C-MinHash: Improving Minwise Hashing with Circulant Permutation

Xiaoyun Li, Ping Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{lixiaoyun996, pingli98}@gmail.com

Abstract

Minwise hashing (MinHash) is an important and practical algorithm for generating random hashes to approximate the Jaccard (resemblance) similarity in massive binary (0/1) data. The basic theory of MinHash requires applying hundreds or even thousands of independent random permutations to each data vector in the dataset, in order to obtain reliable results for (e.g.,) building large-scale learning models or approximate near neighbor search. In this paper, we propose **Circulant MinHash (C-MinHash)** and provide the surprising theoretical results that using only **two** independent random permutations in a circulant manner leads to uniformly smaller Jaccard estimation variance than that of the classical MinHash with K independent permutations. Experiments are conducted to show the effectiveness of the proposed method. We also propose a more convenient C-MinHash variant which reduces two permutations to just **one**, with extensive numerical results to validate that it achieves essentially the same estimation accuracy as using two permutations.

1. Introduction

Given two D-dimensional binary vectors $\boldsymbol{v}, \boldsymbol{w} \in \{0,1\}^D$, the Jaccard similarity is defined as

$$J(\boldsymbol{v}, \boldsymbol{w}) = \frac{\sum_{i=1}^{D} \mathbb{1}\{\boldsymbol{v}_i = \boldsymbol{w}_i = 1\}}{\sum_{i=1}^{D} \mathbb{1}\{\boldsymbol{v}_i + \boldsymbol{w}_i \geq 1\}}, \quad (1)$$

which is a commonly used similarity metric in machine learning and web search applications. The vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ can also be viewed as two sets of items (which represent the locations of non-zero entries), where the Jaccard similarity can be equivalently viewed as the size of set intersection over the size of set union. In large-scale search and learning, directly calculating the pairwise Jaccard similarity among the sample points becomes too costly as the sample size grows. The well-known method of *"minwise hashing"* (MinHash) (Broder, 1997; Broder et al., 1997; 1998; Li and Church, 2005; Li and König, 2011) is a standard technique for computing/estimating the Jaccard similarity in massive binary datasets, with numerous applications in near neighbor search, duplicate detection, malware detection, clustering, large-scale learning, social networks, computer vision, etc. (Charikar, 2002; Fetterly et al., 2003; Henzinger, 2006; Das et al., 2007; Buehrer and Chellapilla, 2008; Gamon et al., 2008; Bendersky and Croft, 2009; Chierichetti et al., 2009; Najork et al., 2009; Pandey et al., 2009; Lee et al., 2010; Li et al., 2011; Deng et al., 2012; Chum and Matas, 2012; Li et al., 2012; Shrivastava and Li, 2012; He et al., 2013; Tamersoy et al., 2014; Shrivastava and Li, 2014; Zamora et al., 2016; Ondov et al., 2016; Zhu et al., 2017; Nargesian et al., 2018; Wang et al., 2019; Lemiesz, 2021; Tseng et al., 2021; Feng and Deng, 2021; Jia et al., 2021).

1.1. A Review of Minwise Hashing (MinHash)

Algorithm 1 Minwise-hashing (MinHash)

Input: Binary data vector $\boldsymbol{v} \in \{0,1\}^D$;
K independent permutations $\pi_1, ..., \pi_K$: $[D] \to [D]$

Output: K hash values $h_1(\boldsymbol{v}), ..., h_K(\boldsymbol{v})$

For $k = 1$ to K

$h_k(\boldsymbol{v}) \leftarrow \min_{i: v_i \neq 0} \pi_k(i)$

End For

We first recap the method of minwise hashing. For simplicity, Algorithm 1 considers just one vector $\boldsymbol{v} \in \{0,1\}^D$. In order to generate K hash values for $\boldsymbol{v}$, we assume K independent permutations: $\pi_1, ..., \pi_K : [D] \mapsto [D]$. For each permutation, the hash value is the first non-zero location in the permuted vector, i.e., $h_k(\boldsymbol{v}) = \min_{i: v_i \neq 0} \pi_k(i)$, $\forall k = 1, ..., K$. Similarly, for another binary vector $\boldsymbol{w} \in \{0,1\}^D$,

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

using the same K permutations, we can also obtain K hash values, $h_k(\boldsymbol{w})$. The estimator of $J(\boldsymbol{v}, \boldsymbol{w})$ is simply

$$\hat{J}_{MH}(\boldsymbol{v}, \boldsymbol{w}) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\{h_k(\boldsymbol{v}) = h_k(\boldsymbol{w})\}, \quad (2)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. By fundamental probability and the independence among the permutations, it is easy to show that

$$\mathbb{E}[\hat{J}_{MH}] = J, \qquad Var[\hat{J}_{MH}] = \frac{J(1-J)}{K}. \quad (3)$$

How large is K? The answer depends on the application domains. For example, for training large-scale machine learning models, it appears that $K = 512$ or $K = 1024$ might be sufficient (Li et al., 2011). However, for approximate near neighbor search using many hash tables (Indyk and Motwani, 1998), it is likely that K might have to be much larger than 1024 (Shrivastava and Li, 2012; 2014).

In the early work of MinHash (Broder, 1997; Broder et al., 1997), actually only one permutation was used by storing the first K non-zero locations after the permutation. Later, Li and Church (2005) proposed better estimators to improve the estimation accuracy. The major drawback of the original scheme was that the hashed values did not form a metric space (e.g., satisfying the triangle inequality) and hence could not be used in many algorithms/applications. We believe this was the main reason why the original authors moved to using K permutations (Broder et al., 1998).

1.2. Hashing for Non-binary Data

We believe the idea of using randomness circulantly, as studied in our paper, might be helpful in broader applications. For example, minwise hashing can also be extended to the non-binary data. For two non-negative data vectors $\boldsymbol{v}, w \in \mathbb{R}_+^D$, the weighted Jaccard similarity is defined as

$$\frac{\sum_{i=1}^{D} \min(\boldsymbol{v}_i, \boldsymbol{w}_i)}{\sum_{i=1}^{D} \max(\boldsymbol{v}_i, \boldsymbol{w}_i)}, \quad (4)$$

which obviously becomes Eq. (1) in binary data. Consistent weighted sampling (CWS) (Manasse et al., 2010; Ioffe, 2010) is the standard hashing method for the weighted Jaccard in massive data. In general, CWS can be applied to the scenarios where MinHash is found useful, and in many cases CWS might be more feasible as real-valued data typically contains more information than binary. As an algorithm, CWS is considerably much more complex than MinHash and essentially reduces to MinHash in binary data. Recently, Li et al. (2021) developed a family of new algorithms for hashing weighted Jaccard based on extremal processes. Li and Zhang (2017) generalized (4) to datasets with negative entries and Li and Zhao (2022) reported their efforts on using CWS and variants for training deep neural networks.

1.3. Outline of Main Results

From K Permutations to two. Using K independent permutations in MinHash has been widely used as the standard approach in textbooks and industry for over two decades. The main idea of this work, is to replace the independent permutations in MinHash with "circulant" permutations. Thus, we name the proposed framework **C-MinHash** (circulant MinHash). The "circulant" trick was used in the literature of random projections. For example, Yu et al. (2017) showed that using circulant projections hurts the estimation accuracy, but not by too much when the data are sparse. In Section 3, we present some (perhaps surprising) theoretical findings that we just need 2 permutations in MinHash and the results (estimation variances) are even more accurate. Basically, with the **initial permutation** (denoted by σ), we randomly shuffle the data to break whatever structure which might exist in the original data, and then the **second permutation** (denoted by π) is applied and re-used K times to generate K hash values, via circulation. This method is called C-MinHash-(σ, π). Before that, in Section 2, we analyze a simpler variant C-MinHash-$(0, \pi)$ without initial permutation σ. Although it is not our recommended method, our analysis for C-MinHash-$(0, \pi)$ provides the necessary preparation for later methods and the intuition to understand the need for the initial permutation.

From two permutations to one. Section 5 provides a convenient variant C-MinHash-(π, π) that only needs one permutation π for both pre-processing and hashing. The resultant estimator is no longer unbiased but the bias is extremely small and has essentially no impact on the estimation accuracy, as verified by extensive numerical experiments.

2. C-MinHash-$(0, \pi)$ Without Initial Permutation

Algorithm 2 C-MinHash-$(0, \pi)$

Input: Binary data vector $\boldsymbol{v} \in \{0, 1\}^D$;
Permutation vector π: $[D] \to [D]$

Output: Hash values $h_1(\boldsymbol{v}), ..., h_K(\boldsymbol{v})$

For $k = 1$ to K

Shift π circulantly rightwards by k units: $\pi_k = \pi_{\to k}$

$h_k(\boldsymbol{v}) \leftarrow \min_{i: v_i \neq 0} \pi_{\to k}(i)$

End For

As shown in Algorithm 2, the C-MinHash-$(0, \pi)$ algorithm has similar operations as MinHash. The difference lies in the permutations used in the hashing process. To generate each hash $h_k(\boldsymbol{v})$, we permute the data vector using $\pi_{\to k}$, which is the permutation shifted k units circulantly towards right based on π. For example, $\pi = [3, 1, 2, 4], \pi_{\to 1} =$

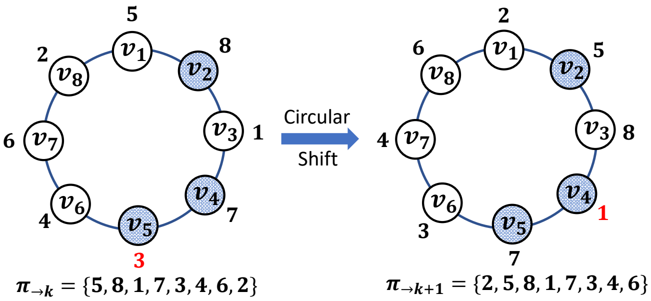

Figure 1. An illustration of the idea of C-MinHash. The data vector has three non-zeros, $v_2 = v_4 = v_5 = 1$. In this example, we get hash values $h_k(\boldsymbol{v}) = 3$, $h_{k+1}(\boldsymbol{v}) = 1$.

$[4, 3, 1, 2], \pi_{\to 2} = [2, 4, 3, 1]$, etc. Conceptually, we may think of circulation as concatenating the first and last elements of a vector to form a circle; see Figure 1 for an illustration. We set the hash value $h_k(\boldsymbol{v})$ as the position of the first non-zero after being permuted by $\pi_{\to k}$. Analogously, we define the C-MinHash-$(0, \pi)$ estimator of the Jaccard similarity $J(\boldsymbol{v}, \boldsymbol{w})$ as

$$\hat{J}_{0,\pi} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\{h_k(\boldsymbol{v}) = h_k(\boldsymbol{w})\}, \tag{5}$$

where h is the hash value output by Algorithm 2. In this paper, for simplicity, we assume $K \leq D$.

Next, we present the theoretical analysis for Algorithm 2, in terms of the expectation (mean) and the variance of the estimator $\hat{J}_{0,\pi}$. Our results reveal that the estimation accuracy depends on the initial data distribution, which may lead to undesirable performance behaviors when real-world datasets exhibit various structures. On the other hand, while it is not our recommended method, the analysis serves a preparation (and insight) for the C-MinHash-(σ, π) which will soon be described.

First, we introduce some notations and definitions. Given $\boldsymbol{v}, \boldsymbol{w} \in \{0, 1\}^D$, we define a and f as

$$a = \sum_{i=1}^{D} \mathbb{1}\{\boldsymbol{v}_i = w_i = 1\}, f = \sum_{i=1}^{D} \mathbb{1}\{\boldsymbol{v}_i + w_i \geq 1\}. \tag{6}$$

We say that $(\boldsymbol{v}, \boldsymbol{w})$ is a (D, f, a)-*data pair*, whose Jaccard similarity can be written as $J = a/f$.

Definition 2.1. Consider $\boldsymbol{v}, \boldsymbol{w} \in \{0, 1\}^D$. Define the **location vector** for $\boldsymbol{v}, \boldsymbol{w}$ as $\boldsymbol{x} \in \{O, \times, -\}^D$, with $\boldsymbol{x}_i$ being "O", "$\times$", "$-$", when $\boldsymbol{v}_i = \boldsymbol{w}_i = 1$, $\boldsymbol{v}_i + \boldsymbol{w}_i = 1$ and $\boldsymbol{v}_i = \boldsymbol{w}_i = 0$, respectively.

The location vector $\boldsymbol{x}$ can fully characterize a hash collision. When a permutation $\pi_{\to k}$ is applied, the hashes $h_k(\boldsymbol{v})$ and $h_k(\boldsymbol{w})$ would collide if after permutation, the first "O" is placed before the first "$\times$" (counting from small to large); the location of "$-$" entries would not affect the collision. This observation will be the key in our theoretical analysis.

Definition 2.2. For $A, B \in \{O, \times, -\}$, let $\{(A, B)|\triangle\}$ denote the set $\{(i, j) : (\boldsymbol{x}_i, \boldsymbol{x}_j) = (A, B), j - i = \triangle\}$. For each $1 \leq \triangle \leq K - 1$, define

$$\mathcal{L}_0(\triangle) = \{(O, O)|\triangle\},\ \mathcal{L}_1(\triangle) = \{(O, \times)\},\ \mathcal{L}_2(\triangle) = \{(O, -)\},$$
$$\mathcal{G}_0(\triangle) = \{(-, O)|\triangle\},\ \mathcal{G}_1(\triangle) = \{(-, \times)\}\ ,\mathcal{G}_2(\triangle) = \{(-, -)\},$$
$$\mathcal{H}_0(\triangle) = \{(\times, O)|\triangle\},\ \mathcal{H}_1(\triangle) = \{(\times, \times)\},\ \mathcal{H}_2(\triangle) = \{(\times, -)\}.$$

Remark 2.3. For the ease of notation, by circulation we write $\boldsymbol{x}_j = \boldsymbol{x}_{j-D}$ when $D < j < 2D$.

Definition 2.2 measures the relative location of different types of points in the location vector, for a specific pair of data vectors. Moreover, one can easily verify that for $\forall 1 \leq \triangle \leq K - 1$,

$$\begin{aligned} &|\mathcal{L}_0| + |\mathcal{L}_1| + |\mathcal{L}_2| = |\mathcal{L}_0| + |\mathcal{G}_0| + |\mathcal{H}_0| = a, \\ &|\mathcal{G}_0| + |\mathcal{G}_1| + |\mathcal{G}_2| = |\mathcal{L}_2| + |\mathcal{G}_2| + |\mathcal{H}_2| = D - f, \\ &|\mathcal{H}_0| + |\mathcal{H}_1| + |\mathcal{H}_2| = |\mathcal{L}_1| + |\mathcal{G}_1| + |\mathcal{H}_1| = f - a, \end{aligned} \tag{7}$$

which is the intrinsic constraints on the size of above sets. We are now ready to analyze the expectation and variance of $\hat{J}_{0,\pi}$. It is easy to see that $\hat{J}_{0,\pi}$ is still unbiased, i.e., $\mathbb{E}[\hat{J}_{0,\pi}] = J$, by linearity of expectation. Lemma 2.4 provides an important quantity that leads to $Var[\hat{J}_{0,\pi}]$ which is given in Theorem 2.5. All the missing proofs in the paper are placed in Appendix A.

Lemma 2.4. *For any $1 \leq s < t \leq K$ with $t - s = \triangle$, we have that*

$$\begin{aligned} &\mathbb{E}_\pi\big[\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}\mathbb{1}\{h_t(\boldsymbol{v}) = h_t(\boldsymbol{w})\}\big] \\ &\qquad = \frac{|\mathcal{L}_0(\triangle)| + (|\mathcal{G}_0(\triangle)| + |\mathcal{L}_2(\triangle)|)J}{f + |\mathcal{G}_0(\triangle)| + |\mathcal{G}_1(\triangle)|}, \end{aligned}$$

where the sets are defined in Definition 2.2 and h_s, h_t are the hash values output by Algorithm 2.

Theorem 2.5. *For C-MinHash-$(0, \pi)$, the variance of $\hat{J}_{0,\pi}$ is given by*

$$Var[\hat{J}_{0,\pi}] = \frac{J}{K} + \frac{2\sum_{s=2}^{K}(s-1)\Theta_{K-s+1}}{K^2} - J^2,$$

where $\Theta_\triangle \triangleq E_\pi\big[\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}\mathbb{1}\{h_t(\boldsymbol{v}) = h_t(\boldsymbol{w})\}\big]$ as in Lemma 2.4 with any $t - s = \triangle$.

Proof. We use $\mathbb{1}_s$ to denote $\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}, \forall 1 \leq s \leq K$. By the expansion of variance formula, since $\mathbb{E}[\mathbb{1}_s^2] = \mathbb{E}[\mathbb{1}_s] = J$, we have

$$Var[\hat{J}_{0,\pi}] = \frac{J}{K} + \frac{\sum_{s=1}^{K}\sum_{t \neq s}^{K}\mathbb{E}[\mathbb{1}_s\mathbb{1}_t]}{K^2} - J^2.$$

Note here that for $\forall t > s$, the t-th hash sample uses π_t as the permutation, which is shifted rightwards by $\triangle = t - s$ from π_s. Thus, we have $\mathbb{E}[\mathbb{1}_s \mathbb{1}_t] = \mathbb{E}[\mathbb{1}_{s-i}\mathbb{1}_{t-i}]$ for $\forall 0 < i < s \wedge t$, which implies $\mathbb{E}[\mathbb{1}_s\mathbb{1}_t] = \mathbb{E}[\mathbb{1}_1\mathbb{1}_{t-s+1}]$, $\forall s < t$. Since by assumption $K \le D$, we have

$$\begin{aligned}
&\sum_{s}^{K}\sum_{t\neq s}^{K}\mathbb{E}[\mathbb{1}_s\mathbb{1}_t]\\
&= 2\mathbb{E}\big[(\mathbb{1}_1\mathbb{1}_2 + \mathbb{1}_1\mathbb{1}_3 + ... + \mathbb{1}_1\mathbb{1}_K)\\
&\qquad + (\mathbb{1}_2\mathbb{1}_3 + ... + \mathbb{1}_2\mathbb{1}_K) + ... + \mathbb{1}_{K-1}\mathbb{1}_K\big]\\
&= 2\sum_{s=2}^{K}(s-1)\mathbb{E}[\mathbb{1}_1\mathbb{1}_{K-s+2}] \triangleq 2\sum_{s=2}^{K}(s-1)\Theta_{K-s+1}.
\end{aligned}$$

The result then follows. □

From Theorem 2.5, we see that the variance of $\hat{J}_{0,\pi}$ depends on a, f, and the sizes of sets $\mathcal{L}$'s and $\mathcal{G}$'s as in Definition 2.1, which are determined by the location vector $\boldsymbol{x}$. Since we use the original data vectors without randomly permuting the entries beforehand, $Var[\hat{J}_{0,\pi}]$ is called "*location-dependent*" as it is dependent on the location of non-zero entries of the original data. Consequently, as will also be shown in our numerical study, $Var[\hat{J}_{0,\pi}]$ may be either smaller or larger than that of MinHash estimate $\hat{J}_{MH}$ up to different structure of the data vectors.

