A MATHEMATICAL ANALYSIS

In this section, we provide detailed proofs of the theorems mentioned in the paper.

A.1 Burst Filter Speedup

THEOREM A.1. *Assume that in multiple time windows T, there are n_{BF} distinct items, and the total number of items is E_{BF}. The Burst Filter contains w buckets, with each bucket having γ cells. Let P_{Bur} denote the probability of the Burst Filter capturing the data stream. We have $P_{Bur} \longrightarrow 1$.*

PROOF. The Burst Filter can hold $w \times \gamma$ distinct items. Since the total number of items is much larger than $w \times \gamma$, we use the appearance frequency of different distinct items to approximate their probabilities in each time window.

For item e_i, let p_i denote its probability of arriving at the Burst Filter. Initially, the probability that e_i is not recorded in the Burst Filter is $1 - p_i$. After m data arrivals, the probability that e_i remains unrecorded is:

$$P_{not-in}(e_i) \approx \prod_{t=1}^{m}(1 - p_i).$$

Using the approximation $\ln(1 - x) \longrightarrow -x$ as $x \longrightarrow 0$, we have

$$\ln(P_{not-in}(e_i)) \approx \sum_{t=1}^{m} -p_i.$$

According to the law of large numbers, when the number of incoming data reaches $\frac{E_{BF} \times w \times \gamma}{n_{BF}}$, the probability that new items will still be inserted into the Burst Filter approaches zero. Therefore, the probability of item e_i being inserted into the Burst Filter is:

$$P_{in}(e_i) \approx 1 - \exp(-p_i \times \frac{E_{BF} \times w \times \gamma}{n_{BF}}).$$

By applying the Hoeffding inequality, we consider n_{BF} and $w \times \gamma$:

$$\mathbb{P}(\sum p_i P_{in}(e_i) > \epsilon_{BF}) \geq 1 - 2\exp(-\frac{2E_{BF} \times (1 - \epsilon_{BF})^2 w\gamma)}{n_{BF}}.$$

From the above proof, we conclude that in practice, there is a substantial volume of data in data streams, allowing the Burst Filter to capture nearly all data effectively. □

A.2 Persistence Estimation

A.2.1 Error Bound.

THEOREM A.2. *Let $\hat{p}_i$ be the estimated persistence of our method. We have*

$$p_i \leq \hat{p}_i \leq T. \tag{1}$$

PROOF. Let the threshold for L_1 be Δ_1 and the threshold for L_2 be Δ_2. The number of time windows in S is T. If $\hat{p}_i > \Delta_1$, the item is transmitted to L_2; if $\hat{p}_i > \Delta_1 + \Delta_2$, it is sent to L_3. The number of hash functions in L_1, L_2, and L_3 is d_1, d_2, and d_3, respectively. Thus, $\hat{p}_i$ can be expressed as:

$$\hat{p}_i = \hat{p}_i^1 + \hat{p}_i^2 + \hat{p}_i^3,$$

where $\hat{p}_i^1$, $\hat{p}_i^2$, and $\hat{p}_i^3$ represent the estimated persistence in L_1, L_2, and L_3. Subsequently, we determine the upper and lower boundaries of $\hat{p}_i$. For each $\hat{p}_i^j$ $(j = 1, 2, 3)$, each arriving item can trigger the mapping, causing $\hat{p}_i^j$ to increase. Consequently, we have $\hat{p}_i^j \geq p_i^j$ $(j = 1, 2, 3)$. In each time window, $\hat{p}_i^j$ can increase by at most 1, which results in $\hat{p}_i \leq T$.

In summary, we have $p_i \leq \hat{p}_i \leq T$. This result indicates that the estimated persistence provided is within a reasonable range and will not introduce significant deviation when utilized. □

Theorem A.3. *Let n, m and L be the number of buckets in layer L_1, L_2 and L_3, $l = \frac{e}{\epsilon}$ and $d = \ln\left(\frac{1}{\delta}\right)$. For small data streams, we obtain*

$$\mathbb{P}\left(\hat{p}_i \leqslant p_i + \epsilon\|p\|_1\right) \geqslant 1 - \delta. \tag{2}$$

