Double-Anonymous Sketch: Achieving Top-K-fairness for Finding Global Top-K Frequent Items

ABSTRACT

Finding top-K frequent items has been a hot topic in data stream processing in recent years, which has a wide range of applications. However, most of existing sketch algorithms focuses on finding local top-K in a single data stream. In this paper, we work on finding global top-K in multiple disjoint data streams. We find that directly deploying prior sketch algorithms is often unfair under global scenarios, which will degrade the accuracy of global top-K. We define *top-K-fairness* and show that it is important for finding global top-K. To achieve top-K-fairness, we propose a new sketch framework, called the Double-Anonymous sketch. The process of finding global top-K items is similar to that of paper reviewing and democratic elections. In these scenarios, double-anonymity is often an effective strategy to achieve top-K-fairness. We also propose two techniques, hot panning, and early freezing, to further improve the accuracy. We theoretically prove that the Double-Anonymous sketch achieves top-K-fairnesswhile keeping high accuracy. We perform extensive experiments to verify top-K-fairness in the scenario of disjoint data streams. The experimental results show that the Double-Anonymous sketch's error is up to 129 times (60 times on average) smaller than the state-of-the-art. All the related source code is open-sourced and available at Github anonymously.

1 INTRODUCTION

1.1 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 [1–4], databases [5–7], networking [8, 9], and network security [10, 11]. Finding top-K frequent items refers to selecting K items with the largest number of frequencies/occurrences, and providing frequency estimation. In the era of big data, the speed and volume of data are growing explosively. Sketches [2–8, 12–30], 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.

For finding top-K frequent items, most of existing sketch algorithms focus on providing statistics over a single data stream [2, 3, 5, 7, 8, 12–16, 31, 32], while a few of them [2, 5] work on merging the statistics over multiple related data streams into one. In this paper, we provide the first sketch that can *compare* the statistics over different *disjoint data streams*. Specifically, given N disjoint data streams, how can we compare their own top-K and select the global top-K. Note that the sizes (volumes) of these data streams are often skewed in practice (*e.g.*, power law distribution) [33].

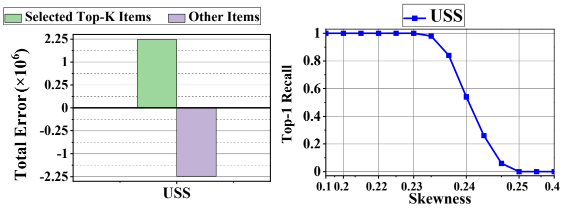
We use an example on network monitoring to explain the problem. For an autonomous system (AS) in a wide-area network (WAN), external traffic enters the AS through multiple border routers [34]. Due to the principle of WAN routing protocol [35], all network packets sent to the AS from the same source IP address must pass through the same border router. In other words, if we regard the source IP address of the network packets as the key, the network 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 IP addresses that send the most packets in a period of time [36]. To find these IP addresses, each border router reports the local top-K frequent source IP address and their frequency within this time period, and operator sorts all local frequent IP 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 to report the most frequent K items globally. However, we find that directly apply existing sketch algorithms for each data stream 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 based on their estimated frequency, 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 [16] to find local top-K items from the N data streams. SpaceSaving is a well-known sketch, which always provides overestimated estimation, and the degree of overestimation is positively correlated to the size of the data stream. 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 items, 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 2.1. When we achieve top-K-fairness, the accuracy of global top-K will rise significantly, especially if the sizes of data streams are highly skewed.

1.2 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 [2, 5]. Unbiasedness is helpful if we want to aggregate the statistics of multiple data streams for all items due to the *Law of Larger Numbers*. However, although the estimation is unbiased over all items, 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 the top-K selection process is not unbiased. In other words, if we select top-K items, we tend to select items which are overestimated, which leads to unfairness. We use two state-of-the-art unbiased sketches, Unbiased SpaceSaving (USS) [5] and WavingSketch (Waving) [2], to illustrate the problem.



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

Figure 1: We demonstrate internal unfairness of USS, and show how external unfairness would severely harm accuracy for finding global top-K items.

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), suppose the global top-1 item e_{top} is in a light data stream with a very small number of items, and we deploy a USS for each data stream. USS provides a slightly overestimated value for e_{top}, which is in the light data stream, and provides significantly overestimated value for frequent items in heavy data streams. As a result, when the size distribution of the distributed disjoint data streams is highly skewed, even the global top-1 item could be ignored, which is often unacceptable in practice. In Section 5.4, we also discuss that such unfairness cannot be alleviated by re-weighting the estimated frequency.

1.3 Our Proposed Solution

To achieve top-K-fairness, we propose a new sketch framework, called the Double-Anonymous sketch. We first propose a basic version which achieves top-K-fairness, and then we optimize the accuracy and throughput through two techniques *hot panning* and *early freezing*. The Double-Anonymous sketch has the following advantages: 1) It is the first work that discusses the fairness problem for comparing multiple disjoint data streams. 2) We provide a formal definition of top-K-fairness and *disjoint data streams* in Section 2.1. We also prove that our sketch can achieve top-K-fairness while keeping high accuracy as prior sketches. 3) It is accurate: The error (average relative error) of our sketch is up to 129 times (60 times on average) smaller than Waving and $3 \sim 4$ orders of magnitude smaller than Frequent, USS, and SS. 4) It is generic: we implement existing four *replacement strategies* in our framework to achieve top-K-fairness and accuracy.

The key technique of our Double-Anonymous sketch to achieve top-K-fairness is called **double-anonymity**. The process of finding global top-K items is similar to that of paper reviewing and democratic elections. Double-anonymity is often an effective strategy to achieve fairness. We leverage this strategy to enable top-K sketches to achieve top-K-fairness in global scenarios. 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 a top-K sketch 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 two unbiased sketches mentioned above do not meet the first condition, and thus are not double-anonymous. Our formal definition of double-anonymity is provided in Section 3.1. We theoretically prove that double-anonymity is a sufficient condition of top-K-fairness. Therefore, we follow this principle to design our solution.

In our basic version, we use a top-K sketch (*e.g.*, SpaveSaving [16]) as the top-K part and use an unbiased sketch (*e.g.*, CMM sketch [37]) 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. Note that the independent condition is stronger than double-anonymity. For an incoming item e, it will be inserted into the two parts independently and respectively. Obviously, our first version is double-anonymous, and thus achieves top-K-fairness.

However, although the first version achieves top-K-fairness, it fails to achieve high accuracy. Therefore, we propose two important optimization methods to significantly improve accuracy: **hot panning** 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. First, the main reason that brings large errors in the first version is information redundancy: the information of hot items is recorded in both two parts. The key idea of **hot panning** is that using the top-K part to pan the hot items, and only record them in the top-K part to remove such redundancy. More details are provided in Section 3.2. Second, the error of a sketch accumulates with more and more items inserted. The key idea of **early freezing** is that using a freezing counter to freeze the continuously accumulating error as early as possible, thus minimizing the error. More details are provided in Section 3.3. According to Section 5.2, the error of *early freezing version* is about 66 × lower than that of the basic version after applying hot panning and early freezing.

We show that the Double-Anonymous sketch is generic. Any replacement strategy independent with the CMM sketch can be applied to the Double-Anonymous sketch, and we choose four [3, 15, 16, 31] as case studies. We also show that the Double-Anonymous sketch is versatile. The Double-Anonymous sketch not only achieves top-K-fairness, but also provides both upper bound and lower bound for item frequency without additional data structures.