3. C-MinHash-(σ, π) with Independent Initial Permutation

Algorithm 3 C-MinHash-(σ, π)

Input: Binary data vector $\boldsymbol{v} \in \{0,1\}^D$;
Permutation vectors π and σ: $[D] \to [D]$

Output: Hash values $h_1(\boldsymbol{v}), ..., h_K(\boldsymbol{v})$

Initial permutation: $\boldsymbol{v}' = \sigma(\boldsymbol{v})$

For $k = 1$ to K

Shift π circulantly rightwards by k units: $\pi_k = \pi_{\to k}$

$h_k(\boldsymbol{v}) \leftarrow \min_{i:v'_i \neq 0} \pi_{\to k}(i)$

End For

Next, we present an improved algorithm by eliminating the "location-dependennce" of C-MinHash-$(0, \pi)$ as analyzed above. The method C-MinHash-(σ, π) is summarized in Algorithm 3, which is very similar to Algorithm 2. This time, as pre-processing, we apply an initial permutation $\sigma \perp\!\!\!\perp \pi$ on the data to break whatever structures which might exist. Analogously, we define the C-MinHash-(σ, π) estimator as

$$\hat{J}_{\sigma,\pi} = \frac{1}{K}\sum_{k=1}^{K}\mathbb{1}\{h_k(\boldsymbol{v}) = h_k(\boldsymbol{w})\}, \tag{8}$$

where h_k's are the hash values output by Algorithm 3. In the remaining part of this section, we will present our detailed theoretical analysis and the main result (Theorem 3.4). First, by linearity of expectation and the fact that σ and π are independent, it is easy to verify that $\hat{J}_{\sigma,\pi}$ is still an unbiased estimator of J. Based on Theorem 2.5, in the following we provide the exact variance formula of $\hat{J}_{\sigma,\pi}$.

Theorem 3.1. *Let a, f be defined as in (6). When $0 < a < f \le D$ ($J \notin \{0,1\}$), we have*

$$Var[\hat{J}_{\sigma,\pi}] = \frac{J}{K} + \frac{(K-1)\tilde{\mathcal{E}}}{K} - J^2, \tag{9}$$

where $l = \max(0, D - 2f + a)$, and

$$\tilde{\mathcal{E}} = \sum_{\Xi}\left(\frac{l_0}{f+g_0+g_1} + \frac{a(g_0+l_2)}{(f+g_0+g_1)f}\right)\left(\sum_{s=l}^{D-f-1}\frac{\binom{D-f}{s}}{\binom{D-a-1}{D-f-1}}\cdot\frac{\binom{f-a-1}{D-f-s-1}\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{a-l_1-l_2}}{\binom{D-1}{a}}\right). \tag{10}$$

The feasible set $\Xi = \{l_0, l_2, g_0, g_1\}$ satisfies the intrinsic constraints (7), and

$$\begin{aligned}
&n_1 = g_0 - (D - f - s - g_1), \qquad n_2 = D - f - s - g_1,\\
&n_3 = l_2 - g_0 + (D - f - s - g_1),\\
&n_4 = l_1 - (D - f - s - g_1).
\end{aligned}$$

When $a = 0$ or $f = a$ ($J = 0$ or 1), $Var[\hat{J}_{\sigma,\pi}] = 0$.

As expected, since the original locational structure of the data is broken by the initial permutation σ, $Var[\hat{J}_{\sigma,\pi}]$ only depends on the values of (D, f, a) but not the specific set sizes as in Theorem 2.5, i.e., it is *"location-independent"*. This would make the performance of C-MinHash-(σ, π) consistent in different tasks. In the sequel, we investigate the statistical properties of $Var[\hat{J}_{\sigma,\pi}]$ in more details. Firstly, same as MinHash, Proposition 3.2 states that given D and f, the variance of $\hat{J}_{\sigma,\pi}$ is symmetric about $J = 0.5$, as illustrated in Figure 2, which also shows that the variance of $\hat{J}_{\sigma,\pi}$ is smaller than the variance of the original MinHash.

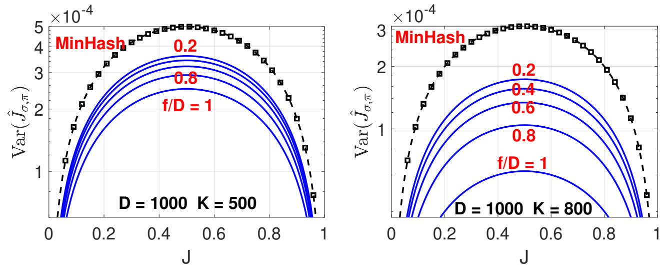

Figure 2. $Var[\hat{J}_{\sigma,\pi}]$ versus J, with $D = 1000$ and varying f. **Left:** $K = 500$. **Right:** $K = 800$.

Proposition 3.2 (Symmetry)**.** *$Var[\hat{J}_{\sigma,\pi}]$ is the same for the (D, f, a)-data pair and the $(D, f, f-a)$-data pair, $\forall 0 \le a \le f \le D$.*

A rigorous comparison of $Var[\hat{J}_{\sigma,\pi}]$ and $Var[\hat{J}_{MH}]$ appears to be a challenging task given the complicated combinatorial form of $Var[\hat{J}_{\sigma,\pi}]$. The following lemma characterizes an important property of $\tilde{\mathcal{E}}$ in (10), that it is monotone in D when a and f are fixed, as illustrated in Figure 3 (left).

Lemma 3.3 (Strict Increment)**.** *Let $f > a > 0$ and K be arbitrary and fixed. Denote $\tilde{\mathcal{E}}_D$ as in (10) in Theorem 3.1, with D is a parameter. Then, $\tilde{\mathcal{E}}_{D+1} > \tilde{\mathcal{E}}_D$ for $\forall D \ge f$.*

Equipped with Lemma 3.3, we arrive at the following main theoretical result of this work, on the uniform variance reduction of C-MinHash-(σ, π).

Theorem 3.4 (Uniform Superiority)**.** *For any two binary vectors $\boldsymbol{v}, \boldsymbol{w} \in \{0,1\}^D$ with $J \ne 0$ or 1, it holds that $Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v}, \boldsymbol{w})] < Var[\hat{J}_{MH}(\boldsymbol{v}, \boldsymbol{w})]$.*

Remark 3.5. In fact, from the proof of Lemma 3.3 and Theorem 3.4, we can show that the collision indicator variables $\mathbb{1}\{h_k(\boldsymbol{v}) = h_k(\boldsymbol{w})\}$, $k = 1, ..., K$, in (8) are *pairwise negatively correlated*. This provides intuition on the source of variance reduction.

Proof. By assumption we have $0 < a < f$. To compare $Var[\hat{J}_{\sigma,\pi}]$ with $Var[\hat{J}_{MH}] = \frac{J(1-J)}{K} = \frac{J}{K} + \frac{(K-1)J^2}{K} - J^2$, it suffices to compare $\tilde{\mathcal{E}}$ with J^2. When $D = f$, we know that the location vector $\boldsymbol{x}$ of $(\boldsymbol{v}, \boldsymbol{w})$ contains no "$-$" elements. It is easy to verify that in this case, $|\mathcal{G}_0| = |\mathcal{G}_1| = |\mathcal{L}_2| = 0$, and $|\mathcal{L}_0|$ follows hyper$(f-1, a, a-1)$. By Theorem 3.1, it follows that when $D = f$,

$$\tilde{\mathcal{E}}_D = \frac{1}{f}\mathbb{E}[|\mathcal{L}_0|] = \frac{a(a-1)}{f(f-1)} = J\tilde{J} < J^2.$$

Recall the definition $\tilde{J} = \frac{a-1}{f-1}$, which is always smaller than J. On the other hand, as $D \to \infty$, we have $|\mathcal{L}_0| \to 0$, $|\mathcal{L}_2| \to a$, $|\mathcal{G}_0| \to a$ and $|\mathcal{G}_1| \to f - a$. We can show that

$$\tilde{\mathcal{E}}_D \to J^2, \quad \text{as } D \to \infty.$$

By Lemma 3.3, the sequence $(\tilde{\mathcal{E}}_f, \tilde{\mathcal{E}}_{f+1}, \tilde{\mathcal{E}}_{f+2}, ...)$ is strictly increasing. Since it is convergent with limit J^2, by the Monotone Convergence Theorem we know that $\tilde{\mathcal{E}}_D < J^2$, $\forall D \ge f$. This completes the proof. $\square$

Theorem 3.4 says that, using merely two permutations as in C-MinHash-(σ, π) improves the Jaccard estimation variance of standard MinHash, in all cases. That said, using two permutations could be strictly better than using K permutations in minwise hashing. How does the variance of $\hat{J}_{\sigma,\pi}$ rely on a,f and K? First, interestingly, in Proposition 3.6, we show that the relative variance reduction of C-MinHash-(σ, π) over MinHash is the same for any a value for given f and K, i.e., the relative improvement is independent of the Jaccard value J at a given sparsity level.

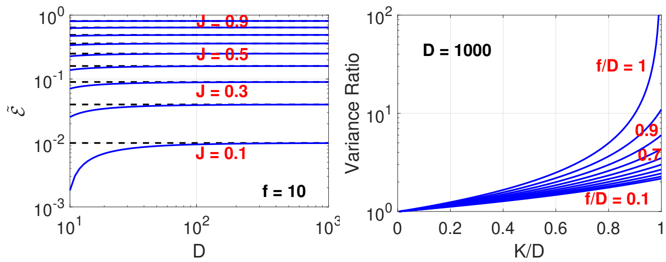

Figure 3. **Left:** Theoretical $\tilde{\mathcal{E}}$, $f = 10$ fixed. Each dash line represents the corresponding J^2. **Right:** Variance ratio $\frac{Var[\hat{J}_{MH}]}{Var[\hat{J}_{\sigma,\pi}]}$, $D = 1000$. This plot holds for all a value (by Proposition 3.6).

Proposition 3.6 (Consistent Improvement)**.** *Suppose f is fixed. In terms of a, the variance ratio $\frac{Var[\hat{J}_{MH}(\boldsymbol{v},\boldsymbol{w})]}{Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v},\boldsymbol{w})]}$ is a constant for any $0 < a < f$.*

To investigate the influence of sparsity f and number of hashes K on the variance, in Figure 3 (right), we plot the variance ratio $\frac{Var[\hat{J}_{MH}]}{Var[\hat{J}_{\sigma,\pi}]}$ with different f and K. The results in Figure 3 again verify Theorem 3.4, as the variance ratio is always greater than 1. We see that the improvement in variance increases both with K (i.e., more hashes) and f (i.e., more non-zero entries). Note that, by Proposition 3.6, here we do not need to consider a since it does not affect the variance ratio.

4. Numerical Experiments

In this section, we provide numerical experiments to validate our theoretical findings and demonstrate that C-MinHash can indeed lead to smaller Jaccard estimation errors.

4.1. Sanity Check: a Simulation Study

A simulation study is conducted on synthetic data to verify the theoretical variances given by Theorem 2.5 and Theorem 3.1. We simulate $D = 128$ dimensional binary vector pairs $(\boldsymbol{v}, \boldsymbol{w})$ with different f and a, which have a special locational structure that the location vector $\boldsymbol{x}$ is such that a "O"'s are followed by $(f-a)$ "$\times$"'s and then followed by $(D-f)$ "$-$"'s sequentially. We plot the empirical and theoretical mean square errors (MSE = variance + bias^2) in Figure 4, and we observe:

- The theoretical variance matches the empirical results, confirming Theorem 2.5 and Theorem 3.1. The variance reduction effect becomes more significant with more number of hashes K.

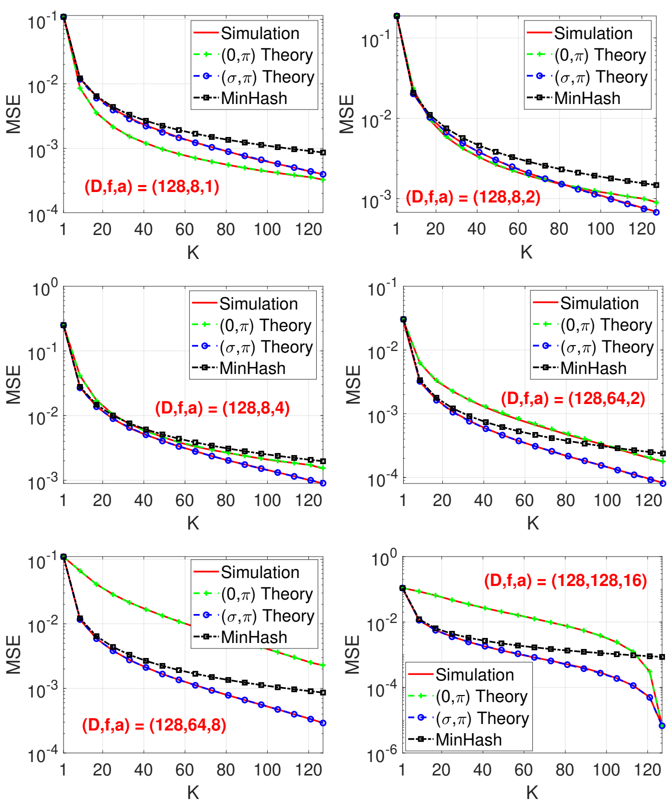

Figure 4. Empirical vs. theoretical variance of $\hat{J}_{0,\pi}$ (C-MinHash-$(0,\pi)$) and $\hat{J}_{\sigma,\pi}$ (C-MinHash-(σ,π)), on synthetic binary data vector pairs with different data statistics.

- $Var[\hat{J}_{\sigma,\pi}]$ is always smaller than $Var[\hat{J}_{MH}]$, as stated by Theorem 3.4. In contrast, $Var[\hat{J}_{0,\pi}]$ (C-MinHash-$(0,\pi)$) varies significantly depending on different data structures, as discussed in Section 2.

4.2. Jaccard Estimation on Text and Image Datasets

We test C-MinHash on four public datasets, including two text datasets: the NIPS full paper dataset from UCI repository (Dua and Graff, 2017), the BBC News dataset (Greene and Cunningham, 2006), and two popular image datasets: the MNIST dataset (LeCun et al., 1998) with hand-written digits, and the CIFAR dataset (Krizhevsky, 2009) containing natural images. All the datasets are processed to be binary. For image data, we first transform the images to gray-scale, then binarize the samples by thresholding at 0.5. For each dataset with n data vectors, there are in total $n(n-1)/2$ data vector pairs. We estimate the Jaccard similarities for all the pairs and report the mean absolute errors (MAE). All the results are averaged over 10 independent repetitions. We report the MAE in Figure 5, from which we see that:

- The MAE of C-MinHash-(σ,π) is consistently smaller than that of MinHash, demonstrating the practical merit of variance reduction (Theorem 3.4) to improve the Jaccard estimation accuracy. The improvements become more substantial with larger K, which is consistent with Figure 3 and Figure 4.

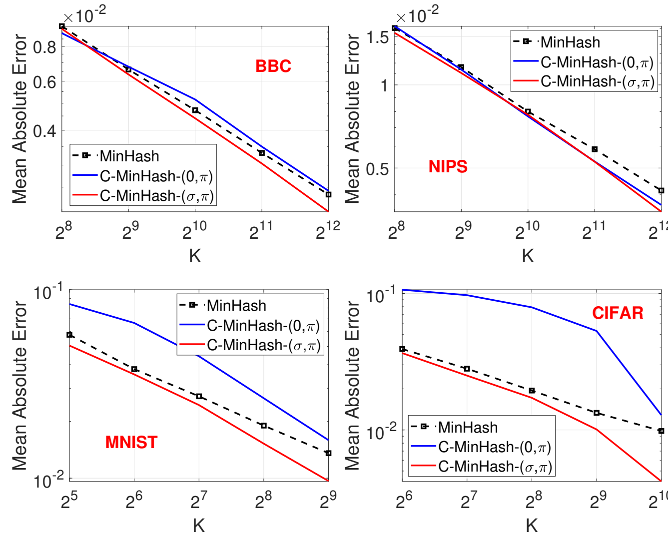

Figure 5. Mean Absolute Error (MAE) of pairwise Jaccard estimation: MinHash vs. C-MinHash on four real-world datasets.

- Without the initial permutation σ, the accuracy of C-MinHash-$(0,\pi)$ depends by the distribution/structure of the original data, and it is worse than C-MinHash-(σ,π) on all these four datasets. In addition, the performance of C-MinHash-$(0,\pi)$ on image data seems much worse than that on text data, which we believe is because the image datasets contain more structural patterns. This again suggests that the initial permutation σ might be needed in practice.

In summary, the simulation study has verified the correctness of our theoretical findings in Theorem 2.5 and Theorem 3.1. The experiments with Jaccard estimation on four real-world datasets confirm that C-MinHash is more accurate than the original MinHash, and the initial permutation σ is recommended.

5. C-MinHash-(π,π): Practically Reducing to One Permutation

In this section, we propose a more convenient variant, C-MinHash-(π,π), which only requires one permutation. That is, π is used for both pre-processing and circulant hashing. The procedure is the same as Algorithm 3, except that the initial permutation σ is replaced by π. Denote the corresponding Jaccard estimator as $\hat{J}_{\pi,\pi}$. The complicated dependency between π (for initial permutation) and $\pi_{\rightarrow k}$ (for hashing) makes the estimator no longer unbiased. Nevertheless, we found through extensive numerical experiments that, the MSE of $\hat{J}_{\pi,\pi}$ is essentially the same as $\hat{J}_{\sigma,\pi}$.