For medium data streams, where $\varepsilon = \frac{e}{n\times m}$ and $\delta = e^{-d_1-d_2}$, it follows that

$$\mathbb{P}(\hat{p}_i \leq p_i + \varepsilon \left\|p\right\|_1 \times \|p\|_1^1) \geq 1 - \delta. \tag{3}$$

For large data streams, $\varepsilon_1 = \frac{e}{n\times m\times L}$ and $\delta = e^{-d_1-d_2-d_3}$, we conclude that

$$\mathbb{P}(\hat{p}_i \leq p_i + \varepsilon_1 \left\|p\right\|_1 \times \|p\|_1^1 \times \|p\|_1^2) \geq 1 - \delta. \tag{4}$$

Proof. For convenience, we define small data streams as those for which $\hat{p}_i < \Delta_1$. Medium data streams are defined as $\Delta_1 < \hat{p}_i \leq \Delta_1 + \Delta_2$, and large data streams correspond to $\hat{p}_i > \Delta_1 + \Delta_2$. We take medium data streams as an example.

For medium data streams, let $\Delta_j p_i = \Delta_j p_i^1 + \Delta_j p_i^2$, where $\Delta_j p_i^1 = C_j^1[h_j(e_i)] - p_i^1$ and $\Delta_j p_i^2 = C_j^2[g_j(e_i)] - p_i^2$. $C_i^1[j]$ denotes the j_{th} counter in the i_{th} array in layer L_1. Let E_t, E_t^1 and E_t^2 be the sets of distinct items arriving in L_1, L_2 and L_3 within the time window T.

The set P_i contains the time windows when e_i occurs, and $\bar{P}_i = \{1, 2, \ldots, T\} - P_i$. Let

$$I_{i,j,t} = \begin{cases} 1, & \text{if } \exists e_k \in E_t, i \neq k \bigvee h_j(e_i) = h_j(e_k) \\ & \quad \bigvee g_j(e_i) = g_j(e_k), \\ 0, & \quad \text{Otherwise.} \end{cases}$$

$$\begin{aligned} E[\Delta_j p_i] &= E[\Delta_j^1 p_i] + E[\Delta_j^2 p_i] \\ &= \sum_{t\in\bar{P}_i} \left[1 - \left(1 - \frac{1}{l_1}\right)^{|E_t|}\right] \times \left[1 - \left(1 - \frac{1}{l_1}\right)^{|E_t^1|}\right], \end{aligned}$$

where $|E_t^1|$ represents the number of items for which $\Delta_1 < \hat{p}_i \leq \Delta_1 + \Delta_2$. We denote:

$$\|p\|_1 = \sum_{i=1}^{N} p_i = \sum_{t=1}^{T} |E_t| = \|E\|_1,$$

and define $\|p\|_1^1 = \sum_{p_i > \Delta_1} p_i$ as the sum of items whose estimated persistence exceeds Δ_1. Therefore, we have

$$E\left[\Delta_j p_i\right] \leq \sum_{t\in\bar{p}_i} \frac{|E_t|}{n} \times \frac{|E_t^1|}{m} \leq \frac{\|p\|_1 \times \|p\|_1^1}{n\times m}.$$

From the increment of the limit function, it can be concluded that

$$E\left[\Delta_j p_i\right] \geq \sum_{t\in\bar{p}_i} \left[1 - \left(\frac{1}{e}\right)^{\frac{|E_t|}{n}}\right] \times \left[1 - \left(\frac{1}{e}\right)^{\frac{|E_t^1|}{m}}\right].$$

Memory ratios can be adjusted by varying n and m so that $\Delta_j p_i$ decreases. We have

$$\mathbb{P}(\hat{p}_i \leq p_i + \varepsilon \|p\|_1 \times \|p\|_1^1) \geq 1 - \mathbb{P}(\forall \Delta_j p_i > e \times \mathbb{E}[\Delta_j p_i]).$$

By Markov's inequality, it follows that

$$1 - \mathbb{P}(\forall \Delta_j p_i > e \times \mathbb{E}[\Delta_j p_i]) \geq 1 - e^{-d_1 - d_2} \geq 1 - \delta.$$