Key Contributions:

- We define a new important property: top-K-fairness. We define and analyze top-K-fairness and derive its sufficient condition.
- We propose the Double-Anonymous sketch, which is accurate, unbiased, and generic. The Double-Anonymous sketch is the first work that achieves top-K-fairness.
- We theoretically prove that the Double-Anonymous sketch achieves top-K-fairness and can keep high accuracy as prior sketches.
- We perform extensive experiments to verify top-K-fairness in the distributed scenario. We also show that the Double-Anonymous sketch's error is much smaller than other existing works.

2 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.

2.1 Formal Definitions and Preliminaries

Definition 2.1. ***(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 2.2. ***(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*

$$f_{(i,j)} = \sum_{k=1}^{m_i} \mathbf{1}_{\{e_{(i,k)}=u_{(i,j)}\}}.$$

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 $\mathcal{T}_i$ of local top-K items 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 the set $\mathcal{T} \subset \mathcal{U}$ of global top-K items.

Definition 2.3. ***(Top-K-fairness)*** *Given a top-K-fair algorithm, for any data stream $\mathcal{S}_i$, let $\mathcal{T}_i$ be the set of local top-K frequent items reported by $\mathcal{S}_i$, and 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. Our definition of fairness is inspired by these work, and adjusted to the scenario of disjoint data streams. We argue that top-K-fairness is an important property for algorithms in the task of finding global top-K items. It can avoid the influence of skewed data streams in the distributed scenarios: 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. If an algorithm achieves top-K-fairness, it means that its degree of overestimation or underestimation *for the selected top-K items* is a constant, *i.e.*, not related to the data stream. Our algorithm achieves top-K-fairness with $\alpha = 1$, $\delta = 0$.

2.2 Unbiasedness *v.s.* Top-K-fairness

Sketches [9, 10, 38, 38–49] 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 [16], Frequent [31], HeavyGuardian [3], Randomized Admission Policy [15], and etc [7, 8, 12, 13, 50]. 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 [2] claim to be unbiased. However, it should be noted that unbiased algorithms are not necessarily top-K-fair. We discuss why both USS and WavingSketch are top-K-unfair through some brief mathematical analysis. We first show the definition of unbiased algorithm.

Definition 2.4. ***(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*

$$E\left(\hat{f}_{(i,j)}\right) = f_{(i,j)} \qquad\qquad \forall u_{(i,j)} \in \mathcal{U}_i,$$

Unbiasedness *v.s.* top-K-fairness: The main difference between our top-K-fairness and unbiasedness is that the top-K-fairness has an additional condition that $u_{(i,j)} \in \mathcal{T}_i$. Take the **USS** for example. Although USS is an unbiased algorithm, it estimates the frequency of all non-top-K items as 0, *i.e.*,

$$E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \notin \mathcal{T}_i\right) = 0$$

$$E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = \frac{f_{(i,j)}}{\Pr\left(u_{(i,j)} \in \mathcal{T}_i\right)}$$

The amplification coefficient $\alpha = \frac{1}{\Pr(u_{(i,j)} \in \mathcal{T}_i)}$ varies largely among data streams, so USS cannot achieve top-K-fairness.

WavingSketch [2] achieves unbiasedness based on the Count sketch [14]. 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 $\hat{f}_{(i,j)}$. This means

$$E\left(\hat{f}_{(i,j)} \mid u_{(i,j)} \in \mathcal{T}_i\right) = f_{(i,j)} + \delta$$

and

$$\delta = \frac{\mathsf{Cov}\left(\hat{f}_{(i,j)}, \Pr\left(u_{(i,j)} \in \mathcal{T}_i \mid \hat{f}_{(i,j)}\right)\right)}{\Pr\left(u_{(i,j)} \in \mathcal{T}_i\right)} > 0.$$

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.

2.3 The CMM Sketch

The CMM sketch [37] 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 $h_i(\cdot)$. Each hash function maps an item to a counter uniformly at random.

Insertion: Given an incoming item e, the CMM maps the counter $\mathcal{A}[i, h_i(e)]$ 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 overestimation $C_{over}(e) = \min_{i=1}^{d} \mathcal{A}[i, h_i(e)]$. The unbiased estimation $C_{unbiased}(e)$ is given by the following formula.

$$C_{unbiased}(e, i) = \mathcal{A}[i, h_i(e)] - \frac{1}{w-1} \cdot (\mathcal{N} - \mathcal{A}[i, h_i(e)]) .$$

$$C_{unbiased}(e) = \frac{1}{d} \cdot \left(\sum_{i=1}^{d} C_{unbiased}(e, i) \right) . \quad (1)$$

Where $\mathcal{N}$ is the sum of the frequencies of all distinct items.

3 THE DOUBLE-ANONYMOUS SKETCH

In this section, we propose the Double-Anonymous sketch. We introduce three techniques of the Double-Anonymous sketch by three progressive versions. We first introduce **double-anonymity**, which is the key technique to achieve top-K-fairness. Then we introduce **hot panning**, a tricky technique that can keep the characteristic of double-anonymity and raise the Double-Anonymous sketch's accuracy at the same time. Finally, we introduce **early freezing**, a technique that can further raise accuracy.

3.1 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, *i.e.*, they are unrelated. A more formal definition of *double-anonymity* is shown in Theorem 3.1. Achieving **double-anonymity** means that the algorithm meets this condition.

Theorem 3.1. ***(Double-anonymity)*** *Given a single data stream $\mathcal{S}_k$ and an item $u_{(k,i)} \in \mathcal{U}_k$, let $\mathcal{K}_{(i)}$ be an indicator random variable indicating whether item $u_{(k,i)}$ is selected as top-K ($u_{(k,i)} \in \mathcal{T}_k$), if there is $E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}$, then $E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) = f_{(k,i)}$ is equivalent to $\mathrm{Cov}\left(\mathcal{K}_i, \hat{f}_{(k,i)}\right) = 0$.*

Proof. Under the condition of $E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}$ (unbiasedness),

$$\left[E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) = f_{(k,i)}\right] \equiv \left[E\left(\hat{f}_{(k,i)} \mid \mathcal{K}_i = 1\right) = E\left(\hat{f}_{(k,i)}\right)\right] .$$

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,

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

In the above formulas, ≡ stands for equivalence. □

The data structure of the basic version has two parts: a Randomized Admission Policy (RA) [15] as the top-K part and a CMM sketch [37] 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. Notice that these two query processes are also independent. Obviously, the basic version is double-anonymous and achieves top-K-fairness.

3.2 The Hot Panning Version

Keeping the characteristic of double-anonymity, the hot panning version aims to pan the hot items, and only record them in the top-K part to remove the redundancy compare to the first version. We first use a top-K part to classify and record hot items, and then use a count part to record the cold items. Because the top-K part pans the hot items, only cold items will be inserted into the count part, which makes the Hot panning 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 $\mathcal{B}[0, \ldots, m-1]$. 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 strategy frequency 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 real frequency 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 [37], which can provide an unbiased estimation and an overestimation value. We detail CMM in Section 3.1.