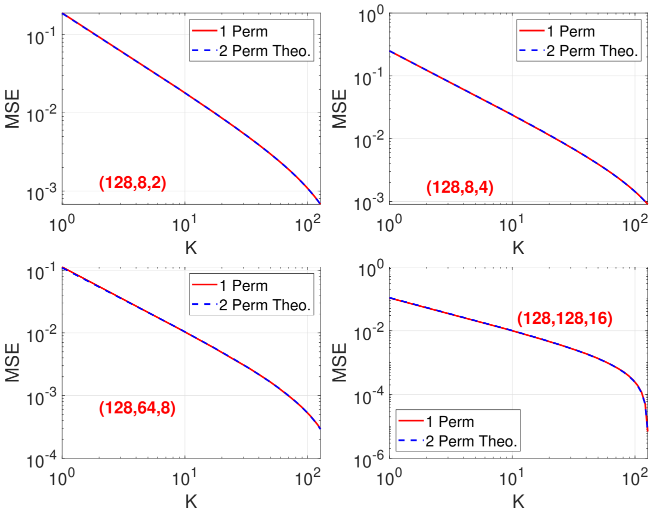

Figure 6. Estimator MSE on simulated data pairs. "1 Perm" is C-MinHash-(π, π), and "2 Perm Theo." is the theoretical variance of C-MinHash-(σ, π) (Theorem 3.1).

Figure 6 compares the empirical MSE of C-MinHash-(π, π) with the theoretical variances of C-MinHash-(σ, π) on simulated data vector pairs. In Figure 7, we present the MAE comparison on real datasets, where we see that the curves for these two estimators ($\hat{J}_{\sigma,\pi}$ and $\hat{J}_{\pi,\pi}$) match well.

To illustrate the bias and variance of specific data pairs in more details, we test C-MinHash-(π, π) on the "Words" dataset (Li and Church, 2005). For each data point, the i-th 0/1 entry indicates whether a word appears in the i-th document, for a total of $D = 2^{16}$ documents. See the key statistics of the 120 selected word pairs in Table 1. Those pairs of words are more or less randomly selected except that we make sure they cover a wide spectrum of data distributions. Denote d as the number of non-zero entries in the vector. Table 1 reports the density $\tilde{d} = d/D$ for each word vector, ranging from 0.0006 to 0.6. The Jaccard similarity J ranges from 0.002 to 0.95.

In Figures 8 - 15 (also see Appendix B), we plot the empirical MSE along with the empirical bias^2 for $\hat{J}_{\pi,\pi}$, as well as the empirical MSE for $\hat{J}_{\sigma,\pi}$. From the results in the Figures, we can observe

- For all the data pairs, the MSE of C-MinHash-(π, π) estimator overlaps with the empirical MSE of C-MinHash-(σ, π) estimator for all K from 1 up to 4096.
- The bias^2 of C-MinHash-(π, π) is several orders of magnitudes smaller than the MSE, in all data pairs. This demonstrates that the bias of $\hat{J}_{\pi,\pi}$ is extremely small and can be safely neglected in practice.

In all figures, the overlapping curves validate our claim that in practice, we just need one permutation π in C-MinHash.

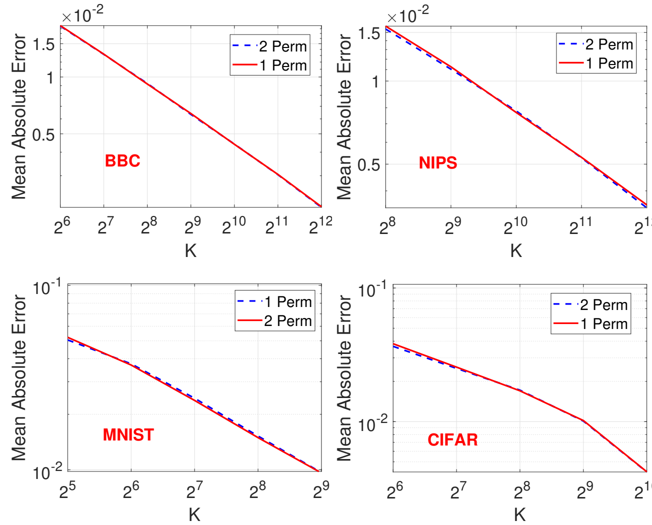

Figure 7. MAE of Jaccard estimation on four datasets. "1 Perm" is C-MinHash-(π, π), and "2 Perm" is C-MinHash-(σ, π).

6. Discussion and Conclusion

The method of *minwise hashing* (MinHash), from the seminal works of Broder and his colleagues, has become standard in industrial practice. One fundamental reason for its wide applicability is that the binary (0/1) high-dimensional representation is convenient and suitable for a wide range of practical scenarios. To estimate the Jaccard similarity on binary data, the standard MinHash requires to use K independent permutations, where K, the number of hashes, can be several hundreds or even thousands in practice.

We have proposed Circulant MinHash (C-MinHash) and present the surprising theoretical results that, with merely 2 permutations, we still obtain an unbiased estimate of the Jaccard similarity with the variance strictly smaller than that of the original MinHash, as confirmed by numerical experiments on simulated and real datasets. The initial permutation is applied to break whatever structure the original data may exhibit. The second permutation is re-used K times in a circulant shifting fashion. Moreover, we propose a more convenient C-MinHash variance which uses only 1 permutation for both pre-processing and circulant hashing. We validate through extensive experiments that it does not result in loss of accuracy in practice.

Practically speaking, our theoretical results may reveal a useful direction for designing hashing methods. For example, in many applications, using permutation vectors of length (e.g.,) 2^{30} might be sufficient. While it is perhaps unrealistic to store (e.g.,) $K = 1024$ such permutation vectors in the memory, one can afford to store two such permutations (even in GPU memory). Using perfectly random permutations in lieu of approximate permutations would simplify the design and analysis of randomized algorithms and ensure that the practical performance strictly matches the theory.

Table 1. 120 selected word pairs from the *Words* dataset (Li and Church, 2005). For each pair, we report the density $\tilde{d}$ (number of non-zero entries divided by $D = 2^{16}$) for each word as well as the Jaccard similarity J. Both $\tilde{d}$ and J cover a wide range of values.

	$\tilde{d}_1$	$\tilde{d}_2$	J		$\tilde{d}_1$	$\tilde{d}_2$	J
ABOUT - INTO	0.302	0.125	0.258	NEW - WEB	0.291	0.194	0.224
ABOUT - LIKE	0.302	0.140	0.281	NEWS - LIKE	0.168	0.140	0.172
ACTUAL - DEVELOPED	0.017	0.030	0.071	NO - WELL	0.220	0.120	0.244
ACTUAL - GRABBED	0.017	0.002	0.016	NOT - IT	0.281	0.295	0.437
AFTER - OR	0.103	0.356	0.220	NOTORIOUSLY - LOCK	0.0006	0.006	0.004
AND - PROBLEM	0.554	0.044	0.070	OF - THEN	0.570	0.104	0.168
AS - NAME	0.280	0.144	0.204	OF - WE	0.570	0.226	0.361
AT - CUT	0.374	0.242	0.052	OPPORTUNITY - COUNTRIES	0.029	0.024	0.066
BE - ONE	0.323	0.221	0.403	OUR - THAN	0.244	0.125	0.245
BEST - AND	0.136	0.554	0.228	OVER - BACK	0.148	0.160	0.233
BRAZIL - OH	0.010	0.031	0.019	OVER - TWO	0.148	0.121	0.289
BUT - MANY	0.167	0.116	0.340	PEAK - SHOWS	0.006	0.033	0.026
CALLED - BUSINESSES	0.016	0.018	0.043	PEOPLE - BY	0.121	0.425	0.228
CALORIES - MICROSOFT	0.002	0.045	0.0003	PEOPLE - INFO	0.121	0.138	0.117
CAN - FROM	0.243	0.326	0.444	PICKS - BOOST	0.007	0.005	0.007
CAN - SEARCH	0.243	0.214	0.237	PLANET - REWARD	0.013	0.003	0.018
COMMITTED - PRODUCTIVE	0.013	0.004	0.029	PLEASE - MAKE	0.168	0.141	0.195
CONTEMPORARY - FLASH	0.011	0.021	0.013	PREFER - PUEDE	0.010	0.003	0.0001
CONVENIENTLY - INDUSTRIES	0.003	0.011	0.009	PRIVACY - FOUND	0.126	0.136	0.053
COPYRIGHT - AN	0.218	0.290	0.209	PROSECUTION - MAXIMIZE	0.002	0.003	0.006
CREDIT - CARD	0.046	0.041	0.285	RECENTLY - INT	0.028	0.007	0.014
DE - WEB	0.117	0.194	0.091	REPLY - ACHIEVE	0.013	0.012	0.023
DO - GOOD	0.174	0.102	0.276	RESERVED - BEEN	0.172	0.141	0.108
EARTH - GROUPS	0.021	0.035	0.056	RIGHTS - FIND	0.187	0.144	0.166
EXPRESSED - FRUSTRATED	0.010	0.002	0.024	RIGHTS - RESERVED	0.187	0.172	0.877
FIND - HAS	0.144	0.228	0.214	SCENE - ABOUT	0.012	0.301	0.029
FIND - SITE	0.144	0.275	0.212	SEE - ALSO	0.138	0.166	0.291
FIXED - SPECIFIC	0.011	0.039	0.054	SEIZE - ANYTHING	0.0007	0.037	0.012
FLIGHT - TRANSPORTATION	0.011	0.018	0.040	SHOULDERS - GORGEOUS	0.003	0.004	0.028
FOUND - DE	0.136	0.117	0.039	SICK - FELL	0.008	0.008	0.085
FRANCISCO - SAN	0.025	0.049	0.476	SITE - CELLULAR	0.275	0.006	0.010
GOOD - BACK	0.102	0.160	0.220	SOLD - LIVE	0.018	0.064	0.055
GROUPS - ORDERED	0.035	0.011	0.034	SOLO - CLAIMS	0.010	0.012	0.007
HAPPY - CONCEPT	0.029	0.013	0.054	SOON - ADVANCE	0.040	0.017	0.057
HAVE - FIRST	0.267	0.151	0.320	SPECIALIZES - ACTUAL	0.003	0.017	0.008
HAVE - US	0.267	0.284	0.349	STATE - OF	0.101	0.570	0.165
HILL - ASSURED	0.020	0.004	0.011	STATES - UNITED	0.061	0.062	0.591
HOME - SYNTHESIS	0.365	0.002	0.003	TATTOO - JEANS	0.002	0.004	0.035
HONG - KONG	0.014	0.014	0.925	THAT - ALSO	0.301	0.166	0.376
HOSTED - DRUGS	0.016	0.013	0.013	THIS - CITY	0.423	0.123	0.132
INTERVIEWS - FOURTH	0.012	0.011	0.031	THEIR - SUPPORT	0.165	0.117	0.189
KANSAS - PROPERTY	0.017	0.045	0.052	THEIR - VIEW	0.165	0.103	0.151
KIRIBATI - GAMBIA	0.003	0.003	0.712	THEM - OF	0.112	0.570	0.187
LAST - THIS	0.135	0.423	0.221	THEN - NEW	0.104	0.291	0.192
LEAST - ROMANCE	0.046	0.007	0.019	THINKS - LOT	0.007	0.040	0.079
LIME - REGISTERED	0.002	0.030	0.004	TIME - OUT	0.189	0.191	0.366
LINKS - TAKE	0.191	0.105	0.134	TIME - WELL	0.189	0.120	0.299
LINKS - THAN	0.191	0.125	0.141	TOP - AS	0.140	0.280	0.217
MAIL - AND	0.160	0.554	0.192	TOP - COPYRIGHT	0.140	0.218	0.149
MAIL - BACK	0.160	0.160	0.132	TOP - NEWS	0.140	0.168	0.192
MAKE - LIKE	0.141	0.140	0.297	UP - AND	0.200	0.554	0.334
MANAGING - LOCK	0.010	0.006	0.010	UP - HAS	0.200	0.228	0.312
MANY - US	0.116	0.284	0.210	US - BE	0.284	0.323	0.335
MASS - DREAM	0.016	0.017	0.048	VIEW - IN	0.103	0.540	0.153
MAY - HELP	0.184	0.156	0.206	VIEW - PEOPLE	0.103	0.121	0.138
MOST - HOME	0.141	0.365	0.207	WALKED - ANTIVIRUS	0.006	0.002	0.002
NAME - IN	0.144	0.540	0.207	WEB - GO	0.194	0.111	0.138
NEITHER - FIGURE	0.011	0.016	0.085	WELL - INFO	0.120	0.138	0.110
NET - SO	0.101	0.154	0.112	WELL - NEWS	0.120	0.168	0.161
NEW - PLEASE	0.291	0.168	0.205	WEEKS - LONDON	0.028	0.032	0.050

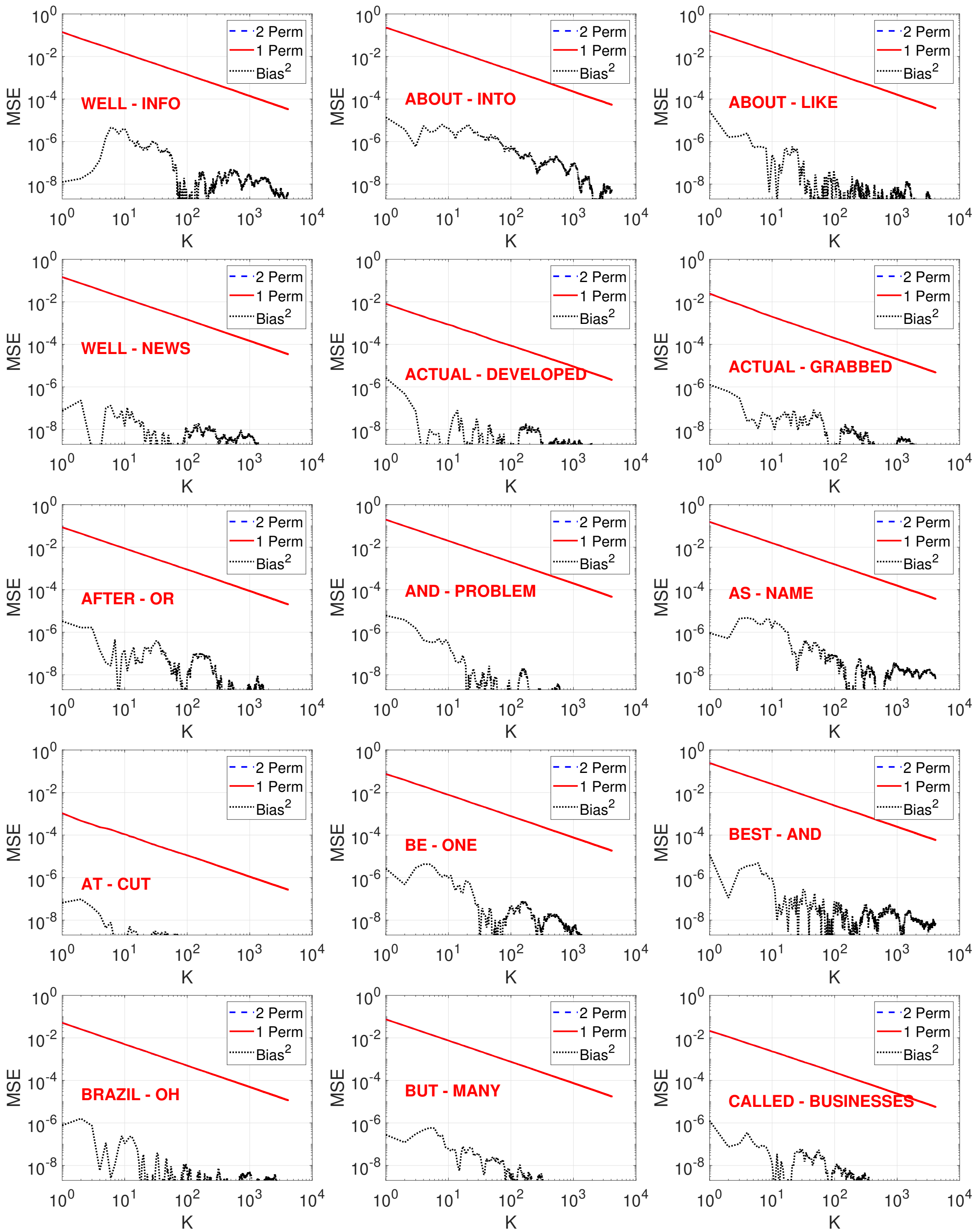

Figure 8. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

References

Michael Bendersky and W. Bruce Croft. Finding text reuse on the web. In *Proceedings of the Second International Conference on Web Search and Web Data Mining (WSDM)*, pages 262–271, Barcelona, Spain, 2009.

Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of the Conference on Compression and Complexity of SEQUENCES*, pages 21–29, Positano, Amalfitan Coast, Salerno, Italy, 1997.

Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Comput. Networks*, 29(8-13):1157–1166, 1997.

Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 327–336, Dallas, TX, 1998.

Gregory Buehrer and Kumar Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 95–106, Stanford, CA, 2008.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 380–388, Montreal, Canada, 2002.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan. On compressing social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 219–228, Paris, France, 2009.

Ondrej Chum and Jiri Matas. Fast computation of min-hash signatures for image collections. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3077–3084, 2012.

Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyamsundar Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pages 271–280, Banff, Alberta, Canada, 2007.

Fan Deng, Stefan Siersdorfer, and Sergej Zerr. Efficient jaccard-based diversity analysis of large document collections. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1402–1411, Maui, HI, 2012.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Weiqi Feng and Dong Deng. Allign: Aligning all-pair near-duplicate passages in long texts. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 541–553, Virtual Event, China, 2021.

Dennis Fetterly, Mark Manasse, Marc Najork, and Janet L. Wiener. A large-scale study of the evolution of web pages. In *Proceedings of the Twelfth International World Wide Web Conference (WWW)*, pages 669–678, Budapest, Hungary, 2003.

Michael Gamon, Sumit Basu, Dmitriy Belenko, Danyel Fisher, Matthew Hurst, and Christian König. BLEWS: using blogs to provide context for news articles. In *Proceedings of the Second International Conference on Weblogs and Social Media (ICWSM)*, Seattle, WA, 2008.

Derek Greene and Padraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML)*, pages 377–384, Pittsburgh, PA, 2006.

