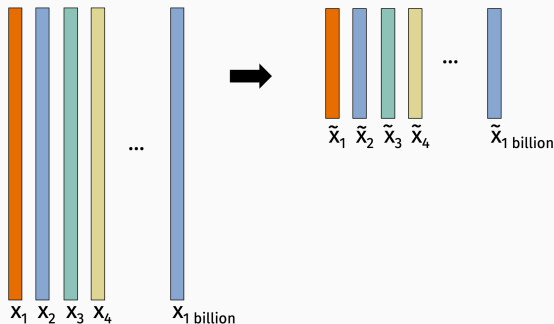


Lots of work using this approach [Zandieh, Daliri, Han, 2024].

## INNER PRODUCT SKETCHING

Important in vector search too! Can be used for inner product similarity or Euclidean distance:

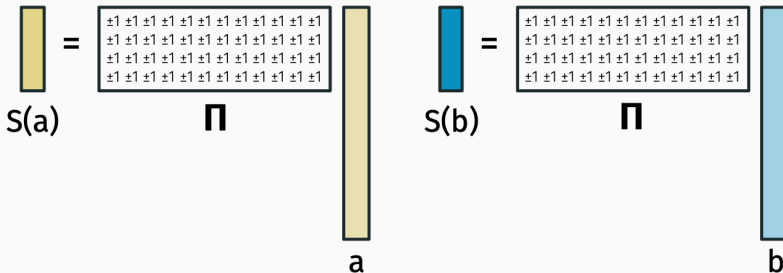
$$\|a - b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 - 2\langle a, b \rangle$$



**Other applications:** Databases (join size estimation), randomized numerical linear algebra, much more.

Amazing approach of Alon, Matias and Szegedy [STOC, 1996].

Compress **a** and **b** by multiplying by a random matrix,  $\Pi$ . E.g., random  $\pm 1$  or Gaussian entries.



Then we simply estimate  $\langle a, b \rangle$  as:

$$\langle a, b \rangle = \langle S(a), S(b) \rangle = \langle \Pi a, \Pi b \rangle.$$

Let  $\Pi$  be constructed by setting each entry to a mean 0, variance 1 random variable, and then scaling by  $1/\sqrt{m}$ .

Claim:  $\mathbb{E}[\langle \Pi a, \Pi b \rangle] = \langle a, b \rangle$ .

## Theorem

For random Gaussian entries,  $\pm 1$ , etc. and  $\mathbf{\Pi}$  scaled by  $1/\sqrt{m}$ .

$$\mathbb{E}[\langle \mathbf{\Pi a}, \mathbf{\Pi b} \rangle] = \langle \mathbf{a}, \mathbf{b} \rangle,$$

and, if  $\mathbf{\Pi}$  is chosen to have  $m$  rows, then:

$$\text{Var}[\langle \mathbf{\Pi a}, \mathbf{\Pi b} \rangle] \leq \frac{2}{m} \|\mathbf{a}\|_2 \|\mathbf{b}\|_2.$$

**Corollary:** If we use sketches of size  $m = O(1/\delta\epsilon^2)$ , then with probability  $(1 - \delta)$ ,

$$|\langle \mathbf{\Pi a}, \mathbf{\Pi b} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \epsilon \cdot \|\mathbf{a}\|_2 \|\mathbf{b}\|_2.$$

Chebyshev's Inequality:

## JOHNSON-LINDENSTRAUSS LEMMA

Dependence on  $\delta$  can be improved using fancier concentration inequalities. In particular, possible to show that with sketches of size  $m = O(\log(1/\delta)/\epsilon^2)$ ,

$$|\langle \mathbf{Pa}, \mathbf{Pb} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \epsilon \cdot \|\mathbf{a}\|_2 \|\mathbf{b}\|_2.$$

Special case:

$$(1 - \epsilon)\|\mathbf{a}\|_2^2 \leq \|\mathbf{Pa}\|_2^2 \leq (1 + \epsilon)\|\mathbf{Pa}\|_2^2$$

.

Can be used to prove the famous Johnson-Lindenstrauss Lemma.

[Dasgupta, **Gupta**, 2003], [Indyk, Motwani 1998], [Arriaga, Vempala 1999], [Achlioptas, 2001].

