Achieving Top-K-fairness for Finding Global Top-K Frequent Items

Yikai Zhao*, Wei Zhou*, Wenchen Han†, Zheng Zhong*, Yinda Zhang‡, Xiuqi Zheng*, Tong Yang*, Bin Cui*
* National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, Beijing, China
† Department of Computer Science, University College London, London, United Kingdom
‡ Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pennsylvania, USA

***Abstract*—Finding top-K frequent items has been a hot topic in data stream processing with wide-ranging applications. However, most existing sketch algorithms focus on finding local top-K in a single data stream. In this paper, we tackle finding global top-K across multiple data streams. We find that using prior sketch algorithms directly is often unfair in global scenarios, degrading global top-K accuracy. We define top-K-fairness and show its importance for finding global top-K. To achieve this, we propose the Double-Anonymous (DA) sketch, where double-anonymity ensures fairness. We also propose two techniques, hot-filtering and early-freezing, to improve accuracy further. We theoretically prove that the DA sketch achieves top-K-fairness while maintaining high accuracy. Extensive experiments verify top-K-fairness in disjoint data streams, showing that the DA sketch's error is up to 129 times (60 times on average) smaller than the state-of-the-art. To enhance the applicability and technical depth, we also investigate how to extend the DA sketch to general distributed data stream scenarios and how to provide a fairer and more accurate global ranking for top-K items. The experimental results show that the extended version of the DA sketch can indeed compute better rankings and still has significant advantages in general data streams.**

***Index Terms*—data streams, global top-K, top-K-fairness, sketch**

I. Introduction

A. Background and Motivation

Finding top-K frequent items has been a hot topic in data stream processing in recent years, which has a wide range of applications, such as data mining [2]–[5], databases [6]–[8], networking [9], [10], and network security [11], [12]. Finding top-K frequent items refers to selecting K items with the largest number of frequencies, and providing frequency estimation. In the era of big data, the speed and volume of data are growing explosively. Sketches [3]–[9], [13]–[32], a kind of probabilistic data structures, have obtained wide acceptance and interests to address the task of finding top-K due to their efficiency in terms of both time and space, although they can have a small error [33]. For finding top-K frequent items, most of existing sketch algorithms focus on providing statistics over a single data stream [3], [4], [6], [8], [9], [13]–[17], [34], [35], while a few of them [3], [6], [36] work on merging the statistics over multiple related data streams into one. In the preliminary version of this paper [1], we provide the first sketch that can *compare* the statistics over different *disjoint data streams* (§III); in the current version, we discuss in greater depth how to extend this sketch to general data streams (§V-A). Specifically, given N data streams, how can we compare their own top-K and select the global top-K. Note that the sizes of these data streams are often skewed in practice (*e.g.*, power law distribution) [37].

We use an example to explain the problem. For an autonomous system (AS) in a wide-area network (WAN), external traffic enters the AS through multiple border routers [38]. Due to the principle of routing protocol [39], all network packets sent to the AS from the same source address must pass through the same border router. In other words, if we regard the source address of the packets as the key, the packets streams on different border routers are *disjoint data streams*. Network operators usually need to monitor the main source of traffic entering the AS, *i.e.*, the K source addresses that send the most packets [40]. To find these addresses, each border router reports the local top-K frequent address and their frequency, and operator sorts all local frequent addresses to get the *global top-K*.

For finding global top-K frequent items, a typical solution is to first use a sketch for each data stream to select local top-K items, and then sort them based on their estimated frequency. However, we find that directly apply existing sketches often leads to **unfairness**. Specifically, the estimated frequency of top-K items in prior sketches is largely influenced by the local environment (*e.g.*, the size of data streams). If we directly sort all the selected local top-K items, the result will be significantly related to the items' local environment rather than its real frequency. For instance, suppose there are N disjoint data streams, some *heavy data streams* have more items, and some *light data streams* have fewer items. Suppose we use N SpaceSaving [17], which always provides overestimated estimation and the degree of overestimation is positively correlated to the size of the data stream, to find local top-K from the N data streams. As a result, the items in the heavy data streams will be overestimated more and get higher chances to be selected as global top-K, while the frequent items in the light data streams will tend to be ignored, which is unfair.

To address this problem, we aim to achieve top-K-fairness: the degree of overestimation or underestimation *for the local selected top-K items* is a constant, *i.e.*, not related to the data stream. The formal definition of top-K-fairness is provided in Section II-A.

Tong Yang (yangtongemail@gmail.com) is the corresponding author.

The preliminary version of this paper titled "Double-Anonymous Sketch: Achieving Fairness for Finding Global Top-K Frequent Items" was published in ACM Special Interest Group on Management of Data (SIGMOD) [1], Seattle, WA, USA, June 18 - 23, 2023.

B. Prior Works

To the best of our knowledge, we are the first work to focus on the top-K-fairness of global top-K items. Many existing work focuses on providing unbiased estimation in distributed scenarios [3], [6]. Unbiasedness is helpful if we want to aggregate the statistics of multiple data streams for all items. However, if we only focus on the estimated frequency of top-K items, we can find that it is often overestimated. The main reason is that if we select top-K items, we tend to select items which are overestimated, which leads to unfairness. We use the state-of-the-art unbiased sketches, Unbiased SpaceSaving (USS) [6], to illustrate the problem.

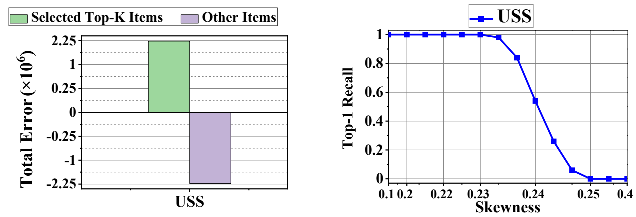
As shown in Figure 1(a), although the estimation of USS is unbiased when considering all items, it overestimates the selected top-K items and underestimates others. Furthermore, such top-K-unfairness in local data streams will cause top-K-unfairness when finding global top-K items. As shown in Figure 1(b)[1], suppose the global top-1 item e_{top} is in a light data stream, and we deploy a USS for each data stream. USS provides a slightly overestimated value for e_{top}, and provides significantly overestimated value for frequent items in heavy data streams. As a result, when the size distribution of the data streams is highly skewed, even the global top-1 item could be ignored, which is often unacceptable in practice. In Section VI-D, we also discuss that such unfairness cannot be alleviated by re-weighting the estimated frequency.

C. Our Proposed Solution

To achieve top-K-fairness, we propose the Double-Anonymous sketch (abbreviated as DA sketch). We first propose a basic version which achieves top-K-fairness, and then we optimize the accuracy through two techniques *hot filtering* and *early freezing*. The DA sketch has the following advantages: 1) It is the first work that discusses the fairness problem for comparing multiple disjoint data streams. 2) It is accurate: The error of our sketch is up to 129 times (60 times on average) smaller than Waving [3] and $3 \sim 4$ orders of magnitude smaller than Frequent [34], USS [6], and SS [17]. 3) It is generic: we implement existing four *replacement strategies* in our framework to achieve top-K-fairness and accuracy.

The key technique of our DA sketch to achieve top-K-fairness is called **double-anonymity**. Double-anonymity is often an effective strategy to achieve fairness. We leverage this strategy to enable top-K sketches to achieve top-K-fairness. A top-K sketch often consists of two parts, a top-K part for finding top-K items and a count part for frequency estimation. If it meets the following two conditions, we consider it achieves double-anonymity: 1) the top-K part finds top-K items independently, and does not know any items' estimated frequency in the count part; 2) the count part estimates item's frequency independently, and does not know which items are top-K. However, the existing unbiased sketches do not meet the first condition. Our formal definition of double-anonymity is provided in Section III-A. We theoretically prove that double-anonymity is a sufficient condition for unbiased sketches to achieve top-K-fairness. Therefore, we follow this principle to design our solution.

[1] Settings for Figure 1(a): We perform the finding local top-K tasks on CAIDA dataset [41] for 1000 times. Memory size is set to be 100KB, and $K = 1000$; Settings for Figure 1(b): We conduct experiments on the Synthetic Dataset [42]. We generate the dataset so that the global Top-1 item is always in the light stream. We set $N = 100$, $K = 50$ and range skewness from 0.1 to 0.4. We allocate an extremely small amount of memory for USS, such that it could only store $K = 50$ local top-K candidates.



(a) Internal unfairness (b) Recall of the Global Top-1 Item

Fig. 1: We demonstrate unfairness of USS, and show how unfairness would harm accuracy for finding global top-K.

In our basic version, we use a top-K sketch (*e.g.*, SpaveSaving [17]) as the top-K part and use an unbiased sketch (*e.g.*, CMM sketch [43]) as the count part. To achieve double-anonymity, our first version makes these two parts work independently, *i.e.*, it forbids any information transmission between them. For an incoming item e, it will be inserted into the two parts independently and respectively. Note that the independent condition is stronger than double-anonymity. Obviously, our first version is double-anonymous, and thus achieves top-K-fairness. However, the first version fails to achieve high accuracy. Therefore, we propose two important optimization methods to significantly improve accuracy: **hot filtering** and **early freezing**. Unlike the first version, in these two versions, we allow some information transmission between the two parts as long as it does not violate double-anonymity. Relaxing the forbidden condition, we can have more opportunities to improve accuracy.

Firstly, the main reason for the significant error in the first version is that, despite identifying the potential top-K items through the top-K part, we still insert the entire frequency of these items into the count part. Considering that the estimation error of sketches is often proportional to the total frequency of inserted items, and the frequencies of the top-K items account for the majority of the total frequency, reducing the frequency of top-K items inserted into the count part can significantly reduce the error. The key idea of the **hot filtering** version is that, while identifying the top-K items using the top-K part, most of the frequency of these potential top-K items can be directly recorded in the top-K part, thus eliminating the need to insert them into the count part. Secondly, the number of items inserted into the count part also grows with the data stream, causing the estimation error to grow as well. The key idea of the **early freezing** version is that, instead of querying the count part after the end of the data stream to get the complete estimated frequency of the local top-K items, it immediately queries the count part and records (*freezes*) the result in the top-K part as soon as an item is recorded in the top-K part, thereby reducing the error.

We show that the DA sketch is generic. Any replacement strategy independent with the CMM sketch can be applied to the DA sketch, and we choose four [4], [16], [17], [34] as case studies. In the current version, we also propose a unified framework to describe the probability guarantees of different algorithms for finding top-K items, and analyze the probability guarantees of the DA sketch using different replacement strategies for finding top-K items.

To further enhance top-k fairness and the applicability and technical depth of the DA sketch, we discuss their extensions from two aspects:

- First, we investigate how to define top-K-fairness in general data streams; why existing unbiased algorithms cannot achieve fairness; and how to extend the DA sketch to general data stream settings to achieve fairness.
- Second, we start with relative ranking to study a more stringent definition of top-K-fairness — ranking-fairness. Using a ranking-fair algorithm, two items with the same true frequency each have an equal probability of having a higher estimated ranking.

Key Contributions: ❶ We define a new and important property: top-K-fairness. We analyze it thoroughly and derive its sufficient condition. Additionally, we propose the DA sketch, an accurate, unbiased, and generic method, marking the first work to achieve top-K-fairness. ❷ We provide a theoretical proof that the DA sketch achieves top-K-fairness while maintaining high accuracy. Furthermore, we analyze its probability guarantees under different replacement strategies for identifying top-K items. ❸ We extend the DA sketch to general data streams, demonstrating its significant advantages in broader settings. We also enhance the sketch to support ranking-fairness and confirm that it achieves higher ranking correlation coefficients. ❹ Through extensive experiments, we validate top-K-fairness and show that the DA sketch achieves significantly smaller errors compared to existing methods.

II. Background and Related Work

In this section, we provide formal definitions of our problem and top-K-fairness. We discuss the difference between unbiasedness and top-K-fairness.

A. Formal Definitions and Preliminaries

Definition 1. ***(Disjoint data streams)*** *Given N data streams $\mathcal{S}_1, \cdots, \mathcal{S}_N$, where $\mathcal{S}_i = \left\{e_{(i,1)}, \cdots, e_{(i,m_i)}\right\}$ contains m_i items, and each item $e_{(i,j)}$ belongs to set $\mathcal{U}_i = \left\{u_{(i,1)}, \cdots, u_{(i,n_i)}\right\}$. N Data streams are disjoint if $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$ for any two different data streams $\mathcal{S}_i$ and $\mathcal{S}_j$.*

Generally speaking, the settings of disjoint data streams require that one item cannot appear in multiple different data streams. Disjoint data streams are common in scenarios such as distributed storage systems and distributed network management. In these scenarios, an item is often placed on only one device, and then only appears in one data stream.

Definition 2. ***(Global top-K items)*** *Given N disjoint data streams $\mathcal{S}_1, \cdots, \mathcal{S}_N$, for data stream $\mathcal{S}_i = \left\{e_{(i,1)}, \cdots, e_{(i,m_i)}\right\}$ and item set $\mathcal{U}_i$, we define that the frequency of item $u_{(i,j)} \in \mathcal{U}_i$ as $f_{(i,j)} = \sum_{k=1}^{m_i} \mathbf{1}_{\left\{e_{(i,k)}=u_{(i,j)}\right\}}$. The global top-$K$ items are the K items with the largest frequency.*

To find global top-K items, each data stream $\mathcal{S}_i$ uses the top-K algorithm to find the set $\mathcal{T}_i = \left\{u_{(i,p_1)}, \cdots, u_{(i,p_K)}\right\}$ of local top-K items and their estimated frequency $\hat{f}_{(i,p_j)}$. Each data stream $\mathcal{S}_i$ reports the set $\mathcal{T}_i$ and frequency of items to a central machine. The central machine obtains the global set $\mathcal{U} = \bigcup_{i=1}^{N} \mathcal{T}_i$, and then uses K items with the largest estimated frequency in $\mathcal{U}$ to form global top-K items.

Definition 3. ***(Top-K-fairness)*** *Given an algorithm, for any data stream $\mathcal{S}_i$, let $\mathcal{T}_i$ be the set of local top-K frequent items reported by $\mathcal{S}_i$. We call the algorithm a top-K-fair algorithm if, for any item $u_{(i,j)} \in \mathcal{T}_i$, the following equation holds: $E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = \alpha \times f_{(i,j)} + \delta$, where $f_{(i,j)}$ and $\hat{f}_{(i,j)}$ are the real frequency and estimated frequency of item $u_{(i,j)}$ respectively, and α and δ are two constants independent of data streams.*

The existing research on fairness and equality mainly focuses on other areas. For example, the previous work in the field of machine learning uses condition probability to define *group fairness*, which requires that each decision has the same probability for members of different groups; the previous work in the field of recommendation system uses ratio to define *ranking fairness*, which requires that the attention received by each object is proportional to its relevance.