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 classic replacement strategies in Section 3.4 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. In other words, the replacement strategy works independently in the top-K part. Then we run the **unbiased operations** of the Double-Anonymous sketch depend on different cases 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 (it can avoid hot items inserting into the count part to minimize the Double-Anonymous sketch's error). Otherwise, we

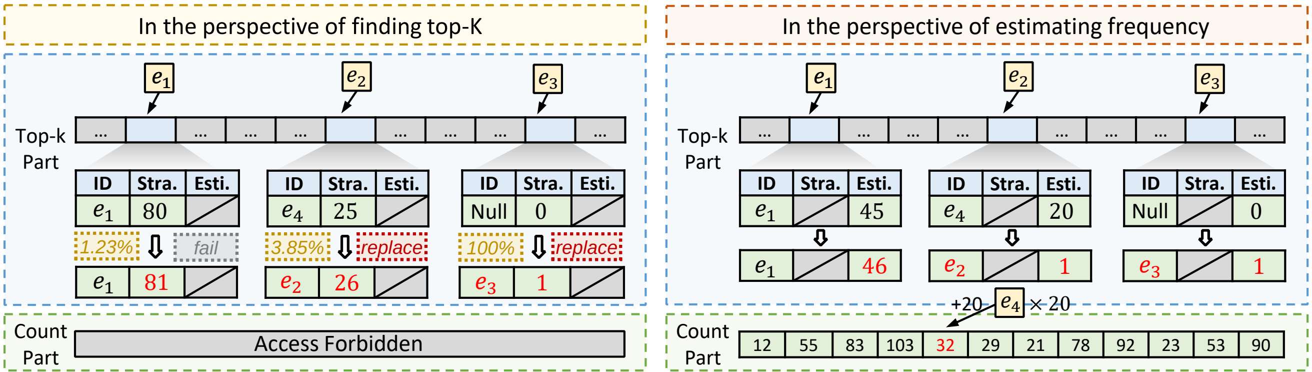


Figure 2: An running example of the Hot panning version of the Double-Anonymous sketch with RA Policy.

use the count part to record this increment. There are three cases as follows.

Case 1: e is in the bucket $\mathcal{B}[h(e)]$. So 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 each item to $[0, M-1]$, and increment the d counters $\mathcal{A}[g_1(.)\dots g_d(.)]$ 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} when it was evicted.

Query: To estimate a local top-K item e, we need to query both the top-K part and the count part. The count part, *i.e.*, CMM sketch[37], reports an overestimated value C_{over} and an unbiased value $C_{unbiased}$. We report three kinds of estimation:

- an unbiased estimation value $\hat{f_i} = C_r + C_{unbiased}$
- an overestimation value $\overline{f_i} = C_r + C_{over}$
- an underestimated value $\underline{f_i} = C_r$

Notice that, if $e.C_{over} = 0$, $\overline{f_i}$ will be equal to $\underline{f_i}$, which means the estimation $\overline{f_i}$ or $\underline{f_i}$ has no error.

Finding Top-K Items: In this task, we query the *strategy frequencies* of items in the top-K part and sort it in descending order. Then we report the largest K items as top-K items.

An running example: Figure 2 shows a running example of Hot panning version of the Double-Anonymous sketch with Randomized Admission Policy. For each item recorded in the top-K part, we record its item ID, strategy frequency (C_s), and real frequency (C_r). 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 $\frac{1}{26}$). 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 $e_3.C_r = 1$.

3.3 The Early Freezing Version

As time goes by, the count part's variance will increase with the increasing number of items inserted into the count part. We propose using a freezing counter ($C_{freezing}$) to freeze the unbiased estimation result in the count part (C_{sketch}) for each frequent item as early as possible, so that we can freeze the error of C_{sketch} and achieve a more accurate estimation. Specially, we add a freezing counter for every cell in the top-K part. Inserting a new incoming item (*i.e.*, an incoming item not in the top-K part before this insertion), we make $C_{freezing} = C_{sketch}$ at that moment. Then the unbiased estimation result of an item e change into $e.C_r + e.C_{freezing}$ instead of $e.C_r + e.C_{sketch}$. Because $C_{freezing}$ is the earlier value of C_{sketch}, the variance of the unbiased estimation result will be smaller.

3.4 Using Different Replacement Policies

The Double-Anonymous sketch can be applied by any top-K algorithm (replacement strategy). We pick four classic top-K strategies: Randomized Replacement Strategy (RA) [15], Spacesaving (SS) [16], Frequent (Freq) [31] and HeavyGuardian (HG) [3] as case studies. For each strategy, we introduce how it works and how to apply it in the Double-Anonymous sketch (different replacement strategies only modify the insertion operation of the Top-K part of the Double-Anonymous sketch). Given an incoming item e, we first hash it into $\mathcal{B}[h(e)]$. Then the strategies work as follows.

RA Policy [15]: *DS+RA* (Double-Anonymous sketch with Randomized Admission Policy) runs the Insertion operation of RA first. Suppose the item whose strategy frequency 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 evict e_{min} with the probability of $\frac{1}{e_{min}.C_s+1}$. If the eviction successes, we record e with its $C_s = e_{min}.C_s + 1$. To make the estimation unbiased, *DS+RA* then runs the Insertion operation of the Double-Anonymous sketch mentioned in Section 3.2.

SpaceSaving (SS) [16]: *DS+SS* (Double-Anonymous sketch with SpaceSaving) runs the Insertion operation of SS first. Suppose the item whose strategy frequency is smallest in the bucket is e_{min}. 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 $C_s = e_{min}.C_s + 1$.

SpaceSaving's estimation is overestimated. To make it unbiased, *DS+SS* then runs the Insertion operation of the Double-Anonymous sketch mentioned in Section 3.2.

Frequent (Freq) [31]: *DS+Freq* (Double-Anonymous sketch with Frequent) runs the Insertion 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 strategy frequency of every item in this bucket by 1. If the strategy frequency of an item e_{evict} is decreased to 0, we evict e_{evict} and record e with its $C_s = 1$. Frequent's estimation is underestimated. To make it unbiased, *DS+Freq* then runs the Insertion operation of the Double-Anonymous sketch mentioned in Section 3.2.

HeavyGardian (HG) [3]: *DS+HG* (Double-Anonymous sketch with HeavyGardian) runs the Insertion operation of HG first. Suppose the item whose strategy frequency 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 $C_s = 1$. HeavyGardian's estimation is underestimated. To make it unbiased, *DS+HG* then runs the Insertion operation of the Double-Anonymous sketch mentioned in Section 3.2.

We further discuss the differences between these four replacement policies based on the experimental results in Section 5, and show that our algorithm is general. Specially, In Section 5.4, we show the degree of top-K-unfairness of these four replacement policies, analyze how top-K-unfairness affects their performance in the task of finding global top-K items, and show that our Double-Anonymous sketch can indeed make them top-K-fair; In Section 5.5, we show a more comprehensive performance comparison of Double-Anonymous sketch using different replacement policies.

4 MATHEMATICAL ANALYSIS

In this section, we analyze the behavior of *our hot panning 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*.

4.1 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.*, $\mathcal{R} \sim \mathcal{P}(\mathcal{S}_k)$.