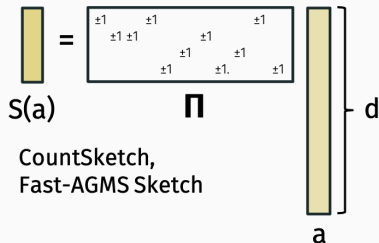
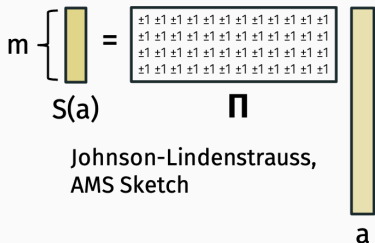
## Lemma (Johnson-Lindenstrauss, 1984)

For any set of  $n$  data points  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  there exists a linear map  $\Pi : \mathbb{R}^d \rightarrow \mathbb{R}^m$  where  $m = O\left(\frac{\log n}{\epsilon^2}\right)$  such that for all  $i, j$ ,

$$(1 - \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \|\Pi\mathbf{x}_i - \Pi\mathbf{x}_j\|_2 \leq (1 + \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2.$$

Main idea:

# FAST LINEAR SKETCHING



Naive cost of dense linear sketching is  $O(d \cdot m)$ .

This can be accelerated to  $O(d)$  (linear) time without sacrificing accuracy by using an ultra-sparse random matrix. [Charikar, Chen, **Farach-Colton**, 2002]. Still achieve with  $m = O(1/\delta\epsilon^2)$ :

$$|\langle \Pi \mathbf{a}, \Pi \mathbf{b} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \epsilon \cdot \|\mathbf{a}\|_2 \|\mathbf{b}\|_2.$$

**Recent developments on JL sketches:** Typically every entry of the compression is a real-value. In your computer, a double or a single precision float taking 32 or 64 bits. Can we reduce cost per dimension down to fewer bits?

Method of [Zandieh, Daliri, Han \[AAAI, 2025\]](#)<sup>1</sup>:

- Let  $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$  be a random Gaussian matrix.
- Let  $\mathcal{S}(\mathbf{a}) = \text{sign}(\mathbf{\Pi}\mathbf{a})$ ,  $\mathcal{S}(\mathbf{b}) = \mathbf{\Pi}\mathbf{a}$
- Estimate inner product via  $\mathcal{F}_{QL} = \frac{\sqrt{\pi}}{m} \cdot \|\mathbf{a}\|_2 \cdot \langle \mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}) \rangle$ .

---

<sup>1</sup>Related to SimHash [Charikar, 2002], [Jacques, 2013], and a few other.

### Theorem

Let  $\mathcal{F}_{QJL}$  be the inner product estimate returned by QJL for vectors  $\mathbf{a}$ ,  $\mathbf{b}$ , we have:

$$\begin{aligned}\mathbb{E}[\mathcal{F}_{QJL}] &= \langle \mathbf{a}, \mathbf{b} \rangle \\ \text{Var}[\mathcal{F}_{QJL}] &\leq \frac{1.6}{m} \|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2.\end{aligned}$$

Slightly better variance than JL! But only using one bit per entry of  $\mathcal{S}(\mathbf{a})$  (and need to store  $\|\mathbf{a}\|_2$ ).

Lots of other recent work on the setting where only  $\mathbf{a}$  needs to be compressed. E.g. RaBitQ sketches of [Gao, Long, 2024] addresses the setting where we want to target error  $\leq \frac{1}{\sqrt{d}}$ .

**Goal:** Present a completely different alternative to JL-style linear sketches that:

1. Is as simple as linear sketching to implement and analyze.
2. Can be applied in linear time (like CountSketch).
3. Matches theoretical bounds for linear sketching in the worst case, better for sparse vectors.
4. Typically beats linear sketching in experiments.

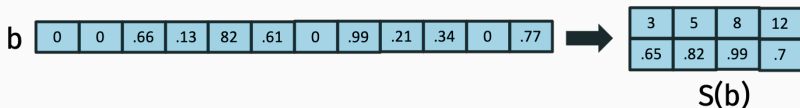
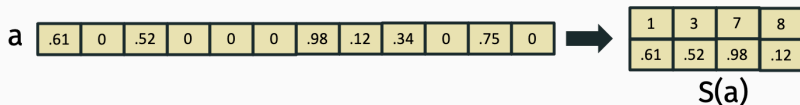
Appeared in [Sampling Methods for Inner Product Sketching](#) [Daliri, Freire, Musco, Santos, Zhang. VLDB 2024].

