Supplementary Materials of WavingSketch

1 Implementation Details of SIMD Acceleration

We present the implementation details of the SIMD acceleration method for WavingSketch. In our implementation, we treat the d cells of the Heavy Part of each bucket as uniform data points, and use SIMD instructions (AVX-512) to process them in parallel. To accelerate WavingSketch with SIMD, we first propose the Heavy Part rearrangement technique to vectorizes the d keys and values in each bucket of WavingSketch, allowing for their parallel processing with SIMD. By utilizing the parallel processing capabilities of SIMD, we further propose two techniques to accelerate the two critical procedures in the insertion/query operation of WavingSketch: 1) finding matched key (used in insertion and query operations); 2) finding the item with the smallest frequency (used in insertion operations). Next, we explain the three techniques in detail.

Heavy Part rearrangement: To make WavingSketch more suitable for SIMD acceleration and reduce the memory accesses in its insertion/query operation, we rearrange the Heavy Part of each bucket by separating it into 3 areas: the key area, the frequency area, and the flag area. Each area contains the corresponding parts of the d cells. In insertion (or query) operation, we first check the key area. If we find a matched key, we update/check the corresponding frequency area and flag area. If we do not find any matched key, we can just move on and do not check the other areas any more. In this way, we omit a lot of unnecessary memory accesses. In typical settings ($d \leqslant 32$), the key area can be fetched in one memory access, and thus, many query operations can be completed with one memory access.

```
1  //input: an item ei
2  //input: an array A of four 32-bit keys
3  int match_key(int ei, int* A) {
4    //broadcast ei to a 128-bit register R1
5    __m128i R1 = _mm_set1_epi32(ei);
6    //load A to a 128-bit register R2
7    __m128i R2 = _mm_load_epi32(A);
8    //compare every 32-bit integer in R1 and R2
9    int match = _mm256_cmpeq_epi32_mask(R1, R2);
10
11   //not match, return -1
12   if(match == 0) return -1;
13   //return the matched index
14   return _tzcnt_u32((uint32_t) match);
15 }
```

Figure 1: Codes of the match interface accelerated with SIMD instructions.

Match optimization: In the insertion/query operation of WavingSketch, we first check whether an incoming item e_i exists in $\mathcal{B}[h(e_i)].heavy$ by matching e_i with the d keys in $\mathcal{B}[h(e_i)].heavy$. This match procedure can be abstracted as an interface with the input of an item e_i and an array $\mathcal{A}$ of d keys, and with the output of an index to the position of e_i in $\mathcal{A}$ (or -1 if e_i is not in $\mathcal{A}$). This interface is also used in another place: In insertion operation, if e_i is not recorded in $\mathcal{B}[h(e_i)].heavy$, we need to check whether $\mathcal{B}[h(e_i)].heavy$ has vacant cells and return the index of a vacant cell (Case 2). This procedure can also be abstracted as a match interface by regarding e_i as 0 and $\mathcal{A}$ as the array of d frequencies.

In basic WavingSketch, we naturally implement the match interface by looping d times. We can unroll this loop using SIMD instructions. Suppose $d = 4$ and each key is of 32-bit length [1], the codes of the SIMD-optimized match interface is shown in Figure 1. First, we load item e_i and array $\mathcal{A}$ into two 128-bit registers $R1$ and $R2$ (line 4-7). Then we use one SIMD instruction to compare every 32-bit integer in $R1$ and $R2$ in parallel (line 8-9). Finally, we return -1 if e_i is not in $\mathcal{A}$ (line 11-12), or we return the index of the matched key by counting the number of trailing zero bits in the comparison result (line 13-14). In this way, we reduce the time complexity of the match interface from $O(d)$ to $O(1)$.

Find-min optimization: In Case 3 of the insertion operation, we need to find the least frequent item in $\mathcal{B}[h(e_i)].heavy$. This find-min procedure can be abstracted to an interface with the input of an array $\mathcal{A}$ of d numbers, and with the output of the smallest number in $\mathcal{A}$ [2]. In basic WavingSketch, we implement the find-min interface by looping d times. We can use SIMD instructions to optimize

[1] For larger d and length of key, we can use the vector registers of larger sizes (AVX-512 supports 512-bit registers at most).

[2] We can further get the index of the smallest number in $\mathcal{A}$ by calling the match interface above.

```
1  //input: an array A of four 32-bit keys
2  int find_min(int* A) {
3    //load A to a 128-bit register R1
4    __m128i R1 = _mm_load_epi32(A);
5
6    //swap the first and second halves of A
7    //R1=[a1,a2,a3,a4], R2=[a3,a4,a1,a2]
8    int mask1 = _MM_SHUFFLE(0, 0, 3, 2);
9    __m128i R2 = _mm_shuffle_epi32(R1, mask1);
10   //compare every 32-bit integers in R1 and R2
11   //and store the minimum values in R2
12   //R2=[min{a1,a3},min{a2,a4}, ...]
13   R2 = _mm_min_epi32(R1, R2);
14
15   //similar procedures as above
16   //R1=[min{a1,a3,a2,a4}, ...]
17   int mask2 = _MM_SHUFFLE(0, 0, 0, 1);
18   R1 = _mm_shuffle_epi32(R2, mask2);
19   R1 = _mm_min_epi32(R1, R2);
20
21   //return the smallest number in A
22   int res = _mm_cvtsi128_si32(R1);
23   return res;
24 }
```

Figure 2: Codes of the find-min interface accelerated with SIMD instructions.

this interface. Suppose $d = 4$ and each number is of 32-bit length, the codes of the SIMD-optimized find-min interface is shown in Figure 2. First, we load array $\mathcal{A}$ into a 128-bit register $R1$ (line 3-4). Then we swap the first and second halves of $\mathcal{A}$ and store the results in $R2$ (line 6-9). Suppose $\mathcal{A} = [a_1, a_2, a_3, a_4]$, then we have $R1 = [a_1, a_2, a_3, a_4]$ and $R2 = [a_3, a_4, a_1, a_2]$. We use one SIMD instruction to compare every 32-bit integers in $R1$ and $R2$ in parallel, and store the minimum values in $R2$ (line 10-13). Now, we have $R2 = [\min\{a_1, a_3\}, \min\{a_2, a_4\}, \cdots]$. Next, we repeat the above procedure to compare the first and second elements in $R2$, and store the minimum values in $R1$ (line 15-19). In this way, we get $R1 = [\min\{a_1, a_3, a_2, a_4\}, \cdots]$. Finally, we return the first element of $R1$, which is the smallest number in $\mathcal{A}$. In our implementation, we reduce the comparison space (length of the array to be compared) by half at one time. Therefore, we reduce the time complexity of the find-min interface from $O(d)$ to $O(log(d))$.

2 Mathematical Proofs

2.1 Proof of Unbiasedness on Frequency Estimation

We prove that for an arbitrary item, its estimated frequency made by WavingSketch is unbiased. We first consider the basic version of WavingSketch in Theorem 1. Then we extend the conclusion to multi-counter WavingSketch in Theorem 2.