4.2 Proof of Top-K-fairness

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

Lemma 4.1. *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*

$$f_{T(k,i,t)} + f_{S(k,i,t)} = f_{(k,i,t)}. \tag{2}$$

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 2 holds for any item $u_{(k,i)}$ and any time $t < t'$. At time $t = t'$, according to Section 3.2, if $e_{(k,t)} = u_{(k,i)}$, we insert frequency $\left(f_{T(k,i,t')} - f_{T(k,i,t'-1)} + 1\right)$ into the CMM sketch of the count part, thus

$$f_{S(k,i,t')} = f_{S(k,i,t'-1)} + f_{T(k,i,t')} - f_{T(k,i,t'-1)} + 1$$

and

$$f_{T(k,i,t')} + f_{S(k,i,t')} = f_{(k,i,t'-1)} + 1 = f_{(k,i,t')};$$

If $e_{(k,t)} \neq u_{(k,i)}$, we insert frequency $\left(f_{T(k,i,t')} - f_{T(k,i,t'-1)}\right)$ into the CMM sketch of the count part, thus

$$f_{S(k,i,t')} = f_{S(k,i,t'-1)} + f_{T(k,i,t')} - f_{T(k,i,t'-1)}$$

and

$$f_{T(k,i,t')} + f_{S(k,i,t')} = f_{(k,i,t'-1)} = f_{(k,i,t')};$$

Therefore, Equation 2 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 4.2. *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 Double-Anonymous 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 4.1, 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 [37] 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 $h_l(\cdot)$. $h_l(\cdot)$ maps each item $u_{(k,i)}$ to one of the w counters uniformly at random.

We define some useful random variables. Let the indicator random variable $I_{(i,j,l)}$ indicates whether $h_l\left(u_{(k,i)}\right)$ and $h_l\left(u_{(k,j)}\right)$ are equal, thus we have

$$\Pr\left(I_{(i,j,l)} = 1\right) = \frac{1}{w}.$$

Let the random variable $X_{(i,l)}$ be the value of the $h_l\left(u_{(k,i)}\right)$-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 have

$$X_{(i,l)} = f_{S(k,i,m_k)} + \sum_{j=1,j\neq i}^{n_k}\left(I_{(i,j,l)}\cdot f_{S(k,j,m_k)}\right).$$

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 again, we have

$$\begin{aligned}&E\left(f_{S'(k,i,m_k)}\mid\mathcal{R}\right)\\ =&\frac{1}{d}\cdot\left(\sum_{k=1}^{d}\left(\frac{w}{w-1}\cdot f_{S(k,i,m_k)} - \frac{1}{w-1}\cdot f_{S(k,i,m_k)}\right)\right) = f_{S(k,i,m_k)}.\end{aligned}$$

□

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

Theorem 4.3 (**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 4.2 and using the law of total expectation, we have

$$E\left(\hat{f}_{(k,i)} = 1\right) = \sum_{\mathcal{R}} E\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right)\cdot\Pr\left(\mathcal{R}\right) = f_{(k,i)}.$$

□

Theorem 4.4 (**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*

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

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

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

□

4.3 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 4.1, 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 Double-Anonymous sketch has tighter error bounds than the sketches of CMM [37] and CM [12].

Lemma 4.5. *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 Double-Anonymous sketch, then we have*

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

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

Proof. We first derive the upper bound of the conditional variance $D\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right)$ of a given item $u_{(k,i)}$ in a given sketching process $\mathcal{R}$. Recalling the definition of $\hat{f}_{(k,i)}$ and $f_{S'(k,i,m_k)}$, we have

$$\begin{aligned}\hat{f}_{(k,i)} = &f_{T(k,i,m_k)}\\ &+\frac{1}{d}\cdot\left(\sum_{k=1}^{d}\left(X_{(i,l)} - \frac{1}{w-1}\cdot\left(\sum_{j=1}^{n_k} f_{S(k,j,m_k)} - X_{(i,l)}\right)\right)\right).\end{aligned} \tag{3}$$

Since $f_{T(k,i,m_k)}$ and $f_{S(k,j,m_k)}$ are constants when then sketching process $\mathcal{R}$ is determined, we have

$$\begin{aligned}D\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right) =&D\left(\frac{1}{d}\cdot\left(\sum_{k=1}^{d}\frac{w}{w-1}\cdot X_{(i,l)}\right)\mid\mathcal{R}\right)\\ =&\frac{1}{d^2}\cdot\left(\sum_{k=1}^{d}\left(\left(\frac{w}{w-1}\right)^2\cdot D\left(X_{(i,l)}\mid\mathcal{R}\right)\right)\right).\end{aligned} \tag{4}$$

By expanding $X_{(i,l)}$ and considering the independence between $I_{(i,j,l)}$, we have

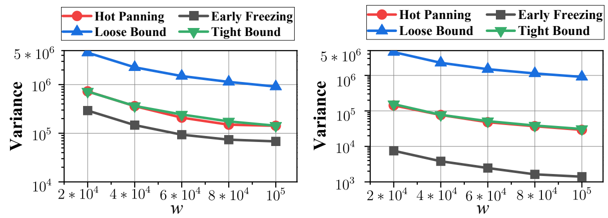
$$\begin{aligned}D\left(X_{(i,l)}\mid\mathcal{R}\right) =&\frac{1}{w}\cdot\left(1-\frac{1}{w}\right)\cdot\left(\sum_{j=1,j\neq i}^{n_k}\left(f_{S(k,j,m_k)}\right)^2\right)\\ \leqslant&\frac{1}{w}\cdot\left(1-\frac{1}{w}\right)\cdot\left(\sum_{j=1}^{n_k}\left(f_{S(k,j,m_k)}\right)^2\right).\end{aligned} \tag{5}$$

In other words

$$D\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right)\leqslant\frac{1}{d\cdot(w-1)}\cdot\left(\sum_{j=1}^{n_k}\left(f_{S(k,j,m_k)}\right)^2\right). \tag{6}$$

Since we derive the unbiasedness and conditional unbiasedness of the estimated frequency $\hat{f}_{(k,i)}$ in Theorem 4.3 and Lemma 4.2, that is

$$E\left(\hat{f}_{(k,i)}\mid\mathcal{R}\right) = E\left(\hat{f}_{(k,i)}\right) = f_{(k,i)}, \tag{7}$$



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

Figure 3: Sample variances and their theoretical upper bounds.

by using the law of total expectation, we have

$$D\left(\hat{f}_{(k,i)}\right)=\sum_{\mathcal{R}} E\left(\left(\hat{f}_{(k,i)}-f_{(k,i)}\right)^2 \mid \mathcal{R}\right)\cdot \Pr(\mathcal{R})$$
$$\leqslant \frac{1}{d\cdot(w-1)}\cdot\left(\sum_{j=1}^{n_k}\left(f_{S(k,j,m_k)}\right)^2\right).$$

According to Lemma 4.1, we have

$$\frac{1}{d\cdot(w-1)}\cdot\left(\sum_{j=1}^{n_k}\left(f_{S(k,j,m_k)}\right)^2\right)\leqslant\frac{1}{d\cdot(w-1)}\cdot\left(\sum_{j=1}^{n_k}f^2_{(k,j)}\right). \quad (8)$$

The left and right sides are the upper bounds of variance of the Double-Anonymous sketch and the CMM sketch, respectively. □

Theorem 4.6. *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 Double-Anonymous sketch, then we have*

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

Theorem 4.7. *Given a data stream $\mathcal{S}_k$, for any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, let $\overline{f}_{(k,i)}$ be the overestimation of $f_{(k,i)}$ given by the Double-Anonymous sketch, then we have*

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

4.4 Analysis on Early Freezing

By using the *early freezing* optimization, the Double-Anonymous 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 framework in Section 4.2 and 4.3 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 of the early freezing version; On the other hand, according to Lemma 4.8 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 4.9.