For large data streams,

$$E\left[\Delta_j p_i\right] \leq \sum_{t \in \hat{p}_i} \frac{|E_t|}{n} \times \frac{|E_t^1|}{m} \times \frac{|E_t^2|}{L} \leq \frac{\|p\|_1 \times \|p\|_1^1 \times \|p\|_1^2}{n \times m \times L},$$

where $\|p\|_1^2$ is the sum of items filtered by L_1 and L_2. Let $\varepsilon_1 = \frac{e}{n \times m \times L}$ and $\delta_1 = e^{-d_1 - d_2 - d_3}$, and the derivation is similar to the above.

$$\mathbb{P}(\hat{p}_i \leq p_i + \varepsilon_1 \|p\|_1 \times \|p\|_1^1 \times \|p\|_1^2) \geq 1 - e^{-d_1 - d_2 - d_3} \geq 1 - \delta_1.$$

From the proof, it is evident that we can obtain the estimated persistence in a finite number of iterations and calculations. The time complexity of our estimation method is $O(\ln(\frac{1}{\delta}))$, while the space complexity is $O\left(\frac{1}{\varepsilon}\ln(\frac{1}{\delta})\right)$. □

A.2.2 Comparison with Related Work.

Theorem A.4. *Let $\hat{p}_i^{OO}$ be the estimated persistence of the On-Off Sketch. Under the same memory conditions, we filter the entire data stream by assigning a different number of counters and d_i for small, medium, and large data streams. For simplicity, we use the same names for the hash functions in the On-Off Sketch as in our method, even though they have different numbers of counters. Let $\Delta_j^{OO} p_i = C_j[h_j(e_i)] - p_i$, where $p_i = \min_{1 \leq j \leq d}\left(C_j[h_j(e_i)]\right)$, and d represents the number of hash functions in the On-Off Sketch. There is:*

$$E\left(\Delta_j^{OO} p_i\right) > E\left(\Delta_j p_i\right).$$

Proof. For small data streams, $E[\Delta_j p_i]$ can be represented by the following formula in both our method and the On-Off Sketch:

$$E[\Delta_j p_i] = \sum_{t \in \bar{P}_i}\left[1 - \left(1 - \frac{1}{n}\right)^{|E_t|}\right],$$

Since we utilize low-byte storage, there is more counters, allowing us to conclude that $E\left(\Delta_j p_i\right) \leq E\left(\Delta_j^{OO} p_i\right)$. For medium data streams, we have

$$1 - \left(1 - \frac{1}{m}\right)^{|E_t^1|} < 1.$$

Therefore, it follows that

$$E\left(\Delta_j^{OO} p_i\right) > 1 - \left(1 - \frac{1}{n}\right)^{|E_t|} > E\left(\Delta_j p_i\right).$$

For large data streams, a similar argument holds. In summary, the above inequalities indicate that our method is superior to the On-Off Sketch. □

Theorem A.5. *Let $\hat{p}_i = B[h_1(e_i)][e_i]$ and denote the On-Off Sketch estimate as $\hat{p}_i^{OO}$, where $B[i]$ is the ith bucket. We have the following inequality:*

$$p_i \leq \hat{p}_i \leq \hat{p}_i^{OO} \leq T. \tag{5}$$

Proof. The On-Off method employs an alternative approach to record persistence. Our method enhances this by using a cold-item filter, which reduces the probability of hash collisions caused by other items when the estimator is recorded. We assume that, in the initial state, $\hat{p}_i = \hat{p}_i^{OO}$. When a new item arrives, several scenarios may occur:

Case 1: When e_i arrives and is present in $B[h_1(e_i)][e_i]$, whether it results in an insertion or a collision, if the number of counters remains the same, then from a probabilistic perspective, (5) is still valid. Under normal circumstances, when e_i arrives, we can consider that (5) holds. If e_i is not found in $B[h_1(e_i)][e_i]$, the occurrence of insertion or hash collision does not affect the validity of (5).

Case 2: When e_j (where $i \neq j$) arrives and $C_1[h_1(e_i)] = C_1[h_1(e_j)]$, this may lead to errors due to collisions. However, our method filters out many cold items, resulting in fewer items reaching the entry point. Thus, we have $|e_j| < |e_j^{OO}|$, where $|e_j|$ and $|e_j^{OO}|$ represent the number of e_j in our method and the On-Off Sketch, respectively. This results in a smaller value of $\hat{p}$. If no collision occurs, there is no change in $\hat{p}$. □

Therefore, our method outperforms the On-Off Sketch in finding persistent items.