Definition of probability space: Given a data stream σ, the difference of different estimation results made by WavingSketch comes from the randomness of its hash functions: $h(\cdot)$ and $s(\cdot)$. Consider an arbitrary item $e \in [m]$ in σ. Its estimated frequency $\hat{f}$ is only affected by the items that are mapped into the same bucket as e, namely the items mapped into $\mathcal{B}[h(e)]$. Thus, in the following, we only need to consider bucket $\mathcal{B}[h(e)]$ and the items mapped into it. In this way, we define our probability space based on the sample space of all possible $s(\cdot)$. Formally, we define the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ by setting $\Omega = \{s : [m] \to \{+1, -1\}\}$, $\mathcal{F} = 2^{\Omega}$, and $\mathbb{P}(A) = \frac{|A|}{|\Omega|}$, where $|A|$ denotes the cardinality of set A.

Theorem 1. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ, the estimated frequency of e made by basic WavingSketch, namely $\hat{f}$, is unbiased, i.e., $\mathbb{E}(\hat{f}) = f$, where f is the real frequency of e.*

Proof. We prove this theorem by induction on each item e_k $(1 \leqslant k \leqslant n)$ that is inserted into WavingSketch.

Base case: When WavingSketch is empty, it is obvious that $\mathbb{E}(\hat{f}) = f = 0$.

Induction step: Let f and $\hat{f}$ be the real and estimated frequency of e before the k_{th} insertion. Suppose the theorem holds for the first $k-1$ inserted items, *i.e.*, we suppose that $\mathbb{E}(\hat{f}) = f$. Now we consider the k_{th} inserted item e_k. Let $f' = f + \Delta f$ and $\hat{f}' = \hat{f} + \Delta\hat{f}$ be the real and estimated frequency of e after the k_{th} insertion. We will discuss four cases to prove that the theorem holds after the k_{th} inserted item e_k, *i.e.*, we will prove that $\mathbb{E}(\hat{f}') = f'$. In the following four cases, we discuss whether e_k is e, and whether e_k is error-free [3] before the k_{th} insertion. Let $\Omega_i \subseteq \Omega$ be the set of all possible $s(\cdot)$ that cause case i, namely the event of case i. By default, in case i, we use $\mathbb{E}$ to denote the conditional expectation defined over Ω_i.

Case 1: $e_k = e$, and e_k is error-free.

[3] An item is error-free if it is recorded in the Heavy Part of WavingSketch with flag $true$.

In this case, we just increment the corresponding frequency of e in the Heavy Part by one. Afterwards, e is still in the Heavy Part and is error-free. Thus, we have $\Delta\hat{f} = \Delta f = 1$, meaning that $\mathbb{E}\big(\hat{f}'\big) = f'$ [4].

Case 2: $e_k = e$, and e_k is not error-free.

We have $\hat{f} = \mathcal{B}[h(e)].count \times s(e)$ [5]. In this case, we add $s(e)$ to $\mathcal{B}[h(e)].count$. If no error-free item is removed from the Heavy Part, we have $\hat{f}' = (\mathcal{B}[h(e)].count + s(e)) \times s(e) = \hat{f} + 1 = \hat{f} + \Delta f$, meaning that $\mathbb{E}\big(\hat{f}'\big) = f'$.

Otherwise, suppose e_r [6] is the error-free item in Heavy Part that is replaced by e. We have $\mathcal{B}[h(e)].count' = \mathcal{B}[h(e)].count + s(e) + f_r \times s(e_r)$. Thus, we have $\hat{f}' = \mathcal{B}[h(e)].count' \times s(e) = \hat{f} + \Delta f + f_r \times s(e_r) \times s(e)$.

Now, we prove the conditional expectation $\mathbb{E}\big(f_r \times s(e_r) \times s(e)\big) = 0$ over Ω_2. Notice that f_r, $s(e_r)$, and $s(e)$ are independent from each other. Consider two subsets of Ω_2: Ω_2^1 and Ω_2^2, where $\Omega_2^1 := \{s : s \in \Omega_2 \wedge s(e_r) = 1\}$, and $\Omega_2^2 := \{s : s \in \Omega_2 \wedge s(e_r) = -1\}$. We can see that Ω_2^1 and Ω_2^2 form a partition of Ω_2. Next, we prove $|\Omega_2^1| = |\Omega_2^2|$. For an arbitrary function $s(\cdot) \in \Omega_2^1$, we can find another function $s'(\cdot) \in \Omega_2^2$ (we call $s'(\cdot)$ the *dual function* of $s(\cdot)$), where we define $s'(e_r) = -1$ and $s'(e_j) = s(e_j)$ for $\forall e_j \in [m] \wedge e_j \neq e_r$. As e_r is always error-free before the k_{th} insertion, the value of $s(e_r)$ does not make any difference to the WavingSketch. Thus, $s'(\cdot)$ also causes case 2, and we have $s'(\cdot) \in \Omega_2^2$. Similarly, for any function $s'(\cdot) \in \Omega_2^2$, we can find its dual function $s''(\cdot)$ in Ω_2^1, and we have $s''(\cdot) = s(\cdot)$. In this way, we get a bijection between Ω_2^1 and Ω_2^2, and thus we have $|\Omega_2^1| = |\Omega_2^2| = \frac{1}{2}|\Omega_2|$. Then we have $\mathbb{E}\left(f_r \times s(e_r) \times s(e)\right) = \sum_{s\in\Omega_2} \mathbb{P}\left(s\right)\left(f_r \times s(e_r) \times s(e)\right) = \left(f_r \times s(e)\right)\left(\sum_{s\in\Omega_2^1} \frac{1}{|\Omega_2|} - \sum_{s\in\Omega_2^2} \frac{1}{|\Omega_2|}\right) = 0$.

Finally, we have $\mathbb{E}\big(\hat{f}'\big) = \mathbb{E}\big(\hat{f}\big) + \Delta f = f + \Delta f = f'$.

Case 3: $e_k \neq e$, and e_k is error-free.

In this case, we just increment the corresponding frequency of e_k in the Heavy Part by one, which does not affect the estimated frequency of e. Thus, we have $\Delta\hat{f} = \Delta f = 0$, meaning that $\mathbb{E}\big(\hat{f}'\big) = f'$.

Case 4: $e_k \neq e$, and e_k is not error-free.

We consider two subcases of case 4, where we discuss whether e is error-free before the k_{th} insertion.

Subcase 4.1: e is error-free.

In this subcase, if e is not removed from the Heavy Part by e_k, it will remain error-free after the k_{th} insertion, meaning that $\mathbb{E}\big(\hat{f}'\big) = f'$.

Otherwise, e is replaced by e_k, and inserted into $\mathcal{B}[h(e)].count$. We have $\mathcal{B}[h(e)].count' = \mathcal{B}[h(e)].count + s(e_k) + s(e) \times f$, and $\hat{f}' = \mathcal{B}[h(e)].count' \times s(e) = f + (\mathcal{B}[h(e)].count + s(e_k)) \times s(e)$. Notice that e is error-free before the k_{th} insertion, meaning that e has not been inserted into $\mathcal{B}[h(e)].count$ before the k_{th} insertion, and thus the value of $s(e)$ does not affect $\mathcal{B}[h(e)].count$. In other words, $\mathcal{B}[h(e)].count$ and $s(e)$ are independent. In addition, $s(e_k)$ and $s(e)$ are also independent. Similar as in case 2, we can prove $\mathbb{E}\left((\mathcal{B}[h(e)].count + s(e_k)) \times s(e)\right) = 0$ by dividing the current event into two equal-sized parts according to the value of $s(e)$. In this way, we have $\mathbb{E}\big(\hat{f}'\big) = f = f'$.