Kaiming He, Fang Wen, and Jian Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2938–2945, Portland, OR, 2013.

Monika Rauch Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 284–291, Seattle, WA, 2006.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 604–613, Dallas, TX, 1998.

Sergey Ioffe. Improved consistent sampling, weighted minhash and L1 sketching. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 246–255, Sydney, Australia, 2010.

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 *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 830–842, Virtual, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

David C. Lee, Qifa Ke, and Michael Isard. Partition min-hash for partial duplicate image discovery. In *Proceedings of the 11th European Conference on Computer Vision (ECCV), Part I*, pages 648–662, Heraklion, Greece, 2010.

Jakub Lemiesz. On the algebra of data sketches. *Proc. VLDB Endow.*, 14(9):1655–1667, 2021.

Ping Li and Kenneth Ward Church. Using sketches to estimate associations. In *Proceedings of the Conference on Human Language Technology and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 708–715, Vancouver, Canada, 2005.

Ping Li and Arnd Christian König. Theory and applications of *b*-bit minwise hashing. *Commun. ACM*, 54(8):101–109, 2011.

Ping Li and Cun-Hui Zhang. Theory of the GMM kernel. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1053–1062, Perth, Australia, 2017.

Ping Li and Weijie Zhao. GCWSNet: Generalized consistent weighted sampling for scalable and accurate training of neural networks. *arXiv preprint arXiv:2201.02283*, 2022.

Ping Li, Anshumali Shrivastava, Joshua Moore, and Arnd Christian König. Hashing algorithms for large-scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, Granada, Spain, 2011.

Ping Li, Anshumali Shrivastava, and Arnd Christian König. GPU-based minwise hashing. In *Proceedings of the 21st World Wide Web Conference (WWW)*, pages 565–566, Lyon, France, 2012.

Ping Li, Xiaoyun Li, Gennady Samorodnitsky, and Weijie Zhao. Consistent sampling through extremal process. In *Proceeding of the Web Conference (WWW)*, pages 1317–1327, Virtual Event / Ljubljana, Slovenia, 2021.

Mark Manasse, Frank McSherry, and Kunal Talwar. Consistent weighted sampling. Technical Report MSR-TR-2010-73, Microsoft Research, 2010.

Marc Najork, Sreenivas Gollapudi, and Rina Panigrahy. Less is more: sampling the neighborhood graph makes salsa better and faster. In *Proceedings of the Second International Conference on Web Search and Web Data Mining (WSDM)*, pages 242–251, Barcelona, Spain, 2009.

Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. Table union search on open data. *Proc. VLDB Endow.*, 11(7):813–825, 2018.

Brian D Ondov, Todd J Treangen, Páll Melsted, Adam B Mallonee, Nicholas H Bergman, Sergey Koren, and Adam M Phillippy. Mash: fast genome and metagenome distance estimation using minhash. *Genome biology*, 17 (1):1–14, 2016.

Sandeep Pandey, Andrei Broder, Flavio Chierichetti, Vanja Josifovski, Ravi Kumar, and Sergei Vassilvitskii. Nearest-neighbor caching for content-match applications. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pages 441–450, Madrid, Spain, 2009.

Anshumali Shrivastava and Ping Li. Fast near neighbor search in high-dimensional binary data. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 474–489, Bristol, UK, 2012.

Anshumali Shrivastava and Ping Li. In defense of minhash over simhash. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 886–894, Reykjavik, Iceland, 2014.

Acar Tamersoy, Kevin A. Roundy, and Duen Horng Chau. Guilt by association: large scale malware detection by mining file-relation graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1524–1533, New York, NY, 2014.

Tom Tseng, Laxman Dhulipala, and Julian Shun. Parallel index-based structural graph clustering and its approximation. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1851–1864, Virtual Event, China, 2021.

Pinghui Wang, Yiyan Qi, Yuanming Zhang, Qiaozhu Zhai, Chenxu Wang, John C. S. Lui, and Xiaohong Guan. A memory-efficient sketch method for estimating high similarities in streaming sets. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 25–33, 2019.

Felix X. Yu, Aditya Bhaskara, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. On binary embedding using circulant matrices. *J. Mach. Learn. Res.*, 18:150:1–150:30, 2017.

Juan Zamora, Marcelo Mendoza, and Héctor Allende. Hashing-based clustering in high dimensional data. *Expert Syst. Appl.*, 62:202–211, 2016.

Erkang Zhu, Ken Q. Pu, Fatemeh Nargesian, and Renée J. Miller. Interactive navigation of open data linkages. *Proc. VLDB Endow.*, 10(12):1837–1840, 2017.

A. Proofs of Technical Results

We first recall the notations and definitions that will be used in our proof.

Notations. In our analysis, for simplicity we will use $\mathbb{1}_s$ to denote $\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}$ for $\forall 1 \leq s \leq K$, where h is the hash value. Given two data vectors $\boldsymbol{v}, \boldsymbol{w} \in \{0,1\}^D$. Recall in (6) that $a = \sum_{i=1}^D \mathbb{1}\{\boldsymbol{v}_i = 1 \text{ and } \boldsymbol{w}_i = 1\}$, $f = \sum_{i=1}^D \mathbb{1}\{\boldsymbol{v}_i = 1 \text{ or } \boldsymbol{w}_i = 1\}$. Thus, the Jaccard similarity is $J = a/f$. We also define $\tilde{J} = (a-1)/(f-1)$.

Definition A.1. Let $\boldsymbol{v}, \boldsymbol{w} \in \{0,1\}^D$. Define the **location vector** as $\boldsymbol{x} \in \{O, \times, -\}^D$, with $\boldsymbol{x}_i$ being "O", "$\times$", "$-$" when $\boldsymbol{v}_i = \boldsymbol{w}_i = 1$, $\boldsymbol{v}_i + \boldsymbol{w}_i = 1$ and $\boldsymbol{v}_i = \boldsymbol{w}_i = 0$, respectively.

Definition A.2. For $A, B \in \{O, \times, -\}$, let $\{(i,j) : (A,B)|\triangle\}$ denote a pair of indices $\{(i,j) : (\boldsymbol{x}_i, \boldsymbol{x}_j) = (A,B), j - i = \triangle\}$. Define

$$
\begin{aligned}
&\mathcal{L}_0(\triangle) = \{(i,j) : (O,O)|\triangle\}, \quad \mathcal{L}_1(\triangle) = \{(i,j) : (O,\times)|\triangle\}, \quad \mathcal{L}_2(\triangle) = \{(i,j) : (O,-)|\triangle\}, \\
&\mathcal{G}_0(\triangle) = \{(i,j) : (-,O)|\triangle\}, \quad \mathcal{G}_1(\triangle) = \{(i,j) : (-,\times)|\triangle\} \quad , \mathcal{G}_2(\triangle) = \{(i,j) : (-,-)|\triangle\}, \\
&\mathcal{H}_0(\triangle) = \{(i,j) : (\times,O)|\triangle\}, \quad \mathcal{H}_1(\triangle) = \{(i,j) : (\times,\times)|\triangle\}, \quad \mathcal{H}_2(\triangle) = \{(i,j) : (\times,-)|\triangle\}.
\end{aligned}
$$

Remark A.3. For the ease of notation, by circulation we write $\boldsymbol{x}_j = \boldsymbol{x}_{j-D}$ when $D < j < 2D$.

One can easily verify that given fixed a, f, D, it holds that for $\forall 1 \leq \triangle \leq K-1$,

$$
\begin{aligned}
&|\mathcal{L}_0(\triangle)| + |\mathcal{L}_1(\triangle)| + |\mathcal{L}_2(\triangle)| = |\mathcal{L}_0(\triangle)| + |\mathcal{G}_0(\triangle)| + |\mathcal{H}_0(\triangle)| = a, \\
&|\mathcal{G}_0(\triangle)| + |\mathcal{G}_1(\triangle)| + |\mathcal{G}_2(\triangle)| = |\mathcal{L}_2(\triangle)| + |\mathcal{G}_2(\triangle)| + |\mathcal{H}_2(\triangle)| = D - f, \\
&|\mathcal{H}_0(\triangle)| + |\mathcal{H}_1(\triangle)| + |\mathcal{H}_2(\triangle)| = |\mathcal{L}_1(\triangle)| + |\mathcal{G}_1(\triangle)| + |\mathcal{H}_1(\triangle)| = f - a.
\end{aligned} \tag{11}
$$

We will refer this as the intrinsic constraints on the sizes of above sets.

A.1. Proof of Lemma 2.4

Lemma 2.4. For any $1 \leq s < t \leq K$ with $t - s = \triangle$, we have

$$
\mathbb{E}_\pi\big[\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}\mathbb{1}\{h_t(\boldsymbol{v}) = h_t(\boldsymbol{w})\}\big] = \frac{|\mathcal{L}_0(\triangle)| + (|\mathcal{G}_0(\triangle)| + |\mathcal{L}_2(\triangle)|)J}{f + |\mathcal{G}_0(\triangle)| + |\mathcal{G}_1(\triangle)|},
$$

where the sets are defined in Definition 2.2 and h_s, h_t are the hash values output by Algorithm 2.

Proof. To check whether a hash sample generated by C-MinHash collides (under some permutation π), it suffices to look at the permuted location vector $\boldsymbol{x}$. If a collision happens if only if the first type "O" point appears before the first "$\times$" point after being permuted by π. That said, the minimal permutation index of "O" elements must be smaller than that of "$\times$" elements. If the hash sample does not collide, then the first "$\times$" must appear before the first "O". Note that "$-$" points does not affect the collision. This observation will be the key to our analysis.

To compute the variance of the C-MinHash-$(0,\pi)$ estimator, it suffices to compute $\mathbb{E}[\mathbb{1}_s \mathbb{1}_t]$. Let $\mathcal{L}$, $\mathcal{G}$ and $\mathcal{H}$ denote the union of $\mathcal{L}$'s, $\mathcal{G}$'s and $\mathcal{H}$'s, respectively. In the following, we say that an index i belongs to a set if i is the first term of an element in that set, e.g., $\{1\}$ belongs to $\mathcal{L}$ if $x_1 = O$. We have

$$
|\mathcal{L}| = a, \quad |\mathcal{H}| = f - a, \quad |\mathcal{G}| = D - f.
$$

One key observation is that, for a pair (i,j) with $|j - i| = \triangle = t - s$ in above sets, the hash index $\pi_s(i)$ will be the hash index of $\pi_t(j)$. We begin by decomposing the expectation of interest into

$$
\begin{aligned}
\mathbb{E}[\mathbb{1}_s \mathbb{1}_t] &= P[\text{collision } s, \text{collision } t] \\
&= \sum_{i_s^* \in \mathcal{L}} P[\text{collision } s \text{ at } i_s^*, \text{collision } t] \\
&= \sum_{p=0}^{2} \sum_{i_s^* \in \mathcal{L}_p} P[\text{collision } s \text{ at } i_s^*, \text{collision } t].
\end{aligned} \tag{12}
$$

where i_s^* is the location of the original "O" in vector x that collides for the s-th hash sample. Note that it is different from the exact location of collision in $x(\pi_s)$. Also, the permutation is totally random, so the location of collision is independent of $\mathbb{1}_s$, and uniformly distributed among all type "O" pairs. We consider the following cases.

1) When $i_s^* \in \mathcal{L}_0$. In this case, the minimum index of the type "O" pair in $x(\pi_s)$, $\pi_s(i_s^*)$, is shifted to another type "O" pair in $x(\pi_t)$. Therefore, the indices of pairs with the first element being "O" or "$\times$" originally in $x(\pi_s)$ will still be greater than $\pi_t(i_s^*)$. If sample s collides at i_s^*, hash sample t will collide when

1. All the points in $\mathcal{G}_1$, after permutation π_s, is greater than $\pi_s(i_s^*)$. In this case, regardless of the permuted $\mathcal{G}_0$, hash t will always collide.
2. There exist points in $\mathcal{G}_1$ after permutation π_s that are smaller than $\pi_s(i_s^*)$, and also points in $\mathcal{G}_0$ that is smaller than the minimum of permuted $\mathcal{G}_1$.

Consequently, we have for $i_s^* \in \mathcal{L}_0$,

$$\begin{aligned}
&P[\text{collision } s \text{ at } i_s^*, \text{collision } t] \\
&= P[\pi_s(i_s^*) < \pi_s(i), \forall i \in \mathcal{H} \cup \mathcal{L}/i_s^*, \text{ and } \min_{j \in \mathcal{G}_1} \pi_s(j) > \pi_s(i_s^*)] \\
&\qquad + P[\pi_s(i_s^*) < \pi_s(i), \forall i \in \mathcal{H} \cup \mathcal{L}/i_s^*, \text{ and } \min_{j \in \mathcal{G}_0} \pi_s(j) < \min_{j \in \mathcal{G}_1} \pi_s(j) < \pi_s(i_s^*)] \\
&= \frac{1}{a} \cdot \frac{a}{f + |\mathcal{G}_1|} + \frac{|\mathcal{G}_0|}{f + |\mathcal{G}_0| + |\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_1|}{f + |\mathcal{G}_1|} \cdot \frac{a}{f} \cdot \frac{1}{a} \\
&= \frac{1}{f + |\mathcal{G}_1|} + \frac{|\mathcal{G}_0| \cdot |\mathcal{G}_1|}{(f + |\mathcal{G}_0| + |\mathcal{G}_1|)(f + |\mathcal{G}_1|) f}.
\end{aligned} \tag{13}$$

This probability holds for $\forall i_s^* \in \mathcal{L}_0$.

2) When $i_s^* \in \mathcal{L}_1$. Similarly, we consider the condition where $i_s^* \in \mathcal{L}_1$, and both hash samples collide. In this case, $\pi_s(i_s^*)$ would be shifted to a "$\times$" pair in $x(\pi_t)$. That is, the indices of pairs with the first element being "O" or "$\times$" originally in $x(\pi_s)$ will all be greater than $\pi_s(i_s^*)$, which now is the location of a "$\times$" pair in $x(\pi_t)$. Thus, to make hash t collide, we need:

- At least one point from $\mathcal{G}_0$ is smaller than any other points in $\mathcal{H} \cup \mathcal{L} \cup \mathcal{G}_1$ after permutation π_s.

Therefore, for any $i_s^* \in \mathcal{L}_1$,

$$\begin{aligned}
&P[\text{collision } s \text{ at } i_s^*, \text{collision } t] \\
&= P[\pi_s(i_s^*) < \pi_s(i), \forall i \in \mathcal{H} \cup \mathcal{L}/i_s^*, \text{ and } \min_{j \in \mathcal{G}_0} \pi_s(j) < \min\{\pi_s(i_s^*), \min_{j \in \mathcal{G}_1} \pi_s(j)\}] \\
&= \frac{|\mathcal{G}_0|}{f + |\mathcal{G}_0| + |\mathcal{G}_1|} \cdot \frac{a}{f} \cdot \frac{1}{a} \\
&= \frac{|\mathcal{G}_0|}{(f + |\mathcal{G}_0| + |\mathcal{G}_1|) f},
\end{aligned} \tag{14}$$

which is true for $\forall i_s^* \in \mathcal{L}_1$.

3) When $i_s^* \in \mathcal{L}_2$.

In this scenario, $\pi_s(i_s^*)$ would be shifted to a "$-$" pair in $x(\pi_t)$. Therefore, if hash s collides, hash t will also collide when:

- After applying π_s, the minimum of $\mathcal{L}_0 \cup \mathcal{H}_0 \cup \mathcal{G}_0$ is smaller than the minimum of $\mathcal{L}_1 \cup \mathcal{H}_1 \cup \mathcal{G}_1$.

Thus, we obtain that for any $i_s^* \in \mathcal{L}_2$,

$$\begin{aligned}
&P[\text{collision } s \text{ at } i_s^*, \text{collision } t] \\
&= P[\pi_s(i_s^*) < \pi_s(i), \forall i \in \mathcal{H} \cup \mathcal{L}/i_s^*, \text{ and } \min_{j \in \mathcal{L}_0 \cup \mathcal{G}_0 \cup \mathcal{H}_0} \pi_s(j) < \min_{j \in \mathcal{L}_1 \cup \mathcal{G}_1 \cup \mathcal{H}_1} \pi_s(j)] \\
&\triangleq P[\Omega].
\end{aligned}$$

Let $\mathbb{1}_{s,i_s^*}$ denote the event $\{\pi_s(i_s^*) < \pi_s(i), \forall i \in \mathcal{H} \cup \mathcal{L}/i_s^*\}$. Then Ω can be separated into the following cases:

1. Ω_1: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) < \min_{j\in\mathcal{G}_1} \pi_s(j)$.
2. Ω_2: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) > \min_{j\in\mathcal{G}_1} \pi_s(j) > \min_{j\in\mathcal{G}_0} \pi_s(j) > \pi_s(i_s^*)$.
3. Ω_3: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) > \min_{j\in\mathcal{G}_1} \pi_s(j) > \pi_s(i_s^*) > \min_{j\in\mathcal{G}_0} \pi_s(j)$.
4. Ω_4: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) > \pi_s(i_s^*) > \min_{j\in\mathcal{G}_1} \pi_s(j) > \min_{j\in\mathcal{G}_0} \pi_s(j)$.
5. Ω_5: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) > \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\pi_s(i_s^*) < \min_{j\in\mathcal{G}_0} \pi_s(j) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1\cup\mathcal{G}_1} \pi_s(j)$.
6. Ω_6: $\mathbb{1}_{s,i_s^*}$, $\min_{j\in\mathcal{L}_0\cup\mathcal{H}_0} \pi_s(j) > \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1} \pi_s(j)$, and $\min_{j\in\mathcal{G}_0} \pi_s(j) < \pi_s(i_s^*) < \min_{j\in\mathcal{L}_1\cup\mathcal{H}_1\cup\mathcal{G}_1} \pi_s(j)$.