Lemma 4.8. *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*

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

Theorem 4.9. *Given a data stream $\mathcal{S}$, for any replacement policy $\mathcal{P}$ and any item $u_{(k,i)}$, we have*

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

4.5 Experimental Verification

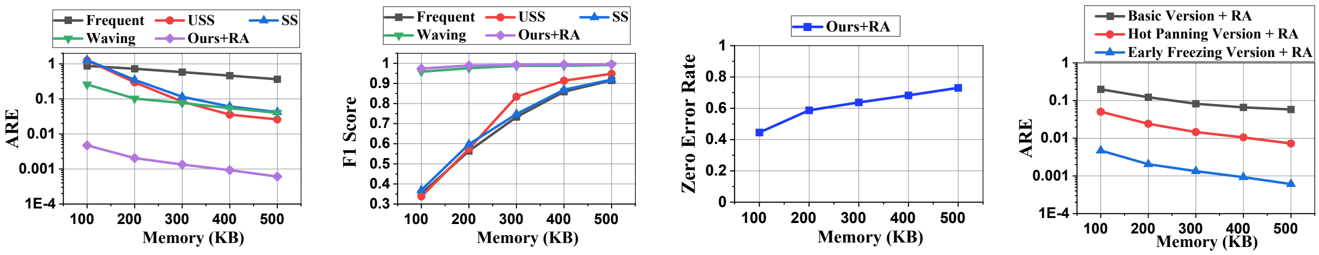
To verify the correctness of Lemma 4.5 and Theorem 4.9, we show two kinds of variance bound. Lemma 4.5 provides a $\mathcal{P}$-independent loss bound, and an $\mathcal{R}$-dependent tight bound. We use SpaceSaving and Randomized Admission Policy as the strategy $\mathcal{P}$, and vary the length w of the count part. As shown in Figure 3, we plot the loose upper bounds, the tight upper bounds, and the sample variances of the hot panning version and the early freezing version. It can be found that the bounds of variances are always greater than the sample variances of the hot panning version, and then greater than the sample variances of the early freezing version, which verifies our theorems and shows the benefits of Early Freezing. It is worth noting two points: 1) The tight bounds are extremely close to the sample variances, which indicates our bounds are accurate. 2) Panning 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.

5 EXPERIMENTAL RESULTS

5.1 Experimental Setup

A. Implementation: We have implemented the Double-Anonymous sketch (DA sketch) and all other algorithms in C++. We apply four replacement strategies to the DA sketch: Randomized Admission Policy (RA) [15], SpaceSaving (SS) [16], Frequent (Freq) [31] and HeavyGuardian (HG) [3]. We find in our experimental results that applying Randomized Admission Policy yields the best results; therefore, we mainly demonstrate the experimental results of DA sketch + RA. We also compare our results with several state-of-the-art top-K sketching algorithms: Frequent [31], SpaceSaving [16], Unbiased SpaceSaving (USS) [5] and WavingSketch (Waving) [2]. All our experiments are repeatedly performed 10 times to ensure statistical stability. Our source code is publicly available at Github [51]. We conduct all our experiments on a machine with two 6-core processors (12 threads, Intel Xeon CPU E5-2620 @2 GHz) and 64 GB DRAM memory.

B. Datasets: We use three real-world datasets and one synthetic dataset for our experiments. The details of the datasets are shown below: 1) IP Trace Dataset (CAIDA) [52]: The IP Trace Dataset 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 for our experiments. 2) Web Page Dataset [53]: The Web page dataset is built from a collection of web pages downloaded from the website. Each item is 4 bytes long. 3) Network Dataset [54]: The network dataset consists of users' posting history on



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

Figure 4: Performance of finding local top-K items.

the StackExchange website. 4) Synthetic Dataset: We generated datasets following the Zip-f distribution [55]. Each dataset contains 32M items, and each item is 4 bytes long. Here we use the generated dataset with skewness=0.6.

C. Metrics:

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

2) 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.

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

4) 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.

5) Relative Bias: This metric is used in section 5.4. 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.

6) 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.

D. Common Settings: Let M 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 $\lambda = 8$, $\frac{M_{top-K}}{M} = 0.55$ in order to maximize the overall performance. 5 For DA sketch, the size of count part's buckets in the Hot Panning version and the Early Freezing is set to be 2 bytes; while for the basic version, the size of count part's buckets is set to be 4 bytes. All other parameters of the baseline top-K algorithms are set according to the recommendations of their authors.

Settings for Figure 1(a) in Section 1.2: We perform the finding local top-K tasks on CAIDA dataset for both USS and Waving, for 1000 times each. Memory size is set to be 100KB, and K is set to be 1000. After insertion, we calculate the total (signed) error for both the selected Top-K items and items that are not selected. We average the results over the 1000 times of experiments.

Settings for Figure 1(b) in Section 1.2: We conduct experiments on the Synthetic Dataset. 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 only allocate an extremely small amount of memory for both USS and Ours+SS, such that they could only store $K = 50$ local top-K candidates for each distributed sketch (3.8KB for USS and 1.4KB for Ours+SS). In such an extremely small amount of memory and high skewness, the estimated frequency of the selected local top-K items in the heavy stream would even be greater than the frequency of the global Top-1 item in the light stream. Therefore, the global Top-1 may be ignored when skewness is high.

5.2 Experiments on Local Top-K

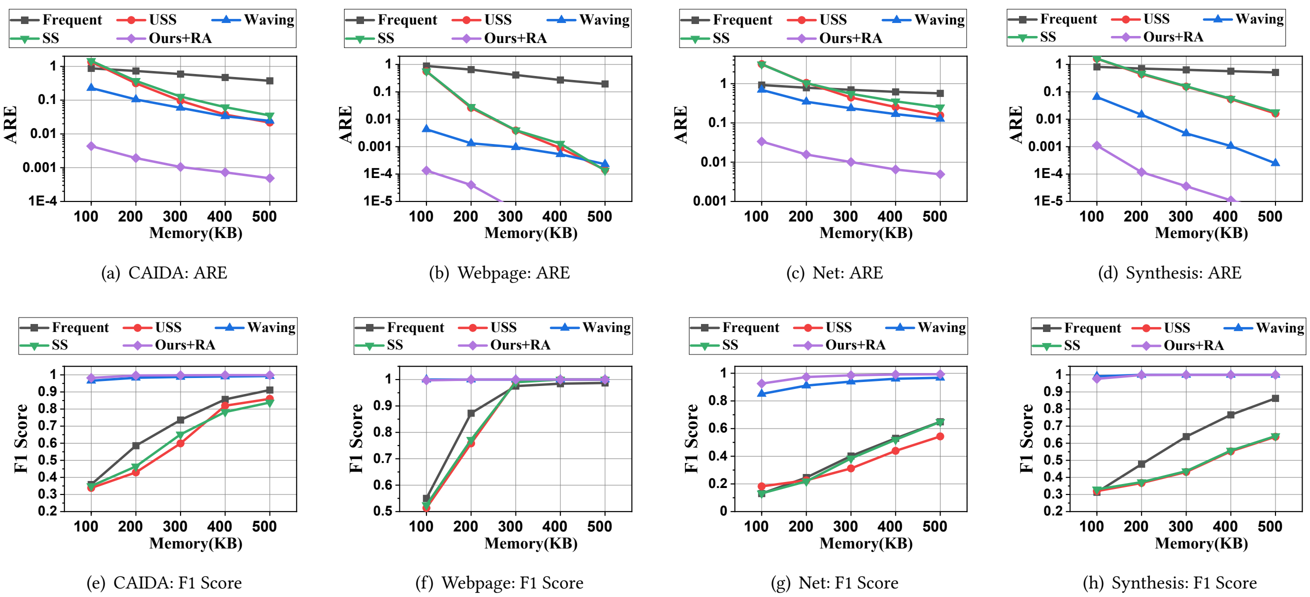
Application Description: We first conduct experiments on finding local top-K items and compare the Double-Anonymous sketch with prior art mentioned in 5.1. We use ARE, F1 Score, and Zero Error rate for evaluation. We also compare the performance of our three versions, *i.e.*, the basic version, the Hot Panning Version, and the Early Freezing Version, and show how hot panning and early freezing improve the performance of our approach.