Subcase 4.2: e is not error-free.

We have $\hat{f} = \mathcal{B}[h(e)].count \times s(e)$. In this subcase, if no error-free item is removed from the Heavy Part after the k_{th} insertion (denote this event as Ω_{421}), we have $\mathcal{B}[h(e)].count' = \mathcal{B}[h(e)].count + s(e_k)$, and $\hat{f}' = \mathcal{B}[h(e)].count' \times s(e) = \hat{f} + s(e_k) \times s(e)$. Thus, we have $\mathbb{E}\big(\hat{f}'|\Omega_{421}\big) = \sum_{s\in\Omega_{421}} \mathbb{P}(s)\left(\hat{f} + s(e_k) \times s(e)\right)$.

Otherwise (denote this event as Ω_{422}. Note that Ω_{421} and Ω_{422} form a partition of the event of subcase 4.2, namely Ω_{42}.), suppose e_r is the replaced error-free item. We have $\mathcal{B}[h(e)].count' = \mathcal{B}[h(e)].count + s(e_k) + f_r \times s(e_r)$, and $\hat{f}' = \mathcal{B}[h(e)].count' \times s(e) = \hat{f} + s(e_k) \times s(e) + f_r \times s(e_r) \times s(e)$. Thus, we have $\mathbb{E}\big(\hat{f}'|\Omega_{422}\big) = \sum_{s\in\Omega_{422}} \mathbb{P}(s)\big(\hat{f} + s(e_k) \times s(e) + f_r \times s(e_r) \times s(e)\big)$.

We can see that $\mathbb{E}\big(\hat{f}'\big) = \mathbb{E}\big(\hat{f}'|\Omega_{421}\big) + \mathbb{E}\big(\hat{f}'|\Omega_{422}\big) = \mathbb{E}\big(\hat{f}\big) + \sum_{s\in\Omega_{42}} \mathbb{P}(s) \times s(e_k) \times s(e) + \sum_{s\in\Omega_{422}} \mathbb{P}(s) \times f_r \times s(e_r) \times s(e) = f' + \mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) + \mathbb{E}\big(f_r \times s(e_r) \times s(e)|\Omega_{422}\big)$. Next, we separately prove that $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$ and $\mathbb{E}\big(f_r \times s(e_r) \times s(e)|\Omega_{422}\big) = 0$.

4.2.1) Proof of $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$.

In subcase 4.2, both e_k and e are not error-free before the k_{th} insertion. Let t_k and t be the time when e_k and e become not error-free, *i.e.*, e_k becomes not error-free at the t_{kth} insertion, and e becomes not error-free at the t_{th} insertion.

[4] Strictly speaking, $\mathbb{E}$ is a conditional expectation over Ω_1, which should be formally written as $\mathbb{E}\big(\hat{f}'|\Omega_1\big)$. For simplicity, we write the conditional expectation in case i (or in subcase $i.j$), namely $\mathbb{E}\big(X|\Omega_i\big)$ (or $\mathbb{E}\big(X|\Omega_{ij}\big)$), as $\mathbb{E}\big(X\big)$.

[5] Let $\mathcal{B}[h(e)].count$ and $\mathcal{B}[h(e)].count'$ be the value of Waving Counter before and after the k_{th} insertion, respectively.

[6] e_r is the least frequent item in the Heavy Part.

Without loss of generality, we first consider the situation when $t_k < t$. Notice that $s(e_k)$ and $s(e)$ are independent from each other. Similar as in case 2, we can divide the event of subcase 4.2, namely Ω_{42} into two equal-sized parts according to the value of $s(e)$: $\Omega_{42}^1 := \{s : s \in \Omega_{42} \wedge s(e) = 1\}$, and $\Omega_{42}^2 := \{s : s \in \Omega_{42} \wedge s(e) = -1\}$. We can see that Ω_{42}^1 and Ω_{42}^2 from a partition of Ω_{42}. Similar as in case 2, for an arbitrary function $s(\cdot) \in \Omega_{42}^1$, we define its *dual function* $s'(\cdot)$ as follows: $s'(e) = -1$ and $s'(e_j) = s(e_j)$ for $\forall e_j \in [m] \wedge e_j \neq e$. As e remains error-free before the t_{th} insertion, the value of $s(e)$ does not make any difference to the WavingSketch before the t_{th} insertion. This means that when using $s'(\cdot)$, e_k and e also become not error-free at t_k and t, respectively. Thus, when using $s'(\cdot)$, e_k and e are also not error-free before the k_{th} insertion, so we have $s'(\cdot) \in \Omega_{42}^2$. Similarly, for any function $s'(\cdot) \in \Omega_{42}^2$, we can find its dual function $s''(\cdot)$ in Ω_{42}^1, and we have $s''(\cdot) = s(\cdot)$. In this way, we get a bijection between Ω_{42}^1 and Ω_{42}^2, and thus we have $|\Omega_{42}^1| = |\Omega_{42}^2| = \frac{1}{2}|\Omega_{42}|$. Then we have $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = \sum_{s\in\Omega_{42}} \mathbb{P}(s) \times s(e_k) \times s(e)$ $= s(e_k)\left(\sum_{s\in\Omega_{42}^1} \frac{1}{|\Omega_{42}|} - \sum_{s\in\Omega_{42}^2} \frac{1}{|\Omega_{42}|}\right) = 0$.

When $t_k > t$ (note that $t_k \neq t$), we can also prove that $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$ by dividing Ω_{42} into a partition according to the value of $s(e_k)$. In this way, we have proved that $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$.

4.2.2) Proof of $\mathbb{E}\big(f_r \times s(e_r) \times s(e)|\Omega_{422}\big) = 0$.

Notice that f_r, $s(e_r)$, and $s(e)$ are independent from each other. Similar as in case 2, we can prove that the conditional expectation is zero by dividing Ω_{422} into two equal-sized parts according to the value of $s(e_r)$.

Now, we have proved that $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$ and $\mathbb{E}\big(f_r \times s(e_r) \times s(e)|\Omega_{422}\big) = 0$. Thus, we finally have $\mathbb{E}\big(\hat{f}'\big) = f'$ in subcase 4.2.

Conclusion: We have proved that the theorem holds for both the base case and the induction step. Therefore, the theorem holds for any k, and thus holds for the entire data stream σ. □

Theorem 2. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ, the estimated frequency of e made by multi-counter WavingSketch is also unbiased.*

Proof. We prove this theorem by making small modifications to the proof of Theorem 1:

1) In case 2 and subcase 4.1, we replace $\mathcal{B}[h(e)].count$ with $\mathcal{B}[h(e)].count[g(e)]$.

2) In subcase 4.2, we modify Ω_{421} to the event of no error-free item mapped into $\mathcal{B}[h(e)].count[g(e)]$ is removed from the Heavy Part after the insertion of e_k. We modify Ω_{422} to the event of an error-free item e_r mapped into $\mathcal{B}[h(e)].count[g(e)]$ is removed from the Heavy Part after the insertion of e_k (we have $g(e_r) = g(e)$). In this way, Ω_{421} and Ω_{422} also form a partition of Ω_{42}, and similarly, we can also separately prove that $\mathbb{E}\big(s(e_k) \times s(e)|\Omega_{42}\big) = 0$ and $\mathbb{E}\big(f_r \times s(e_r) \times s(e)|\Omega_{422}\big) = 0$ as above.