Although these fairness concerns focus on different areas, we can find a commonality among them: *the evaluation of different individuals should depend on their intrinsic attributes rather than their environment.* Our definition of top-K-fairness originates from this common point as well: we want the selection of global top-k items to depend on their true frequencies (*i.e.*, *intrinsic attributes*) rather than the statistical characteristics of their local data streams (*i.e.*, *environment*). In fact, overestimated algorithms will make frequent items in small data streams be easily ignored, while underestimated algorithms will make frequent items in large data streams be easily ignored, as verified in Section VI-D. However, for top-K-fair algorithms, we can ensure that two items with the same true frequency have the same expected estimated frequency when listed as global top-K candidates. After discussing and analyzing ranking fairness in Section V-B, we can even ensure that these two items each have a 50% probability of having a higher estimated frequency. Our algorithm achieves top-K-fairness with $\alpha = 1$, $\delta = 0$, which, for an unbiased algorithm, is equivalent to $E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = E\left(\hat{f}_{(i,j)}\right)$.

B. Unbiasedness v.s. Top-K-fairness

Sketches [10], [11], [44]–[55] are a kind of probabilistic algorithm which is often used to find top-K items due to its high speed and small memory consumption. There are two kinds of top-K sketch algorithms, biased algorithm and unbiased algorithm. Biased top-K algorithms include SpaceSaving [17], Frequent [34], HeavyGuardian [4], Randomized Admission Policy [16], and etc [8], [9], [13], [14], [56]. Because all these biased algorithm's biases are highly related to the data streams, they cannot achieve top-K-fairness. Among all existing works, USS and WavingSketch [3] claim to be unbiased. However, it should be noted that unbiased algorithms are not necessarily top-K-fair. We first use USS as an example to discuss why an algorithm being unbiased does not imply that the algorithm is top-K-fair, and further discuss its behavior in general data streams. We will also discuss the relationship between disjoint data streams and general data streams. We show the definition of unbiased algorithm.

Definition 4. *(**Unbiased algorithm**) When finding local top-K items in a single data stream $\mathcal{S}_i$, the top-K algorithm maintains the estimated frequency $\hat{f}_{(i,j)}$ of each item $u_{(i,j)}$. The algorithm is unbiased if $\forall u_{(i,j)} \in \mathcal{U}_i$, $E\left(\hat{f}_{(i,j)}\right) = f_{(i,j)}$.*

*1) **Case Study of USS**:* The main difference between our top-K-fairness and unbiasedness is that the top-K-fairness has an additional condition that $u_{(i,j)} \in \mathcal{T}_i$. That is to say, we only consider the behavior of an item $u_{(i,j)}$ when it is listed as a global top-K candidate. Although USS is an unbiased algorithm, it estimates the frequency of all non-top-K items as 0, *i.e.*, $E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \notin \mathcal{T}_i\right) = 0$, $E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = \frac{f_{(i,j)}}{\Pr\left(u_{(i,j)} \in \mathcal{T}_i\right)}$.

- **USS is an unbiased algorithm:** $E\left(\hat{f}_{(i,j)}\right) = E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \notin \mathcal{T}_i\right) \times \Pr\left(u_{(i,j)} \notin \mathcal{T}_i\right) + E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) \times \Pr\left(u_{(i,j)} \in \mathcal{T}_i\right) = f_{(i,j)}$
- **USS is not a top-K-fair algorithm:** The amplification coefficient $\alpha = \frac{1}{\Pr\left(u_{(i,j)} \in \mathcal{T}_i\right)}$ varies largely among data streams, so USS cannot achieve top-K-fairness.
- **Unfairness brought by USS in general data streams:** Taking USS as an example, we demonstrate why unbiased but top-K-unfairness sketch algorithms still fail in the context of general data streams. Consider an item u, and without loss of generality, assume it appears in the first m data streams $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_m$. When u is considered a candidate for global top-K, that is, $u \in \mathcal{U}$, it is reported as local top-K by at least one of the local data streams. In other words, only when $\mathcal{S}_1, \cdots, \mathcal{S}_m$ all do not report item u, then it is $u \notin \mathcal{U}$, and in this case, $E\left(\hat{f}_u \mid u \notin \mathcal{U}\right) = 0$. Therefore, we have $E\left(\hat{f}_u \mid u \in \mathcal{U}\right) = \frac{f_u}{1 - \prod_{i=1}^{m} \Pr(u \notin \mathcal{T}_i)}$, where f_u is the true frequency of u, and $\hat{f}_u$ is the estimated frequency of u. For USS, $\hat{f}_u$ is the sum of the estimated frequencies reported by each data stream.

Relationship with Disjoint Data Streams: Through the formula above, we can see that as m increases, the denominator gradually approaches 1, and thus $E\left(\hat{f}_u \mid u \in \mathcal{U}\right)$ gradually approaches f_u. This means that for USS, when $m = 1$, that is, when item u appears in only one unique data stream, the estimated frequency of the item is most affected by the data stream. Therefore, we believe that for algorithms without top-K-fairness, disjoint data streams represent the scenario where top-K-unfairness is most severe.

*2) **Extension to other unbiased algorithms**:* In fact, for any unbiased algorithm, $E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = f_{(i,j)} + \delta$ and $\delta = \frac{\mathtt{Cov}\left(\hat{f}_{(i,j)}, \Pr\left(u_{(i,j)} \in \mathcal{T}_i \mid \hat{f}_{(i,j)}\right)\right)}{\Pr\left(u_{(i,j)} \in \mathcal{T}_i\right)}$. WavingSketch [3] achieves unbiasedness based on the C sketch [15]. When an item's estimated frequency is large, WavingSketch uses the heavy part to record its ID and frequency. However, WavingSketch tends to favor recording the overestimated items in the heavy part, *i.e.*, $\Pr\left(u_{(i,j)} \in \mathcal{T}_i \mid \hat{f}_{(i,j)}\right)$ increases with $\hat{f}_{(i,j)}$, meaning $\delta > 0$. Moreover, the deviation δ depends on not only the frequency distribution of the data stream, but also the arrival order of the items. Therefore, WavingSketch cannot achieve top-K-fairness. In conclusion, no existing work achieves top-K-fairness in the task of finding global top-K items.

C. The CMM Sketch

The CMM sketch [43] can provide an unbiased estimation of items' frequency. Since we use the CMM sketch as a component of our algorithm, we describe the data structure and operators of the CMM sketch in detail in this section.

Data Structure: A CMM sketch consists of d arrays, each of which includes w counters $\mathcal{A}[i,j]$ $(1 \leqslant i \leqslant d, 1 \leqslant j \leqslant w)$ and is associated with a hash function $g_i(\cdot)$. Each hash function maps an item to a counter uniformly at random.

Insertion: Given an incoming item e, the CMM maps e to the counter $\mathcal{A}[i, g_i(e)]$ in each array and increments each of them by 1.

Query: Given a query about item e, the CMM can give the overestimation and unbiased estimation of its frequency. The unbiased estimation $C_{unbiased}(e)$ is given by the following formula. $C_{unbiased}(e,i) = \mathcal{A}[i, g_i(e)] - \frac{1}{w-1} \cdot (\mathcal{N} - \mathcal{A}[i, g_i(e)])$. $C_{unbiased}(e) = \frac{1}{d} \cdot \left(\sum_{i=1}^{d} C_{unbiased}(e,i)\right)$. Where $\mathcal{N}$ is the sum of the frequencies of all distinct items.

III. The Double-Anonymous Sketch

In this section, we propose the Double-Anonymous sketch. We first introduce **double-anonymity**, which is the key technique to achieve top-K-fairness. Then we introduce **hot filtering**, a tricky technique that can keep the characteristic of double-anonymity and raise the accuracy. Finally, we introduce **early freezing**, a technique that can further raise accuracy.

A. The Basic Version

Definition of double-anonymity: Suppose the estimation has already been unbiased, one *sufficient condition* of top-K-fairness is that the covariance of the result of finding top-K items and estimating frequency is 0. A more formal definition is shown in Definition 5. Achieving **double-anonymity** means that the algorithm meets this condition.

Definition 5. *(**Double-anonymity**) Given an algorithm, for a single data stream $\mathcal{S}_k$ and an item $u_{(k,i)}$, let $\mathcal{K}_{(i)}$ be an indicator indicating whether item $u_{(k,i)}$ is selected as top-K ($u_{(k,i)} \in \mathcal{T}_k$). We call the algorithm has double-anonymity if, for any item $u_{(k,i)}$, $\mathtt{Cov}\left(\mathcal{K}_i, \hat{f}_{(k,i)}\right) = 0$.*

Theorem 1. *(**Sufficient Condition**) Given an algorithm, if it is unbiased, i.e., $E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}$, then it being top-$K$-fair[2],*

[2]Here we only consider top-K-fairness with $\alpha = 1$ and $\delta = 0$

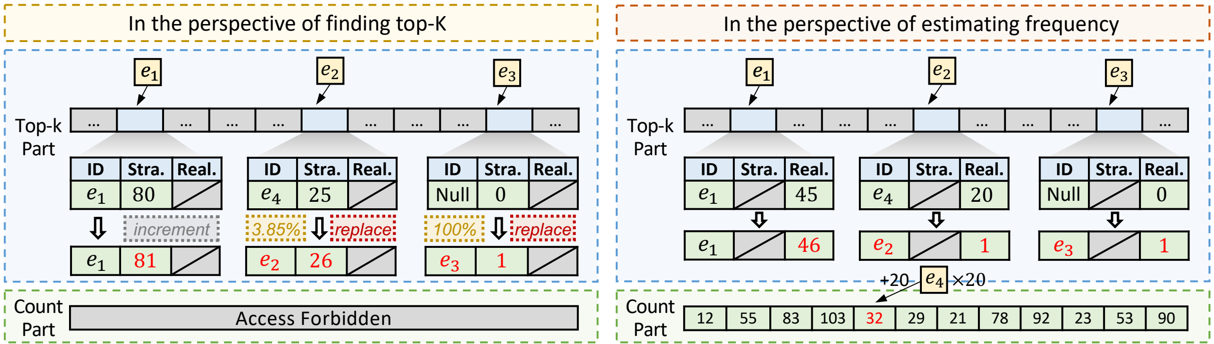


Fig. 2: A running example of the Hot filtering version of the Double-Anonymous sketch with RA Policy.

i.e., $E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) = f_{(k,i)}$*, is equivalent to it having double-anonymity, i.e.,* $\texttt{Cov}\left(\mathcal{K}_i, \hat{f}_{(k,i)}\right) = 0$.

Proof. Expanding $E\left(\hat{f}_{(k,i)} \cdot \mathcal{K}_i\right)$, we have $E\left(\hat{f}_{(k,i)} \cdot \mathcal{K}_i\right) = E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) \cdot E\left(\mathcal{K}_i\right)$. Therefore, under the condition of $E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}$ (unbiasedness),

$$\begin{aligned} &\left[E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) = f_{(k,i)}\right] \\ \equiv &\left[E\left(\hat{f}_{(k,i)} \cdot \mathcal{K}_i\right) = E\left(\hat{f}_{(k,i)}\right) \cdot E\left(\mathcal{K}_i\right)\right] \\ \equiv &\left[\texttt{Cov}\left(\mathcal{K}_i, \hat{f}_{(k,i)}\right) = 0\right]. \end{aligned}$$

In the above formulas, $\equiv$ stands for equivalence. □

The data structure of the basic version has two parts: a Randomized Admission Policy (RA) [16] as the top-K part and a CMM sketch [43] as the count part. For an incoming item e, e will be inserted into the RA and the CMM sketch independently. To find top-K items, we query the RA and report the result. To query an item e's frequency, we query the CMM sketch and report the result. Obviously, the basic version is double-anonymous and achieves top-K-fairness.

B. The Hot Filtering Version

Keeping the characteristic of double-anonymity, the hot filtering version aims to filter the hot items, and only record them in the top-K part to remove the redundancy. We first use a top-K part to classify and record hot items. Because the top-K part filters the hot items, only cold items will be inserted into the count part, which makes the Hot filtering version accurate.

Data Structure: As shown in Figure 2, the Double-Anonymous sketch has two parts: a top-K part and a count part. The top-K part is an array of buckets $\mathcal{B}[0, \dots, M-1]$. Each item will be hashed into a bucket using $h(.)$, a hash function that maps each item to $[0, M-1]$ uniformly at random. Each bucket has λ cells. Each cell records the information of one item: the item ID (key), the strategy frequency C_s, and the real frequency C_r. The C_s is a counter used to decide whether this item should be evicted according to different replacement strategies. It is often biased, *i.e.*, overestimated or underestimated. The C_r is another counter used to record the number of appearances of this item after it was inserted into the top-k part. The count part is a CMM sketch [43], which can provide an unbiased estimation.

Insertion: We first try inserting the incoming item into the top-K part. If the replacement strategy thinks the item is frequent, we record it in the top-K part. Otherwise, we insert it into the count part. Given an incoming item e, we hash it into the bucket $\mathcal{B}[h(e)]$. For any case, we first run the *replacement strategy* of the Double-Anonymous sketch to find the top-K frequent items (we implement four strategies in Section III-D for case study). Usually, the replacement strategy (*e.g.*, SpaceSaving) will find the top-K frequent items and keep their ID in the top-K part according to their strategy frequency C_s. To guarantee that the replacement strategy works properly, the Double-Anonymous sketch rules that the ID and the strategy frequency can only be changed by the replacement strategy. Then we run the **unbiased operations** of the Double-Anonymous sketch to provide unbiased estimation for top-K items. The unbiased operations are following this principle: if the incoming item e is in the top-K part at that time, we use the top-K part to record this increment. Otherwise, we use the count part to record this increment. There are three cases as follows.

Case 1: e is in the bucket $\mathcal{B}[h(e)]$. We increment $e.C_r$ by 1.

Case 2: e is not in the bucket $\mathcal{B}[h(e)]$. We insert e into the count part: we use d other hash functions $g_1(.) \dots g_d(.)$ to map item to $[0, w-1]$, and increment the d counters $\mathcal{A}[1, g_1(e)], \dots \mathcal{A}[d, g_d(e)]$ by 1, which are called the d *mapped counters*.

Case 3: An item e_{evict} is evicted by the replacement strategy. We increase the d *mapped counters* in the count part by $e_{evict}.C_r$, *i.e.*, the real frequency of e_{evict} before the eviction. This operation can transfer the frequency of e_{evict} from the top-K part to the count part. Therefore, we would not lose the frequency information of e_{evict}.

Identifying local top-K items: We traverse the top-K part of the DA sketch and collect all the $\langle$item e, strategy frequency $C_s\rangle$ pairs from all buckets. We then sort all collected pairs by C_s. The K items with the highest strategy frequency C_s are selected as the local top-K items, forming the local top-K set $\mathcal{T}$.