---

Similar to “End-Biased Sampling” [Estan, Naughton, 2006]. Also very related to the MinHash Sketch [Broder, 1997] and KMV Sketches [Bar-Yossef, Jayram, Kumar, Sivakumar, Trevisan, 2002], [Haas, Reinwald, Sismanis, Gemulla, 2007].

## BASIC IDEA

Sketch consists of subset of index/value pairs from **a** and **b**.

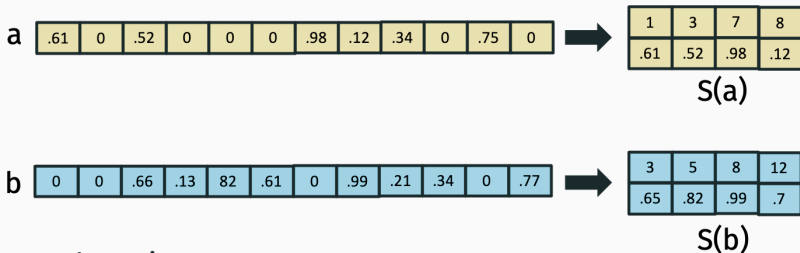


Let  $\mathcal{T}$  be the set of indices common to  $\mathcal{S}(\mathbf{a})$ ,  $\mathcal{S}(\mathbf{b})$ . Estimate:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^d a_i b_i \approx \sum_{i \in \mathcal{T}} w_i \cdot a_i b_i,$$

where  $w_i > 1$  is an appropriately chosen weight.

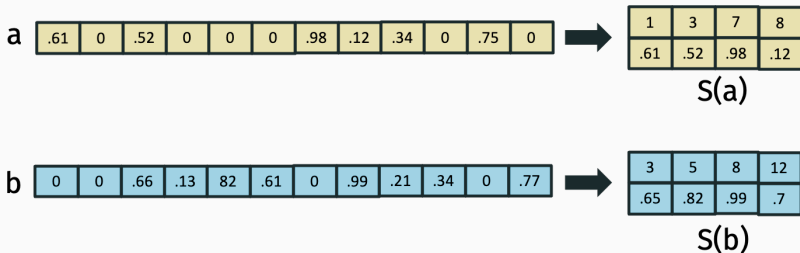
## BASIC IDEA



Natural tension:

- Larger entries in **a** and **b** contribute more to  $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^d a_i b_i$ . I.e., choice of indices should depend on magnitude of entries in vector being sketched.
- Want  $\mathcal{S}(\mathbf{a})$  and  $\mathcal{S}(\mathbf{b})$  to have many of the same indices. I.e., choice of indices should be coordinated between vectors.

## BASIC IDEA



**Coordinate Random Sampling:** Collect a sample from two different distributions while maximizing probability the samples are the same.

## Threshold Sampling:

- Set target sketch size  $m$ .
- Draw uniform random numbers  $u_1, \dots, u_d \sim [0, 1]$ .
- For  $i \in 1, \dots, d$ :
  - Add  $(i, a_i)$  to  $\mathcal{S}(\mathbf{a})$  if  $u_i \leq m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}$ .
  - Add  $(i, b_i)$  to  $\mathcal{S}(\mathbf{b})$  if  $u_i \leq m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2}$ .

## Estimation:

- Let  $\mathcal{T}$  be the set of indices common to  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$ .
- Return  $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) = \sum_{i \in \mathcal{T}} \frac{1}{p_i} a_i b_i$ , where
 
$$p_i = \min \left( 1, m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}, m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2} \right).$$

**Theorem**

*Let  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$  be sketches for  $\mathbf{a}, \mathbf{b}$  obtained via Threshold Sampling and let  $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))$  be the corresponding estimate for  $\langle \mathbf{a}, \mathbf{b} \rangle$  obtained from those sketches.*

*We have that  $\mathbb{E}[|\mathcal{S}(\mathbf{a})|] = m$ ,  $\mathbb{E}[|\mathcal{S}(\mathbf{b})|] = m$ , and:*

$$\begin{aligned}\mathbb{E}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &= \langle \mathbf{a}, \mathbf{b} \rangle \\ \text{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &\leq \frac{2}{m} \max(\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2, \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2)\end{aligned}$$