Experimental Settings: In this experiment, we use the CAIDA dataset for our experiments. 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 4(a)): Results show that our approach achieves much more accurate unbiased frequency estimation thanks to the hot panning and early freezing technique. When $M = 100$KB, our approach is around 500-1000 times more accurate than USS, SS, and Frequent and around 50-100 times more accurate than Waving on the CAIDA dataset.

F1 Score (Figure 4(b)): When applying RA to our approach, the Double-Anonymous sketch achieves sufficiently high F1 Score ($\geq$ 95%) even when memory is extremely tight. This is because for the Double-Anonymous 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 5.3.

Zero Error Rate (Figure 4(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



(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

Figure 5: Performance of finding global top-K items.

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 M = 500KB. The results suggest that for the majority of items, our algorithm could tell with 100% confidence that their estimated frequencies are perfectly accurate, which is useful in practice.

Comparison between the three versions (Figure 4(d)): We find that both the hot panning 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 panning version.

5.3 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 5.1 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 5(a) - 5(d)): We find that our approach could achieve much lower ARE than prior art. On CAIDA dataset, when M = 100KB, 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.

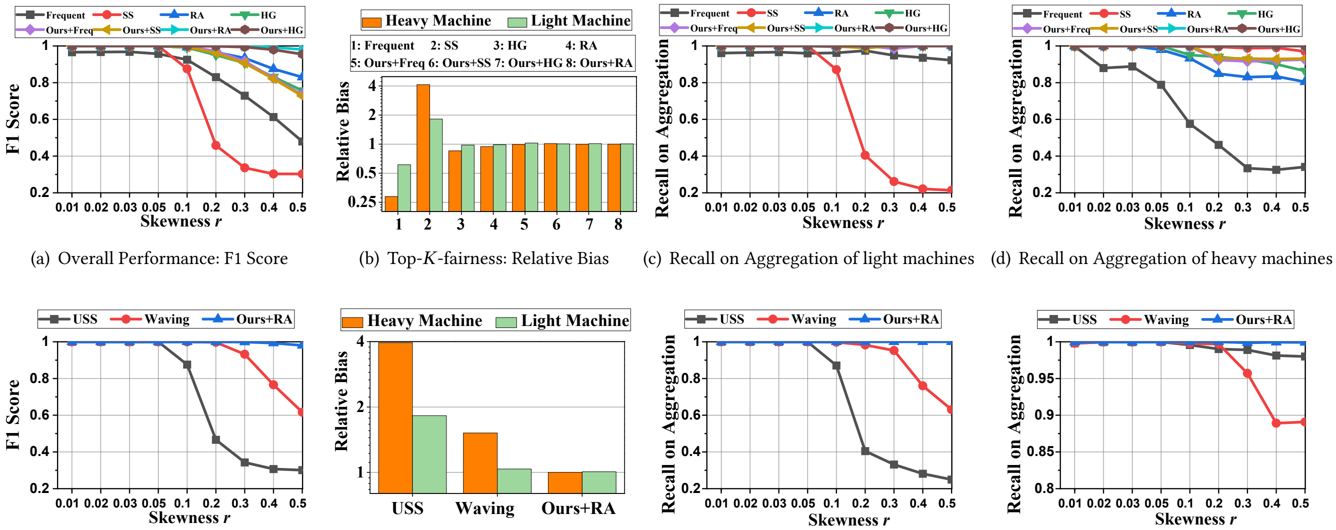
F1 Score (Figure 5(e) - 5(h)): Results show that in this scenario, our approach could achieve a high F1 Score on both datasets even when M is small. When M = 100KB, the F1 Score of our approach is greater than 90% on both datasets, while the F1 Score of Frequent, USS, and SS is lower than 60% on the 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 1): 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 1: Throughput (Mops) of finding top-K frequent items.

Analysis: 1) Our approach is accurate in frequency estimation on global top-K items even with extremely small memory. 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. Waving sketch is also not as accurate as our approach because when memory is tight, Waving counters tend to be highly inaccurate. 2) F1 Score of our approach



(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

Figure 6: Performance and fairness for finding global top-K items comparing our approaches with baseline algorithms.

is mainly determined by the top-K replacement strategy, and when applying Randomized Admission Policy (RA) replacement strategy to our approach, the Double-Anonymous sketch could achieve a high F1 Score on both the local top-K task and the global top-K task. F1 Score of Frequent, USS, and SS is significantly lower than our approach since all of them use the Stream Summary [16] data structure, which consumes more memory to store one item than our approach, and those replacement strategies are not as accurate as the RA replacement strategy. 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.

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

5.4.1 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 apply four replacement policies to the Double-Anonymous sketch and 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 of those algorithms. **Relative bias** is used to directly 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 items. Among those real global top-K items proposed by sketch i, only a proportion of them survive and are selected as global top-K items. **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, regardless of which local sketch it comes from.*

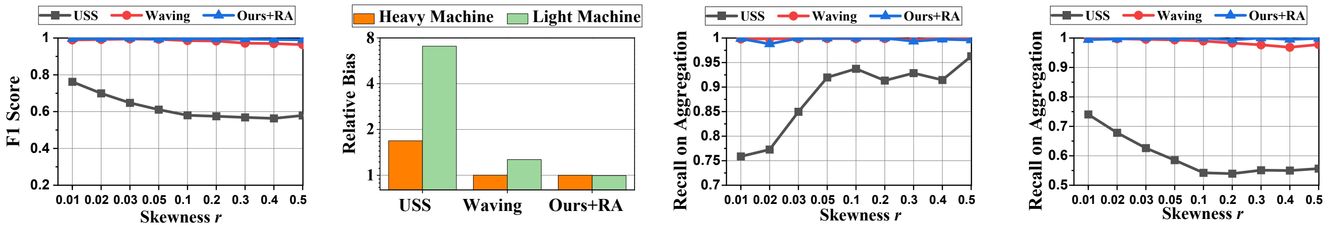
Frequent	SS	HG	RA	USS	Waving	Ours
40KB	40KB	15KB	15KB	40KB	15KB	15KB

Table 2: Memory size configurations in Section 5.4

Experimental Settings: We set $N = 100$, $K = 1000$, and vary the skewness r from 0.01 to 0.5. In order to better demonstrate how top-K-fairness affects the performance and eliminate the effects of selecting local top-K items itself on the performance of finding global top-K items, we adjust the memory sizes for different algorithms so that they could store exactly the same number of local top-K candidates. The configurations on memory size are shown in Table 2. We use the synthetic dataset with skewness=0.9, which is relatively low in skewness, to better demonstrate the concept of "top-K-fairness" and illustrate our results.