Estimating frequencies of local top-K items: For each selected local top-K item e, part of its frequency is recorded in the true frequency C_r, and the remaining part is recorded in the count part, which is the CMM sketch. Therefore, we query the CMM for an unbiased estimate $C_{unbiased}$ of the remaining part of the frequency using its ID e, and take $\hat{f} = C_r + C_{unbiased}$ as the estimated frequency of item e.

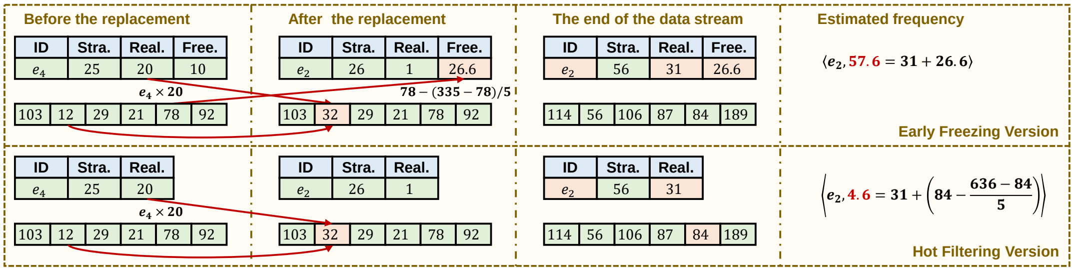


Fig. 3: The process after item e_2 replaces item e_4: we immediately query the count part and obtain the current estimated frequency of e_2 as 26.6 using the CMM sketch's query method and record it in $C_{freezing}$. When querying, the early freezing version returns the estimated frequency as 57.6 based on the real frequency Cr and the frozen frequency $C_{freezing}$; in contrast, the hot filtering version queries the CMM at this point and gets an estimated frequency of 4.6.

Identifying global top-K items: We obtain the local top-K set $\mathcal{T}_i$ reported by each local data stream $\mathcal{S}_i$, along with the estimated frequency $\hat{f}$ of each local top-K item in the set. The K local top-K items with the highest estimated frequencies $\hat{f}$ (selected from $\mathcal{U} = \bigcup \mathcal{T}_i$) are chosen as the global top-K items.[3]

A running example: Figure 2 shows a running example of Hot filtering version of the DA sketch with Randomized Admission Policy. Notice that the process of finding top-K and estimating frequency are Double-Anonymous, *i.e.*, information that may influence their covariance is not shared between these two processes. In the perspective of finding top-K, 1) To insert e_1, it successes, so we increment $e_1.C_s$ by 1. 2) To insert e_2, it evicts e_4 successfully (according to the Randomized Admission Policy, the chance of success is $\frac{1}{26} \approx 3.85\%$). Then we record e_2 and make $e_2.C_s = 26$. 3) To insert e_3, we find an empty cell, so we just record e_3 and make $e_3.C_s = 1$. In the perspective of estimating frequency, 1) To insert e_1, it successes, so we increment $e_1.C_r$ by 1. 2) To insert e_2, it successes, so we make $e_2.C_r$ to 1. At the same time, e_4 is evicted, so we insert $e_4 \times 20$ into the count part, *i.e.*, the mapped counters in the CMM sketch are increased by 20. 3) To insert e_3, we find an empty cell, so we just record e_3 and make $e_3.C_r = 1$.

C. The Early Freezing Version

As shown in Figure 3, the early freezing version of the DA sketch uses an additional frozen frequency counter $C_{freezing}$ in each cell of the top-K part. If a newly arrived item e is not originally recorded in the top-K part but is decided to be recorded under the replacement strategy used in the top-K part, we immediately query the count part to obtain an unbiased estimate of the current frequency of item e and record it in the frozen counter $C_{freezing}$.

When querying the frequency of top-K items, we use $C_r + C_{freezing}$ instead of $C_r + C_{unbiased}$. Since the error of the sketch in the count part increases with the number of inserted items, $C_{freezing}$ is an earlier and lower-error version of $C_{unbiased}$.

D. Using Different Replacement Policies

The DA sketch can be applied by any top-K algorithm (replacement strategy). We pick four classic top-K strategies: Randomized Replacement Strategy (RA) [16], Spacesaving (SS) [17], Frequent (Freq) [34] and HeavyGuardian (HG) [4] as case studies. For each strategy, we introduce how it works and how to apply it in the DA sketch. Given an incoming item e, we first hash it into $\mathcal{B}[h(e)]$. Then the strategies work as follows. Suppose the item whose C_s is smallest in the bucket is e_{min}.

RA Policy [16]: *DS+RA* (Double-Anonymous sketch with Randomized Admission Policy) runs the operation of RA first. If e is in the bucket, we increment $e.C_s$ by 1. If e is not in the bucket, we evict e_{min} with the probability of $\frac{1}{e_{min}.C_s+1}$. If the eviction successes, we record e with its $C_s = e_{min}.C_s + 1$. *DS+RA* then runs the unbiased operation of the DA sketch.

SpaceSaving (SS) [17]: *DS+SS* (Double-Anonymous sketch with SpaceSaving) runs the operation of SS first. If e is in the bucket, we just increment $e.C_s$ by 1. If e is not in the bucket, we evict e_{min} and record e with its $C_s = e_{min}.C_s+1$. *DS+SS* then runs the unbiased operation of the DA sketch.

Frequent (Freq) [34]: *DS+Freq* (Double-Anonymous sketch with Frequent) runs the operation of Freq first. If e is in the bucket, we increment $e.C_s$ by 1. If e is not in the bucket, we decrement the C_s of every item in this bucket by 1. If the C_s of an item e_{evict} is decreased to 0, we evict e_{evict} and record e with its $C_s = 1$. *DS+Freq* then runs the unbiased operation of the DA sketch.

HeavyGardian (HG) [4]: *DS+HG* (Double-Anonymous sketch with HeavyGardian) runs the operation of HG first. Suppose the item whose C_s is smallest in the bucket is e_{min}. If e is in the bucket, we increment $e.C_s$ by 1. If e is not in the bucket, we decrement $e_{min}.C_s$ by 1 with a probability of $1.08^{-e_{min}.C_s}$. If $e_{min}.C_s$ is decreased to 0, we evict e_{min} and insert e with its $C_s = 1$. *DS+HG* then runs the unbiased operation of the DA sketch.

We further discuss the differences between these replacement policies based on the experimental results in Section VI, and show that our algorithm is general. Specially, In Section VI-D, we show the degree of top-K-unfairness of these four replacement policies, analyze how top-K-unfairness affects their performance, and show that our DA sketch can indeed make them top-K-fair.

E. Probability Guarantees for Finding Local Top-K

Although the DA sketch is designed for global top-K identification, its initial step is to precisely identify the local

[3] We describe this process in Section II-A and reiterate it here.

top-K within the local data stream, and then provide fair frequency estimates for these items. Therefore, it is crucial to offer theoretical guarantees for the DA sketch in finding local top-k and to configure parameters based on these theoretical assurances.

In this section, we analyze the probability guarantees of identifying local Top-K frequent items using the DA sketch under various replacement strategies. To achieve this, we first generalize and formulate a unified representation of the probability guarantees for existing Top-K algorithms. After examining the probability guarantee forms of several Top-K algorithms, we define an $(\psi, 1-\phi)$-Top-K algorithm as follows.

Definition 6. *Given a data stream $\mathcal{S} = \{e_1, \cdots, e_m\}$ comprising m items and a Top-K algorithm $\mathcal{A}$ with λ cells, where each cell captures the information of an item. For any item u_i, if its frequency $f_i \geqslant \psi \cdot \frac{m}{\lambda}$, then the probability of it being ultimately recorded in one of the cells is at least $1-\phi$. Under these conditions, we categorize algorithm $\mathcal{A}$ as an $(\psi, 1-\phi)$-Top-K algorithm.*

For instance, both the SpaceSaving [17] and Frequent [34] algorithms are $(1,1)$-Top-K algorithms, which means they are guaranteed to record items with a frequency exceeding $\frac{m}{\lambda}$. On the other hand, the Unbiased SpaceSaving [6] algorithm is an $(\psi, 1-e^{-\psi})$-Top-K algorithm. Next, we extend the proof approach presented in [57] to provide probability guarantees for the DA sketch under both general frequency distributions and Zipfian frequency distributions.

Theorem 2. *Given a data stream $\mathcal{S} = \{e_1, \cdots, e_m\}$ comprising m items, and a DA sketch whose top-k part contains M buckets, each bucket containing λ cells, and employing an $(\psi, 1-\phi)$-Top-K algorithm as the replacement strategy. Without assuming any frequency distribution, we can assert that the DA sketch is an $\left(\psi', \left(1-\frac{\psi\lambda}{(\lambda-\psi)\psi'}\right)(1-\phi)\right)$-Top-$K$ algorithm.*

Proof. For any item e with frequency $f_i \geqslant \psi' \cdot \frac{m}{M\lambda}$, and the sum of the frequencies $\hat{f}$ of the other items mapped to the same bucket, the expected value of $\hat{f}$ satisfies: $E\left(\hat{f}\right) < \frac{m}{M}$. Then, let $\mathcal{T} = \left(\lambda\frac{\psi'}{\psi} - \psi'\right)\frac{m}{M\lambda}$, according to Markov's inequality, we have: $\Pr\left(\hat{f} > \mathcal{T}\right) \leqslant \frac{\psi\lambda}{(\lambda-\psi)\psi'}$. When $\hat{f} \leqslant \mathcal{T}$, we have $f_i \geqslant \psi\frac{(f_i+\hat{f})}{\lambda}$, which implies that the item e has at least a $1-\phi$ probability of being ultimately recorded. That is, the probability $\Pr$ of item e being ultimately recorded satisfies: $\Pr \geqslant \Pr\left(\hat{f} \leqslant \mathcal{T}\right)(1-\phi) \geqslant \left(1-\frac{\psi\lambda}{(\lambda-\psi)\psi'}\right)(1-\phi)$. □

Lemma 1. *Given a data stream $\mathcal{S}$ comprising m items, and a DA sketch whose top-k part contains M buckets, each bucket containing λ cells. Without loss of generality, assume $f_1 > f_2 > \cdots$. Assuming the frequencies follow a Zipfian distribution with parameter $s > 1$, i.e., $f_i = \frac{m}{\zeta(s)}i^{-s}$, the expected sum of frequencies $\hat{f}$ of all items mapped to a particular bucket satisfies the following with a probability of at least $3^{-\frac{k}{M}}$: $E\left(\hat{f}\right) \leqslant \frac{mk^{1-s}}{M}$.*

Proof. For any positive integer k, when the number of buckets M is sufficiently large, the probability that no top-k item is mapped to this bucket is $\left(1-\frac{1}{M}\right)^k > 3^{-\frac{k}{M}}$. In this case, the expected sum of frequencies $\hat{f}$ of all non-top-k items mapped to this bucket satisfies:

$$\begin{aligned} E\left(\hat{f}\right) &= \frac{\sum_{i=k+1} f_i}{M} = \frac{\sum_{i=k+1}\frac{m}{\zeta(s)}i^{-s}}{M} \\ &\leqslant \frac{m}{M}\left(\int_k^{+\infty} x^{-s}dx\right)\left(\int_1^{+\infty} x^{-s}dx\right)^{-1} \\ &\leqslant \frac{m}{M}\frac{k^{1-s}}{s-1}(s-1) = \frac{mk^{1-s}}{M}. \end{aligned}$$

□

Theorem 3. *Given a data stream $\mathcal{S}$ comprising m items, and a DA sketch whose top-k part contains M buckets, each bucket containing λ cells, and employing an $(\psi, 1-\phi)$-Top-K algorithm as the replacement strategy. Without loss of generality, assume $f_1 > f_2 > \cdots$. Assuming the frequencies follow a Zipfian distribution with parameter $s > 1$, we can assert that the DA sketch is an $\left(\psi', \sup_{\eta>0}\left(3^{-\eta}(1-\phi)\left(1-\frac{\psi\lambda}{(\lambda-\psi)\psi'(\eta M)^{s-1}}\right)\right)\right)$-Top-$K$ algorithm.*

Proof. For any item e with frequency $f_i \geqslant \psi' \cdot \frac{m}{M\lambda}$, and the sum of the frequencies $\hat{f}$ of the other items mapped to the same bucket, we define condition $\mathcal{C}$ as: no other top-k item is mapped to the same bucket as item e. Then, let $\mathcal{T} = \left(\lambda\frac{\psi'}{\psi} - \psi'\right)\frac{m}{M\lambda}$, according to Lemma 1, we have

$$\Pr\left(\hat{f} > \mathcal{T} \mid \mathcal{C}\right) \leqslant \frac{\psi\lambda k^{1-s}}{(\lambda-\psi)\psi'}.$$

Thus, for any positive integer k, we have

$$\begin{aligned} \Pr\left(\hat{f} > \mathcal{T}\right) &\leqslant \Pr\left(\hat{f} > \mathcal{T}, \mathcal{C}\right) + \Pr(\neg\mathcal{C}) \\ &\leqslant 3^{-\frac{k}{M}}\left(\frac{\psi\lambda k^{1-s}}{(\lambda-\psi)\psi'} - 1\right) + 1. \end{aligned}$$

For simplicity of form, let $k = \eta M$, iterating over all η, we obtain

$$\Pr \geqslant \sup_{\eta>0}\left(3^{-\eta}(1-\phi)\left(1-\frac{\psi\lambda}{(\lambda-\psi)\psi'(\eta M)^{s-1}}\right)\right).$$

□

Comparison of DS+USS and USS: As shown in Figure 4, we compare the probability guarantees of USS and the DA sketch using USS (DS+USS). For DS+USS, we set the parameters as $\lambda = 16$ and $M = 10^5$; for the Zipfian distribution, we set its parameter as $s = 1.5$. There are two points to explain.

- In Theorem 2 and 3, for a given ψ', ψ can take any value. Thus, for the probability guarantee of DS+USS under general distributions, we present the curve $\left(\psi', \sup_{\psi>0}\left(\left(1-\frac{\psi\lambda}{(\lambda-\psi)\psi'}\right)(1-e^{-\psi})\right)\right)$; the same logic applies to the probability guarantee under Zipfian distribution. For example, when $\psi' = 5$, for general distribution, we take $\psi = 1.353$; for Zipfian distribution, we take $\psi = 4.535$ and $\eta = 0.0152$.
- USS is an $(\psi, 1-e^{-\psi})$-Top-K algorithm. However, in the implementation of USS, each cell needs to record the item's key (4 bytes), frequency (4 bytes), and four

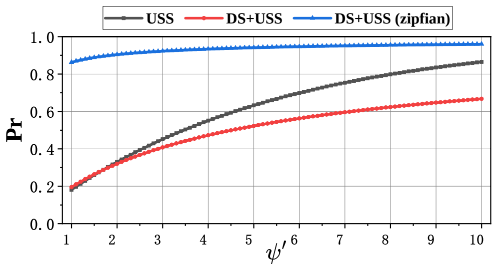


Fig. 4: Probability guarantees for DS+USS and USS in finding local top-K items.

pointers (32 bytes) [8], requiring a total of 40 bytes. In the implementation of DS+USS, if only considering the lookup of local top-K, each cell only needs to record the key (4 bytes) and C_s (4 bytes)[4], requiring only 8 bytes. Therefore, with the same memory, if DS+USS can use λM cells, USS can only use $\frac{\lambda M}{5}$ cells. Hence, for USS, we present the curve $\left(\psi', 1-e^{\frac{-\psi'}{5}}\right)$.