**Corollary:** If  $m = O(1/\epsilon^2)$ , then with high probability,  
 $|\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \epsilon \cdot \max(\|\mathbf{a}_{\mathcal{I}}\|_2 \|\mathbf{b}\|_2, \|\mathbf{a}\|_2 \|\mathbf{b}_{\mathcal{I}}\|_2).$

## THEORETICAL GUARANTEE

$$|\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) - \langle \mathbf{a}, \mathbf{b} \rangle| \leq \epsilon \cdot \max(\|\mathbf{a}_{\mathcal{I}}\|_2 \|\mathbf{b}\|_2, \|\mathbf{a}\|_2 \|\mathbf{b}_{\mathcal{I}}\|_2).$$

|                      |     |   |     |     |    |     |     |     |     |     |     |     |
|----------------------|-----|---|-----|-----|----|-----|-----|-----|-----|-----|-----|-----|
| <b>a</b>             | .61 | 0 | .52 | 0   | 0  | 0   | .98 | .12 | .34 | 0   | .75 | 0   |
| <b>b</b>             | 0   | 0 | .66 | .13 | 82 | .61 | 0   | .99 | .21 | .34 | 0   | .77 |
| <b>a<sub>I</sub></b> | 0   | 0 | .52 | 0   | 0  | 0   | 0   | .12 | .34 | 0   | 0   | 0   |
| <b>b<sub>I</sub></b> | 0   | 0 | .66 | 0   | 0  | 0   | 0   | .99 | .21 | 0   | 0   | 0   |

So always tighter than the AMS/JL/CountSketch bound of  $\epsilon \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$ . Some implications for vector search involving sparse embeddings (e.g., SPLADE embeddings [Formal, Piwowarski, Clinchant, 2021]).

### Threshold Sampling:

- Set target sketch size  $m$ .
  - Draw uniform random numbers  $u_1, \dots, u_d \sim [0, 1]$ .
  - For  $i \in 1, \dots, d$ :
    - Add  $(i, a_i)$  to  $\mathcal{S}(\mathbf{a})$  if  $u_i \leq m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}$ .
- 

Claim:  $\mathbb{E} [|\mathcal{S}(\mathbf{a})|] \leq m$ .

### Threshold Sampling:

- Set target sketch size  $m$ .
  - Draw uniform random numbers  $u_1, \dots, u_d \sim [0, 1]$ .
  - For  $i \in 1, \dots, d$ :
    - Add  $(i, a_i)$  to  $\mathcal{S}(\mathbf{a})$  if  $u_i \leq m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}$ .
- 

Claim:  $\mathbb{E}[|\mathcal{S}(\mathbf{a})|] \leq m$ .

$$\begin{aligned}\mathbb{E}[|\mathcal{S}(\mathbf{a})|] &= \sum_{i=1}^d \Pr \left[ u_i \leq m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2} \right] = \sum_{i=1}^d \min \left( 1, m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2} \right) \\ &\leq \frac{m}{\|\mathbf{a}\|_2^2} \sum_{i=1}^d a_i^2 \\ &= m.\end{aligned}$$

Can also be shown to hold with high probability.

### Threshold Sampling:

- Draw uniform random numbers  $u_1, \dots, u_d \sim [0, 1]$ .
  - For  $i \in 1, \dots, d$ :
    - Add  $(i, a_i)$  to  $\mathcal{S}(\mathbf{a})$  if  $u_i \leq m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}$ .
    - Add  $(i, b_i)$  to  $\mathcal{S}(\mathbf{b})$  if  $u_i \leq m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2}$ .
- 

### Claim:

$$\Pr[i \in \mathcal{S}(\mathbf{a}) \text{ and } i \in \mathcal{S}(\mathbf{b})] = p_i = \min \left( 1, m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}, m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2} \right).$$

Claim:

$$\Pr[i \in \mathcal{S}(\mathbf{a}) \text{ and } i \in \mathcal{S}(\mathbf{b})] = p_i = \min \left( 1, m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}, m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2} \right).$$

---

Estimation:

- Let  $\mathcal{T}$  be the set of indices common to  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$ .
  - Return  $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) = \sum_{i \in \mathcal{T}} \frac{1}{p_i} a_i b_i$ .
- 

Claim:  $\mathbb{E}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] = \langle \mathbf{a}, \mathbf{b} \rangle$ .