The rest of the proof is the same as the proof of Theorem 1. □

2.2 Proof of Estimation Variance on Frequency Estimation

We derive the variance of the estimated frequency of WavingSketch. We first consider the basic version of WavingSketch in Theorem 3. Then we extend the formula to multi-counter WavingSketch in Theorem 4. We use the same probability space defined in § 2.1.

Theorem 3. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ (suppose e is not error-free, and let Ω' be the current event). Consider the basic version of WavingSketch. Let $S_1 \subseteq [m]$ be the set of all items mapped into $\mathcal{B}[h(e)]$ that are not error-free. Let $S_1' = S_1 \backslash \{e\}$. The variance of the estimated frequency of e, namely $Var\big(\hat{f}\big)$, satisfies the following bound:*

$$Var\big(\hat{f}\big) \leqslant |S_1'| \times \sum_{e_j \in S_1'} f_j^2$$

where $|S_1'|$ denotes the cardinality of set S_1'.

Proof. We have

$$Var\big(\hat{f}\big) = \mathbb{E}\big((\hat{f} - f)^2\big) = \sum_{s\in\Omega'} \mathbb{P}(s) \times \big(\mathcal{B}[h(e)].count \times s(e) - f\big)^2$$

Notice that $\mathcal{B}[h(e)].count = \sum_{e_j\in S_1} f_j \times s(e_j) = f \times s(e) + \sum_{e_j\in S_1'} f_j \times s(e_j)$.

We have

$$Var\big(\hat{f}\big) = \sum_{s\in\Omega'} \mathbb{P}(s) \left(\sum_{e_j\in S_1'} f_j \times s(e_j) \times s(e)\right)^2$$

Thus, we have

$$Var(\hat{f}) \leqslant \sum_{s\in\Omega'} \mathbb{P}(s) \left(|S_1'| \times \sum_{e_j \in S_1'} f_j^2 \right) = |S_1'| \times \sum_{e_j \in S_1'} f_j^2$$

□ □

Theorem 4. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ (suppose e is not error-free). Consider the multi-counter version of WavingSketch. Let $S_2 \subseteq [m]$ be the set of all items mapped into $\mathcal{B}[h(e)].count[g(e)]$ that are not error-free. Let $S_2' = S_2 \backslash \{e\}$.*

The variance of the estimated frequency of e, namely $Var(\hat{f})$, satisfies the following bound:

$$Var(\hat{f}) \leqslant |S_2'| \times \sum_{e_j \in S_2'} f_j^2$$

where $|S_2'|$ denotes the cardinality of set S_2'.

The proof is similar to the proof of Theorem 4. Notice that $S_2' \subseteq S_1'$. Thus, the variance of multi-counter WavingSketch is smaller than that of the basic WavingSketch.

2.3 Proof of Estimation Error Bound on Frequency Estimation

We first derive the general error bound of WavingSketch without distribution assumption in Theorem 5-6. Then we derive the error bound of WavingSketch under Zipf distribution in Theorem 7-8. We directly consider multi-counter WavingSketch in this subsection.

We first derive the error bound of WavingSketch without distribution assumption. We use L2-norm and L1-norm to derive Theorem 5 and Theorem 6.

Theorem 5. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ. Let $||F_e||_2 = \sqrt{\sum_{e_j \in S_2'} f_j^2}$, where S_2' is defined in Theorem 4. The estimated frequency of item e, namely $\hat{f}$, satisfies the following error bound:*

$$\mathbb{P}\left(\left| \hat{f} - f \right| \geqslant \epsilon \sqrt{|S_2'|} \cdot ||F_e||_2 \right) \leqslant \frac{1}{\epsilon^2}$$

Proof. According to Chebyshev's inequality, we have

$$\mathbb{P}\left(\left| \hat{f} - f \right| \geqslant \epsilon \sqrt{|S_2'|} \cdot ||F_e||_2 \right) \leqslant \frac{Var(\hat{f})}{\epsilon^2 |S_2'| \cdot \sum_{e_j \in S_2'} f_j^2} \leqslant \frac{1}{\epsilon^2}$$

□

Theorem 6. *Given a data stream σ and an arbitrary item $e \in [m]$ in σ. Let $||F_e||_1 = \left| \sum_{e_j \in S_2'} f_j \right|$, where S_2' is defined in Theorem 4. The estimated frequency of item e, namely $\hat{f}$, satisfies the following error bound: $\mathbb{P}\left(\left| \hat{f} - f \right| \geqslant \epsilon \cdot ||F_e||_1 \right) \leqslant \frac{1}{\epsilon}$.*

Proof. We have

$$\mathbb{E}\left(\left| \hat{f} - f \right| \right) = \mathbb{E}\left(\left| \sum_{e_j \in S_2'} f_j \times s(e_j) \right| \right) \leqslant \mathbb{E}\left(\left| \sum_{e_j \in S_2'} f_j \right| \right) = ||F_e||_1$$

According to Markov's inequality, we have

$$\mathbb{P}\left(\left| \hat{f} - f \right| \geqslant \epsilon \cdot ||F_e||_1 \right) \leqslant \frac{\mathbb{E}\left(\left| \hat{f} - f \right| \right)}{\epsilon \cdot ||F_e||_1} \leqslant \frac{1}{\epsilon}$$

□

We then derive the error bound of WavingSketch under Zipf distribution. We can see that Theorem 5-6 are only dependent on the items mapped into the same Waving Counter as e, and they do not consider the parameters of WavingSketch. Next, we define a more complete probability space considering the parameters of WavingSketch, in which we derive the error bound of WavingSketch.

Consider a fixed data stream σ. The randomness of multi-counter WavingSketch comes from its three hash functions: $h(\cdot)$, $g(\cdot)$, and $s(\cdot)$. Thus, we define our probability based on the sample space of all possible $h(\cdot)$, $g(\cdot)$, and $s(\cdot)$. Formally, we define the probability

space $(\overline{\Omega}, \mathcal{F}, \mathbb{P})$ by setting $\overline{\Omega} = \mathcal{H} \times \mathcal{G} \times \mathcal{S}$, [7], $\mathcal{F} = 2^{\overline{\Omega}}$, and $\mathbb{P}(A) = \frac{|A|}{|\overline{\Omega}|}$, where $\mathcal{H} := \{h : [m] \to [l]\}$, $\mathcal{G} := \{g : [m] \to [c]\}$, and $\mathcal{S} := \{s : [m] \to \{+1, -1\}\}$. We suppose the items in data stream σ come from a skewed Zipf [1] distribution: the k_{th} most frequent item in $[m]$ shows up $\frac{n}{k^\alpha \zeta(\alpha)}$ times, where α [8] is the parameter of Zipf distribution and $\zeta(\alpha) = \sum_{i=1}^{m} \frac{1}{i^\alpha}$.

Next, we use L2-norm and L1-norm to derive Theorem 7 and Theorem 8.