Corollary 1. *Given a data stream $\mathcal{S}$ comprising m items, assuming the frequencies follow a Zipfian distribution with parameter $s>1$, to maximize the probability of recording the local top-K frequent items in the DA sketch, we recommend setting the number of cells per bucket to $\lambda=\frac{\psi s}{s-1}$.*

Proof. According to Theorem 3, to maximize the probability guarantee Pr, when the total number of cells $\lambda M=N$ is fixed, we should minimize $\frac{\lambda}{(\lambda-\psi)M^{s-1}}$. The derivative of this expression is:

$$\left[\frac{\lambda}{(\lambda-\psi)M^{s-1}}\right]'=\frac{\lambda^{s-1}(\lambda+s(\psi-\lambda))}{(\lambda-\psi)^2N^{s-1}}$$

Consequently, the optimal λ should be $\lambda=\frac{\psi s}{s-1}$. □

IV. Mathematical Analysis

In this section, we analyze the behavior of *our hot filtering version* on a single data stream, and prove that it meets top-K-fairness. We then give some conclusions about the error of the algorithm. We also discuss how to apply the proof process to the *early freezing version*.

A. Preliminary

We then define the state $s_{(k,t)}$ of the Double-Anonymous sketch on data stream $\mathcal{S}_k$ at time t as $s_{(k,t)}=\{s_{(k,1,t)},\cdots,s_{(k,n_k,t)}\}$, where $s_{(k,i,t)}=\langle f_{T(k,i,t)},f_{S(k,i,t)}\rangle$. In general, let $f_{T(k,i,t)}$ be the frequency of item $u_{(k,i)}$ recorded in the top-K part at time t, and let $f_{S(k,i,t)}$ be the frequency of item i recorded in the count part at time t. In particular, if item $u_{(k,i)}$ is not recorded in the top-K part at time j, let $f_{T(k,i,t)}=0$.

Given a data stream $\mathcal{S}_k$, let a *sketching process* $\mathcal{R}$ be a sequence of states of the Double-Anonymous sketch at each time, *i.e.*, $\mathcal{R}=\{s_{(k,1)},s_{(k,2)},\cdots,s_{(k,m_k)}\}$. The replacement policy $\mathcal{P}$ determines the distribution of the sketching process, *i.e.*, $\mathcal{R}\sim\mathcal{P}(\mathcal{S}_k)$.

[4]We do not consider the memory occupied by C_r and $C_{freezing}$

B. Proof of Top-K-fairness

In this section, we prove that the DA sketch achieves top-K-fairness. We first give a lemma about the sketching process.

Lemma 2. *Given a data stream $\mathcal{S}_k$ and a sketching process $\mathcal{R}=\{s_{(k,1)},\cdots,s_{(k,m_k)}\}$, for any item $u_{(k,i)}$ and any time j, there is*

$$f_{T(k,i,t)}+f_{S(k,i,t)}=f_{(k,i,t)}. \tag{1}$$

Proof. When time $t=0$, for any item $u_{(k,i)}$, there is $f_{T(k,i,0)}=f_{S(k,i,0)}=f_{(k,i,t)}=0$, so there is $f_{T(k,i,0)}+f_{S(k,i,0)}=f_{(k,i,0)}$. Suppose that Equation 1 holds for any item $u_{(k,i)}$ and any time $t<t'$. At time $t=t'$, according to Section III-B[5], if $e_{(k,t)}=u_{(k,i)}$, we insert frequency $\left(f_{T(k,i,t'-1)}-f_{T(k,i,t')}+1\right)$ into the CMM sketch of the count part, thus $f_{T(k,i,t')}+f_{S(k,i,t')}=f_{(k,i,t'-1)}+1=f_{(k,i,t')}$; If $e_{(k,t)}\neq u_{(k,i)}$, we insert frequency $\left(f_{T(k,i,t'-1)}-f_{T(k,i,t')}\right)$ into the CMM sketch of the count part, thus $f_{T(k,i,t')}+f_{S(k,i,t')}=f_{(k,i,t'-1)}=f_{(k,i,t')}$; Therefore, Equation 1 also holds for $t=t'$, so it holds for any time $1\leqslant t\leqslant m_k$. □

Now we prove the following lemma holds for any replacement policy $\mathcal{P}$.

Lemma 3. *Given a data stream $\mathcal{S}_k$. For any item $u_{(k,i)}$, let $f_{S'(k,i,t)}$ be the estimate of $f_{S(k,i,t)}$ given by the count part, and let $\hat{f}_{(k,i)}=f_{T(k,i,m)}+f_{S'(k,i,m)}$ be the estimation of $f_{(k,i)}$ given by the DA sketch. For any replacement policy $\mathcal{P}$, any sketching process $\mathcal{R}$, there is $E\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right)=f_{(k,i)}$.*

Proof. According to Lemma 2, in the sketching process $\mathcal{R}$, $f_{T(k,i,m_k)}+f_{S(k,i,m_k)}=f_{(k,i,m_k)}$. Since $\hat{f}_{(k,i)}=f_{T(k,i,m_k)}+f_{S'(k,i,m_k)}$, and $f_{T(k,i,m_k)}$ is determined by sketching process $\mathcal{R}$, we only need to prove $E\left(f_{S'(k,i,m_k)}\mid\mathcal{R}\right)=f_{S(k,i,m_k)}$, Recall that we use a CMM [43] sketch as the count part. Specifically, assume that the count part uses d counter arrays, each of which has w counters and is associated with a hash function $g_l(\cdot)$.

Let the indicator random variable $I_{(i,j,l)}$ indicates whether $g_l\left(u_{(k,i)}\right)$ and $g_l\left(u_{(k,j)}\right)$ are equal, thus $\Pr\left(I_{(i,j,l)}=1\right)=\frac{1}{w}$. Let the random variable $X_{(i,l)}$ be the value of the $g_l\left(u_{(k,i)}\right)$-th counter in the l-th array, thus we have

$$f_{S'(k,i,m_k)}=\frac{1}{d}\cdot\left(\sum_{k=1}^{d}\left(X_{(i,l)}-\frac{1}{w-1}\cdot\left(\sum_{j=1}^{n_k}f_{S(k,j,m_k)}-X_{(i,l)}\right)\right)\right).$$

According to the rules of CMM, we can obtain the conditional expectation of $X_{i,k}$, *i.e.*,

$$E\left(X_{(i,l)}\mid\mathcal{R}\right)=f_{S(k,i,m_k)}+\frac{1}{w}\cdot\left(\sum_{j=1,j\neq i}^{n_k}f_{S(k,j,m_k)}\right).$$

Using the linear property of expectation, we have

$$E\left(f_{S'(k,i,m_k)}\mid\mathcal{R}\right)=f_{S(k,i,m_k)}.$$

□

Now we prove that the DA sketch achieves both **unbiasedness** and **Double-anonymity**, thus achieving top-K-fairness.

[5]$e_{(k,t)}$ is defined in Section II-A; in Section III-B, we simplify $e_{(k,t)}$ to e.

Theorem 4 (unbiasedness). *Given a data stream $\mathcal{S}_k$. For any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, there is*

$$E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}.$$

Proof. According to Lemma 3 and using the law of total expectation, we have

$$E\left(\hat{f}_{(k,i)}\right) = \sum_{\mathcal{R}} E\left(\hat{f}_{(k,i)} \mid \mathcal{R}\right) \cdot \Pr\left(\mathcal{R}\right) = f_{(k,i)}.$$

□

Theorem 5 (Double-anonymity). *Given a data stream $\mathcal{S}_k$. For any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, let $\mathcal{K}_i$ be an indicator random variable indicating whether item $u_{(k,i)}$ is selected as top-K, there is*

$$\texttt{Cov}\left(\hat{f}_{(k,i)}, \mathcal{K}_i\right) = 0.$$

Proof. Because sketching process $\mathcal{R}$ determines whether item $u_{(k,i)}$ is selected as top-K, all $\mathcal{R}$ can be divided into two kinds: $\mathcal{R} \in \mathcal{G}_0$ makes $\mathcal{K}_i = 0$, and $\mathcal{R} \in \mathcal{G}_1$ makes $\mathcal{K}_i = 1$. Therefore, we expand $E(\hat{f}_{(k,i)}\mathcal{K}_i)$ as follows:

$$\begin{aligned} E\left(\hat{f}_{(k,i)} \cdot \mathcal{K}_i\right) &= \sum_{\mathcal{R}\in\mathcal{G}_1} E(\hat{f}_{(k,i)} \cdot \mathcal{K}_i | \mathcal{R}) \cdot \Pr(\mathcal{R}) \\ &= \left(\sum_{\mathcal{R}\in\mathcal{G}_1} \Pr(\mathcal{R})\right) \cdot f_{(k,i)} = E(\mathcal{K}_i) \cdot f_{(k,i)}. \end{aligned}$$

Combined with unbiasedness, we have

$$\texttt{Cov}\left(\hat{f}_{(k,i)}, \mathcal{K}_i\right) = E\left(\hat{f}_{(k,i)} \cdot \mathcal{K}_i\right) - E\left(\hat{f}_{(k,i)}\right) E\left(\mathcal{K}_i\right) = 0.$$

□

C. Error Bounds of Estimations

In this section, we give some theorems about the error bounds of estimations. The item frequencies which are inserted into the count part are $f_{S(k,1,m_k)}, \cdots, f_{S(k,n_k,m_k)}$. According to lemma 2, they are less than or equal to $f_{(k,1,m_k)}, \cdots, f_{(k,n_k,m_k)}$, *i.e.*, $f_{(k,1)}, \cdots, f_{(k,n_k)}$. Based on this insight, we give the following lemmas and theorems, which show that the DA sketch has tighter error bounds than the sketches of CMM [43].

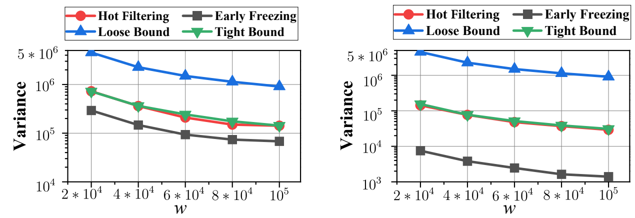
Lemma 4. *Given a data stream $\mathcal{S}_k$, for any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, let $\hat{f}_{(k,i)}$ be the unbiased estimation of $f_{(k,i)}$ given by the DA sketch, then we have*

$$\texttt{Var}\left(\hat{f}_{(k,i)}\right) \leqslant \frac{1}{d \cdot (w-1)} \cdot \left(\sum_{j=1}^{n_k} f^2_{S(k,j,m_k)}\right).$$

Where d and w are parameters of the count part (CMM).

Theorem 6. *Given a data stream $\mathcal{S}_k$, for any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, let $\hat{f}_{(k,i)}$ be the unbiased estimation of $f_{(k,i)}$ given by the DA sketch, then we have*

$$\begin{aligned} \Pr\left(\left|\hat{f}_{(k,i)} - f_{(k,i)}\right| \geqslant \varepsilon\right) &\leqslant \frac{1}{\varepsilon^2 \cdot d \cdot (w-1)} \cdot \left(\sum_{j=1}^{n_k} f^2_{S(k,j,m_k)}\right) \\ &< \frac{1}{\varepsilon^2 \cdot d \cdot (w-1)} \cdot \left(\sum_{j=1}^{n_k} f^2_{(k,j)}\right). \end{aligned}$$



(a) $\mathcal{P}$: SpaceSaving (b) $\mathcal{P}$: RA Policy

Fig. 5: Sample variances and their theoretical upper bounds.

D. Analysis on Early Freezing

By using the *early freezing* optimization, the DA sketch gives a more accurate item frequency estimation $\tilde{f}_{(k,i)} = f_{T(k,i,m_k)} + f_{S'(k,i,t_i)}$, where t_i is the time when item $u_{(k,i)}$ is recorded in the top-K part. In particular, $t_i = m_k$ when item $u_{(k,i)}$ is not recorded. On the one hand, following the proof in Section IV-B and IV-C and replacing $f_{S'(k,i,m_k)}$ with $f_{S'(k,i,t_i)}$, we can still prove the top-K-fairness and derive the error bound; On the other hand, according to Lemma 5 shown below, we know that the variance of $f_{S'(k,i,t_i)}$ is smaller than that of $f_{S'(k,i,m_k)}$ in any sketching process $\mathcal{R}$, so we have Theorem 7.

Lemma 5. *Given a data stream $\mathcal{S}_k$ and a sketching process $\mathcal{R} = \{s_{(k,1)}, \cdots, s_{(k,m)}\}$, for any item $u_{(k,i)}$ and any time j, there is*

$$f_{S(k,i,j-1)} \leqslant f_{S(k,i,j)}.$$

Theorem 7. *Given a data stream $\mathcal{S}$, for any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, we have*

$$\texttt{Var}\left(\tilde{f}_{(k,i)}\right) \leqslant \texttt{Var}\left(\hat{f}_{(k,i)}\right).$$

E. Experimental Verification

To verify the correctness of Lemma 4 and Theorem 7, we show two kinds of variance bound. The CMM sketch itself provides a loose $\mathcal{P}$-independent bound: $\frac{1}{d\cdot(w-1)} \cdot \left(\sum_{j=1}^{n_k} f^2_{(k,j)}\right)$, while Lemma 4 offers a tight $\mathcal{R}$-dependent bound. We use SpaceSaving and Randomized Admission Policy as the strategy $\mathcal{P}$, and vary the length w of the count part. As shown in Figure 5, it is worth noting two points: 1) The tight bounds are extremely close to the sample variances, which indicates our bounds are accurate. 2) Filtering hot items to reduce the redundancy is beneficial to reduce variance, and the strategy of finding top-K frequent items more accurately has a smaller variance.

V. Discussion

A. Top-K-fairness for General Data Streams

We focus on the problem of identifying global top-K items in disjoint data streams, which dictates that an item appears in only one unique data stream. However, a more general scenario in distributed data streams is where an item may be present in multiple data streams. Thus, a question arises: how is top-K-fairness defined in general data streams[6]? Is it possible to

[6]In general data streams, it is not required that $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$.

adapt the DA sketch, as proposed in this paper, to general data streams? We thoroughly discuss and answer these questions in this section.