Claim:

$$\Pr[i \in \mathcal{S}(\mathbf{a}) \text{ and } i \in \mathcal{S}(\mathbf{b})] = p_i = \min \left( 1, m \cdot \frac{a_i^2}{\|\mathbf{a}\|_2^2}, m \cdot \frac{b_i^2}{\|\mathbf{b}\|_2^2} \right).$$


---

Estimation:

- Let  $\mathcal{T}$  be the set of indices common to  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$ .
  - Return  $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})) = \sum_{i \in \mathcal{T}} \frac{1}{p_i} a_i b_i$ .
- 

Claim:  $\mathbb{E}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] = \langle \mathbf{a}, \mathbf{b} \rangle$ .

$$\begin{aligned} \mathbb{E}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &= \mathbb{E} \left[ \sum_{i=1}^d \mathbb{1}[i \in \mathcal{T}] \cdot \frac{1}{p_i} a_i b_i \right] = \sum_{i=1}^d \Pr[i \in \mathcal{T}] \cdot \frac{1}{p_i} a_i b_i \\ &= \sum_{i=1}^d p_i \cdot \frac{1}{p_i} a_i b_i = \sum_{i=1}^d a_i b_i = \langle \mathbf{a}, \mathbf{b} \rangle. \end{aligned}$$

Claim:  $\text{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] \leq \frac{2}{m} \max(\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2, \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2).$

Claim:  $\text{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] \leq \frac{2}{m} \max(\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2, \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2).$

**Takeaway:** Just elementary calculations.

$$\begin{aligned}\mathrm{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &= \sum_{i \in \mathcal{I}} \mathrm{Var} \left[ \mathbb{1}[i \in \mathcal{T}] \cdot \frac{\mathbf{a}_i \mathbf{b}_i}{p_i} \right] = \sum_{i \in \mathcal{I}} \frac{(\mathbf{a}_i \mathbf{b}_i)^2}{p_i^2} \mathrm{Var}[\mathbb{1}[i \in \mathcal{T}]] \\ &\leq \sum_{i \in \mathcal{I}, p_i \neq 1} \frac{(\mathbf{a}_i \mathbf{b}_i)^2}{p_i}.\end{aligned}$$

$$\begin{aligned}\sum_{i \in \mathcal{I}, p_i \neq 1} \frac{(\mathbf{a}_i \mathbf{b}_i)^2}{p_i} &\leq \sum_{i \in \mathcal{I}, p_i \neq 1} \|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2 \frac{(\mathbf{a}_i^2 / \|\mathbf{a}\|_2^2)(\mathbf{b}_i^2 / \|\mathbf{b}\|_2^2)}{m \cdot \min(\mathbf{a}_i^2 / \|\mathbf{a}\|_2^2, \mathbf{b}_i^2 / \|\mathbf{b}\|_2^2)} \\ &= \sum_{i \in \mathcal{I}, p_i \neq 1} \|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2 \frac{\max(\mathbf{a}_i^2 / \|\mathbf{a}\|_2^2, \mathbf{b}_i^2 / \|\mathbf{b}\|_2^2)}{m} \\ &\leq \frac{\|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2}{m} \sum_{i \in \mathcal{I}} \frac{\mathbf{a}_i^2}{\|\mathbf{a}\|_2^2} + \frac{\mathbf{b}_i^2}{\|\mathbf{b}\|_2^2} \\ &= \frac{1}{m} (\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2 + \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2).\end{aligned}$$

### Theorem

*Let  $\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})$  be sketches for  $\mathbf{a}, \mathbf{b}$  obtained via Threshold Sampling and let  $\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))$  be the corresponding estimate for  $\langle \mathbf{a}, \mathbf{b} \rangle$  obtained from those sketches.*

*We have that  $\mathbb{E}[|\mathcal{S}(\mathbf{a})|] = m$ ,  $\mathbb{E}[|\mathcal{S}(\mathbf{b})|] = m$ , and:*

$$\begin{aligned}\mathbb{E}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &= \langle \mathbf{a}, \mathbf{b} \rangle \\ \text{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] &\leq \frac{2}{m} \max(\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2, \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2)\end{aligned}$$