Theorem 7. *Given a data stream σ that comes from a Zipf distribution with the parameter $\alpha > 1$. Let $||F||_2 = \sqrt{\sum_{e_j \in [m]} f_j^2}$. Let $Z = \left(\frac{m}{\zeta(\alpha)}\right)^{\frac{1}{\alpha}}$ [9], meaning that the frequency of the Z_{th} most frequent items is $\frac{n}{m}$. For an arbitrary item $e \in [m]$ in σ, its estimated frequency $\hat{f}$ has the following error bound:*

$$\mathbb{P}\left(\left|\hat{f} - f\right| \geqslant \epsilon ||F||_2\right) \leqslant \frac{Z}{lc} + \frac{4m}{\epsilon^2 l^2 c^2} + \frac{2lc}{m} \tag{1}$$

Proof. Let $Q \subseteq \overline{\Omega}$ be the event of $|\hat{f} - f| \geqslant \epsilon ||F||_2$. Let $R \subseteq \overline{\Omega}$ be the event of $||F_e||_2 > \sqrt{\frac{2}{lc}} \cdot ||F||_2$, where $||F_e||_2$ is defined in Theorem 5. Next, we derive the upper bound of $\mathbb{P}(Q)$ by dividing Q into two parts using R and $\Omega \backslash R$ as conditions:

$$\begin{aligned} \mathbb{P}(Q) =& \mathbb{P}(R) \cdot \mathbb{P}(Q|R) + \mathbb{P}(\Omega \backslash R) \cdot \mathbb{P}(Q|\Omega \backslash R) \\ \leqslant& \mathbb{P}(R) \cdot 1 + \mathbb{P}((\Omega \backslash R) \cap Q) \end{aligned} \tag{2}$$

Part 1: upper bound of $\mathbb{P}(R)$.

Let $S_Z \subseteq [m]$ be the set of the most frequent Z items, meaning that for $\forall e_j \in S_Z$, $f_j \geqslant \frac{n}{m}$. We consider the following two events: 1) the event of $S_2' \cap S_Z \neq \varnothing$, which is denoted as $R1$; and 2) the event of $|S_2| \geqslant \frac{2m}{lc} + 1$, which is denoted as $R2$. We consider using $R1 \cup R2$ to cover R, so that to derive the upper bound of $\mathbb{P}(R)$. Now, we first prove that $R \subseteq R1 \cup R2$ by proving $R \backslash R1 \subseteq R2$ as follows. When $R \backslash R1$ happens, *i.e.*, $S_2' \cap S_Z = \varnothing$, we have $|S_2| = 1 + |S_2'| \geqslant 1 + \sum_{e_j \in S_2'} \frac{f_j^2}{(n/m)^2} = 1 + \frac{||F_e||_2^2}{(n/m)^2}$. Thus, we have

$$\begin{aligned} |S_2| &\geqslant 1 + \frac{\frac{2}{lc} \cdot ||F||_2^2}{(n/m)^2} = 1 + \frac{2m^2 \cdot ||F||_2^2}{n^2 \cdot lc} \\ &\geqslant 1 + \frac{2m \cdot ||F||_1^2}{n^2 \cdot lc} = 1 + \frac{2m}{lc} \end{aligned} \tag{3}$$

where $||F||_1 = \sum_{e_j \in [m]} f_j = n$.

From Equation 14, we conclude that $R \backslash R1 \subseteq R2$, and thus $R \subseteq R1 \cup R2$. In this way, we have $\mathbb{P}(R) \leqslant \mathbb{P}(R1) + \mathbb{P}(R2)$. Next, we separately derive the upper bound of $R1$ and $R2$.

Subpart 1.1: upper bound of $\mathbb{P}(R1)$.

We have $\mathbb{P}(R1) = 1 - \mathbb{P}(\overline{\Omega} \backslash R1)$. When $\overline{\Omega} \backslash R1$ happens, *i.e.*, $S_2' \cap S_Z = \varnothing$, all the Z items in S_Z are not mapped into the Waving Counter of e, namely $\mathcal{B}[h(e)].count[g(e)]$. Consider an arbitrary item $e_j \in S_Z$, the probability that e_j is not mapped into $\mathcal{B}[h(e)].count[g(e)]$ is $1 - \frac{1}{lc}$. Thus, we have $\mathbb{P}(\overline{\Omega} \backslash R1) = \left(1 - \frac{1}{lc}\right)^Z$. According to Bernoulli's inequality, we have

$$\mathbb{P}(R1) = 1 - \mathbb{P}\left(\overline{\Omega} \backslash R1\right) = 1 - \left(1 - \frac{1}{lc}\right)^Z \leqslant \frac{Z}{lc} \tag{4}$$

Subpart 1.2: upper bound of $\mathbb{P}(R2)$.

First, we can treat the cardinality of set S_2' as a binomial distribution with the parameters of $m - 1$ and $\frac{1}{lc}$, *i.e.*, $|S_2'| \sim B(m-1, \frac{1}{lc})$. Thus, we have $\mathbb{E}(|S_2'|) = \frac{m-1}{lc}$, and $Var\left(|S_2'|\right) = (m-1)\left(\frac{1}{lc}\right)\left(1 - \frac{1}{lc}\right)$.

We have $\mathbb{E}(|S_2|) = 1 + \mathbb{E}(|S_2'|) = 1 + \frac{m-1}{lc}$. We have $\mathbb{P}(R2) = \mathbb{P}\left(|S_2| \geqslant \frac{2m}{lc} + 1\right) = \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m+1}{lc}\right) \leqslant \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m}{lc}\right)$. According to Chebyshev's inequality, we have:

[7] In this subsection, we use $t = \langle h(\cdot), g(\cdot), s(\cdot) \rangle$ to denote a triple in sample space $\overline{\Omega}$, which is a set of configuration of the hash functions of a multi-counter WavingSketch. In other words, the randomness of the estimation results of WavingSketch is completely determined by t.

[8] We assume $\alpha > 1$ so that the series $\sum_{i=1}^{\infty} \frac{1}{i^\alpha}$ converges.

[9] Notice that Z is a constant determined by data stream σ.

$$\begin{aligned}\mathbb{P}(R2) &\leqslant \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m}{lc}\right) \leqslant \frac{Var\left(|S_2'|\right)}{\left(\frac{m}{lc}\right)^2} \\ &= (m-1)\left(\frac{1}{lc}\right)\left(1-\frac{1}{lc}\right)\left(\frac{lc}{m}\right)^2 \leqslant \frac{lc}{m}\end{aligned} \tag{5}$$

From Equation 15-16, we finally have

$$\mathbb{P}(R) \leqslant \mathbb{P}(R1) + \mathbb{P}(R2) \leqslant \frac{Z}{lc} + \frac{lc}{m} \tag{6}$$

Part 2: upper bound of $\mathbb{P}((\Omega \backslash R) \cap Q)$.