Top-K-fairness in General Data Streams: Reflecting on our method for finding global top-K frequent items in distributed data streams, we require each data stream $\mathcal{S}_i$ to report a set of local top-K items $\mathcal{T}_i$. Thus, we obtain a candidate set for global top-K items $\mathcal{U} = \bigcup_{i=1}^{N} \mathcal{T}_i$. Different algorithms can provide an estimated frequency $\hat{f}_u$ for each candidate item $u \in \mathcal{U}$ in various ways. Top-K-fairness demands that for any item $u \in \mathcal{U}$, $\hat{f}_u = \alpha \times f_u + \delta$, where α and δ are constants consistent across all items.

Using DA sketch for Finding Global Top-K: For each local data stream $\mathcal{S}_i$, we use a DA sketch to find the local top-K, and form a set $\mathcal{T}_i$ based on the frequency of items in the top-K part. However, beyond reporting $\mathcal{T}_i$, we also need to report the entire DA sketch, referred to as $\mathcal{D}_i$. Then, on the central server, we first generate the candidate set $\mathcal{U}$. For each item u in the set $\mathcal{U}$, we go through the N local DA sketches $\mathcal{D}_i$, and query the unbiased estimated frequency $\hat{f}_{(i,u)}$ of u in the data stream $\mathcal{S}_i$ using the method described in Section III-B[7], summing these query results to obtain its estimated frequency $\hat{f}_u = \sum_{i=1}^{N} \hat{f}_{(i,u)}$. Then, we can consider the K items u with the largest estimated frequencies $\hat{f}_u$ as the global top-K frequent items.

Brief Proof of Top-K-fairness: First, let's revisit two important properties of the DA sketch $\mathcal{D}_i$: $E\left(\hat{f}_{(i,u)} \mid u \in \mathcal{T}_i\right) = f_{(i,u)}$ and $E\left(\hat{f}_{(i,u)} \mid u \notin \mathcal{T}_i\right) = f_{(i,u)}$, where the former represents double anonymity and the latter directly results from the combination of double anonymity and unbiasedness. Therefore, for the data stream $\mathcal{S}_i$, if u appears in $\mathcal{S}_i$ and $u \in \mathcal{T}_i$, then $E\left(f_{(i,u)}\right) = f_{(i,u)}$; if u appears in $\mathcal{S}_i$ and $u \notin \mathcal{T}_i$, then $E\left(f_{(i,u)}\right) = f_{(i,u)}$ as well; if u does not appear in $\mathcal{S}_i$, then $E\left(f_{(i,u)}\right) = 0$; thus, we have $E\left(\hat{f}_u \mid u \in \mathcal{U}\right) = f_u$. This satisfies top-$K$ fairness with $\alpha = 1$ and $\delta = 0$.

B. Top-K-fairness for Rankings

In Section III, we discussed in detail the top-K-fairness in frequency estimation. In this section, we delve deeper into the fairness of the process of obtaining global top-K items through ranking. During the ranking process, when comparing the estimated frequencies of two items, not only the mathematical expectation of the estimated frequency affects fairness, but its distribution also impacts fairness. As shown in Figure 6[8], in case 1, even through both distributions have the same expectation, there is more than a 50%[9] chance that a value from orange distribution is larger than a value from the blue distribution during comparison; whereas in case 2, this probability is exactly 50%. Therefore, we can say that the comparisons in case 2 are fairer than those in case 1. Of course, we need a more formalized definition of fairness under ranking. We attempt to define the following ranking-fairness, a stricter form of top-K-fairness.

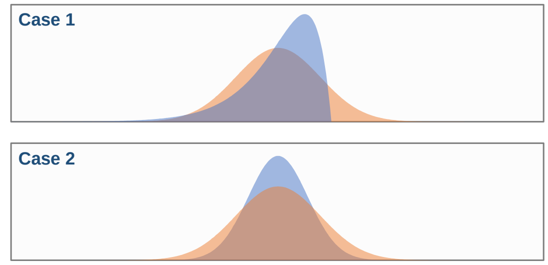


Fig. 6: Case studies of two distributions. The second case is fairer in terms of ranking.

Definition 7. *(**Ranking-fairness**) Given any two data stream $\mathcal{S}_{i_1}$ and $\mathcal{S}_{i_2}$, and any two items $u_{(i_1,j_1)}$ and $u_{(i_2,j_2)}$, if $f_{(i_1,j_1)} \geq \lambda f_{(i_2,j_2)}$, then for any monotonically non-decreasing function g, it holds that*

$$E\left[g\left(\hat{f}_{(i_1,j_1)} - \lambda \hat{f}_{(i_2,j_2)}\right)\right] \geq E\left[g\left(\lambda \hat{f}_{(i_2,j_2)} - \hat{f}_{(i_1,j_1)}\right)\right].$$

Intuitively, we hope after calculating $\hat{f}_{(i_1,j_1)} - \lambda \hat{f}_{(i_2,j_2)}$, regardless of the standard g used to judge this difference, the central analyzer will be more inclined to choose $\hat{f}_{(i_1,j_1)}$ over $\lambda \hat{f}_{(i_2,j_2)}$. Additionally, ranking-fairness implies the following properties:

- Let $g(x) = \mathbf{1}_{x>0}$, then it can be deduced that if $f_{(i_1,j_1)} \geq f_{(i_2,j_2)}$, then $\Pr\left(\hat{f}_{(i_1,j_1)} > \hat{f}_{(i_2,j_2)}\right) \geq \Pr\left(\hat{f}_{(i_2,j_2)} > \hat{f}_{(i_1,j_1)}\right)$.
- Let $g(x) = \mathbf{1}_{x>0}$, then it can be deduced that if $f_{(i_1,j_1)} = f_{(i_2,j_2)}$, then $\Pr\left(\hat{f}_{(i_1,j_1)} > \hat{f}_{(i_2,j_2)}\right) = \Pr\left(\hat{f}_{(i_2,j_2)} > \hat{f}_{(i_1,j_1)}\right) = 0.5$.
- Let $g(x) = x$, then it can be deduced that $E[\hat{f}_{(i,j)}] = \alpha \times f_{(i,j)} + \delta$. This indicates that ranking-fairness is a stricter version of top-K-fairness.

The proofs of the first two properties are obvious, and here we provide the proof for the last property.

Proof. We first set $g(x) = x, \lambda = 1$, and find two items $u_{(i_1,j_1)}$ and $u_{(i_1,j_1)}$ such that $f_{(i_1,j_1)} = f_{(i_2,j_2)}$, we then can obtain

$$\begin{aligned} &E\left[\hat{f}_{(i_1,j_1)} - \hat{f}_{(i_2,j_2)}\right] \geq E\left[\hat{f}_{(i_2,j_2)} - \hat{f}_{(i_1,j_1)}\right] \\ \Rightarrow &E\left[\hat{f}_{(i_1,j_1)}\right] \geq E\left[\hat{f}_{(i_2,j_2)}\right]. \end{aligned}$$

Similarly, we also have $E\left[\hat{f}_{(i_1,j_1)}\right] \leq E\left[\hat{f}_{(i_2,j_2)}\right]$, therefore we can conclude $E\left[\hat{f}_{(i_1,j_1)}\right] = E\left[\hat{f}_{(i_2,j_2)}\right]$. This indicates that $E\left[\hat{f}_{(i,j)}\right]$ can be considered a function of $f_{(i,j)}$, that is, $E\left[\hat{f}_{(i,j)}\right] = h\left(f_{(i,j)}\right)$.

Next, we proof that $h\left(f_{(i,j)}\right)$ is a linear function. We find two items $u_{(i,j_1)}$ and $u_{(i_1,j_1)}$ satisfying $f_{(i_1,j_1)} = \lambda f_{(i_2,j_2)}$, then through a process similar to the above, we can obtain $E\left[\hat{f}_{(i_1,j_1)}\right] = \lambda E\left[\hat{f}_{(i_2,j_2)}\right]$, that is, $h\left(f_{(i_1,j_1)}\right) = h\left(\lambda f_{(i_2,j_2)}\right) = \lambda h\left(f_{(i_2,j_2)}\right)$. Consider the arbitrariness of $f_{(i_2,j_2)}$, we have $h\left(\lambda f_{(i,j)}\right) = \lambda h\left(f_{(i,j)}\right)$. By differentiating

[7] Note that even if item u does not appear in data stream $\mathcal{S}_i$, it can still obtain a non-zero estimated frequency $\hat{f}_{(i,u)}$ through the sketch $\mathcal{D}_i$.

[8] The orange distribution is $X \sim N(4, 10)$, the blue distribution in case 1 is $Y \sim \Gamma(2, 2)$, and in case 2, the blue distribution is $Z \sim N(4, 5)$.

[9] $\Pr(X > Y) \approx 0.525 > 0.5$, $\Pr(X > Z) = 0.5$.

both sides of this equation, we can get

$$\frac{\partial h\left(\lambda f_{(i,j)}\right)}{\partial f_{(i,j)}} = \lambda h'\left(\lambda f_{(i,j)}\right) = \lambda h'\left(f_{(i,j)}\right) = \frac{\partial h\lambda\left(f_{(i,j)}\right)}{\partial f_{(i,j)}}$$

Due to the arbitrariness, we know $h'\left(f_{(i,j)}\right)$ is a constant function, which means $h'\left(f_{(i,j)}\right) = \alpha$, thus we have $h\left(f_{(i,j)}\right) = \alpha \times f_{(i,j)} + \delta$. □

Sufficient Condition: After defining ranking-fairness, a question arises: what kind of algorithms satisfy ranking-fairness? We have identified a sufficient condition, namely that *the distribution of estimated frequencies of items given by the algorithm is symmetric with respect to their true frequencies*, *i.e.*, $\Pr\left(\hat{f}_{(i,j)} > f_{(i,j)} + \delta\right) = \Pr\left(\hat{f}_{(i,j)} < f_{(i,j)} - \delta\right)$. We briefly summarize its proof.

Proof. If the distributions of $\hat{f}_{(i_1,j_1)}$ and $\hat{f}_{(i_2,j_2)}$ are both symmetric, then the distribution of $X = \hat{f}_{(i_1,j_1)} - \lambda\hat{f}_{(i_2,j_2)}$ is also symmetric, *i.e.*, $F(E(X)+x) = 1 - F(E(X)-x)$, where $F(x)$ is the cumulative distribution function (CDF). If $E(X) > 0$, *i.e.*, $f_{(i_1,j_1)} - \lambda f_{(i_2,j_2)} \geq 0$, then $g(x+E(x)) - g(x-E(X)) > 0$, and

$$E(g(X)) - E(g(-X)) = \int g(x)dF(x) - \int g(-x)dF(x)$$

$$= \int (g(x+E(x)) - g(x-E(X)))dF(x+E(x)) \geq 0.$$

□

Symmetric Distribution Version of DA Sketch: We only need to replace the count part of the DA sketch with the C sketch [15], an unbiased courting sketch that provides symmetric distribution of estimated frequencies, to make the estimated frequency distribution of DA sketch symmetric and thus achieving ranking-fairness. The data structure of the C sketch is completely identical to the CMM sketch, except that each array of counters is also associated with a hash function $h_i(\cdot)$, which maps each item uniformly and randomly to $\{-1, 1\}$. During insertion, the C sketch increments the mapped counter $\mathcal{A}[i, g_i(e)]$ by $h_i(e)$. For querying, the C sketch use the median $median\{h_1(e)\mathcal{A}[1, g_1(e)], \cdots, h_d(e)\mathcal{A}[d, g_d(e)]\}$ of the d mapped counters as the unbiased estimated value.

Analysis: Proving that *the distribution of the estimated frequencies for any item e in the DA sketch with C sketch is symmetric* is trivial: in any given sketching process $\mathcal{R}$, the randomness of the estimated frequency of item e is entirely derived from the C sketch, hence its frequency distribution is symmetric. Thus, considering all possible $\mathcal{R}$, the distribution of e's estimated frequency remains symmetric about its true frequency. We have empirically validated in Section VI-F that the DA sketch with C sketch outperforms the DA sketch with CMM sketch under several ranking-related metrics.

VI. Experimental Results

A. Experimental Setup

*1) **Implementation**:* We have implemented the DA sketch and all other algorithms in C++. We apply four replacement strategies to the DA sketch: Randomized Admission Policy (RA) [16], SpaceSaving (SS) [17], Frequent (Freq) [34] and HeavyGuardian (HG) [4]. We find that applying RA Policy yields the best results; therefore, we mainly demonstrate the experimental results of DA sketch + RA. We also compare our results with state-of-the-art top-K sketching algorithms: Frequent [34], SpaceSaving [17], Unbiased SpaceSaving (USS) [6] and WavingSketch (Waving) [3]. Our source code is publicly available at Github [58].

*2) **Datasets**:* We use three real-world datasets and one synthetic dataset. The details are shown below: 1) The *IP Trace Dataset* (CAIDA) [41] consists of streams of anonymous IP traces collected by CAIDA in 2016. Each item is identified by its 13-byte "5-Tuple". We use the first 20M items. 2) The *Web Page Dataset* [59] is built from a collection of web pages downloaded from the website. Each item is 4 bytes long. 3) The *Network Dataset* [60] consists of users' posting history on the StackExchange website. 4) We generated *Synthetic Dataset* following the Zip-f distribution [42]. Each dataset contains 32M items, and each item is 4 bytes long. Here we use the generated dataset with skewness=0.6.

*3) **Metrics**:*

Average Relative Error (ARE): $\frac{1}{|\mathbf{\Psi}|}\sum_{e_i\in\mathbf{\Psi}}\frac{|f_i-\hat{f}_i|}{f_i}$, where f_i is the ground truth frequency of item e_i, $\hat{f}_i$ is its estimated frequency, and $\mathbf{\Psi}$ is the query set.

F1 Score: $\frac{2*CR*PR}{CR+PR}$, where PR (Precision rate) represents the proportion of the correctly selected items among all the selected items, and CR (Recall rate) represents the proportion of the correctly selected items among all the real top-K items.

Throughput: The number of operations (insertions) in million per second (Mops). It indicates the overall speed of insertion.

Zero Error Rate: The proportion of items selected by our sketch whose estimated frequency is guaranteed to be exactly the same as its ground truth frequency.

Relative Bias: This metric is used in section VI-D. For the local sketch i, the relative bias is defined $\frac{\sum_{e_j\in\Psi}\hat{f}_j}{\sum_{e_j\in\Psi}f_j}$, where Ψ is the set of items that local sketch i returns as the local top-K items.

Recall on Aggregation: $\frac{||\{(\hat{\mathcal{T}}_i\cap\mathcal{T})\cap\hat{\mathcal{T}}\}||}{||\{\hat{\mathcal{T}}_i\cap\mathcal{T}\}||}$ for local sketch i, where $\mathcal{T}$ denotes the set of global top-K items, $\hat{\mathcal{T}}$ denotes the set of *predicted* global top-K items (after aggregation), and $\hat{\mathcal{T}}_i$ denotes the selected local top-K items from sketch i.