A.3 Skewness-Aware Error Bound

Theorem A.6. ***Skewness-Aware Error Bound*** *Assume the persistence of items follows a Zipf distribution with parameter s, i.e., the persistence of the i-th most frequent item is:*

$$p_i = \frac{1}{i^s H_N^{(s)}}, \quad \text{where } H_N^{(s)} = \sum_{k=1}^{N} \frac{1}{k^s}.$$

The expected error upper bound of the Hypersistent Sketch satisfies:

$$\mathbb{E}[\hat{p}_i - p_i] \leq \underbrace{\frac{H_N^{(s)}}{n}}_{\text{Cold-item error}} + \underbrace{\frac{H_N^{(s-1)}}{m}}_{\text{Medium-hot item error}},$$

where n and m are the number of counters in L_1 and L_2 layers of the Cold Filter, respectively.

Proof. **Stage 1: Cold Items ($p_i \leq \Delta_1$)**

Collision Probability: For items processed in L_1, the hash collision probability is approximated via Poisson distribution:

$$\mathbb{P}_{\text{coll}}^{(1)} = 1 - \left(1 - \frac{1}{n}\right)^{H_N^{(s)}} \approx \frac{H_N^{(s)}}{n}\left(1 - \frac{H_N^{(s)}}{2n}\right).$$

Expected Error: Summing over all cold items:

$$\mathbb{E}[\epsilon_{\text{cold}}] = \sum_{i=1}^{N} p_i \cdot \mathbb{P}_{\text{coll}}^{(1)} = \frac{1}{H_N^{(s)}} \cdot \frac{H_N^{(s)}}{n} \sum_{i=1}^{N} \frac{1}{i^s} = \frac{H_N^{(s)}}{n}.$$

Stage 2: Medium-hot Items ($\Delta_1 < p_i \leq \Delta_1 + \Delta_2$)

Adjusted Distribution: After filtering by L_1, the remaining items follow a truncated Zipf distribution with parameter $s - 1$:

$$p_i' \propto \frac{1}{i^{s-1}}, \quad H_N^{(s-1)} = \sum_{k=1}^{N} \frac{1}{k^{s-1}}.$$

Collision Probability: Collision probability in L_2 becomes:

$$\mathbb{P}_{\text{coll}}^{(2)} \approx \frac{H_N^{(s-1)}}{m}.$$

Expected Error:

$$\mathbb{E}[\epsilon_{\text{mid}}] = \sum_{i=1}^{N} p_i' \cdot \mathbb{P}_{\text{coll}}^{(2)} = \frac{H_N^{(s-1)}}{m}.$$

Stage 3: Extreme-hot Items ($p_i > \Delta_1 + \Delta_2$)
Full ID storage in the Hot Part eliminates hash collisions. Replacement errors decay as:

$$\mathbb{P}_{\text{replace}} \sim \frac{1}{\Delta_2 + 1} \to 0 \quad (s \to \infty).$$

Total Error Bound: Combining all stages:

$$\mathbb{E}[\hat{p}_i - p_i] \leq \frac{H_N^{(s)}}{n} + \frac{H_N^{(s-1)}}{m}.$$

Skewness Sensitivity Analysis The error bound varies with s as follows:
Low Skewness ($s \to 0$):

$$\epsilon(s) \approx \frac{N}{n} + \frac{N^2}{m} \quad \text{(matches uniform distribution)},$$

moderate Skewness ($1 < s < 2$):

$$\epsilon(s) \approx \frac{\zeta(s)}{n} + \frac{N^{2-s}}{m(2-s)},$$

high Skewness ($s \geq 2$):

$$\epsilon(s) \approx \frac{\zeta(s)}{n} + \frac{\zeta(s-1)}{m}.$$

For $s_1 > s_2 \geq 0$, the improvement ratio satisfies:

$$\frac{\epsilon(s_2)}{\epsilon(s_1)} \sim N^{s_1 - s_2}.$$

□

In conclusion, the above proof indicates that our method has good adaptability to data with different skewness. **When the skewness is large, our method performs better; when the skewness is not large, the data distribution is relatively uniform, and our method is not inferior to the comparison method.**