We can compute the probability of each event as

$$\begin{aligned}
P[\Omega_1] &= \frac{1}{a} \cdot \frac{a}{f+|\mathcal{G}_1|} \cdot \frac{|\mathcal{L}_0|+|\mathcal{H}_0|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|+|\mathcal{G}_1|}, \\
&= \frac{a-|\mathcal{G}_0|}{(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)}, \\
P[\Omega_2] &= \frac{1}{a} \cdot \frac{a}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_0|}{|\mathcal{G}_0|+|\mathcal{G}_1|+|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&\qquad \cdot \frac{|\mathcal{G}_1|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{L}_0|+|\mathcal{H}_0|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&= \frac{1}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_0|}{f} \cdot \frac{|\mathcal{G}_1|}{f-|\mathcal{G}_0|} \cdot \frac{a-|\mathcal{G}_0|}{f-|\mathcal{G}_0|-|\mathcal{G}_1|} \\
&= \frac{|\mathcal{G}_0|\cdot|\mathcal{G}_1|\cdot(a-|\mathcal{G}_0|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)f}, \\
P[\Omega_3] &= \frac{|\mathcal{G}_0|}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{1}{f+|\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_1|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{L}_0|+|\mathcal{H}_0|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&= \frac{|\mathcal{G}_0|\cdot|\mathcal{G}_1|\cdot(a-|\mathcal{G}_0|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f+|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)}, \\
P[\Omega_4] &= \frac{|\mathcal{G}_0|}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_1|}{f+|\mathcal{G}_1|} \cdot \frac{1}{f} \cdot \frac{|\mathcal{L}_0|+|\mathcal{H}_0|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&= \frac{|\mathcal{G}_0|\cdot|\mathcal{G}_1|\cdot(a-|\mathcal{G}_0|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)f}, \\
P[\Omega_5] &= \frac{1}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{|\mathcal{G}_0|}{|\mathcal{G}_0|+|\mathcal{G}_1|+|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \cdot \frac{|\mathcal{L}_1|+|\mathcal{H}_1|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&= \frac{|\mathcal{G}_0|\cdot(f-a-|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)f}, \\
P[\Omega_6] &= \frac{|\mathcal{G}_0|}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} \cdot \frac{1}{f} \cdot \frac{|\mathcal{L}_1|+|\mathcal{H}_1|}{|\mathcal{L}_0|+|\mathcal{H}_0|+|\mathcal{L}_1|+|\mathcal{H}_1|} \\
&= \frac{|\mathcal{G}_0|\cdot(f-a-|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)f}.
\end{aligned}$$

Note that

$$\begin{aligned}
&P[\Omega_2]+P[\Omega_3]+P[\Omega_4] \\
&= \frac{|\mathcal{G}_0|\cdot|\mathcal{G}_1|\cdot(a-|\mathcal{G}_0|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)} \left[\frac{1}{(f-|\mathcal{G}_0|)f} + \frac{1}{(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)} + \frac{1}{f(f+|\mathcal{G}_1|)}\right] \\
&= \frac{|\mathcal{G}_0|\cdot|\mathcal{G}_1|\cdot(a-|\mathcal{G}_0|)(3f-|\mathcal{G}_0|+|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)f(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)}.
\end{aligned}$$

Summing up all the terms together, we obtain $P[\Omega]$ as

$$\begin{aligned}\sum_{n=1}^{6} P[\Omega_n] &= \frac{f(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(a-|\mathcal{G}_0|)+|\mathcal{G}_0||\mathcal{G}_1|(a-|\mathcal{G}_0|)(3f-|\mathcal{G}_0|+|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)f} \\ &\quad + \frac{2|\mathcal{G}_0|(f-a-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)f} \\ &= \frac{(a-|\mathcal{G}_0|)(f+|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)+2|\mathcal{G}_0|(f-a-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)f} \\ &= \frac{(a+|\mathcal{G}_0|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f-|\mathcal{G}_0|-|\mathcal{G}_1|)(f-|\mathcal{G}_0|)(f+|\mathcal{G}_1|)f} \\ &= \frac{a+|\mathcal{G}_0|}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)f}, \end{aligned} \tag{15}$$

which holds for $\forall i_s^* \in \mathcal{L}_2$. Now combining (13), (14), (15) with (12), we obtain

$$\mathbb{E}[\mathbb{1}_s\mathbb{1}_t] = \frac{|\mathcal{L}_0|}{f+|\mathcal{G}_1|} + \frac{|\mathcal{G}_0||\mathcal{G}_1||\mathcal{L}_0|}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)(f+|\mathcal{G}_1|)f} + \frac{|\mathcal{G}_0||\mathcal{L}_1|}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)f} + \frac{(a+|\mathcal{G}_0|)|\mathcal{L}_2|}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)f}. \tag{16}$$

Here, recall that the sets are associated with all $1 \le s < t \le K$ such that $\triangle = t - s$. Using the intrinsic constraints (11), after some calculation we can simplify (16) as

$$\mathbb{E}_\pi[\mathbb{1}_s\mathbb{1}_t] = \frac{|\mathcal{L}_0|}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} + \frac{a(|\mathcal{G}_0|+|\mathcal{L}_2|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)f} = \frac{|\mathcal{L}_0(\triangle)|+(|\mathcal{G}_0(\triangle)|+|\mathcal{L}_2(\triangle)|)J}{f+|\mathcal{G}_0(\triangle)|+|\mathcal{G}_1(\triangle)|},$$

which completes the proof. □

A.2. Proof of Theorem 2.5

Theorem 2.5. Under the same setting as in Lemma 2.4, the variance of $\hat{J}_{0,\pi}$ is

$$Var[\hat{J}_{0,\pi}] = \frac{J}{K} + \frac{2\sum_{s=2}^{K}(s-1)\Theta_{K-s+1}}{K^2} - J^2,$$

where $\Theta_\triangle \triangleq \mathbb{E}_\pi\big[\mathbb{1}\{h_s(\boldsymbol{v}) = h_s(\boldsymbol{w})\}\mathbb{1}\{h_t(\boldsymbol{v}) = h_t(\boldsymbol{w})\}\big]$ as in Lemma 2.4 with any $t - s = \triangle$.

Proof. By the expansion of variance formula, since $\mathbb{E}[\mathbb{1}_s^2] = \mathbb{E}[\mathbb{1}_s] = J$, we have

$$Var[\hat{J}_{0,\pi}] = \frac{J}{K} + \frac{\sum_{s=1}^{K}\sum_{t\neq s}^{K}\mathbb{E}[\mathbb{1}_s\mathbb{1}_t]}{K^2} - J^2. \tag{17}$$

Note here that for $\forall t > s$, the t-th hash sample uses π_t as the permutation, which is shifted rightwards by $\triangle = t - s$ from π_s. Thus, we have $\mathbb{E}[\mathbb{1}_s\mathbb{1}_t] = \mathbb{E}[\mathbb{1}_{s-i}\mathbb{1}_{t-i}]$ for $\forall 0 < i < s \wedge t$, which implies $\mathbb{E}[\mathbb{1}_s\mathbb{1}_t] = \mathbb{E}[\mathbb{1}_1\mathbb{1}_{t-s+1}], \forall s < t$. Since by assumption $K \le D$, we have

$$\begin{aligned}\sum_{s}^{K}\sum_{t\neq s}^{K}\mathbb{E}[\mathbb{1}_s\mathbb{1}_t] &= 2\mathbb{E}\big[(\mathbb{1}_1\mathbb{1}_2 + \mathbb{1}_1\mathbb{1}_3 + ... + \mathbb{1}_1\mathbb{1}_K) + (\mathbb{1}_2\mathbb{1}_3 + ... + \mathbb{1}_2\mathbb{1}_K) + ... + \mathbb{1}_{K-1}\mathbb{1}_K\big] \\ &= 2\mathbb{E}\big[(\mathbb{1}_1\mathbb{1}_2 + \mathbb{1}_1\mathbb{1}_3 + ... + \mathbb{1}_1\mathbb{1}_K) + (\mathbb{1}_1\mathbb{1}_2 + ... + \mathbb{1}_1\mathbb{1}_{K-1}) + ... + \mathbb{1}_1\mathbb{1}_2\big] \\ &= 2\sum_{s=2}^{K}(s-1)\mathbb{E}[\mathbb{1}_1\mathbb{1}_{K-s+2}] \\ &\triangleq 2\sum_{s=2}^{K}(s-1)\Theta_{K-s+1}. \end{aligned} \tag{18}$$

Finally, integrating (17), (18) and Lemma 2.4 completes the proof. □

A.3. Proof of Theorem 3.1

Theorem 3.1. Let a, f be defined as in (6). When $0 < a < f \leq D$ ($J \notin \{0,1\}$), we have

$$Var[\hat{J}_{\sigma,\pi}] = \frac{J}{K} + \frac{(K-1)\tilde{\mathcal{E}}}{K} - J^2, \tag{19}$$

where $l = \max(0, D-2f+a)$, and

$$\tilde{\mathcal{E}} = \sum_{\{l_0,l_2,g_0,g_1\}} \Bigg\{ \left(\frac{l_0}{f+g_0+g_1} + \frac{a(g_0+l_2)}{(f+g_0+g_1)f} \right) \times \sum_{s=l}^{D-f-1} \frac{\binom{D-f}{s}}{\binom{D-a-1}{D-f-1}} \frac{\binom{f-a-1}{D-f-s-1}\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{a-l_1-l_2}}{\binom{D-1}{a}} \Bigg\}. \tag{20}$$

The feasible set $\{l_0, l_2, g_0, g_1\}$ satisfies the intrinsic constraints (7), and

$$n_1 = g_0 - (D-f-s-g_1), \qquad n_2 = D-f-s-g_1,$$
$$n_3 = l_2 - g_0 + (D-f-s-g_1), \quad n_4 = l_1 - (D-f-s-g_1).$$

Also, when $a = 0$ or $f = a$ ($J = 0$ or $J = 1$), we have $Var[\hat{J}_{\sigma,\pi}] = 0$.

Proof. Similar to the proof of Theorem 2.5, we denote $\Theta_\triangle = \mathbb{E}_{\sigma,\pi}[\mathbb{1}_s \mathbb{1}_t]$ with $|t-s| = \triangle$. Note that now the expectation is taken w.r.t. both two independent permutations σ and π. Since σ is random, we know that $\Theta_1 = \Theta_2 = \cdots = \Theta_{K-1}$. Then by the variance formula, we have

$$Var[\hat{J}_{\sigma,\pi}] = \frac{J^2}{K} - \frac{(K-1)\Theta_1}{K} - J^2 \tag{21}$$

Hence, it suffices to consider Θ_1. In this proof, we will set $\triangle = 1$ and drop the notation $\triangle$ for conciseness, and denote $\tilde{\mathcal{E}} = \Theta_1$ from now on. First, we note that Lemma 2.4 gives the desired quantity conditional on σ. By the law of total probability, we have

$$\tilde{\mathcal{E}} = \mathbb{E}_\sigma \left[\frac{|\mathcal{L}_0|}{f+|\mathcal{G}_0|+|\mathcal{G}_1|} + \frac{a(|\mathcal{G}_0|+|\mathcal{L}_2|)}{(f+|\mathcal{G}_0|+|\mathcal{G}_1|)f} \right], \tag{22}$$

where the sizes of sets are random depending on the initial permutation σ (i.e. counted after permuting by σ). As a result, the problem turns into deriving the distribution of $|\mathcal{L}_0|, |\mathcal{L}_1|, |\mathcal{L}_2|, |\mathcal{G}_0|$ and $|\mathcal{G}_1|$ under random permutation σ, and then taking expectation of (22) with respect to this additional randomness.

When $a = 0$, we know that $|\mathcal{L}_0| = |\mathcal{L}_2| = |\mathcal{G}_0| = 0$, hence the expectation $\tilde{\mathcal{E}}$ is trivially 0. Thus, the $Var[\hat{J}_{\sigma,\pi}] = 0$. When $f = a$, $|\mathcal{G}_1| = 0$, and the constraint on the sets becomes

$$|\mathcal{L}_0| + |\mathcal{G}_0| = |\mathcal{L}_0| + |\mathcal{L}_2| = f,$$
$$|\mathcal{L}_2| + |\mathcal{G}_2| = |\mathcal{G}_0| + |\mathcal{G}_2| = D - f.$$

Then (22) becomes

$$\tilde{\mathcal{E}} = \mathbb{E}_\sigma \left[\frac{|\mathcal{L}_0|}{f+|\mathcal{G}_0|} + \frac{|\mathcal{G}_0|+|\mathcal{L}_2|}{f+|\mathcal{G}_0|} \right] = \mathbb{E}_\sigma \left[\frac{|\mathcal{L}_0|+|\mathcal{G}_0|+|\mathcal{L}_2|}{f+|\mathcal{G}_0|} \right] \equiv 1.$$

Therefore, when $f = a$, we also have $Var[\hat{J}_{\sigma,\pi}] = 0$.

Next, we will consider the general case where $0 < a < f \leq D$. This can be considered as a combinatorial problem where we randomly arrange a type "O", $(f-a)$ type "$\times$" and $(D-f)$ type "$-$" points in a circle. We are interested in the distribution of the number of $\{O,O\}$, $\{O,\times\}$, $\{O,-\}$, $\{-,O\}$ and $\{-,\times\}$ pairs of consecutive points in clockwise direction. We consider this procedure in two steps, where we first place "$\times$" and "$-$" points, and then place "O" points.

Step 1. Randomly place "$\times$" and "$-$" points on the circle.

In this step, four types of pairs may appear: $\{-,-\}$, $\{-,\times\}$, $\{\times,\times\}$ and $\{\times,-\}$. Denote $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ as the collections of above pairs. Since

$$|\mathcal{C}_1| + |\mathcal{C}_4| = |\mathcal{C}_1| + |\mathcal{C}_2| = D - f,$$
$$|\mathcal{C}_2| + |\mathcal{C}_3| = |\mathcal{C}_2| + |\mathcal{C}_4| = f - a,$$

knowing the size of one set gives information on the size of all the sets. Thus, we can characterize the joint distribution by analyzing the distribution of $|\mathcal{C}_1|$. First, placing $(D-f)$ "$-$" points on a circle leads to $(D-f)$ number of $\{-,-\}$ pairs. This $(D-f)$ elements can be regarded as the borders that split the circle into $(D-f)$ bins. Now, we randomly throw $(f-a)$ number of "$\times$" points into these bins. If at least one "$\times$" falls into one bin, then the number of $\{-,-\}$ pairs ($|\mathcal{C}_1|$) would reduce by 1, while $|\mathcal{C}_2|$ and $|\mathcal{C}_4|$ would increase by 1. If z "$\times$" points fall into one bin, then the number of $\{\times,\times\}$ ($|\mathcal{C}_3|$) would increase by $(z-1)$. Notice that since $s \leq D-f$ and $D-f-s \leq f-a$, we have $\max(0, D-2f+a) \leq s \leq D-f$. Consequently, for s in this range, we have

$$\begin{aligned} P\Big\{|\mathcal{C}_1| = s\Big\} &= P\Big\{|\mathcal{C}_1| = s, |\mathcal{C}_3| = f-a-(D-f-s)\Big\} \\ &= \frac{\binom{D-f}{D-f-s}\binom{f-a-1}{D-f-s-1}}{\binom{D-a-1}{D-f-1}} \\ &= \frac{\binom{D-f}{s}\binom{f-a-1}{D-f-s-1}}{\binom{D-a-1}{D-f-1}}. \end{aligned} \tag{23}$$

The second line is due to the stars and bars problem that the number of ways to place n unlabeled balls in m distinct bins such that each bin has at least one ball is $\binom{n-1}{m-1}$. For $|\mathcal{C}_1| = s$, we need $n = f-a$ (number of "$\times$") and $m = |\mathcal{C}_2| = D-f-s$. Moreover, the number of ways to place n balls in m distinct bins is $\binom{n+m-1}{m-1}$. When counting the total number of possibilities, we have $n = f-a$, $m = D-f$. This gives the denominator. Note that (23) is a hyper-geometric distribution.

Step 2. Randomly place "O" points on the circle.

We have the probability mass function

$$\begin{aligned} P[\Psi] &\triangleq P\Big\{|\mathcal{L}_1| = l_1, |\mathcal{L}_2| = l_2, |\mathcal{G}_0| = g_0, |\mathcal{G}_1| = g_1\Big\} \\ &= \sum_{s=D-2f+a}^{D-f-1} P\Big\{|\mathcal{L}_1| = l_1, |\mathcal{L}_2| = l_2, |\mathcal{G}_0| = g_0, |\mathcal{G}_1| = g_1 \Big| |\mathcal{C}_1| = s\Big\} P\Big\{|\mathcal{C}_1| = s\Big\}. \end{aligned} \tag{24}$$

It remains to compute the distribution conditional on $|\mathcal{C}_1|$. Here we drop $|\mathcal{L}_0|$ since it is intrinsically determined by $|\mathcal{L}_1|$, $|\mathcal{L}_2|$. Again, given a placement of all "$\times$" and "$-$" points, each consecutive pair can be regarded as a distinct bin. The problem is hence to randomly throw a type "O" points into that $(D-a)$ bins, given that we have placed type "$\times$" and "$-$" points on the circle with $|\mathcal{C}_1| = s$ (thus $|\mathcal{C}_2| = |\mathcal{C}_3| = D-f-s$ and $|\mathcal{C}_4| = f-a-(D-f-s)$ are also determined correspondingly). In the following, we count the number of "O" points that fall in $\mathcal{C}_i$, $i = 1$ to 4, to make the event Ψ happen. Note that

- When at least one "O" point falls into $\mathcal{C}_1$ (between $\{-,-\}$), $|\mathcal{L}_2|$ and $|\mathcal{G}_0|$ increase by 1.
- When at least one "O" point falls into $\mathcal{C}_2$ (between $\{-,\times\}$), $|\mathcal{L}_1|$ and $|\mathcal{G}_0|$ increase by 1, while $|\mathcal{G}_1|$ decreases by 1.
- When at least one "O" point falls into $\mathcal{C}_3$ (between $\{\times,-\}$), $|\mathcal{L}_2|$ increases by 1.
- When at least one "O" point falls into $\mathcal{C}_4$ (between $\{\times,\times\}$), $|\mathcal{L}_1|$ increases by 1.