**One annoying feature of Threshold Sampling:** Sketch size is random. Ideally it would be exactly equal to  $m$ .

Priority Sampling<sup>2</sup>

- Set sketch size  $m$ .
- Draw uniform random numbers  $u_1, \dots, u_d \sim [0, 1]$ .
- Let  $i_1, \dots, i_m$  be the indices corresponding to the  $m$  smallest values of  $u_i/a_i^2$ .
- Add  $(i_1, a_{i_1}), (i_2, a_{i_2}), \dots, (i_m, a_{i_m})$  to  $\mathcal{S}(\mathbf{a})$ .

**Bound:**  $\text{Var}[\mathcal{F}(\mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b}))] \leq \frac{2}{m-1} \max(\|\mathbf{a}_{\mathcal{I}}\|_2^2 \|\mathbf{b}\|_2^2, \|\mathbf{a}\|_2^2 \|\mathbf{b}_{\mathcal{I}}\|_2^2)$

Almost identical to the bound given by Threshold Sampling.

---

<sup>2</sup>[Duffield, Lund, Thorup, 2004], [Ohlsson, 1998].

# PARALLELIZING LARGE LANGUAGE MODELS

Modern large language models are inherently sequential.

NYU is a private research university \_\_\_\_\_ .

Even if cost per word can come down, limits the speed at which text can be generated.

Introduced concurrently by [Leviathan, Kalman, Matias, 2023] at Google and [Chen, Borgeaud, Irving, Lepiau, Sifre, Jumper, 2023] at Google Deepmind

**Key idea:** Use small model to “draft” a response, and verify with multiple instances of a large model.

# SPECULATIVE DECODING

**Draft:** NYU is a private research university in the city of New York .



**Desired Output:** NYU is a private research university in New York City.



NYU is a private research university



NYU is a private research university in



NYU is a private research university in New



NYU is a private research university in the city

....

Above, we got three tokens from one batch of parallel invocations of the large model.

# SPECULATIVE DECODING

**Draft:** NYU is a private research university in New York City .



**Desired Output:** NYU is a private research university in New York City.



NYU is a private research university in New York



NYU is a private research university in New York City

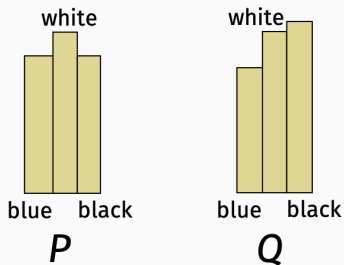


NYU is a private research university in New York City .

....

## SPECULATIVE DECODING

**Obvious issue:** Even if the next token distribution for the drafter model,  $\mathcal{P}$ , and the product model,  $\mathcal{Q}$  are very similar, it could be unlikely for the draft to be correct.



If  $a \sim P$  and  $b \sim Q$ ,  $\Pr[a = b] \approx$

**Solution:** Coordinate the sampling!

### Definition (Coupling)

Let  $\mathcal{P}$  and  $\mathcal{Q}$  be distributions over  $\{1, \dots, n\}$ . A coupling between  $\mathcal{P}$  and  $\mathcal{Q}$  is any distribution over pairs  $(a, b) \in \{1, \dots, n\} \times \{1, \dots, n\}$  such that  $a$ 's marginal distribution is  $\mathcal{P}$  and  $b$ 's marginal distribution is  $\mathcal{Q}$ .

**Goal:** Efficiently sample from a coupling  $\mathcal{C}$  between the small and large model distributions which maximizes

$$\Pr[a = b].$$

**Simple case:** If  $\mathcal{P} = \mathcal{Q}$ , can find a  $\mathcal{C}$  for which  $\Pr[a = b]$ . Do you see how?

### Definition

The **total variation distance**,  $D_{TV}(\mathcal{P}, \mathcal{Q})$ , between two distributions equals:

$$D_{TV}(\mathcal{P}, \mathcal{Q}) = 1 - \max_{\text{couplings } \mathcal{C}} \left[ \Pr_{(a,b) \sim \mathcal{C}} [a = b]. \right]$$