Let $W \subseteq \overline{\Omega}$ be the event of $\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_2$. We first prove that $(\Omega \backslash R) \cap Q \subseteq W$ as follows. When $(\Omega \backslash R) \cap Q$ happens, we have $\left|\hat{f} - f\right| \geqslant \epsilon||F||_2$, and $||F||_2 \geqslant \sqrt{\frac{lc}{2}} \cdot ||F_e||_2$. Thus, we have $\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_2$, meaning that $(\Omega \backslash R) \cap Q \subseteq W$. Thus, we have

$$\mathbb{P}((\Omega \backslash R) \cap Q) \leqslant \mathbb{P}(W) \tag{7}$$

We further divide event W into two parts using $R2$, which is defined in part 1. In this way, we have

$$\begin{aligned}\mathbb{P}(W) &= \mathbb{P}(R2) \cdot \mathbb{P}(W|R2) + \mathbb{P}\left(\overline{\Omega} \backslash R2\right) \cdot \mathbb{P}\left(W|\overline{\Omega} \backslash R2\right) \\ &\leqslant \mathbb{P}(R2) + \mathbb{P}\left(W|\overline{\Omega} \backslash R2\right)\end{aligned} \tag{8}$$

Next, we derive the upper bound of $\mathbb{P}\left(W|\overline{\Omega} \backslash R2\right)$ in the probability space of event $\overline{\Omega} \backslash R2$. Formally, we define the probability space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$ by setting $\hat{\Omega} = \overline{\Omega} \backslash R2$, $\hat{\mathcal{F}} = 2^{\hat{\Omega}}$, and $\hat{\mathbb{P}}(A) = \frac{|A|}{|\hat{\Omega}|}$. According to Chebyshev's inequality, we have

$$\begin{aligned}\mathbb{P}\left(W|\overline{\Omega} \backslash R2\right) &= \hat{\mathbb{P}}\left(\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_2\right) \\ &\leqslant \frac{Var(\hat{f})}{\epsilon^2\left(\frac{lc}{2}\right) \cdot ||F_e||_2^2} \leqslant \frac{|S_2'| \cdot ||F_e||_2^2}{\epsilon^2\left(\frac{lc}{2}\right) \cdot ||F_e||_2^2}\end{aligned} \tag{9}$$

According to the definition of $\overline{\Omega} \backslash R2$, we have

$$\mathbb{P}\left(W|\overline{\Omega} \backslash R2\right) \leqslant \frac{\frac{2m}{lc}}{\epsilon^2\left(\frac{lc}{2}\right)} \leqslant \frac{4m}{\epsilon^2 l^2 c^2} \tag{10}$$

From Equation 16 and Equation 18-10, we have

$$\mathbb{P}((\Omega \backslash R) \cap Q) \leqslant \frac{lc}{m} + \frac{4m}{\epsilon^2 l^2 c^2} \tag{11}$$

Finally, from Equation 13, Equation 17 and Equation 21, we prove that

$$\mathbb{P}\left(\left|\hat{f} - f\right| \geqslant \epsilon||F||_2\right) \leqslant \frac{Z}{lc} + \frac{4m}{\epsilon^2 l^2 c^2} + \frac{2lc}{m}$$

□

Theorem 8. *Given a data stream σ that comes from a Zipf distribution with the parameter $\alpha > 1$. Let $||F||_1 = \left|\sum_{e_j \in [m]} f_j\right|$. Let $Z = \left(\frac{m}{\zeta(\alpha)}\right)^{\frac{1}{\alpha}}$, meaning that the frequency of the Z_{th} most frequent items is $\frac{n}{m}$. For an arbitrary item $e \in [m]$ in σ, its estimated frequency $\hat{f}$ has the following error bound:*

$$\mathbb{P}\left(\left|\hat{f} - f\right| \geqslant \epsilon||F||_1\right) \leqslant \frac{Z}{lc} + \frac{\sqrt{2}}{\epsilon\sqrt{lc}} + \frac{2lc}{m} \tag{12}$$

Proof. Let $Q \subseteq \overline{\Omega}$ be the event of $|\hat{f} - f| \geqslant \epsilon||F||_1$. Let $R \subseteq \overline{\Omega}$ be the event of $||F_e||_1 > \sqrt{\frac{2}{lc}} \cdot ||F||_1$, where $||F_e||_1$ is defined in Theorem 6. Next, we derive the upper bound of $\mathbb{P}(Q)$ by dividing Q into two parts using R and $\Omega \backslash R$ as conditions:

$$\begin{aligned}\mathbb{P}(Q) =& \mathbb{P}(R) \cdot \mathbb{P}(Q|R) + \mathbb{P}(\Omega \backslash R) \cdot \mathbb{P}(Q|\Omega \backslash R) \\ \leqslant& \mathbb{P}(R) \cdot 1 + \mathbb{P}((\Omega \backslash R) \cap Q)\end{aligned} \tag{13}$$

Part 1: upper bound of $\mathbb{P}(R)$.

Let $S_Z \subseteq [m]$ be the set of the most frequent Z items, meaning that for $\forall e_j \in S_Z$, $f_j \geqslant \frac{n}{m}$. We consider the following two events: 1) the event of $S_2' \cap S_Z \neq \varnothing$, which is denoted as $R1$; and 2) the event of $|S_2| \geqslant \frac{2m}{lc} + 1$, which is denoted as $R2$. We consider using $R1 \cup R2$ to cover R, so that to derive the upper bound of $\mathbb{P}(R)$. Now, we first prove that $R \subseteq R1 \cup R2$ by proving $R \backslash R1 \subseteq R2$ as follows. When $R \backslash R1$ happens, *i.e.*, $S_2' \cap S_Z = \varnothing$, we have $|S_2| = 1 + |S_2'| \geqslant 1 + \sum_{e_j \in S_2'} \frac{f_j}{(n/m)} = 1 + \frac{||F_e||_1}{(n/m)}$. Thus, we have

$$\begin{aligned}|S_2| &\geqslant 1 + \frac{\frac{2}{lc} \cdot ||F||_1}{(n/m)} = 1 + \frac{2m \cdot ||F||_1}{nlc} \\ &= 1 + \frac{2m}{lc}\end{aligned} \tag{14}$$

From Equation 14, we conclude that $R \backslash R1 \subseteq R2$, and thus $R \subseteq R1 \cup R2$. In this way, we have $\mathbb{P}(R) \leqslant \mathbb{P}(R1) + \mathbb{P}(R2)$. Next, we separately derive the upper bound of $R1$ and $R2$.

Subpart 1.1: upper bound of $\mathbb{P}(R1)$.

We have $\mathbb{P}(R1) = 1 - \mathbb{P}(\overline{\Omega} \backslash R1)$. When $\overline{\Omega} \backslash R1$ happens, *i.e.*, $S_2' \cap S_Z = \varnothing$, all the Z items in S_Z are not mapped into the Waving Counter of e, namely $\mathcal{B}[h(e)].count[g(e)]$. Consider an arbitrary item $e_j \in S_Z$, the probability that e_j is not mapped into $\mathcal{B}[h(e)].count[g(e)]$ is $1 - \frac{1}{lc}$. Thus, we have $\mathbb{P}(\overline{\Omega} \backslash R1) = \left(1 - \frac{1}{lc}\right)^Z$. According to Bernoulli's inequality, we have

$$\mathbb{P}(R1) = 1 - \mathbb{P}\left(\overline{\Omega} \backslash R1\right) = 1 - \left(1 - \frac{1}{lc}\right)^Z \leqslant \frac{Z}{lc} \tag{15}$$

Subpart 1.2: upper bound of $\mathbb{P}(R2)$.

First, we can treat the cardinality of set S_2' as a binomial distribution with the parameters of $m - 1$ and $\frac{1}{lc}$, *i.e.*, $|S_2'| \sim B(m-1, \frac{1}{lc})$. Thus, we have $\mathbb{E}(|S_2'|) = \frac{m-1}{lc}$, and $Var\left(|S_2'|\right) = (m-1)\left(\frac{1}{lc}\right)\left(1 - \frac{1}{lc}\right)$.