We denote the number of bins in $\mathcal{C}_i$, $i = 1,2,3,4$ that contain at least one "O" point as n_1, n_2, n_3, n_4, respectively. As a result of above reasoning, in the event Ψ, we have

$$\begin{cases} n_1 + n_3 = l_2, \\ n_2 + n_4 = l_1, \\ n_1 + n_2 = g_0, \\ D - f - s - n_2 = g_1. \end{cases}$$

Solving the equations gives

$$\begin{cases} n_1 = g_0 - (D - f - s - g_1), \\ n_2 = D - f - s - g_1, \\ n_3 = l_2 - g_0 + (D - f - s - g_1), \\ n_4 = l_1 - (D - f - s - g_1). \end{cases}$$

Note that $\sum_{i=1}^4 n_i = l_1 + l_2$. Therefore, event Ψ is equivalent to randomly picking n_1, n_2, n_3 and n_4 bins in $\mathcal{C}_1$,...,$\mathcal{C}_4$, and then allocate a type "O" points in these $(l_1 + l_2)$ bins such that each bin contains at least one "O". Hence, we obtain

$$\begin{aligned} P\Big\{|\mathcal{L}_1| = l_1, |\mathcal{L}_2| = l_2, |\mathcal{G}_0| = g_0, |\mathcal{G}_1| = g_1 \Big| |\mathcal{C}_1| = s\Big\} &= \frac{\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{l_1+l_2-1}}{\binom{D-1}{D-a-1}} \\ &= \frac{\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{a-l_1-l_2}}{\binom{D-1}{a}}, \end{aligned} \tag{25}$$

which is also a multi-variate hyper-geometric distribution. Now combining (23), (24) and (25), we obtain the joint distribution of $|\mathcal{L}_0|, |\mathcal{L}_1|, |\mathcal{L}_2|, |\mathcal{G}_0|$ and $|\mathcal{G}_1|$ as

$$\begin{aligned} &P\Big\{|\mathcal{L}_1| = l_1, |\mathcal{L}_2| = l_2, |\mathcal{G}_0| = g_0, |\mathcal{G}_1| = g_1\Big\} \\ &= \sum_{s=\max(0,D-2f+a)}^{D-f-1} \frac{\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{a-l_1-l_2}}{\binom{D-1}{a}} \cdot \frac{\binom{D-f}{s}\binom{f-a-1}{D-f-s-1}}{\binom{D-a-1}{D-f-1}}. \end{aligned} \tag{26}$$

Now let Ξ be the feasible set of $(l_0, l_1, g_0, g_1, g_2)$ that satisfies the intrinsic constraints (11). The desired expectation w.r.t. both π and σ can thus be written as

$$\begin{aligned} \tilde{\mathcal{E}} = \sum_{\Xi} &\left(\frac{l_0}{f + g_0 + g_1} + \frac{a(g_0 + l_2)}{(f + g_0 + g_1)f}\right) \cdot \\ &\left(\sum_{s=\max(0,D-2f+a)}^{D-f-1} \frac{\binom{s}{n_1}\binom{D-f-s}{n_2}\binom{D-f-s}{n_3}\binom{f-a-(D-f-s)}{n_4}\binom{a-1}{a-l_1-l_2}}{\binom{D-1}{a}} \cdot \frac{\binom{D-f}{s}\binom{f-a-1}{D-f-s-1}}{\binom{D-a-1}{D-f-1}}\right). \end{aligned}$$

The desired result then follows from (21). □

A.4. Proof of Proposition 3.2

Proposition 3.2 (Symmetry). $Var[\hat{J}_{\sigma,\pi}]$ is the same for the (D, f, a)-data pair and the $(D, f, f - a)$-data pair, for $\forall 0 \leq a \leq f \leq D$.

Proof. For fixed a, f, D, let $\tilde{\mathcal{E}}_1$ be the expectation defined in Theorem 3.1 for $(\boldsymbol{v}_1, \boldsymbol{w}_1)$, and $\tilde{\mathcal{E}}_2$ be that for $(\boldsymbol{v}_2, \boldsymbol{w}_2)$. From Theorem 3.1 we know that

$$\tilde{\mathcal{E}}_1 = \mathbb{E}_{(l_0,l_2,g_0,g_1)}\Big[\frac{l_0}{f + g_0 + g_1} + \frac{a(g_0 + l_2)}{(f + g_0 + g_1)f}\Big],$$

where (l_0, l_2, g_0, g_1) follows the distribution of $(|\mathcal{L}_0|, |\mathcal{L}_2|, |\mathcal{G}_0|, |\mathcal{G}_1|)$ associated with the location vector $\boldsymbol{x}_1$ of $(\boldsymbol{v}_1, \boldsymbol{v}_2)$. For data pair $(\boldsymbol{v}_2, \boldsymbol{w}_2)$, we can consider its location vector $\boldsymbol{x}_2$ as swapping the "O" and "$\times$" entries of $\boldsymbol{x}_1$. Now we denote the size of the corresponding sets (Definition 2.2) of $\boldsymbol{x}_2$ as $l_i's, g_i's, h_i's$, for $i = 0, 1, 2$. Since σ is applied before hashing, by symmetry there is a one-to-one correspondence between the two location vectors. More specifically, l_0' corresponds to h_1, g_0' corresponds to g_1, g_1' corresponds to g_0, and l_2' corresponds to h_2. Therefore, in probability we can write

$$\begin{aligned}\tilde{\mathcal{E}}_2 &= \mathbb{E}_{(l_0',l_2',g_0',g_1')}\Big[\frac{l_0'}{f+g_0'+g_1'} + \frac{a(g_0'+l_2')}{(f+g_0'+g_1')f}\Big] \\ &= \mathbb{E}_{(h_1,h_2,g_0,g_1)}\Big[\frac{h_1}{f+g_0+g_1} + \frac{(f-a)(g_1+h_2)}{(f+g_0+g_1)f}\Big].\end{aligned}$$

Consequently, we have

$$\tilde{\mathcal{E}}_1 - \tilde{\mathcal{E}}_2 = \mathbb{E}_{(l_0,l_2,h_1,h_2,g_0,g_1)}\Big[\frac{l_0-h_1}{f+g_0+g_1} + \frac{a(g_0+l_2)-(f-a)(g_1+h_2)}{(f+g_0+g_1)f}\Big].$$

In the sequel, the subscript of expectation is suppressed for conciseness. Exploiting the constraints (11), we deduce that $h_1 = (f-a) - l_1 - g_1$, $h_2 = l_0 + g_0 + l_1 + g_1 - a$ and $l_0 + l_1 = a - l_2$. Using these facts we obtain

$$\begin{aligned}\tilde{\mathcal{E}}_1 - \tilde{\mathcal{E}}_2 &= \mathbb{E}\Big[\frac{(l_0-(f-a)+l_1+g_1)f + a(g_0+l_2) - (f-a)(l_0+g_0+l_1+2g_1-a)}{(f+g_0+g_1)f}\Big] \\ &= \mathbb{E}\Big[\frac{(2a-f+g_1-l_2)f + a(g_0+l_2) - (f-a)(2g_1+g_0-l_2)}{(f+g_0+g_1)f}\Big] \\ &= \mathbb{E}\Big[\frac{2(f+g_0+g_1)a - (f+g_0+g_1)f}{(f+g_0+g_1)f}\Big] \\ &= 2J-1.\end{aligned}$$

Comparing the variances of $\hat{J}_{\sigma,\pi}(\boldsymbol{v}_1,\boldsymbol{w}_1)$ and $\hat{J}_{\sigma,\pi}(\boldsymbol{v}_2,\boldsymbol{w}_2)$, we derive

$$\begin{aligned}&Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v}_1,\boldsymbol{w}_1)] - Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v}_2,\boldsymbol{w}_2)] \\ &= (\frac{J}{K} + \frac{(K-1)\tilde{\mathcal{E}}_1}{K} - J^2) - (\frac{1-J}{K} + \frac{(K-1)\tilde{\mathcal{E}}_2}{K} - (1-J)^2) \\ &= -\frac{K-1}{K}(2J-1) + \frac{K-1}{K}(\tilde{\mathcal{E}}_1 - \tilde{\mathcal{E}}_2) = 0.\end{aligned}$$

This completes the proof. □

A.5. Proof of Lemma 3.3

Lemma 3.3 (Strict Increment). Assume $a > 0$ and $f > a$ are arbitrary and fixed. Denote $\tilde{\mathcal{E}}_D$ as in (20) in Theorem 3.1, with D treated as a parameter. Then we have $\tilde{\mathcal{E}}_{D+1} > \tilde{\mathcal{E}}_D$ for $\forall D \geq f$.

Proof. The lemma basically says that $\tilde{\mathcal{E}}$ is monotonically increasing when we append more "−" entries to the data vector. Let the probability mass function (26) parameterized by a, f and dimensionality D be $P_{a,f,D}(l_0,l_2,g_0,g_1)$. Conditional on l_0, l_2, g_0, g_1 with D elements, there are several cases for the possible values l_0', l_2', g_0', g_1' when adding a "−":

- $g_0' = g_0 + 1, l_0' = l_0, l_2' = l_2, g_1' = g_1$. This is true when the new elements falls between a pair of $(\times, O)$, with probability $\frac{l_1+l_2-g_0}{D}$.
- $g_1' = g_1 + 1, l_0' = l_0, l_2' = l_2, g_0' = g_0$, when the new elements falls between a pair of $(\times, \times)$, with probability $\frac{f-a-l_1-g_1}{D}$.
- $g_1' = g_1 + 1, l_2' = l_2 + 1, l_0' = l_0, g_0' = g_0$, when the new elements falls between a pair of $(O, \times)$, with probability $\frac{l_1}{D}$.
- $l_0' = l_0 - 1, l_2' = l_2 + 1, g_0' = g_0 + 1, g_1' = g_1$, when the new elements falls between a pair of (O, O). The probability of this event is $\frac{l_0}{D}$.
- All values unchanged, when the "−" falls between other types of pairs, with probability $\frac{D-f+g_0+g_1}{D}$.

Denote Ξ_D as the feasible set satisfying (11) with dimension $D \geq f$. Above reasoning builds a correspondence between Ξ_D and Ξ_{D+1}. More precisely, we have

$$\begin{aligned}
\tilde{\mathcal{E}}_{D+1} &= \sum_{\Xi_{D+1}} \left(\frac{l_0'}{f+g_0'+g_1'} + \frac{a(g_0'+l_2')}{(f+g_0'+g_1')f} \right) P_{a,f,D+1}(l_0', l_2', g_0', g_1') \\
&= \sum_{\Xi_D} \Big\{ \Big(\frac{l_0}{f+g_0+g_1+1} + \frac{a(g_0+l_2+1)}{(f+g_0+g_1+1)f} \Big) \frac{l_1+l_2-g_0}{D} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&\quad + \Big(\frac{l_0}{f+g_0+g_1+1} + \frac{a(g_0+l_2)}{(f+g_0+g_1+1)f} \Big) \frac{f-a-l_1-g_1}{D} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&\quad + \Big(\frac{l_0}{f+g_0+g_1+1} + \frac{a(g_0+l_2+1)}{(f+g_0+g_1+1)f} \Big) \frac{l_1}{D} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&\quad + \Big(\frac{l_0-1}{f+g_0+g_1+1} + \frac{a(g_0+l_2+2)}{(f+g_0+g_1+1)f} \Big) \frac{l_0}{D} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&\quad + \Big(\frac{l_0}{f+g_0+g_1} + \frac{a(g_0+l_2)}{(f+g_0+g_1)f} \Big) \frac{D-f+g_0+g_1}{D} P_{a,f,D}(l_0,l_2,g_0,g_1) \Big\}.
\end{aligned}$$

Therefore, the increment can be computed as

$$\begin{aligned}
\tilde{\delta}_D &\triangleq \tilde{\mathcal{E}}_{D+1} - \tilde{\mathcal{E}}_D \\
&= \sum_{\Xi_D} \Big\{ \frac{f-g_0-g_1}{D} \Big[\big(\frac{l_0}{f+g_0+g_1+1} - \frac{l_0}{f+g_0+g_1} \big) + \big(\frac{a(g_0+l_2+1)}{f+g_0+g_1+1} - \frac{a(g_0+l_2)}{f+g_0+g_1} \big) \Big] \\
&\qquad - \frac{l_0}{D(f+g_0+g_1+1)} - \frac{a(f-a-l_1-g_1)-al_0}{Df(f+g_0+g_1+1)} \Big\} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&= \sum_{\Xi_D} \Big\{ \frac{(f-g_0-g_1)[a(f+g_1-l_2)-fl_0]}{Df(f+g_0+g_1)(f+g_0+g_1+1)} - \frac{(f-a)l_0 + a(f-a-l_1-g_1)}{Df(f+g_0+g_1+1)} \Big\} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&= \sum_{\Xi_D} \frac{2af(l_1+g_1) - 2f(f-a)l_0 - 2a(f-a)(g_0+g_1)}{Df(f+g_0+g_1)(f+g_0+g_1+1)} P_{a,f,D}(l_0,l_2,g_0,g_1) \\
&= \mathbb{E}\Big[\frac{2af(l_1+g_1) - 2f(f-a)l_0 - 2a(f-a)(g_0+g_1)}{Df(f+g_0+g_1)(f+g_0+g_1+1)} \Big] \\
&= \mathbb{E}\Big[\frac{2af(f-a-h_1) - 2f(f-a)l_0 - 2a(f-a)(g_0+g_1+f-f)}{Df(f+g_0+g_1)(f+g_0+g_1+1)} \Big] \\
&= \mathbb{E}\Big[\frac{4a(f-a)}{D(f+g_0+g_1)(f+g_0+g_1+1)} \Big] - \mathbb{E}\Big[\frac{2ah_1 + 2(f-a)l_0}{D(f+g_0+g_1)(f+g_0+g_1+1)} \Big] - \mathbb{E}\Big[\frac{2a(f-a)}{Df(f+g_0+g_1+1)} \Big] \\
&\triangleq 4a(f-a)E_0 - 2aE_1 - 2(f-a)E_2 - 2a(f-a)E_3,
\end{aligned} \tag{27}$$

where

$$\begin{aligned}
&E_0 = \mathbb{E}\Big[\frac{1}{D(f+g_0+g_1)(f+g_0+g_1+1)} \Big], \ \ E_1 = \mathbb{E}\Big[\frac{h_1}{D(f+g_0+g_1)(f+g_0+g_1+1)} \Big], \\
&E_2 = \mathbb{E}\Big[\frac{l_0}{D(f+g_0+g_1)(f+g_0+g_1+1)} \Big], \ \ E_3 = \mathbb{E}\Big[\frac{g_2}{Df(f+g_0+g_1+1)} \Big].
\end{aligned}$$

Note that here the expectations are taken w.r.t. the set size distribution under (a, f, D). We can expand the terms of density function (26) to derive

$$\begin{aligned}
&P_{a,f,D}(l_0,l_2,g_0,g_1)\\
&=\sum_{s=\max(0,D-2f+a)}^{D-f-1}\frac{(D-f-s)(D-f)!(f-a-1)!}{[D-(f+g_0+g_1)]![(f+g_0+g_1)-D+s]!g_1!(D-f-s-g_1)!}\\
&\frac{(a-1)!}{(g_0+g_1-l_2)![D-s+l_2-(f+g_0+g_1)]!(f-a-l_1-g_1)!(f+g_1+l_1-D+s)!l_0!(a-l_0-1)!}\\
&\frac{a!(f-a)!(D-f-1)!}{(D-1)!}.
\end{aligned}$$

Denote $a'=a-1$, $f'=f-1$, $D'=D-1$ and $l_0'=l_0-1$. We have

$$\begin{aligned}
E_2&=\sum_{\Xi_D}\frac{l_0}{D(f+g_0+g_1)(f+g_0+g_1+1)}P_{a,f,D}(l_0,l_2,g_0,g_1)\\
&=\sum_{\Xi_D}\frac{a(a-1)}{D-1}\cdot\frac{1}{D(f+g_0+g_1)(f+g_0+g_1+1)}\\
&\qquad\sum_{s=\max(0,D'-2f'+a')}^{D'-f'-1}\frac{(D'-f'-s)(D'-f')!(f'-a'-1)!}{[D'-(f'+g_0+g_1)]![(f'+g_0+g_1)-D'+s]!g_1!(D'-f'-s-g_1)!}\\
&\frac{(a'-1)!}{(g_0+g_1-l_2)![D'-s+l_2-(f'+g_0+g_1)]!(f'-a'-l_1-g_1)!(f'+g_1+l_1-D'+s)!l_0'!(a'-l_0'-1)!}\\
&\frac{a'!(f'-a')!(D'-f'-1)!}{(D'-1)!}\\
&=\sum_{\Xi_{D-1}}\frac{a(a-1)}{D-1}\frac{1}{D(f+g_0+g_1)(f+g_0+g_1+1)}P_{a-1,f-1,D-1}(l_0,l_2,g_0,g_1)\\
&=\frac{a(a-1)}{D-1}\mathbb{E}_{a-1,f-1,D-1}\Big[\frac{1}{D(f+g_0+g_1)(f+g_0+g_1+1)}\Big]\\
&\triangleq\frac{a(a-1)}{D-1}\bar{E}.
\end{aligned}$$

Here, the subscript means that we are taking expectation w.r.t the set sizes when the numbers of "O", "$\times$" and "$-$" points are $(a-1,f-1,D-1)$. By symmetry, it can be shown similarly that

$$E_1=\frac{(f-a)(f-a-1)}{D-1}\mathbb{E}_{a,f-1,D-1}\Big[\frac{1}{D(f+g_0+g_1)(f+g_0+g_1+1)}\Big]=\frac{(f-a)(f-a-1)}{D-1}\bar{E}.$$

Substituting above results into (27), we obtain

$$\tilde{\delta}_D=2a(f-a)[2E_0-\frac{f-2}{D-1}\bar{E}-E_3].$$

To compute E_0, note that with a,f and D fixed, variable g_2 is distributed as hyper$(D-1,D-f,D-f-1)$. For $\bar{E}$, the distribution becomes hyper$(D-2,D-f,D-f-1)$. Since $f+g_0+g_1=D-g_2$, we know that

$$\begin{aligned}
E_0&=\sum_{s=\max(0,D-2f)}^{D-f-1}\frac{1}{D(D-s)(D-s+1)}\frac{\binom{D-f-1}{s}\binom{f}{D-f-s}}{\binom{D-1}{D-f}}\\
&=\sum_{s=\max(0,D-2f)}^{D-f-1}\frac{1}{D(D-s)(D-s+1)}\frac{(D-f-1)!f!}{s!(D-f-s-1)!(D-f-s)!(-D+2f+s)!}\frac{(D-f)!(f-1)!}{(D-1)!},
\end{aligned}$$

and

$$\bar{E} = \sum_{s=\max(0,D-2f+1)}^{D-f-1} \frac{1}{D(D-s)(D-s+1)} \frac{\binom{D-f-1}{s}\binom{f-1}{D-f-s}}{\binom{D-2}{D-f}}$$
$$= \sum_{s=\max(0,D-2f+1)}^{D-f-1} \frac{1}{D(D-s)(D-s+1)} \frac{(D-f)!(f-2)!}{(D-2)!} \cdot \frac{(D-f-1)!(f-1)!}{s!(D-f-s-1)!(D-f-s)!(-D+2f+s-1)!}.$$

For $\forall D \geq f$, we have

$$\frac{f-2}{D-1}\bar{E} \leq \sum_{s=\max(0,D-2f)}^{D-f-1} \frac{(f-2)(D-1)(-D+2f+s)}{D(D-1)f(f-1)(D-s)(D-s+1)} \frac{(D-f-1)!f!}{s!(D-f-s-1)!(D-f-s)!(-D+2f+s)!} \frac{(D-f)!(f-1)!}{(D-1)!}$$
$$\leq \mathbb{E}\Big[\frac{(f-2)(f-(D-f-g_2))}{Df(f-1)(D-g_2)(D-g_2+1)}\Big]$$
$$< \mathbb{E}\Big[\frac{(f-g_0-g_1)}{Df(f+g_0+g_1)(f+g_0+g_1+1)}\Big].$$

Consequently, we have

$$\tilde{\delta}_D > 2a(f-a)\mathbb{E}\Big[\frac{2}{D(f+g_0+g_1)(f+g_0+g_1+1)} - \frac{f-g_0-g_1}{Df(f+g_0+g_1)(f+g_0+g_1+1)} - \frac{f+g_0+g_1}{Df(f+g_0+g_1)(f+g_0+g_1+1)}\Big]$$
$$= 0,$$

and note that this holds for $\forall D \geq K$. The proof is now complete. □

A.6. Proof of Theorem 3.4

Theorem 3.4 (Uniform Superiority). For any two binary vectors $\boldsymbol{v}, \boldsymbol{w} \in \{0,1\}^D$ with $J \neq 0$ or 1, it holds that $Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v},\boldsymbol{w})] < Var[\hat{J}_{MH}(\boldsymbol{v},\boldsymbol{w})]$.