**Claim:** Let  $\mathcal{P}$  and  $\mathcal{Q}$  be discrete distributions over  $\{1, \dots, n\}$  represented by length  $n$  probability vectors  $\mathbf{p}, \mathbf{q} \in [0, 1]^n$ .

$$D_{TV}(\mathcal{P}, \mathcal{Q}) = 1 - \sum_{i=1}^n \min(p_i, q_i).$$

**Claim:** Let  $\mathcal{P}$  and  $\mathcal{Q}$  be discrete distributions over  $\{1, \dots, n\}$  represented by length  $n$  probability vectors  $\mathbf{p}, \mathbf{q} \in [0, 1]^n$ .

$$D_{TV}(\mathcal{P}, \mathcal{Q}) \geq 1 - \sum_{i=1}^n \min(p_i, q_i).$$

**Proof:** Under any coupling  $\mathcal{C}$ ,  $\Pr[a = b] =$

## OPTIMAL COUPLING

There is a simple procedure that achieves this bound in the speculative decoding setting.

**Drafter:**

- Sample  $a \sim \mathcal{P}$ . Sends both  $a$  and  $\mathbf{p}$  to FullModel.

**FullModel:**

- Await  $(a, \mathbf{p})$  from Drafter.
- With probability  $\min(1, q_a/p_a)$  return  $b = a$ .
- Otherwise, sample  $b$  from  $\mathcal{Q}' = \{q'_1, \dots, q'_n\}$ , where:

$$q'_i = \frac{\max(0, q_i - p_i)}{\sum_{j=1}^n \max(0, q_j - p_j)}$$

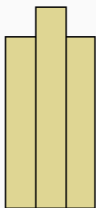
# OPTIMAL COUPLING

Drafter:

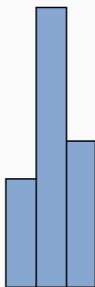
- Sample  $a \sim \mathcal{P}$ . Sends both  $a$  and  $\mathbf{p}$  to FullModel.

FullModel:

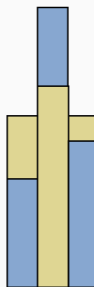
- With probability  $\min(1, q_a/p_a)$  return  $b = a$ .
- Otherwise, set  $b = i$  w.p.  $q'_i = \frac{\max(0, q_i - p_i)}{\sum_{i=1}^n \max(0, q_i - p_i)}$ .



$P$



$Q$



$Q'$

## OPTIMAL COUPLING

**Drafter:**

- Sample  $a \sim \mathcal{P}$ . Sends both  $a$  and  $\mathbf{p}$  to FullModel.

**FullModel:**

- With probability  $\min(1, q_a/p_a)$  return  $b = a$ .
- Otherwise, set  $b = i$  w.p.  $q'_i = \frac{\max(0, q_i - p_i)}{\sum_{i=1}^n \max(0, q_i - p_i)}$ .

**Claim:** Procedure samples from a coupling. I.e.,  $a \sim \mathcal{P}, b \sim \mathcal{Q}$ .

**Case to check:**  $q_a > p_a$ .

## OPTIMAL COUPLING

**Drafter:**

- Sample  $a \sim \mathcal{P}$ . Sends both  $a$  and  $\mathbf{p}$  to FullModel.

**FullModel:**

- With probability  $\min(1, q_a/p_a)$  return  $b = a$ .
- Otherwise, set  $b = i$  w.p.  $q'_i = \frac{\max(0, q_i - p_i)}{\sum_{i=1}^n \max(0, q_i - p_i)}$ .

**Claim:**  $\Pr[a = b] = \sum_{i=1}^n \min(p_i, q_i)$ .

Earlier showed that  $D_{\text{TV}}(\mathcal{P}, \mathcal{Q}) \geq 1 - \sum_{i=1}^n \min(p_i, q_i)$ , so this is optimal.

### FullModel:

- With probability  $\min(1, q_a/p_a)$  return  $b = a$ .
- Otherwise, set  $b = i$  w.p.  $q'_i = \frac{\max(0, q_i - p_i)}{\sum_{i=1}^n \max(0, q_i - p_i)}$ .

**Seemingly small but annoying issue:** The output of the FullModel is always sampled from  $\mathcal{Q}$ , but the exact value sampled depends on the Drafter distribution  $\mathcal{P}$ .