A.4 Threshold Sensitivity Analysis

Theorem IV.7: Threshold Sensitivity and Pareto Optimality Let the Cold Filter thresholds be parameterized as:

$$\Delta_1 = k_1 \cdot \frac{\log n}{\log \log n}, \quad \Delta_2 = k_2 \cdot \Delta_1 = k_1 k_2 \cdot \frac{\log n}{\log \log n}.$$

where k_1, k_2 are tunable constants. The memory-error trade-off satisfies:

$$\text{Memory Efficiency} \propto \frac{1}{k_1 k_2}.$$

$$\text{Relative Error} \propto \frac{\sqrt{k_1}}{n^{1/2}} + \frac{\sqrt[3]{k_2}}{m^{1/3}}.$$

Pareto optimality is achieved when:

$$k_1 = \Theta\left(\sqrt{\frac{n}{\log n}}\right), \quad k_2 = \Theta\left(\sqrt[3]{\frac{m}{\log m}}\right).$$

Proof. The Cold Filter memory consumption consists of two layers:

$$M_{\text{cold}} = n \cdot \lceil \log_2 \Delta_1 \rceil + m \cdot \lceil \log_2 \Delta_2 \rceil.$$

Substituting the threshold parameterization:

$$\begin{aligned} M_{\text{cold}} &\approx n \log(k_1 \log n) + m \log(k_1 k_2 \log n) \\ &= n(\log k_1 + \log \log n) + m(\log k_1 + \log k_2 + \log \log n) \\ &\approx n \log k_1 + m(\log k_1 + \log k_2), \end{aligned}$$

under fixed total memory $M_{\text{total}} = M_{\text{cold}} + M_{\text{hot}}$:

$$k_1 k_2 \propto \frac{1}{M_{\text{cold}}}.$$

We analyze the relationship between the error and the thresholds, from Theorem IV.6, the error bound can be expressed as:

$$\epsilon \propto \frac{H_N^{(s)}}{n} + \frac{H_N^{(s-1)}}{m}.$$

For general distributions, using moment bounds:

$$\begin{aligned} H_N^{(s)} &\propto N^{1-s} \\ H_N^{(s-1)} &\propto N^{2-s}, \end{aligned}$$

substituting the threshold relationships:

$$\begin{aligned} \epsilon &\propto \frac{N^{1-s}}{n} + \frac{N^{2-s}}{m} \\ &= \frac{N^{1-s}}{\Delta_1^{1/2}} + \frac{N^{2-s}}{\Delta_2^{1/3}} \\ &= \frac{\sqrt{k_1}}{n^{1/2}} + \frac{\sqrt[3]{k_2}}{m^{1/3}}. \end{aligned}$$

Pareto Optimality Condition, Define the optimization problem:

$$\min_{k_1,k_2} \left(\frac{\sqrt{k_1}}{n^{1/2}} + \frac{\sqrt[3]{k_2}}{m^{1/3}}\right) \quad \text{s.t.} \quad k_1 k_2 = C.$$

Using Lagrange multipliers with $\mathcal{L} = \frac{\sqrt{k_1}}{n^{1/2}} + \frac{\sqrt[3]{k_2}}{m^{1/3}} + \lambda(k_1 k_2 - C)$:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial k_1} &= \frac{1}{2n^{1/2}k_1^{1/2}} + \lambda k_2 = 0 \\ \frac{\partial \mathcal{L}}{\partial k_2} &= \frac{1}{3m^{1/3}k_2^{2/3}} + \lambda k_1 = 0, \end{aligned}$$

dividing the two equations:

$$\frac{3m^{1/3}k_2^{2/3}}{2n^{1/2}k_1^{1/2}} = \frac{k_2}{k_1},$$

solving yields the optimal scaling:

$$k_1 \propto \sqrt{\frac{n}{\log n}}, \quad k_2 \propto \sqrt[3]{\frac{m}{\log m}}.$$

We discuss Pareto optimality of threshold parameters, for real-world deployment with $n = m$, the optimal thresholds satisfy:

$$\frac{\Delta_2}{\Delta_1} = \Theta\left((\log n)^{1/6}\right).$$

□

This ratio automatically adapts to data scale while maintaining near-optimal performance. The above proof shows that **our method behaves differently for different thresholds, and gives the memory allocation and the relationship between error and threshold parameters. In addition, the Pareto optimality of the theoretical threshold is given in the end, which can guide us to set the corresponding threshold parameters in the experiment.**