We have $\mathbb{E}(|S_2|) = 1 + \mathbb{E}(|S_2'|) = 1 + \frac{m-1}{lc}$. We have $\mathbb{P}(R2) = \mathbb{P}\left(|S_2| \geqslant \frac{2m}{lc} + 1\right) = \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m+1}{lc}\right) \leqslant \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m}{lc}\right)$. According to Chebyshev's inequality, we have:

$$\begin{aligned}\mathbb{P}(R2) &\leqslant \mathbb{P}\left(|S_2| - \mathbb{E}(|S_2|) \geqslant \frac{m}{lc}\right) \leqslant \frac{Var\left(|S_2'|\right)}{\left(\frac{m}{lc}\right)^2} \\ &= (m-1)\left(\frac{1}{lc}\right)\left(1 - \frac{1}{lc}\right)\left(\frac{lc}{m}\right)^2 \leqslant \frac{lc}{m}\end{aligned} \tag{16}$$

From Equation 15-16, we finally have

$$\mathbb{P}(R) \leqslant \mathbb{P}(R1) + \mathbb{P}(R2) \leqslant \frac{Z}{lc} + \frac{lc}{m} \tag{17}$$

Part 2: upper bound of $\mathbb{P}((\Omega \backslash R) \cap Q)$.

Let $W \subseteq \overline{\Omega}$ be the event of $\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_1$. We first prove that $(\Omega \backslash R) \cap Q \subseteq W$ as follows. When $(\Omega \backslash R) \cap Q$ happens, we have $\left|\hat{f} - f\right| \geqslant \epsilon||F||_1$, and $||F||_1 \geqslant \sqrt{\frac{lc}{2}} \cdot ||F_e||_1$. Thus, we have $\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_2$, meaning that $(\Omega \backslash R) \cap Q \subseteq W$. Thus, we have

$$\mathbb{P}((\Omega \backslash R) \cap Q) \leqslant \mathbb{P}(W) \tag{18}$$

We further divide event W into two parts using $R2$, which is defined in part 1. In this way, we have

$$\begin{aligned}\mathbb{P}(W) &= \mathbb{P}(R2) \cdot \mathbb{P}(W|R2) + \mathbb{P}\left(\overline{\Omega} \backslash R2\right) \cdot \mathbb{P}\left(W|\overline{\Omega} \backslash R2\right) \\ &\leqslant \mathbb{P}(R2) + \mathbb{P}\left(W|\overline{\Omega} \backslash R2\right)\end{aligned} \tag{19}$$

Next, we derive the upper bound of $\mathbb{P}\left(W|\overline{\Omega}\backslash R2\right)$ in the probability space of event $\overline{\Omega}\backslash R2$. Formally, we define the probability space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$ by setting $\hat{\Omega} = \overline{\Omega}\backslash R2$, $\hat{\mathcal{F}} = 2^{\hat{\Omega}}$, and $\hat{\mathbb{P}}(A) = \frac{|A|}{|\hat{\Omega}|}$. According to Markov's inequality, we have

$$\begin{aligned}\mathbb{P}\left(W|\overline{\Omega}\backslash R2\right) &= \hat{\mathbb{P}}\left(\left|\hat{f} - f\right| \geqslant \epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_1\right) \\ &\leqslant \frac{\mathbb{E}\left(\left|\hat{f} - f\right|\right)}{\epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_1} \leqslant \frac{||F_e||_1}{\epsilon\sqrt{\frac{lc}{2}} \cdot ||F_e||_1} = \frac{\sqrt{2}}{\epsilon\sqrt{lc}}\end{aligned} \tag{20}$$

From Equation 16 and Equation 18-20, we have

$$\mathbb{P}((\Omega\backslash R) \cap Q) \leqslant \frac{lc}{m} + \frac{\sqrt{2}}{\epsilon\sqrt{lc}} \tag{21}$$

Finally, from Equation 13, Equation 17 and Equation 21, we prove that

$$\mathbb{P}\left(\left|\hat{f} - f\right| \geqslant \epsilon||F||_1\right) \leqslant \frac{Z}{lc} + \frac{\sqrt{2}}{\epsilon\sqrt{lc}} + \frac{2lc}{m}$$

□

2.4 Proof of Unbiasedness on Join-aggregate Estimation

Theorem 11. *Given two data streams σ_1 and σ_2, the estimated result of their join-aggregate query made by WavingSketch is unbiased, namely we have $\mathbb{E}\big(\hat{J}(\sigma_1, \sigma_2)\big) = J(\sigma_1, \sigma_2)$.*

Proof. Consider $\mathcal{B}_1[k]$ and $\mathcal{B}_2[k]$. In the following, we separately prove the three parts of $\hat{J}_k(\sigma_1, \sigma_2)$ are unbiased.

Part 1: For any item $e_i \in \Psi_{k,1}$, its recorded frequencies in $\mathcal{B}_1[k]$ and $\mathcal{B}_2[k]$ are accurate, meaning that $\hat{f}_i = f_i$ and $\hat{g}_i = g_i$. Thus, we directly have $\hat{J}_{k,1}(\sigma_1, \sigma_2) = \sum_{e_i \in \Psi_{k,1}} f_i \cdot g_i$.

Part 2: Consider the first part of $\hat{J}_{k,2}(\sigma_1, \sigma_2)$, namely $\hat{J}^1_{k,2}(\sigma_1, \sigma_2)$. For any item $e_i \in \Psi^1_{k,2}$, its recorded frequency in $\mathcal{B}_1[k].heavy$ is accurate, meaning that $\hat{f}_i = f_i$.

In § 2.1, we have proved that the estimated frequency made by WavingSketch is unbiased, meaning that $\mathbb{E}\big(\hat{g}_i\big) = g_i$.

We have

$$\mathbb{E}\big(\hat{J}^1_{k,2}(\sigma_1, \sigma_2)\big) = \sum_{e_i \in \Psi^1_{k,2}} f_i \cdot \mathbb{E}\big(\hat{g}_i\big) = \sum_{e_i \in \Psi^1_{k,2}} f_i \cdot g_i$$

Similarly, we can prove that

$$\mathbb{E}\big(\hat{J}^2_{k,2}(\sigma_1, \sigma_2)\big) = \sum_{e_i \in \Psi^2_{k,2}} f_i \cdot g_i$$

In this way, we prove that $\mathbb{E}\big(\hat{J}_{k,2}(\sigma_1, \sigma_2)\big) = \sum_{e_i \in \Psi_{k,2}} f_i \cdot g_i$.

Part 3: Let $F^1_k = \sum_{e_i \in \Psi_{k,3}} f_i \times s(e_i)$, $F^2_k = \sum_{e_i \in \Psi^2_{k,2}} f_i \times s(e_i)$, $G^1_k = \sum_{e_i \in \Psi_{k,3}} g_i \times s(e_i)$, and $G^2_k = \sum_{e_i \in \Psi^1_{k,2}} g_i \times s(e_i)$.

We have $\mathcal{B}_1[k].count = F^1_k + F^2_k$, and $\mathcal{B}_2[k].count = G^1_k + G^2_k$.