- Cannot immediately verify that adding speculative decoding did not change the model distribution.
- Model output is not deterministic from the user's point of view given a fixed random seed.

**Key idea:** Basic protocol requires communication between the Drafter and FullModel. Try to couple samples without communication at all but just using shared randomness.

“Coupling without Communication and Drafter-Invariant Speculative Decoding” [Daliri, Musco, Suresh, ISIT 2025].

**Basically the same idea appeared in:**

- Anari, Gao, Rubinstein, STOC 2024
- Liu, Yin, STOC 2022
- Bavarian, Ghazi, Haramaty, Kamath, Rivest, Sudan, 2020.

## WEIGHTED MINHASH COUPLING

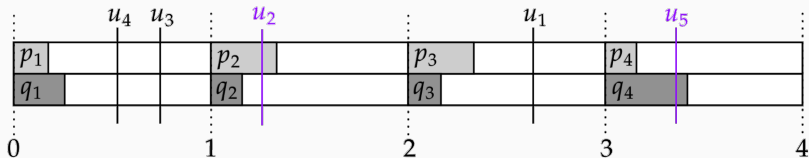
Fix public random variables  $u_1, u_2, \dots \sim \text{Unif}[0, n]$ .

**Drafter:**

- For  $k = 1, 2, \dots$ ,
  - If  $k \in [j - 1, j - 1 + p_j]$  for some  $j$ , return  $a = j$ .

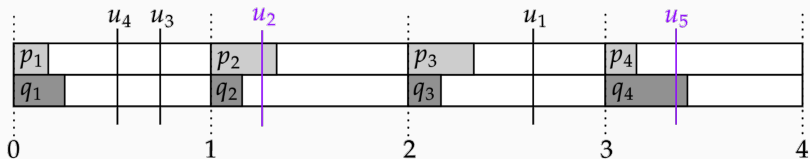
**FullModel:**

- For  $k = 1, 2, \dots$ ,
  - If  $k \in [j - 1, j - 1 + q_j]$  for some  $j$ , return  $b = j$ .



## WEIGHTED MINHASH COUPLING

Claim:  $\Pr[a = b] \geq \frac{\sum_{i=1}^n \min(p_i, q_i)}{\sum_{i=1}^n \max(p_i, q_i)}$



Optimal Coupling:

$$\Pr[a = b] = 1 - D_{TV}(\mathcal{P}, \mathcal{Q})$$

. Communication-Free Coupling:

$$\Pr[a = b] \geq \frac{\sum_{i=1}^n \min(p_i, q_i)}{\sum_{i=1}^n \max(p_i, q_i)} = \frac{1 - D_{TV}(\mathcal{P}, \mathcal{Q})}{1 + D_{TV}(\mathcal{P}, \mathcal{Q})}$$

.

**Takeaway: Pay very little for drafter-invariance!**

Possible to show that this is optimal in some sense. No communication-free protocol can achieve

$\Pr[a = b] > \frac{1 - D_{TV}(\mathcal{P}, \mathcal{Q})}{1 + D_{TV}(\mathcal{P}, \mathcal{Q})}$  for all distributions simultaneously [Bavarian, Ghazi, Haramaty, Kamath, Rivest, Sudan, 2020].

## GUMBEL SAMPLING

Fix public random variables  $u_1, u_2, \dots \sim \text{Unif}[0, 1]$ .

**Drafter:**

- Return  $a = \arg \min_{i \in \{1, \dots, n\}} \frac{-\ln(u_i)}{p_i}$ .

**FullModel:**

- Return  $b = \arg \min_{i \in \{1, \dots, n\}} \frac{-\ln(u_i)}{q_i}$ .

This is already how samples are typically obtained! In particular, standard to use the “Gumbel Max Trick”:

$$b = \arg \max_{i \in \{1, \dots, n\}} [-\ln(-\ln(1/u_i)) + \ln(q_i)].$$

### Theorem (pareto improvement)

*For any two distributions  $\mathcal{P}, \mathcal{Q}$ ,*

$$\Pr_{(a,b) \sim \text{Gumbel}} [a = b] \geq \Pr_{(a,b) \sim \text{MinHash}} [a = b],$$

*and there exist distributions where inequality is strict.*

**Question one group is studying for the project:** Is Gumbel pareto optimal?