5.4.2 Overall Performance & Top-K-fairness

F1 Score (Figure 6(a) and 6(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.



(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

Figure 7: Performance and fairness of weighted algorithms for finding global top-K items.

Relative Bias on Top-K items (Figure 6(b) and 6(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 Double-Anonymous 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 in heavy machines is low in real frequency, its estimated frequency is still high enough to be falsely selected as a global top-K item. 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 Double-Anonymous sketch to this top-K algorithm. Meanwhile, the F1 Score is also much improved because our approach is top-K-fair in global top-K aggregation. Specifically, for top-K algorithms Frequent and SS with significant top-K-unfairness, our DA sketch can improve their F1 scores by up to 25.5% and 42.5%; and for top-K algorithms RA and HG with slight top-K-unfairness, our DA sketch can still improve their F1 scores by 15.0% and 19.8%.

5.4.3 Recall on Aggregation

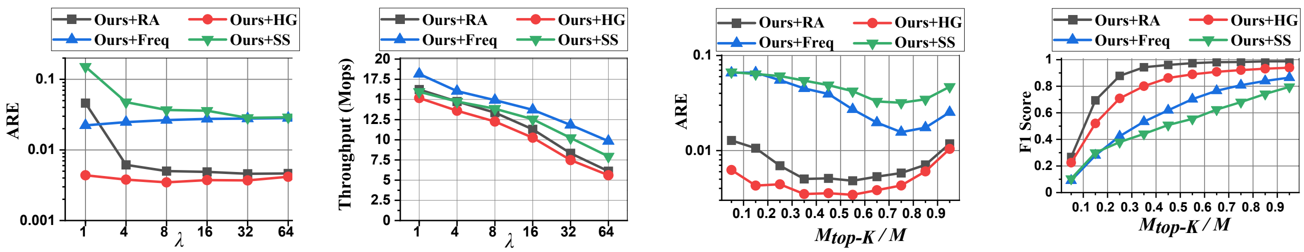
Recall on Aggregation (Figure 6(c) - 6(d) and 6(g) - 6(h)): For light machines, we find that Recall on Aggregation of overestimation algorithms, like SS, USS, and Waving, decreases fast as r increases, while that of other algorithms keeps at a high level ($\geq 90\%$). Conversely, for heavy machines, 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 survive the global aggregation and 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 on top-K items. 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 (Recall on Aggregation on light machines tends to be small). 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 in many real-world applications since the global top-K selector should not be partial to items from any machine.

5.4.4 Other Baseline Algorithms

Comparison algorithms: In this section, 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 data structures 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 data structures 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 7): We compare weighted USS, weighted Waving, and weighted Ours+RA. As shown in Figure 7(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 7(a), 7(c), and 7(d), weighting can indeed improve the performance of USS and Waving,



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

Figure 8: Experiments on different parameter settings (λ and $\frac{M_{top-K}}{M}$) of Double-Anonymous Sketch.

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 (Table 3): We compare with the sampling algorithms using different sampling rates and different global data structures on global top-K selectors. The experimental results show that higher sampling rate means higher accuracy, but the performance of the sampling algorithm using 0.1 sampling rate and precise global data structure is still inferior to Ours+RA. Furthermore, using a sampling algorithm with a rate of $p = 0.1$, in total all machines need to transmit 8MB of data to the global top-K selector; using a sampling algorithm with a rate of $p = 0.02$, all machines need to transmit 1.6MB of data. By contrast, using Ours+RA, all machines only need to transmit $100 \times 15KB = 1.5MB$ of data.

Algorithms	F1 Score	ARE
Sampling ($p = 0.02$) + Precise	85.8%	0.1234
Sampling ($p = 0.1$) + Precise	94.3%	0.0576
Sampling ($p = 0.02$) + RA	68.3%	0.1367
Sampling ($p = 0.1$) + RA	73.4%	0.0958
Ours + RA ($r = 0.5$)	98.0%	0.0069

Table 3: Comparisons between the sampling approach and our approach, where "precise" indicates that we use a deterministic algorithm to precisely record every sampled item.

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 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. However, as shown in Section 5.4.2, the overestimation and underestimation of different algorithms are not consistent. On the one hand, this algorithm is difficult to practice, and on the other hand, it cannot achieve the exact top-K-fairness.

5.5 Experiments on Parameter Settings

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

Varying λ (Figure 8(a)-8(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 by 6.8 times and 3.2 times respectively and then remains steady. For Ours+HG and Ours+Freq, ARE keeps roughly steady. However, as λ increases, the throughput of all Double-Anonymous sketch applications drops severely: on average, throughput when $\lambda = 64$ is 2.3 times smaller than throughput when $\lambda = 1$. Therefore in practice, we choose $\lambda = 8$ as the best setting.

Varying $\frac{M_{top-K}}{M}$ (Figure 8(c)-8(d)): We find that F1 scores grow as $\frac{M_{top-K}}{M}$ increases, since F1 scores are only determined by the top-K part. However, we find that when $\frac{M_{top-K}}{M} \geq 0.55$, growth rate of F1 scores of all Double-Anonymous sketch applications becomes slow if $\frac{M_{top-K}}{M}$ continues to increase. In addition, in this experiment, $M = 100$KB is tight, and if M becomes larger, growth of F1 scores contributed by $\frac{M_{top-K}}{M}$ will become more negligible. Besides, Ours+HG and Ours+RA reach their respective minimal ARE score when $\frac{M_{top-K}}{M} \approx 0.55$ (2.7 and 1.8 times smaller than when $\frac{M_{top-K}}{M} = 0.05$ and 2.5 and 3.0 times smaller than when $\frac{M_{top-K}}{M} = 0.95$), while Ours+Freq and Ours+SS reach their minimal ARE when $\frac{M_{top-K}}{M} \approx 0.75$. In practice, we choose $\frac{M_{top-K}}{M} = 0.55$ 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.

6 CONCLUSION

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

REFERENCES

[1] Pinghui Wang, Yiyan Qi, Yuanming Zhang, and etal. A memory-efficient sketch method for estimating high similarities in streaming sets. In *SIGKDD*, 2019.

[2] Jizhou Li, Zikun Li, Yifei Xu, and etal. Wavingsketch: An unbiased and generic sketch for finding top-k items in data streams. In *SIGKDD*, 2020.

[3] Tong Yang, Junzhi Gong, Haowei Zhang, and etal. Heavyguardian: Separate and guard hot items in data streams. In *SIGKDD*, 2018.

[4] Xiangyang Gou, Long He, Yinda Zhang, and etal. Sliding sketches: A framework using time zones for data stream processing in sliding windows. In *SIGKDD*, 2020.

[5] Daniel Ting. Data sketches for disaggregated subset sum and frequent item estimation. In *SIGMOD Conference*, 2018.