First, we have

$$\mathbb{E}\big(F^1_k \cdot G^1_k\big) = \sum_{e_i \in \Psi_{k,3}} f_i \cdot g_i + \mathbb{E}\Big(\sum_{e_i \in \Psi_{k,3}} \sum_{e_j \in \Psi_{k,3} \backslash \{e_i\}} f_i \cdot s(e_i) \cdot g_j \cdot s(e_j)\Big) = \sum_{e_i \in \Psi_{k,3}} f_i \cdot g_i$$

Notice that we have $\Psi_{k,3} \cap \Psi^1_{k,2} = \varnothing$, $\Psi^2_{k,2} \cap \Psi_{k,3} = \varnothing$, and $\Psi^2_{k,2} \cap \Psi^1_{k,2} = \varnothing$. Thus, we can prove that $\mathbb{E}\big(F^1_k \cdot G^2_k\big) = 0$, $\mathbb{E}\big(F^2_k \cdot G^1_k\big) = 0$, and $\mathbb{E}\big(F^2_k \cdot G^2_k\big) = 0$.

Finally, we prove that

$$\mathbb{E}\big(\hat{J}_{k,3}(\sigma_1, \sigma_2)\big) = \mathbb{E}\big(\mathcal{B}_1[k].count \times \mathcal{B}_2[k].count\big) = \mathbb{E}\big(F^1_k \cdot G^1_k\big) + \mathbb{E}\big(F^1_k \cdot G^2_k\big) + \mathbb{E}\big(F^2_k \cdot G^1_k\big) + \mathbb{E}\big(F^2_k \cdot G^2_k\big) = \sum_{e_i \in \Psi_{k,3}} f_i \cdot g_i$$

In this way, we prove that for $\forall k \in [l]$, $\mathbb{E}\big(\hat{J}_k(\sigma_1, \sigma_2)\big) = \hat{J}_k(\sigma_1, \sigma_2)$. Thus, we have $\mathbb{E}\big(\hat{J}(\sigma_1, \sigma_2)\big) = J(\sigma_1, \sigma_2)$. □

3 Details of the Experimental Setup

Platform: We conduct experiments on a 36-core CPU server (Intel i9-10980XE) with 128GB DDR4 memory and 25.4MB L3 cache. We set the CPU frequency to 4.2GHz, and set the memory frequency to 3200MHz.

Implementation: We implement WavingSketch and the other algorithms with C++ and build them with g++ 7.5.0. We use 32-bit Murmur Hash (obtained from the open-source website [2]) with different initial seeds.

Datasets:

1) Synthetic datasets: We use Web Polygraph [3], an open-source performance testing tool, to generate 10 synthetic datasets that follow the Zipf [1] distribution. The skewness of the datasets varies from 0.0 to 3.0. Each dataset has 32 million items, each of which has 4-byte ID. We use these datasets to evaluate the performance of WavingSketch on the data with different skewness (§ 4.3). In other experiments using the synthetic dataset, we use the dataset with the skewness of $\alpha = 1.5$ by default.

2) IP trace dataset: The IP trace dataset is a collection of IP traces collected on backbone links by CAIDA 2018 [4]. By default, we treat each packet in the traces as one item, and use its source IP address (4 bytes) as the ID field. We use two traces with different sizes: 1) a small-scale 1-minute trace containing about 30M items (used by default); 2) a large-scale 1-hour trace containing about 1.5G items (used in Figure 11).

3) Webpage dataset: The webpage dataset is built from a collection of web pages, which are downloaded from the website [5]. This dataset contains 512M items, each of which (4 bytes) represents the number of distinct terms in a web page.

4) Network dataset: The network dataset contains users' posting history on a stack exchange website [6]. This dataset contains 10M items, each of which has three fields: (u, v, t), which means user u answered user v's question at time t. We use u as the ID field.

Metrics: We evaluate the performance of WavingSketch using the following metrics. All experiments are repeated 100 times and the average results are reported.

1) Average Relative Error (ARE): $\frac{1}{|\mathbf{\Psi}|}\sum_{e_i \in \mathbf{\Psi}} |f_i - \hat{f}_i|/f_i$, where f_i is the real frequency of item e_i, $\widehat{f_i}$ is the estimated frequency of e_i, and $\mathbf{\Psi}$ is the query set. By default, the query set $\mathbf{\Psi}$ contains all ground-truth frequent items (defined in § 2.1).

2) Recall Rate (RR): The ratio of the number of correctly reported instances to the number of true instances.

3) Precision Rate (PR): The ratio of the number of correctly reported instances to the number of reported instances.

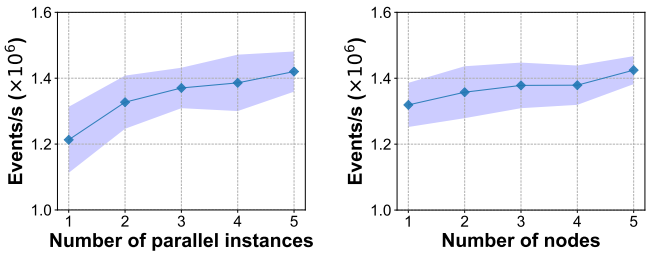
4) F1 Score: $\frac{2\times RR\times PR}{RR+PR}$

5) Throughput: Million operations (insertions/queries) per second (Mops).

4 Details of the Experiments on Apache Flink

We implement WavingSketch on top of Apache Flink [7], showing that our solution can be easily integrated into modern stream processing framework and work in distributed systems. Next, we present the details of the experimental setup and discuss the experimental results.

Setup: We build a Flink cluster with 1 master node and 5 worker nodes, and conduct experiments using the CAIDA [4] dataset. To feed the data into the Flink application, we deploy a Hadoop Distributed File System (HDFS) in our Flink cluster, in which we set the master node as NameNode and the worker nodes as DataNodes. Each of the master node or the worker node has 4 virtual CPU cores of Intel XEON Platinum 8369B, and 8 GB main memory. The job manager and each task manager of Flink are configured with 1 GB of memory. Each node uses Flink 1.13.1, Java 11 and Hadoop 2.8.3 running on Ubuntu 20.04 LTS. We conduct both local experiments and cluster experiments. For local experiments, we run the experiments only in the local mode of the master node. All experiments are repeated 10 times and the average (±std) throughput is plotted.



(a) Local mode (b) Cluster mode

Figure 3: Throughput of WavingSketch on Apache Flink.

Experimental results (Figure 3): We find that WavingSketch can smoothly work on top of Flink framework. As shown in Figure 3(a), in local mode experiments, as the number of parallel instances (called parallelism in Flink) grows from 1 to 5, the throughput of WavingSketch increases from 1.20 million events per second to 1.45 million events per second. As shown in Figure 3(b), in cluster mode, as the number of nodes grows from 1 to 5, the throughput of WavingSketch increases from 1.31 million events per second to 1.73 million events per second. The reason why the throughput grows slowly with the increase of parallelism is because besides the process running WavingSketch, there are other processes such as job manager, Hadoop NameNode running in the system. When the parallelism is 3, the CPU utilization is already high, and after which the increase of parallelism contributes little to the overall throughput. In summary, WavingSketch achieves satisfactory throughput ($1.2 \sim 1.8$ million events per second) in our Flink cluster.

References

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

[2] Murmur hashing source codes. `https://github.com/aappleby/smhasher/blob/master/src/MurmurHash3.cpp`.

[3] Alex Rousskov and Duane Wessels. High-performance benchmarking with web polygraph. *Software: Practice and Experience*, 2004.

[4] CAIDA [on line]. Available: `http://www.caida.org/home`.

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

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

[7] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4), 2015.