Proof. By assumption we have $0 < a < f$. To compare $Var[\hat{J}_{\sigma,\pi}]$ with $Var[\hat{J}_{MH}] = \frac{J(1-J)}{K} = \frac{J}{K} + \frac{(K-1)J^2}{K} - J^2$, it suffices to compare $\tilde{\mathcal{E}}$ with J^2. When $D = f$, we know that the location vector $\boldsymbol{x}$ of $(\boldsymbol{v},\boldsymbol{w})$ contains no "$-$" elements. It is easy to verify that in this case, $|\mathcal{G}_0| = |\mathcal{G}_1| = |\mathcal{L}_2| = 0$, and $|\mathcal{L}_0|$ follows hyper$(f-1, a, a-1)$. By Theorem 3.1, it follows that when $D = f$,

$$\tilde{\mathcal{E}}_D = \frac{1}{f}\mathbb{E}[|\mathcal{L}_0|] = \frac{a(a-1)}{f(f-1)} = J\tilde{J} < J^2.$$

Recall the definition $\tilde{J} = \frac{a-1}{f-1}$, which is always smaller than J. On the other hand, as $D \to \infty$, we have $|\mathcal{L}_0| \to 0$, $|\mathcal{L}_2| \to a$, $|\mathcal{G}_0| \to a$ and $|\mathcal{G}_1| \to f - a$. We can show that

$$\tilde{\mathcal{E}}_D \to J^2, \quad \text{as } D \to \infty.$$

By Lemma 3.3, the sequence $(\tilde{\mathcal{E}}_f, \tilde{\mathcal{E}}_{f+1}, \tilde{\mathcal{E}}_{f+2}, ...)$ is strictly increasing. Since it is convergent with limit J^2, by the Monotone Convergence Theorem we know that $\tilde{\mathcal{E}_D} < J^2, \forall D \geq f$. □

A.7. Proof of Proposition 3.6

Proposition 3.5 (Consistent Improvement). Suppose f is fixed. In terms of a, the variance ratio $\rho(a) = \frac{Var[\hat{J}_{MH}(\boldsymbol{v},\boldsymbol{w})]}{Var[\hat{J}_{\sigma,\pi}(\boldsymbol{v},\boldsymbol{w})]}$ is constant for any $0 < a < f$.

Proof. Let $\tilde{\mathcal{E}}$ be defined as in Theorem 3.1. Assume that D and f are fixed and a is variable. Firstly, we can write the variance ratio explicitly as

$$\rho(a) = \frac{\frac{J-J^2}{K}}{\frac{J}{K} + \frac{(K+1)\tilde{\mathcal{E}}}{K} - J^2} = \frac{1-J}{1-J-(K-1)(J-\frac{\tilde{\mathcal{E}}}{J})}.$$

We now show that the term $J - \frac{\tilde{\mathcal{E}}}{J} = C(1-J)$, where C is some constant independent of J (i.e., a). Then, for fixed D and f, by cancellation $\rho(a)$ would be constant for all $0 < a < f$. We have

$$\begin{aligned} J - \frac{\tilde{\mathcal{E}}}{J} &= \frac{a}{f} - \mathbb{E}_{a,f,D}\Big[\frac{fl_0}{a(f+g_0+g_1)} + \frac{g_0+l_2}{f+g_0+g_1}\Big] \\ &= \mathbb{E}\Big[\frac{a^2(f+g_0+g_1) - f^2 l_0 - af(g_0+l_2)}{af(f+g_0+g_1)}\Big] \\ &= \mathbb{E}\Big[\frac{a(a-f)(g_0+g_1) + a^2 f + afg_1 - f^2 l_0 - afl_2}{af(f+g_0+g_1)}\Big] \\ &= \mathbb{E}\Big[\frac{a(a-f)(g_0+g_1) + af(l_0+l_1) + afg_1 - f^2 l_0}{af(f+g_0+g_1)}\Big] \\ &= \mathbb{E}\Big[\frac{a(a-f)(g_0+g_1) + f(a-f)l_0 + af(f-a-h_1)}{af(f+g_0+g_1)}\Big], \end{aligned} \tag{28}$$

where we use the constraints (11) that $l_0 + l_1 + l_2 = a$ and $l_1 + g_1 + h_1 = f - a$. We now study the three terms respectively. We have

$$\mathbb{E}\Big[\frac{a(a-f)(g_0+g_1)}{af(f+g_0+g_1)}\Big] = -(1-J)\mathbb{E}\Big[\frac{g_0+g_1}{f+g_0+g_1}\Big] \triangleq -E'(1-J).$$

We have shown in the proof of Lemma 3.3 that

$$\mathbb{E}_{a,f,D}\Big[\frac{l_0}{f+g_0+g_1}\Big] = \frac{a(a-1)}{D-1}\mathbb{E}_{a-1,f-1,D-1}\Big[\frac{1}{f+g_0+g_1}\Big] \triangleq \frac{a(a-1)}{D-1}E^*,$$

and by symmetry it holds that

$$\mathbb{E}_{a,f,D}\Big[\frac{h_1}{f+g_0+g_1}\Big] = \frac{(f-a)(f-a-1)}{D-1}E^*.$$

Since f is fixed, $(|\mathcal{G}_0| + |\mathcal{G}_1|)$ is distributed independently of a. Consequently, E' and E^* are both independent of a. Thus, it follows that

$$\mathbb{E}\Big[\frac{f(a-f)l_0}{af(f+g_0+g_1)}\Big] = -(1-J)\frac{f(a-1)}{D-1}E^*,$$

additionally,

$$\mathbb{E}\Big[\frac{af(f-a-h_1)}{af(f+g_0+g_1)}\Big] = (1-J)fE^* - (1-J)\frac{f(f-a-1)}{D-1}E^*.$$

Summing up the terms and substituting into (28), we derive

$$J - \frac{\tilde{\mathcal{E}}}{J} = C(1-J),$$

where $C = -E' + (f - \frac{f(f-2)}{D-1})E^*$, which is independent of a. Taking into $\rho(a)$, we get

$$\rho(a) = \frac{1-J}{1-J-(K-1)C(1-J)} = \frac{1}{1-(K-1)C},$$

which is a constant only depending on f, D and K. This completes the proof. □

B. More Numerical Justification on C-MinHash-(π, π)

The "Words" dataset (Li and Church, 2005) (which is publicly available) contains a large number of word vectors, with the i-th entry indicating whether this word appears in the i-th document, for a total of $D = 2^{16}$ documents. The key statistics of the 120 selected word pairs are presented in Table 1. Those 120 pairs of words are more or less randomly selected except that we make sure they cover a wide spectrum of data distributions. Denote d as the number of non-zero entries in the vector. Table 1 reports the density $\tilde{d} = d/D$ for each word vector, ranging from 0.0006 to 0.6. The Jaccard similarity J ranges from 0.002 to 0.95.

In Figures 8 - 15, we plot the empirical MSE along with the empirical bias^2 for $\hat{J}_{\pi,\pi}$, as well as the empirical MSE for $\hat{J}_{\sigma,\pi}$. Note that for D this large, it is numerically difficult to evaluate the theoretical variance formulas. From the results in the Figures, we can observe

- For all the data pairs, the MSE of C-MinHash-(π, π) estimator overlaps with the empirical MSE of C-MinHash-(σ, π) estimator for all K from 1 up to 4096.
- The bias^2 is several orders of magnitudes smaller than the MSE, in all data pairs. This verifies that the bias of $\hat{J}_{\pi,\pi}$ is extremely small in practice and can be safely neglected.

We have many more plots on more data pairs. Nevertheless, we believe the current set of experiments on this "Words" dataset should be sufficient to verify that, the proposed C-MinHash-(π, π) could give indistinguishable Jaccard estimation accuracy in practice compared with C-MinHash-(σ, π).