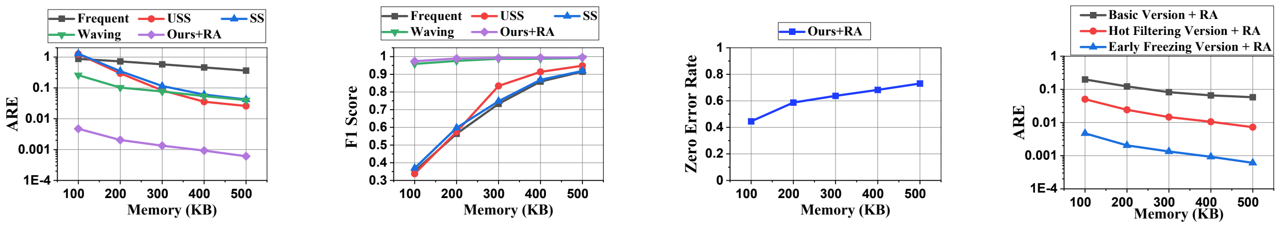
*4) **Common Settings**:* Let Mem denote the total amount of memory allocated to the sketches, M_{top-K} denote the amount of memory allocated to the top-K part for the DA sketch, K denote that we query the top-K frequent items, and λ denotes the number of cells in each bucket of the top-K part. For the DA sketch, we set $\lambda = 8$, $\frac{M_{top-K}}{Mem} = 0.55$. For DA sketch, the size of count part's buckets is set to be 2 bytes[10]. All other parameters of the baseline top-K algorithms are set according to the recommendations of their authors.

B. Experiments on Local Top-K

Experimental Settings: In this experiment, we use the CAIDA dataset. We set $K = 1000$, and range the memory size from 100KB to 500KB for all sketches to see how different sketches perform in different amounts of memory.

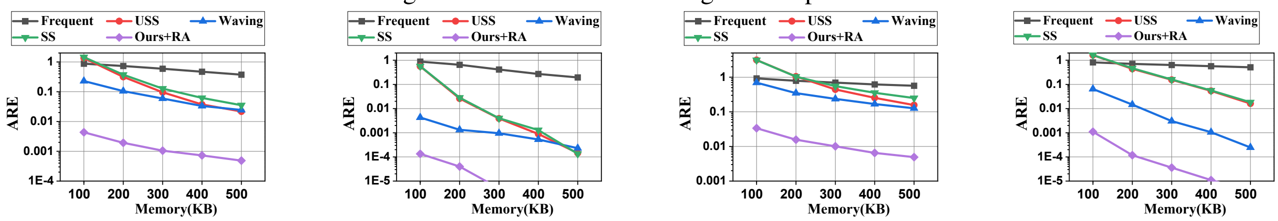
ARE (Figure 7(a)): Results show that our approach achieves much more accurate estimation thanks to the hot filtering and

[10]For the basic version, the size of count part's buckets is set to be 4 bytes.

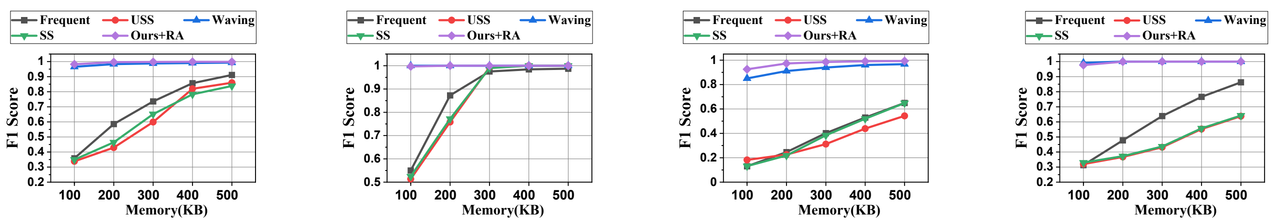


(a) CAIDA: ARE (b) CAIDA: F1 Score (c) Zero Error Rate (d) ARE of three versions

Fig. 7: Performance of finding local top-K items.



(a) CAIDA: ARE (b) Webpage: ARE (c) Net: ARE (d) Synthesis: ARE



(e) CAIDA: F1 Score (f) Webpage: F1 Score (g) Net: F1 Score (h) Synthesis: F1 Score

Fig. 8: Performance of finding global top-K items.

early freezing technique. When $Mem = 100$KB, our approach is around 500-1000 times more accurate than USS, SS, and Frequent and around 50-100 times more accurate than Waving.

F1 Score (Figure 7(b)): When applying RA to our approach, the DA sketch achieves sufficiently high F1 Score ($\geq 95\%$) even when memory is extremely tight. This is because for the DA sketch, local top-K items' selection is determined by only the replacement policy, and RA itself is accurate in selecting local top-K items. In contrast, Frequent, USS, and SS are much more inaccurate in finding top-K items. The discussion will be further elaborated in section VI-C.

Zero Error Rate (Figure 7(c)): We demonstrate the proportion of items of which we are confident that frequency estimation error is guaranteed to be 0 (as denoted by zero error rate). We could determine this because $C_{freezing} = 0$ indicates that such item has never been evicted from the Top-K part throughout the process. The results show that our approach achieves a zero error rate greater than 40% when memory is as tight as 100KB, and greater than 72% when $Mem = 500$KB. The results suggest that for the majority of items, our algorithm could tell with 100% confidence that their estimated frequencies are perfectly accurate.

Comparison between the three versions (Figure 7(d)): We find that both the hot filtering and early freezing significantly improve the accuracy of our unbiased frequency estimation. On average, the final version — the early freezing version is approximately 66 times more accurate than the first version — the basic version and approximately 10 times more accurate than the second version — the hot filtering version.

C. Experiments on Global Top-K with Same Sizes across Different Data Streams

Application Description: In a distributed scenario, there are N data streams $\mathcal{S}_1, \cdots, \mathcal{S}_N$. Data stream $\mathcal{S}_i$ contains m_i items. Each data stream is measured by a sketch on one machine. Memory sizes of all the sketches on different data streams are set the same. We denote $\mathcal{S} = \bigcup_{i=1}^{N} \mathcal{S}_i$. In different scenarios, the skewness of the size distribution across different data streams could be small or large. We set $m_1 = r * |\mathcal{S}|$, and $m_i = \frac{1-r}{N-1}|\mathcal{S}|, i \geq 2$, where $r \geq \frac{1}{N}$ represents the skewness of the size distribution across different data streams. We denote $\mathcal{S}_1$ as a heavy stream, and other data streams as light streams. In this subsection, we focus on the case when the sizes of different data streams are the same, *i.e.*, $r = \frac{1}{N}$.

Experimental settings: We use all the four datasets mentioned in VI-A2 for our experiments. There are in total $N = 10$ data streams, and we select $K = 1000$ global top-K items. We allocate the same amount of memory for each sketch on different machines, and the total memory size for the $N = 10$ sketches in total ranges from 100KB to 500KB.

ARE (Figure 8(a) - 8(d)): We find that our approach could achieve much lower ARE than prior art. On CAIDA dataset, when $Mem = 100$KB, ARE of our approach is 3 orders of magnitude times lower than Frequent, USS, SS, and 70 times lower than Waving. We observe similar results on the other three datasets.

F1 Score (Figure 8(e) - 8(h)): Results show that in this scenario, our approach could achieve a high F1 Score on both datasets even when Mem is small. When $Mem = 100$KB, the F1 Score of our approach is greater than 90% on both datasets, while the F1 Score of Frequent, USS, and SS is lower than 60% on the Webpage dataset and lower than 40% on the rest of the datasets. We also find that our approach achieves a slightly better F1 Score than Waving .

Throughput (Table I): Our approach achieves higher or comparable throughput compared with prior art. Specifically, the throughput of our approach is on average 3.19, 2.89 and 3.15 times higher than Frequent, USS, and SS respectively over the four datasets, and is comparable with Waving.

	CAIDA	Webpage	Net	Synthesis
Frequent (300KB)	5.3	6.2	4.5	5.1
USS (300KB)	5.4	6.9	5.3	5.7
SS (300KB)	5.9	6.4	4.8	4.7
Waving (300KB)	14.8	21.2	**13.4**	**16.8**
Ours + RA (300KB)	**14.9**	**25.5**	12.7	15.6

TABLE I: Throughput (Mops) of finding top-K frequent items.

Analysis: 1) Our approach is accurate in frequency estimation on global top-K items. Prior works, like Frequent, USS, and SS tend to provide highly underestimated or overestimated frequency estimation, so their frequency estimation tends to be significantly inaccurate. 2) F1 Score of our approach is mainly determined by the top-K replacement strategy, and when applying RA replacement strategy, the DA sketch could achieve a high F1 Score. F1 Score of Frequent, USS, and SS is significantly lower than our approach since all of them use the Stream Summary [17], which consumes more memory to store one item than our approach, and those strategies are not as accurate as the RA. 3) Both our approach and Waving sketch use bucket-array data structure, which is cache-friendly and requires fewer memory access, resulting in higher insertion throughput. For Frequent, USS, and SS, frequent pointer operations would lead to cache misses, making the insertion much slower.

D. Experiments on Top-K-fairness with Highly Skewed Data Streams' Sizes

*1) **Experimental Setup**:*

In this subsection, we focus on the case when the size distribution is highly skewed. We show why top-K-fairness is important in finding global top-K items in this scenario. We compare our results with four biased algorithms: Frequent, SS, HG, and RA, and two unbiased algorithms: USS, and Waving. **F1 Score** is used to demonstrate the overall performance. **Relative bias** is used to demonstrate the top-K-fairness of our approach and the top-K-unfairness of prior art. Considering the global top-K aggregation: before that, sketch i proposes several local top-K candidates, and some of them are real global top-K. Among those real global top-K proposed by sketch i, only a proportion of them are selected as global top-K. **Recall on aggregation**, which refers to the proportion mentioned above, is used to demonstrate the top-K-fairness of the global top-K selector on aggregation. Specifically, we use this metric to answer our questions: *does the global top-K selector favors items from heavy machines or from light machines, or is the global top-K-fair so that it selects global top-K items solely based only on their real frequency.*

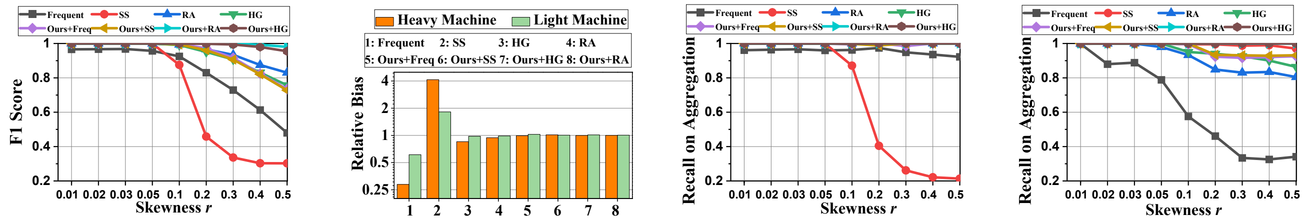
Experimental Settings: We set $N = 100$, $K = 1000$, and vary the skewness r from 0.01 to 0.5. In order to eliminate the effects of selecting local top-K itself on the performance of finding global top-K, we adjust the memory sizes for different algorithms so that they could store exactly the same number of local top-K candidates. For Frequent, SS, and USS, we use 40KB; for HG, RA, Waving, and DA sketch, we use 15KB. We use the synthetic dataset with skewness=0.9, which is relatively low in skewness, to better demonstrate the concept of ”top-K-fairness”.

*2) **Overall Performance & Top-K-fairness**:*

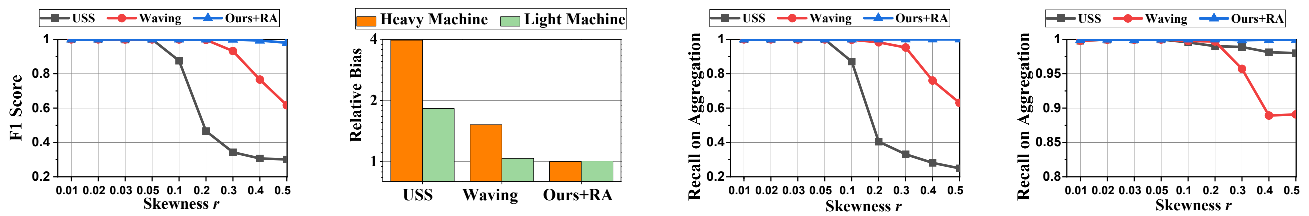
F1 Score (Figure 9(a) and 9(e)): Results show that when skewness increases, our F1 Score degradation is much slower than all the prior art. Specifically, when skewness $r = 0.5$, Ours + Frequent achieves F1 Score $\geq 73\%$, while Frequent itself only achieves F1 Score $\leq 48\%$; Ours + SS achieves F1 Score $\geq 72\%$, while SS itself only achieves F1 Score $\leq 31\%$. Ours + RA achieves F1 Score $\geq 98\%$, while RA itself only achieves F1 Score $\leq 83\%$. Ours + HG achieves F1 Score $\geq 95\%$, while HG itself only achieves F1 Score $\leq 76\%$. F1 Score of Waving Sketch and USS is 62%, 30% respectively, which is also significantly lower than that of our approach.

Relative Bias on Top-K items (Figure 9(b) and 9(f)): Results show that SS, USS and Waving tend to provide overestimated frequency. For these algorithms, items in heavy machines tend to be overestimated much more than light machines, so the global top-K selector tends to favor items in heavy machines. Similarly, Frequent, RA, and HG tend to provide underestimated frequency, and items in heavy machines tend to be underestimated much more, so the global top-K selector tends to favor items in light machines. More detailed recall rates on aggregation are shown in Section 5.4.3.

Analysis: 1) One of the desired properties that top-K-fairness brings is that the F1 Score of top-K-fair algorithms, like our DA sketch, tends to be higher than top-K-unfair algorithms. For example, for SS and USS, local top-K candidates in heavy machines tend to be highly overestimated, so even if an item is low in real frequency, its estimated frequency is still high enough to be falsely selected as a global top-K. With items in heavy machines falsely selected as global top-K items and items in light machines ignored, the F1 Scores of SS and USS become unacceptably low when skewness is large. 2) The degree of top-K-unfairness of algorithms is often negatively related to their F1 scores. Specifically, the top-K-unfairness of SS, USS, and Frequent is very significant, so their F1 scores are lower than other algorithms. Although Waving, RA, and HG are also top-K-unfair, their top-K-unfairness is relatively slight, so they have higher F1 scores. For top-K-fair algorithms, the accuracy of the replacement policy they use determines their performance, so Ours+RA and Ours+HG have the highest F1 scores. 3) Our approach is generic: we can make *any* top-K algorithm top-K-fair simply by applying the DA sketch to this top-K algorithm. Meanwhile, the F1 Score is also much improved. Specifically, for Frequent and SS with significant top-K-unfairness, DA sketch can improve their F1 scores by up to 25.5% and 42.5%; and for RA and HG with slight top-K-unfairness, DA sketch can still improve their F1



(a) Overall Performance: F1 Score (b) Top-K-fairness: Relative Bias (c) Recall on Aggregation of light machines (d) Recall on Aggregation of heavy machines



(e) Overall Performance: F1 Score (f) Top-K-fairness: Relative Bias (g) Recall on Aggregation of light machines (h) Recall on Aggregation of heavy machines

Fig. 9: Performance and fairness for finding global top-K items comparing our approaches with baseline algorithms.

scores by 15.0% and 19.8%.