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

[7] Pratanu Roy, Arijit Khan, and Gustavo Alonso. Augmented sketch: Faster and more accurate stream processing. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1449–1463, 2016.

[8] Tong Yang, Jie Jiang, Peng Liu, and etal. Elastic sketch: adaptive and fast network-wide measurements. In *SIGCOMM*, 2018.

[9] Yuliang Li, Rui Miao, Changhoon Kim, and etal. Flowradar: a better netflow for data centers. In *NSDI*, 2016.

[10] Haipeng Dai, Muhammad Shahzad, Alex X Liu, and etal. Finding persistent items in data streams. *VLDB Endowment*, 2016.

[11] S. Venkataraman, D. Xiaodong Song, P. B. Gibbons, and A. Blum. New streaming algorithms for fast detection of superspreaders. In *NDSS*, 2005.

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

[13] Cristian Estan and George Varghese. New directions in traffic measurement and accounting. *SIGMCOMM*, 2002.

[14] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming*. Springer, 2002.

[15] Ran Ben-Basat, Gil Einziger, Roy Friedman, and etal. Randomized admission policy for efficient top-k and frequency estimation. In *INFOCOM*, 2017.

[16] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT*, 2005.

[17] Tao Li, Shigang Chen, and Yibei Ling. Per-flow traffic measurement through randomized counter sharing. *TON*.

[18] Peiqing Chen, Dong Chen, Lingxiao Zheng, Jizhou Li, and Tong Yang. Out of many we are one: Measuring item batch with clock-sketch. SIGMOD, 2021.

[19] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. Correlation sketches for approximate join-correlation queries. In *SIGMOD*, 2021.

[20] Rundong Li, Pinghui Wang, Jiongli Zhu, Junzhao Zhao, Jia Di, Xiaofei Yang, and Kai Ye. Building fast and compact sketches for approximately multi-set multi-membership querying. In *SIGMOD*, 2021.

[21] Y. Izenov, A. Datta, F. Rusu, and JH Shin. Online sketch-based query optimization. 2021.

[22] Zheng Zhong, Shen Yan, Zikun Li, Decheng Tan, Tong Yang, and Bin Cui. Burstsketch: Finding bursts in data streams. In *SIGMOD*, 2021.

[23] Daniel Ting and Rick Cole. Conditional cuckoo filters. In *SIGMOD*, 2021.

[24] Prashant Pandey, Alex Conway, Joe Durie, Michael A Bender, Martin Farach-Colton, and Rob Johnson. Vector quotient filters: Overcoming the time/space trade-off in filter design. In *SIGMOD*, 2021.

[25] Gaurav Gupta, Minghao Yan, Benjamin Coleman, Bryce Kille, RA Leo Elworth, Tharun Medini, Todd Treangen, and Anshumali Shrivastava. Fast processing and querying of 170tb of genomics data via a repeated and merged bloom filter (rambo). In *SIGMOD*, 2021.

[26] Yesdaulet Izenov, Asoke Datta, Florin Rusu, and Jun Hyung Shin. Compass: Online sketch-based query optimization for in-memory databases. In *SIGMOD*, 2021.

[27] Benwei Shi, Zhuoyue Zhao, Yanqing Peng, Feifei Li, and Jeff M Phillips. At-the-time and back-in-time persistent sketches. In *SIGMOD*, 2021.

[28] Peng Jia, Pinghui Wang, Junzhou Zhao, Shuo Zhang, Yiyan Qi, Min Hu, Chao Deng, and Xiaohong Guan. Bidirectionally densifying lsh sketches with empty bins. In *SIGMOD*, 2021.

[29] Zhenwei Dai, Aditya Desai, Reinhard Heckel, and Anshumali Shrivastava. Active sampling count sketch (ascs) for online sparse estimation of a trillion scale covariance matrix. In *SIGMOD*, 2021.

[30] Kangfei Zhao, Jeffrey Xu Yu, Hao Zhang, Qiyan Li, and Yu Rong. A learned sketch for subgraph counting. In *SIGMOD*, 2021.

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

[32] M. Gurmeet Singh and M. Rajeev. Approximate frequency counts over data streams. In *VLDB*, 2002.

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

[34] Yakov Rekhter, Tony Li, and Susan Hares. A border gateway protocol 4 (bgp-4). Technical report, 2006.

[35] Joao Luis 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*, pages 49–60, 2003.

[36] Zaoxing Liu, Antonis Manousis, and et al. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proc. ACM SIGCOMM*, 2016.

[37] Fan Deng and Davood Rafiei. New estimation algorithms for streaming data: Count-min can do more. *Webdocs. Cs. Ualberta. Ca*, 2007.

[38] Kai Sheng Tai, Vatsal Sharan, Peter Bailis, and etal. Sketching linear classifiers over data streams. In *SIGMOD*, 2018.

[39] Anshumali Shrivastava, Arnd Christian Konig, and Mikhail Bilenko. Time adaptive sketches (ada-sketches) for summarizing data streams. In *SIGMOD*, 2016.

[40] Yanqing Peng, Jinwei Guo, Feifei Li, and etal. Persistent bloom filter: Membership testing for the entire history. In *SIGMOD*, 2018.

[41] T. Nan, C. Qing, and M. Prasenjit. Graph stream summarization: From big bang to big crunch. In *SIGMOD*, 2016.

[42] Zhewei Wei, Ge Luo, Ke Yi, and etal. Persistent data sketching. In *SIGMOD*, 2015.

[43] Daniel Ting. Count-min: Optimal estimation and tight error bounds using empirical error distributions. In *SIGKDD*, 2018.

[44] Prashant Pandey, Michael A. Bender, Rob Johnson, and Rob Patro. A general-purpose counting filter: Making every bit count. In *SIGMOD*, 2017.

[45] Alex D. Breslow and Nuwan S. Jayasena. Morton filters: Faster, space-efficient cuckoo filters via biasing, compression, and decoupled logical sparsity. 2018.

[46] Minmei Wang, Mingxun Zhou, Shouqian Shi, and etal. Vacuum filters: More space-efficient and faster replacement for bloom and cuckoo filters. *VLDB Endowment*, 2019.

[47] Graham Cormode and Shanmugavelayutham Muthukrishnan. What's new: Finding significant differences in network data streams. *IEEE/ACM Transactions on Networking*, 13(6):1219–1232, 2005.

[48] Robert Schweller, Zhichun Li, Yan Chen, and etal. Reversible sketches: enabling monitoring and analysis over high-speed data streams. *TON*, 2007.

[49] K. Balachander, S. Subhabrata, Z. Yin, and C. Yan. Sketch-based change detection: methods, evaluation, and applications. In *SIGCOMM*, 2003.

[50] Tong Yang, Yang Zhou, Hao Jin, , and etal. Pyramid sketch: A sketch framework for frequency estimation of data streams. *VLDB Endowment*, 2017.

[51] Source code related to double-anonymous sketch. https://github.com/Arimase97/Double-Anonymous-Sketch.

[52] The caida anonymized 2016 internet traces. http://www.caida.org/data/overview/.

[53] Real-life transactional dataset. http://fimi.ua.ac.be/data/.

[54] The Network dataset Internet Traces. http://snap.stanford.edu/data/.

[55] David MW Powers. Applications and explanations of Zipf's law. In *Proc. EMNLP-CoNLL*. Association for Computational Linguistics, 1998.