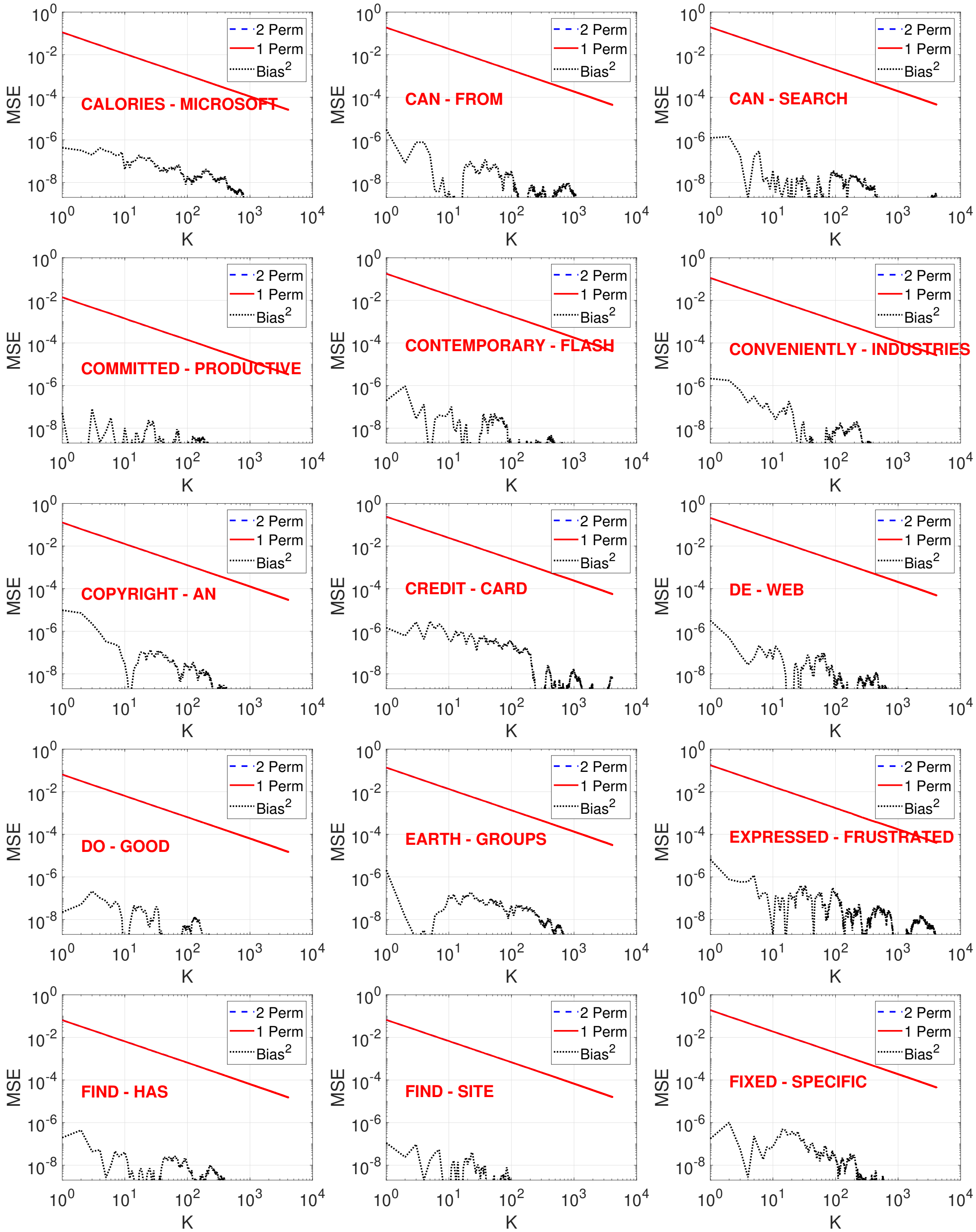

Figure 9. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

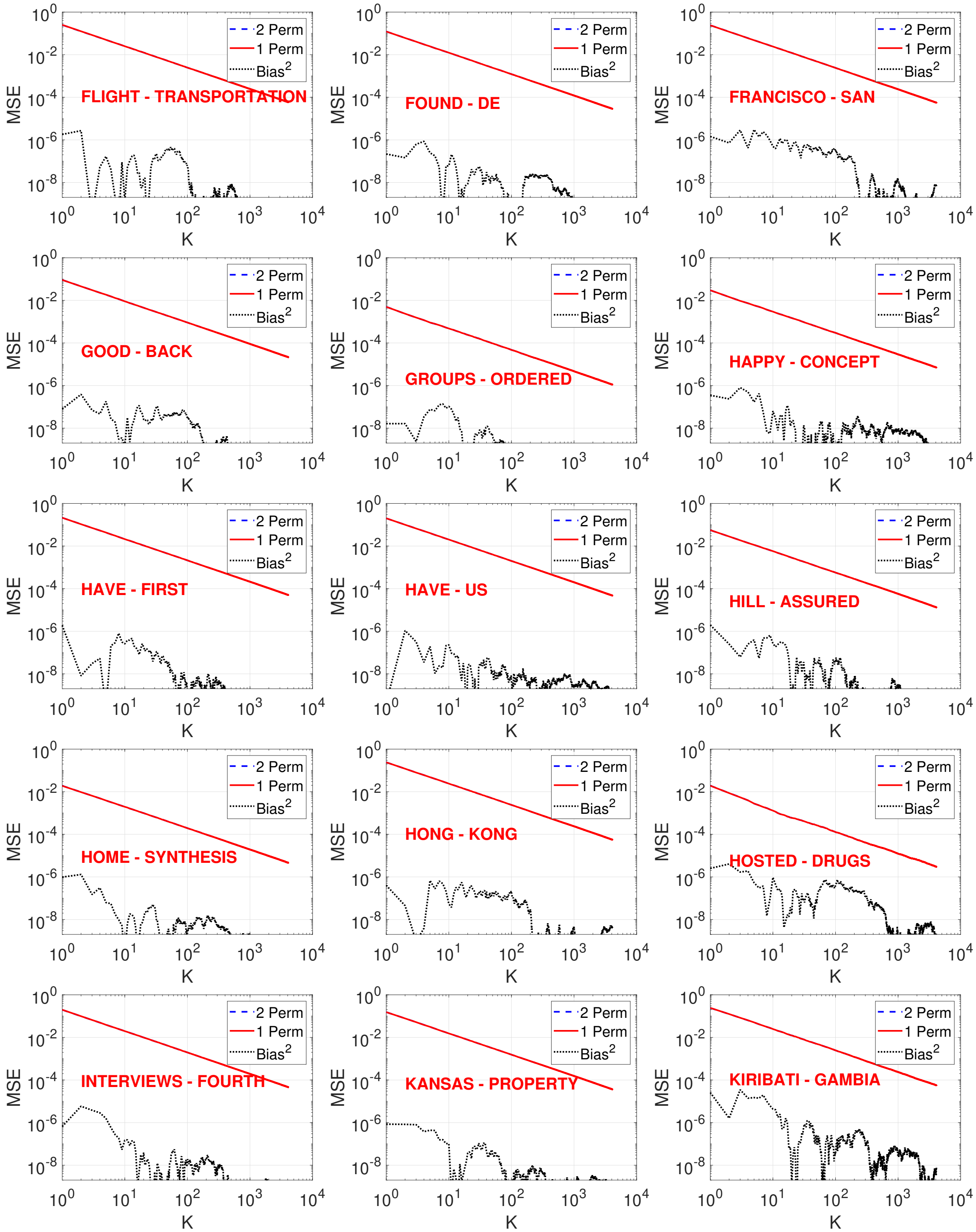

Figure 10. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

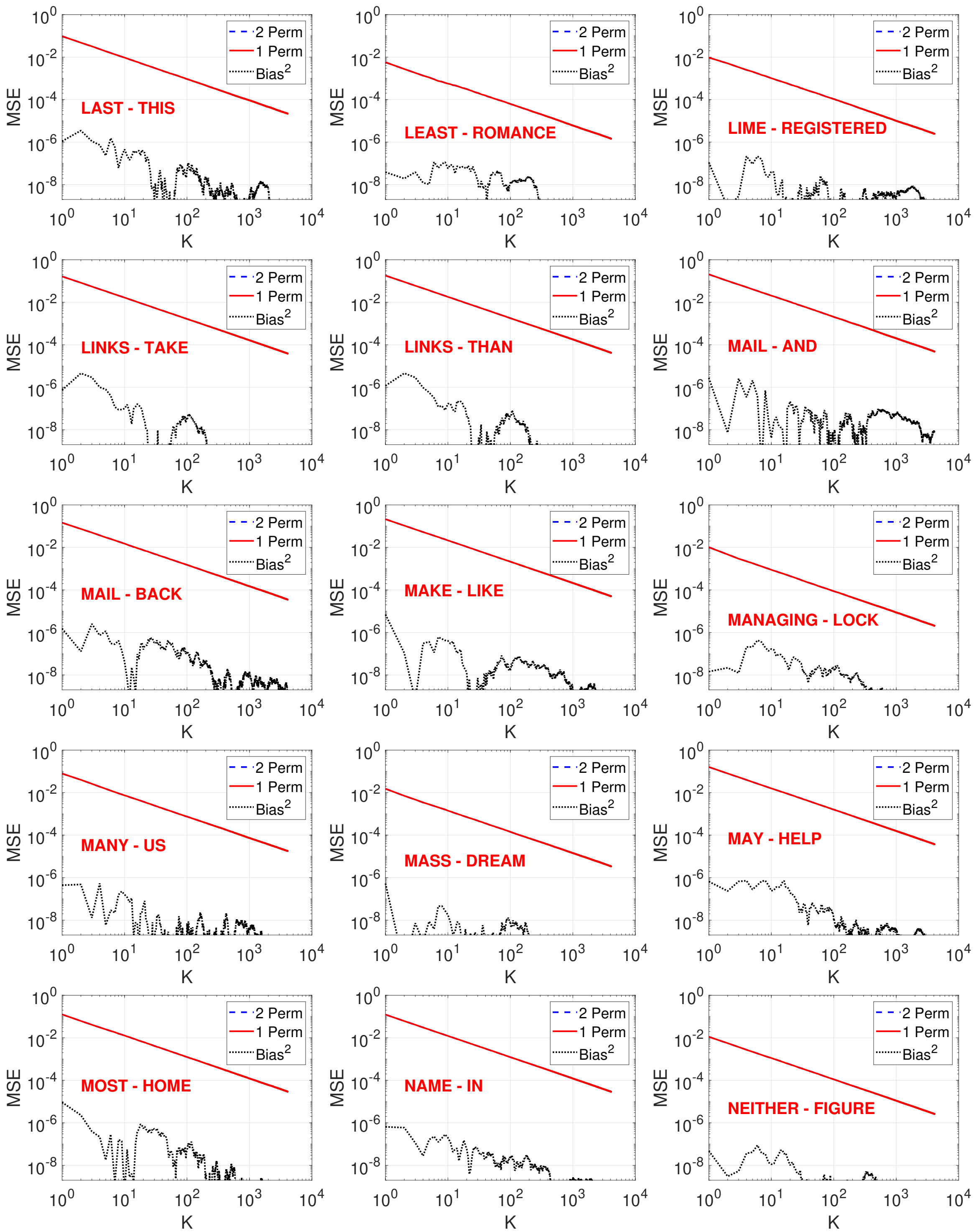

Figure 11. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

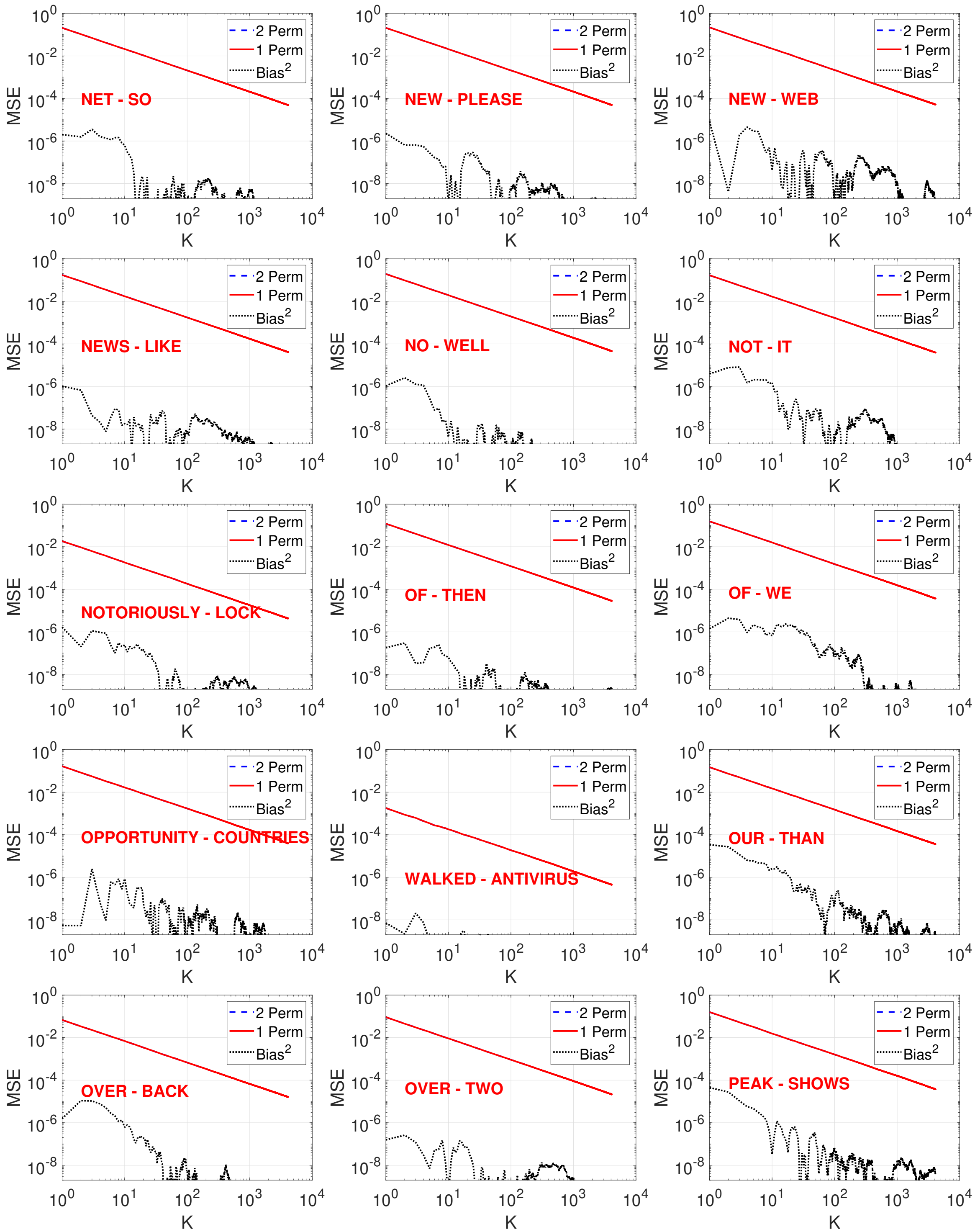

Figure 12. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

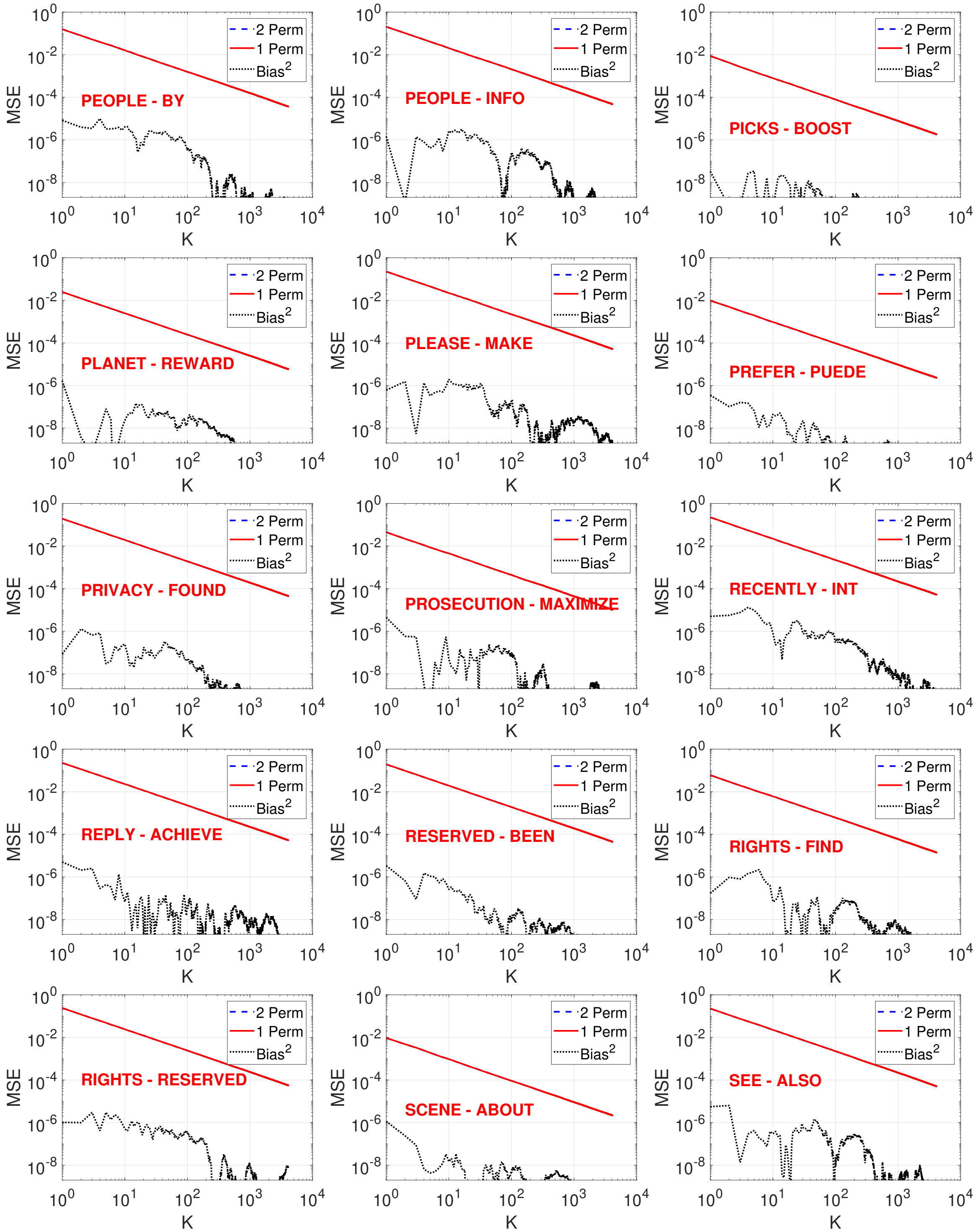

Figure 13. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

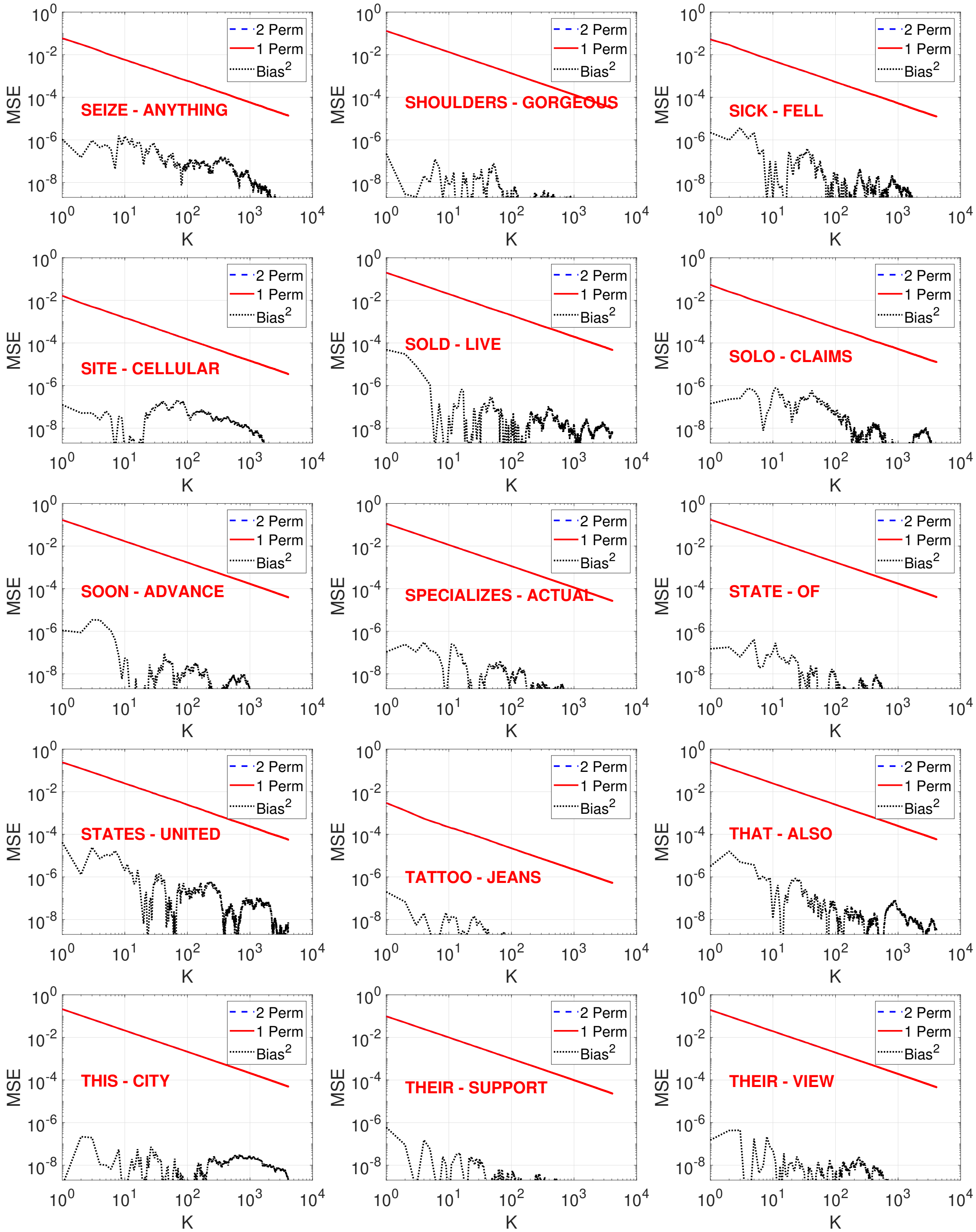

Figure 14. Empirical MSEs of C-MinHash-(π, π) ("1 Perm", red, solid) vs. C-MinHash-(σ, π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π, π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.

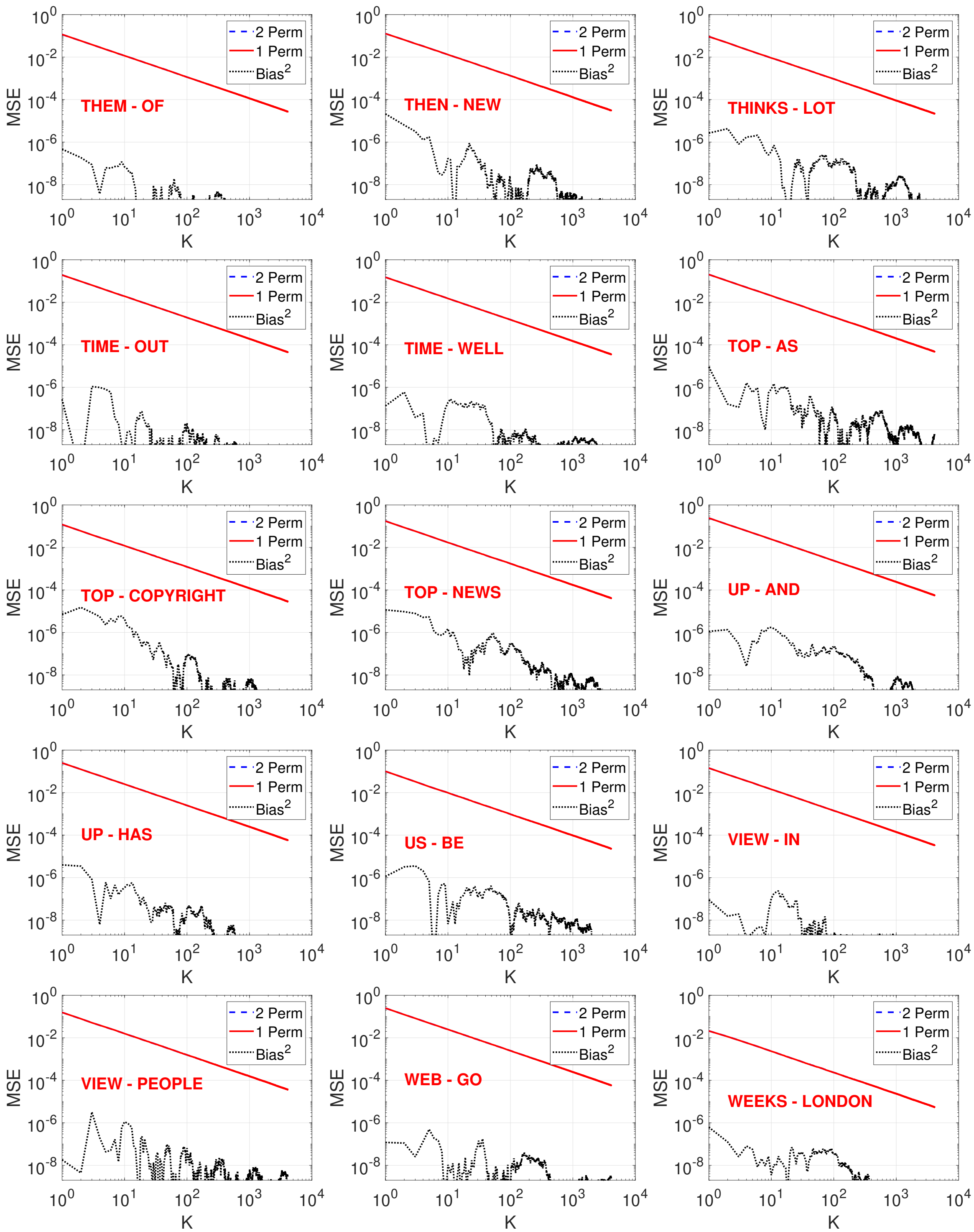

Figure 15. Empirical MSEs of C-MinHash-(π,π) ("1 Perm", red, solid) vs. C-MinHash-(σ,π) ("2 Perm", blue, dashed) on various data pairs from the *Words* dataset. We also report the empirical bias^2 for C-MinHash-(π,π) to show that the bias is so small that it can be safely neglected. The empirical MSE curves for both estimators essentially overlap for all data pairs.