3) Recall on Aggregation:

Recall on Aggregation (Figure 9(c) - 9(d) and 9(g) - 9(h)): For light machines, we find that Recall on Aggregation of overestimation algorithms, like SS, USS, and Waving, decreases fast as r increases, while that of other algorithms keeps at a high level ($\geq 90\%$). Conversely, for heavy machines, Recall on Aggregation of underestimation algorithms like Frequent, RA, and HG, decreases as r increases, while other algorithms remain $\geq 90\%$. It can be concluded that for overestimation algorithms, it is more difficult for items in light machines to be selected as global top-K items; for underestimation algorithms, it is more difficult for items in heavy machines to be selected as global top-K items.

Analysis: Top-K-fairness is determined by the bias of frequency estimation. For overestimation sketches like SS, USS, and Waving, many local top-K candidates from light machines that are supposed to become global top-K items would actually be evicted during aggregation. It can be concluded that the global top-K selector favors items from heavy machines. Conversely, for underestimation sketches like Frequent, RA, and HG, global top-K selector tends to favors items from light machines. We argue that top-K-unfair aggregation is unacceptable since the global top-K selector should not be partial to items from any machine.

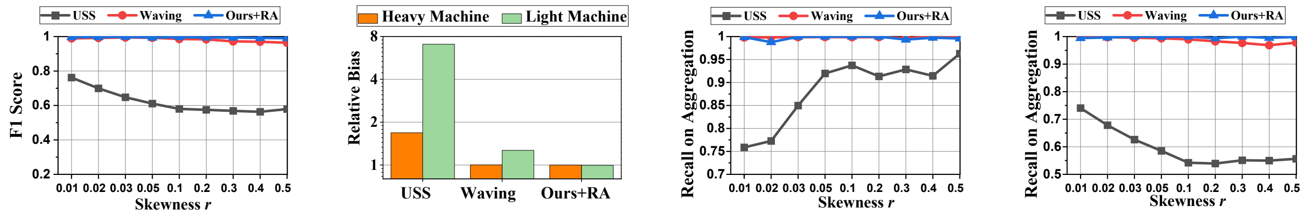
4) Other Baseline Algorithms:

Comparison algorithms: We compare two other baseline algorithms designed for skewed data streams: algorithms based on global sampling and algorithms based on weighting. For sampling algorithms, we use the same sampling rate for each data stream to sample items and send them to the global top-K selector. On the global top-K selector, we use sketch or directly use deterministic data structures (*e.g.*, maps) to find global top-K items in the sampled data stream. For weighted algorithms, we maintain sketch of different sizes on different machines according to the number of items contained in the data stream. Specifically, if the data stream on the heavy machine contains 10 times as many items as that on the light machine, the sketch size on the heavy machine is set to be 10 times as large as that on the light machine.

DA sketch *v.s.* weighted algorithms (Figure 10): We compare weighted USS, weighted Waving, and weighted Ours+RA. As shown in Figure 10(b), for weighted USS and weighted Waving, their overestimation on heavy machines is reduced, but their overestimation on light machines is significantly increased. This is due to the non-linear relationship between their overestimation and the size of the data stream. However, as shown in Figures 10(a), 10(c), and 10(d), weighting can indeed improve the performance of USS and Waving, especially when the distribution is particularly skewed. Specifically, when $r = 0.5$, the F1 score of weighted USS is 57.9%, that of weighted Waving is 96.4%, and that of weighted Ours+RA is 99.0%.

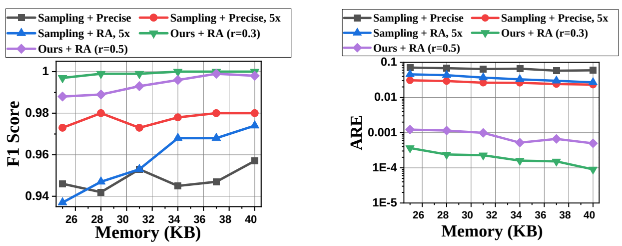
DA sketch *v.s.* sampling algorithms (Figure 11): We compare with the sampling algorithms using different sampling rates and different global data structures on global top-K selectors. The sampling rate we default to ensures that the amount of data transmitted to the global analyzer from all machines is equal to the total memory usage of the Ours+RA sketch data structures on all machines. For example, a sampling rate of $p = 0.04$ would transfer 3.2MB of data, whereas using a 32KB Ours+RA sketch would require a total memory of $100 \times 32\text{KB} = 3.2\text{MB}$. The label "5×" indicates that we have used a sampling rate that is five times the default value. "Sampling+Precise" indicates that we use a precise data structure, such as a hash table, on the global analyzer to count global frequency information and find global top-K elements; "Sampling+RA" means we use an RA sketch on the global analyzer to find global top-K elements. The experimental results show that higher sampling rate means higher accuracy, but the performance of the sampling algorithm using "5×" sampling rate and precise global data structure is still inferior to Ours+RA.

Analysis: For the two comparison algorithms, the sampling algorithms are top-K-fair, and the weighted algorithms can indeed improve the performance. However, our algorithm still



(a) Overall Performance: F1 Score (b) Top-K-fairness: Relative Bias (c) Recall on Aggregation of light machines (d) Recall on Aggregation of heavy machines

Fig. 10: Performance and fairness of weighted algorithms for finding global top-K items.



(a) F1 Score (b) ARE

Fig. 11: Comparisons between the sampling approach and our approach.

shows its superiority over the two algorithms. In addition, there is another artificial weighted algorithm: manually correct the overestimation or underestimation of reported top-K items from different data streams. On the one hand, this algorithm is difficult to practice, and on the other hand, it cannot achieve the exact top-K-fairness.

E. Experiments on Parameter Settings

In order to find the optimal parameter settings, we conduct experiments on finding local top-K items and vary λ and $\frac{M_{top-K}}{Mem}$ to see how AAE, ARE, F1 Score and Throughput change. We set Mem to be 100KB, λ to range from 1 to 64, and $\frac{M_{top-K}}{Mem}$ to range from 0.05 to 0.95.

Varying λ (Figure 12(a)-12(b)): We find that, as λ increases from 1 to 64, ARE of Our+RA and Ours+SS first decreases when λ grows from 1 to 8 and then remains steady. For Ours+HG and Ours+Freq, ARE keeps roughly steady. However, as λ increases, the throughput of all DA sketch applications drops severely. Therefore in practice, we choose $\lambda = 8$ as the best setting.

Varying $\frac{M_{top-K}}{Mem}$ (Figure 12(c)-12(d)): We find that F1 scores grow as $\frac{M_{top-K}}{Mem}$ increases, since F1 scores are only determined by the top-K part. However, we find that when $\frac{M_{top-K}}{Mem} \geq 0.55$, growth rate of F1 scores becomes slow[11]. Besides, Ours+HG and Ours+RA reach their respective minimal ARE score when $\frac{M_{top-K}}{Mem} \approx 0.55$, while Ours+Freq and Ours+SS reach their minimal ARE when $\frac{M_{top-K}}{Mem} \approx 0.75$. In practice, we choose $\frac{M_{top-K}}{Mem} = 0.55$ as the default parameter setting.

Analysis: 1) Among the four replacement policies, Ours+RA and Ours+HG often have higher performance than Ours+Freq and Ours+SS. Specifically, Ours+RA has more advantages in F1 score, while Ours+HG has more advantages in ARE. Considering that Ours+RA has higher throughput, we recommend using Ours+RA in practice. 2) However, although Ours+Freq and Ours+SS are slightly inferior in accuracy, Freq and SS are famous for their formal and comprehensive error theories and error bounds. Benefiting from their theories, we suggest that Ours+Freq and Ours+SS should be considered in scenarios where exact error guarantees are required.

F. Experiments on Ranking Fairness

In this section, we verify the impact on ranking fairness after replacing the CMM sketch [43] used in the counting part of the DA sketch with the C sketch [15]. To measure ranking fairness, we introduce two different rank correlation coefficients:

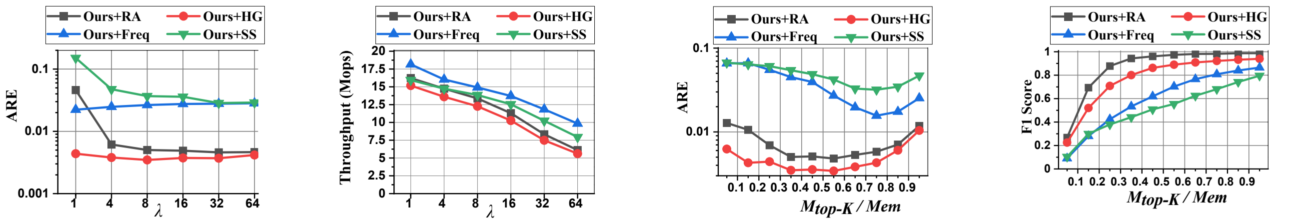
- Kendall's τ rank correlation coefficient: $\tau = \frac{n_1 - n_2}{\binom{|\hat{\mathcal{T}}|}{2}}$. Where n_1 and n_2 are the numbers of concordant and discordant pairs, respectively, in the global top-K set $\hat{\mathcal{T}}$ reported by the algorithm. A concordant pair $\langle u_i, u_j \rangle$ refers to $\hat{f}_i \geqslant \hat{f}_j$ and $f_i \geqslant f_j$ or $\hat{f}_i < \hat{f}_j$ and $f_i < f_j$, while a discordant pair refers to otherwise.
- Spearman's ρ rank correlation coefficient: $\rho = \frac{\sum (r_i - r_j)(s_i - s_j)}{(r_i - r_j)^2} = 1 - \frac{6 \sum_{i=1}^{|\hat{\mathcal{T}}|} d_i^2}{n(n^2-1)}$. Where r_i is the true ranking of item u_i, and s_i is the ranking of item u_i in the global top-K set $\hat{\mathcal{T}}$ reported by the algorithm, with $d_i = s_i - r_i$.

We vary the skewness of distributed data streams on synthetic datasets, CAIDA datasets, and Webpage datasets, and assess the τ and ρ rank correlation coefficients of different algorithms. It can be observed that the rank correlation coefficient of all algorithms decreases as the skewness of the data stream increases.

Kendall's τ coefficient (Figure 13(a), 13(e), 13(g)): For Ours+CMM and Ours+C, we can see that the τ coefficients of both algorithms consistently exceed 0.9; when the skewness of the data stream is 0.5, the τ coefficient of Ours+C is higher than that of Ours+CMM by 1.58%, 2.57%, and 1.65% on the three datasets respectively; although this number is small, taking the Webpage dataset as an example, τ increased from 0.958 to 0.974, and the number of discordant pairs decreased by 38.7%.

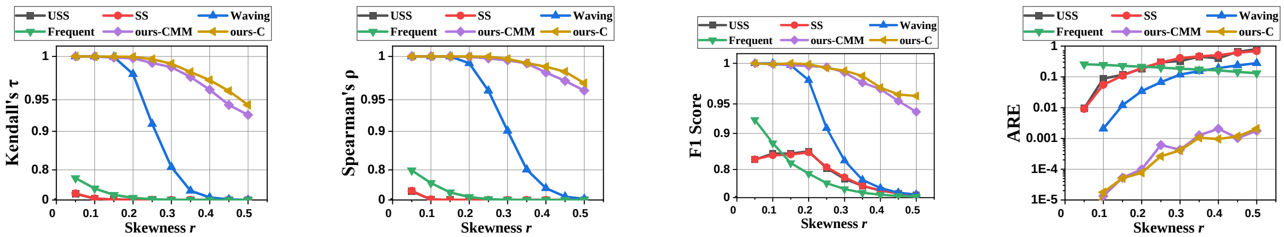
Spearman's ρ coefficient (Figure 13(b), 13(f), 13(h)): For Ours+CMM and Ours+C, we can see that the ρ coefficients of both algorithms consistently exceed 0.95; when the skewness of the data stream is 0.5, the ρ coefficient of Ours+C is higher than that of Ours+CMM by 1.32%, 0.94%, and 1.89% on the three datasets respectively; although this number is small, taking the Webpage dataset as an example, ρ increased

[11]In addition, in this experiment, $Mem = 100$KB is tight, and if Mem becomes larger, growth of F1 scores contributed by $\frac{M_{top-K}}{Mem}$ will become more negligible.

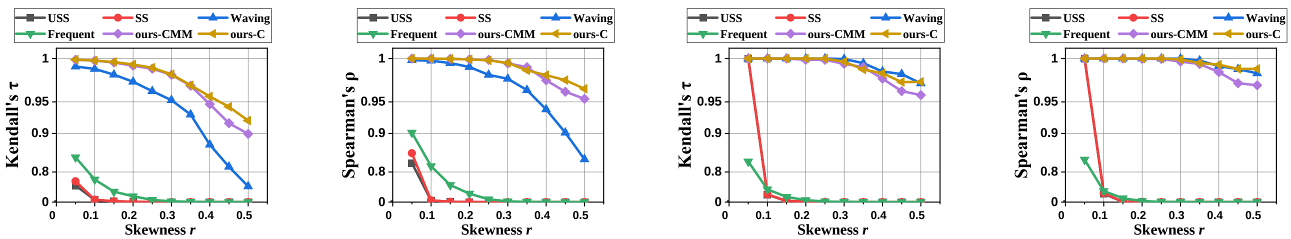


(a) Varying λ: ARE (b) Varying λ: Throughput (c) Varying $\frac{M_{top-K}}{Mem}$: ARE (d) Varying $\frac{M_{top-K}}{Mem}$: F1 Score

Fig. 12: Experiments on different parameter settings (λ and $\frac{M_{top-K}}{Mem}$) of Double-Anonymous Sketch.



(a) Synthesis: Kendall's τ (b) Synthesis: Spearman's ρ (c) Synthesis: F1 Score (d) Synthesis: ARE



(e) CAIDA: Kendall's τ (f) CAIDA: Spearman's ρ (g) Webpage: Kendall's τ (h) Webpage: Spearman's ρ

Fig. 13: Performance of ranking fairness.

from 0.971 to 0.989, and the sum of squared rank differences decreased by 63.4%.

An interesting phenomenon is that on the Webpage dataset, the τ and ρ coefficients of Waving are even the highest, which is because Waving is based on the C sketch [15] and naturally benefits from the symmetric distribution of frequency estimates.

F1 Score and ARE (Figure 13(c), 13(d)): It can be seen that the ARE of Ours+C and Ours+CMM does not show a significant difference, but the F1 score of Ours+C is higher than that of Ours+CMM; we believe this is because Ours+C achieves better ranking fairness, thus being able to more accurately identify the global Top-K items. As for other comparison algorithms, their F1 score trends are consistent with those in Figures 9(a) and 9(e).

G. Experiments on General Data Streams

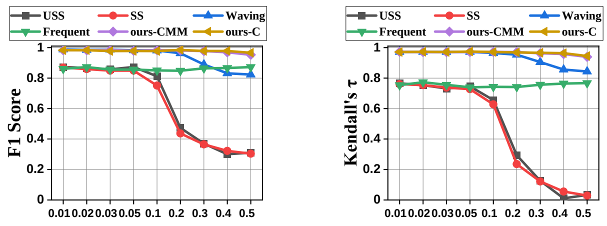
In this section, we demonstrate that the DA sketch still exhibits better performance under general distributed data stream settings. We construct general data streams based on synthetic datasets: for each item u, each occurrence is uniformly and randomly assigned to one of the $N = 100$ data streams. In this setup, the number of distinct items in each data stream is larger compared to disjoint data streams. Therefore, we allocate 160KB of memory to SS, USS, and Frequent, while the other algorithms use 60KB of memory.

F1 Score and Kendall's τ (Figure 14): As shown in Figure 14, when we gradually increase the skewness among the distributed data streams, the F1 scores and Kendall's τ coefficients of algorithms like Waving, SS, and USS significantly decrease, while ours+C and ours+CMM consistently maintain scores and coefficients close to 100%. An interesting finding is that the scores and coefficients of Frequent also remain almost unchanged. We believe this is because, in the design of the Frequent algorithm, the underestimation of the frequencies of local top-K items is almost proportional to the scale of the data stream, and the global top-K items generally appear in all data streams. Therefore, under the experimental setup, the underestimation of the frequencies of the global top-K candidates is proportional to the overall scale of the data stream and is almost independent of the skewness among the distributed data streams, leading to the almost unchanged F1 scores and Kendall's coefficients for the Frequent algorithm.

Variation of Parameters λ and M (Figure 15): As λ increases, the F1 score improves significantly before stabilizing, while the ARE decreases substantially and then stabilizes. When M increases, we reduce the memory allocated to the count part to maintain constant total memory. It can be observed that the performance of the DA sketch deteriorates when M is either too small or too large. The variation in the F1 score is not entirely consistent with Figure 12(d), as finding the global top-K depends more heavily on accurate frequency estimation than identifying the local top-K.

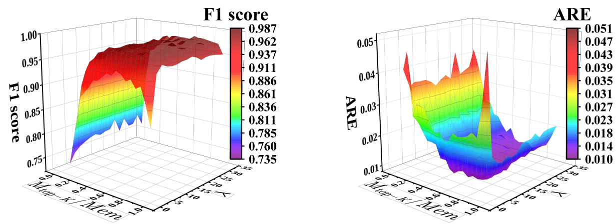
VII. Conclusion

In this paper, we propose the Double-Anonymous sketch, which is the first work that achieves top-K-fairness of global



(a) F1 Score (b) Kendall's τ

Fig. 14: Comparison under general distributed data stream.



(a) F1 Score (b) ARE

Fig. 15: Experiments on different parameter settings.

top-K. We theoretically prove that the DA sketch achieves both unbiasedness and double-anonymity, so as to achieve top-K-fairness. We conduct extensive experiments on three real and one synthetic dataset. Our experimental results show that compared with the state-of-the-art, our algorithm improves the accuracy 129 times.

Acknowledgment

This work is supported by National Key R&D Program of China (No. 2022YFB2901504), and National Natural Science Foundation of China (NSFC) (No. U20A20179, 62372009, 623B2005).

References

[1] Y. Zhao, W. Han, Z. Zhong, Y. Zhang, T. Yang, and B. Cui, "Double-anonymous sketch: Achieving top-k-fairness for finding global top-k frequent items," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–26, 2023.

[2] P. Wang, Y. Qi, Y. Zhang, and etal, "A memory-efficient sketch method for estimating high similarities in streaming sets," in *SIGKDD*, 2019.

[3] J. Li, Z. Li, Y. Xu, and etal, "Wavingsketch: An unbiased and generic sketch for finding top-k items in data streams," in *SIGKDD*, 2020.

[4] T. Yang, J. Gong, H. Zhang, and etal, "Heavyguardian: Separate and guard hot items in data streams," in *SIGKDD*, 2018.

[5] X. Gou, L. He, Y. Zhang, and etal, "Sliding sketches: A framework using time zones for data stream processing in sliding windows," in *SIGKDD*, 2020.

[6] D. Ting, "Data sketches for disaggregated subset sum and frequent item estimation," in *SIGMOD Conference*, 2018.

[7] Z. Haida, H. Zengfeng, W. Zhewei, and etal, "Tracking matrix approximation over distributed sliding windows," in *ICDE*, 2017.

[8] P. Roy, A. Khan, and G. Alonso, "Augmented sketch: Faster and more accurate stream processing," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 1449–1463.

[9] T. Yang, J. Jiang, P. Liu, and etal, "Elastic sketch: adaptive and fast network-wide measurements," in *SIGCOMM*, 2018.

[10] Y. Li, R. Miao, C. Kim, and etal, "Flowradar: a better netflow for data centers," in *NSDI*, 2016.

[11] H. Dai, M. Shahzad, A. X. Liu, and etal, "Finding persistent items in data streams," *VLDB Endowment*, 2016.

[12] S. Venkataraman, D. X. Song, P. B. Gibbons, and A. Blum, "New streaming algorithms for fast detection of superspreaders," in *NDSS*, 2005.

[13] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, 2005.

[14] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," *SIGMCOMM*, 2002.

[15] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Automata, Languages and Programming*. Springer, 2002.

[16] R. Ben-Basat, G. Einziger, R. Friedman, and etal, "Randomized admission policy for efficient top-k and frequency estimation," in *INFOCOM*, 2017.

[17] A. Metwally, D. Agrawal, and A. El Abbadi, "Efficient computation of frequent and top-k elements in data streams," in *ICDT*, 2005.

[18] T. Li, S. Chen, and Y. Ling, "Per-flow traffic measurement through randomized counter sharing," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1622–1634, 2012.

[19] P. Chen, D. Chen, L. Zheng, J. Li, and T. Yang, "Out of many we are one: Measuring item batch with clock-sketch." SIGMOD, 2021.

[20] A. Santos, A. Bessa, F. Chirigati, C. Musco, and J. Freire, "Correlation sketches for approximate join-correlation queries," in *SIGMOD*, 2021.

[21] R. Li, P. Wang, J. Zhu, J. Zhao, J. Di, X. Yang, and K. Ye, "Building fast and compact sketches for approximately multi-set multi-membership querying," in *SIGMOD*, 2021.

[22] Y. Izenov, A. Datta, F. Rusu, and J. H. Shin, "Compass: Online sketch-based query optimization for in-memory databases," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 804–816.

[23] Z. Zhong, S. Yan, Z. Li, D. Tan, T. Yang, and B. Cui, "Burstsketch: Finding bursts in data streams," in *SIGMOD*, 2021.

[24] D. Ting and R. Cole, "Conditional cuckoo filters," in *SIGMOD*, 2021.

[25] P. Pandey, A. Conway, J. Durie, M. A. Bender, M. Farach-Colton, and R. Johnson, "Vector quotient filters: Overcoming the time/space trade-off in filter design," in *SIGMOD*, 2021.

[26] G. Gupta, M. Yan, B. Coleman, B. Kille, R. L. Elworth, T. Medini, T. Treangen, and A. Shrivastava, "Fast processing and querying of 170tb of genomics data via a repeated and merged bloom filter (rambo)," in *SIGMOD*, 2021.

[27] B. Shi, Z. Zhao, Y. Peng, F. Li, and J. M. Phillips, "At-the-time and back-in-time persistent sketches," in *SIGMOD*, 2021.

[28] P. Jia, P. Wang, J. Zhao, S. Zhang, Y. Qi, M. Hu, C. Deng, and X. Guan, "Bidirectionally densifying lsh sketches with empty bins," in *SIGMOD*, 2021.

[29] Z. Dai, A. Desai, R. Heckel, and A. Shrivastava, "Active sampling count sketch (ascs) for online sparse estimation of a trillion scale covariance matrix," in *SIGMOD*, 2021.

[30] K. Zhao, J. X. Yu, H. Zhang, Q. Li, and Y. Rong, "A learned sketch for subgraph counting," in *SIGMOD*, 2021.

[31] Y. Zhao, K. Yang, Z. Liu, T. Yang, L. Chen, S. Liu, N. Zheng, R. Wang, H. Wu, Y. Wang *et al.*, "Lightguardian: A full-visibility, lightweight, in-band telemetry system using sketchlets." in *NSDI*, 2021, pp. 991–1010.

[32] Y. Zhao, Y. Zhang, P. Yi, T. Yang, B. Cui, and S. Uhlig, "The stair sketch: Bringing more clarity to memorize recent events," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 164–177.

[33] P. Chen, Y. Wu, T. Yang, J. Jiang, and Z. Liu, "Precise error estimation for sketch-based flow measurement," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021, pp. 113–121.

[34] G. Lukasz, D. David, D. E. D, L. Alejandro, and M. J. Ian, "Identifying frequent items in sliding windows over on-line packet streams," in *IMC*, 2003.

[35] M. G. Singh and M. Rajeev, "Approximate frequency counts over data streams," in *VLDB*, 2002.

[36] Y. Zhao, Z. Zhong, Y. Li, Y. Zhou, Y. Zhu, L. Chen, Y. Wang, and T. Yang, "Cluster-reduce: Compressing sketches for distributed data streams," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2316–2326.

[37] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," *SIGCOMM computer communication review*, 2001.

[38] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (bgp-4)," Tech. Rep., 2006.

[39] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 49–60.

[40] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with univmon," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 101–114.

[41] "Anonymized Internet Traces 2016," https://catalog.caida.org/dataset/passive_2016_pcap, 2016.
[42] D. M. Powers, "Applications and explanations of zipf's law," in *New methods in language processing and computational natural language learning*, 1998.
[43] F. Deng and D. Rafiei, "New estimation algorithms for streaming data: Count-min can do more," *Webdocs. Cs. Ualberta. Ca*, 2007.
[44] K. S. Tai, V. Sharan, P. Bailis, and etal, "Sketching linear classifiers over data streams," in *SIGMOD*, 2018.
[45] A. Shrivastava, A. C. Konig, and M. Bilenko, "Time adaptive sketches (ada-sketches) for summarizing data streams," in *SIGMOD*, 2016.
[46] Y. Peng, J. Guo, F. Li, and etal, "Persistent bloom filter: Membership testing for the entire history," in *SIGMOD*, 2018.
[47] T. Nan, C. Qing, and M. Prasenjit, "Graph stream summarization: From big bang to big crunch," in *SIGMOD*, 2016.
[48] Z. Wei, G. Luo, K. Yi, and etal, "Persistent data sketching," in *SIGMOD*, 2015.
[49] D. Ting, "Count-min: Optimal estimation and tight error bounds using empirical error distributions," in *SIGKDD*, 2018.
[50] P. Pandey, M. A. Bender, R. Johnson, and R. Patro, "A general-purpose counting filter: Making every bit count," in *SIGMOD*, 2017.
[51] A. D. Breslow and N. S. Jayasena, "Morton filters: Faster, space-efficient cuckoo filters via biasing, compression, and decoupled logical sparsity," 2018.
[52] M. Wang, M. Zhou, S. Shi, and etal, "Vacuum filters: More space-efficient and faster replacement for bloom and cuckoo filters," *VLDB Endowment*, 2019.
[53] G. Cormode and S. Muthukrishnan, "What's new: Finding significant differences in network data streams," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1219–1232, 2005.
[54] R. Schweller, Z. Li, Y. Chen, and etal, "Reversible sketches: enabling monitoring and analysis over high-speed data streams," *TON*, 2007.
[55] K. Balachander, S. Subhabrata, Z. Yin, and C. Yan, "Sketch-based change detection: methods, evaluation, and applications," in *SIGCOMM*, 2003.
[56] T. Yang, Y. Zhou, H. Jin, , and etal, "Pyramid sketch: A sketch framework for frequency estimation of data streams," *VLDB Endowment*, 2017.
[57] H. Zhang, Z. Liu, B. Chen, Y. Zhao, T. Zhao, T. Yang, and B. Cui, "Cafe: Towards compact, adaptive, and fast embedding for large-scale recommendation models," in *SIGMOD Conference*, 2024.
[58] "Source code related to double-anonymous sketch," https://github.com/Arimase97/Double-Anonymous-Sketch, 2023.
[59] "Real-life transactional dataset." http://fimi.ua.ac.be/data/, 2004.
[60] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

Yikai Zhao received the B.S. degree in computer science from Peking University in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science, Peking University, advised by Tong Yang. His research interest lies in the intersection of distributed systems and probabilistic algorithms. He published papers in SIGMOD, SIGCOMM, NSDI, *etc.*

Wei Zhou received his bachelor degree in Computer Science from Peking University in 2023. His research interests include data structures and algorithms in network measurement.

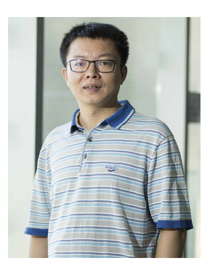
Wenchen Han received his B.S. degree in Intelligence Science and Technology from Peking University in 2022. He is currently a second-year PhD student in Computer Science at the University College London. under the supervision of Ran Ben Basat and Brad Karp. His research interests are on algorithmic designs for networking and distributed machine learning systems.

Zheng Zhong is an undergraduate student of Peking University. His research interests include Streaming processing, Bloom filters, and data structure. He published papers in SIGMOD and SIGKDD. He is also the reviewer of SIGKDD2022.

Yinda Zhang is a Ph.D. student at the University of Pennsylvania. He graduated with a bachelor's degree from Peking University and a master's degree from the University of Chicago. His research interests include network measurements and streaming algorithms.

Xiuqi Zheng received her bachelor degree from Peking University in 2024.

Tong Yang received the PhD degree in computer science from Tsinghua University in 2013. He visited the Institute of Computing Technology, Chinese Academy of Sciences (CAS). He is currently an Associate Professor with School of Computer Science, Peking University. His research interests include network measurements, sketches, IP lookups, Bloom filters, and KV stores. He has published more than ten papers in SIGCOMM, SIGKDD, SIGMOD, NSDI, etc.



Bin Cui is a professor in School of Computer Science and Director of Institute of Network Computing and Information Systems, at Peking University. His research interests include database system architectures, query and index techniques, big data management and mining. He is a fellow of IEEE, member of ACM and distinguished member of CCF.