APPENDIX

A Additional Motivation Examples

Example 3: *On platforms like Twitter, hashtags appear as a rapid, high-volume data stream. Automated or troll-bot hashtags can generate millions of occurrences per hour, while rare but potentially critical hashtags may surface only a few thousand times. In a CM, a small additive error can severely distort counts for the rarer tags. Although this may seem like a "free boost", the resulting collision-induced overcounts undermine reliability, making it hard to determine whether a rare hashtag is truly trending or merely inflated by overlap with high-volume tags. As a result, trend detection, resource allocation, and moderation decisions such as filtering become noisier—false signals may emerge while genuinely urgent topics are delayed or misclassified.*

Example 4: *In network monitoring, consider two groups of IPs: malicious and benign IPs. While benign IP addresses (e.g., Google DNS) generate massive traffic, the malicious IPs might only appear sporadically. A CM with a small additive error has a negligible impact on the counts of the benign group. This error, on the other hand, can artificially inflate the counts for the malicious IPs, hiding meaningful anomalies. This makes it harder to detect subtle but critical security threats.*

Example 5: *In online advertising, platforms track how often each ad or user click event occurs. Popular ads may generate millions of impressions, while niche ads may only get a few hundred. With CM, small additive errors hardly affect big advertisers but can completely distort the performance metrics for small advertisers — for example, inflating a niche ad's 50 clicks into 500 due to collisions. This creates unfair reporting, potentially wasting budgets or hiding click fraud.*

B Price of Fairness: Randomness vs. Uniformity

In Section 5, we observed a counterintuitive result for $d = 1$, proving a negative price of fairness (PoF) for FCM, compared to a regular CM sketch that uses a random hash function. This result can be explained by the fact that a random hash function is unlikely to *uniformly* distribute the elements to the buckets (an interesting related topic is *the occupant problem* [59]). As a result, some of the buckets will have more than $\frac{n}{w}$ elements in them. For example, Figure 33 illustrates a random hashing of $n = 100$ element types into $w = 10$ buckets. While a uniform distribution would allocate 10 elements to each bucket, the random hashing allocated up to 18 elements in one bucket.

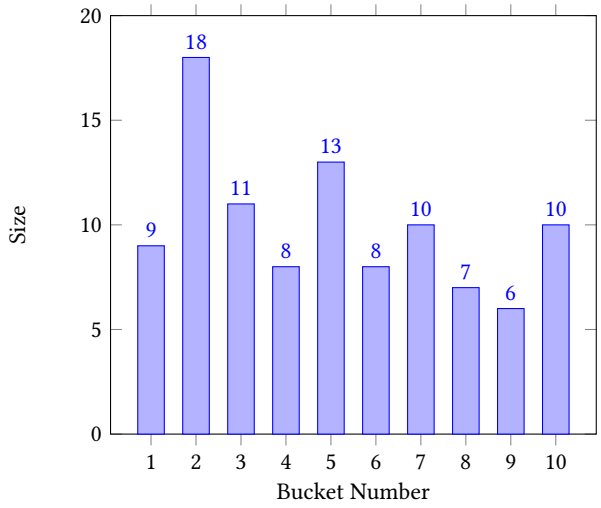

Figure 33: Illustration of a random distribution of $n = 100$ elements to $w = 10$ buckets.

Elements in large buckets will suffer from a large additive error, and there are a large number of them in such buckets. Hence, such buckets significantly increase the total additive error of the CM.

FCM increases the size-uniformity of the buckets by reserving $w_{\mathbf{g}} = \frac{n_{\mathbf{g}}}{n} w$ for each group $\mathbf{g}$. For example, Figure 34 illustrates the allocation of the 100 elements to the 10 buckets, by dividing (arbitrarily) the elements into two groups of size $n_{\mathbf{g}_1} = 50$, $n_{\mathbf{g}_2} = 50$. As a result, each group is allocated 5 buckets, and the distribution of the elements is more uniform, reducing the maximum size of the buckets (in this example) to 14. Increasing the uniformity (reducing the change of large-size buckets) helps to reduce the total additive error of FCM versus CM.

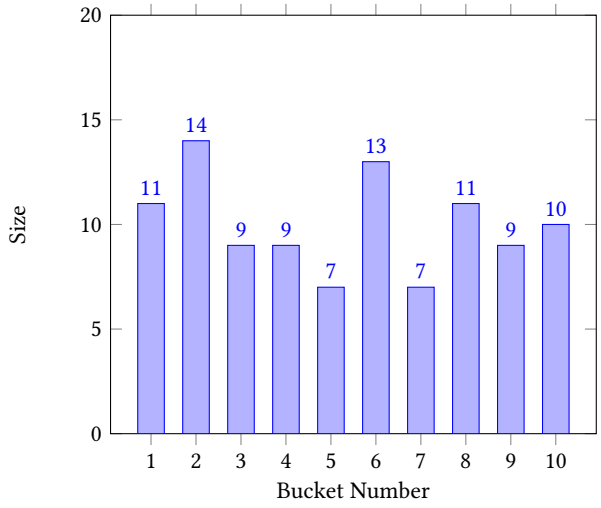

Figure 34: Illustration of a random distribution of $n = 100$ elements with two groups of sizes $n_{\mathbf{g}_1} = 50$ and $n_{\mathbf{g}_2} = 50$ to $w = 10$ buckets, based on FCM.

B.1 PoF for Uniform hashing ($d = 1$)

Using a hashing scheme (such as data-informed hashmaps [52]) that ensures uniformity by allocating exactly $\frac{n}{w}$ elements to each bucket can resolve the non-uniformity issue in the CM sketches. In the following, we compute the PoF of FCM for $d = 1$ when a uniform hashing scheme is used.

Since the hashing is uniform, the size of each bucket i in the CM sketch is

$$C_i = \frac{n}{w}$$

Therefore, each element e_j collides with exactly $\frac{n}{w} - 1$ other elements. In other words, the frequency of each element contributes to the additive error of exactly $\frac{n}{w} - 1$ other elements.

As a result, the total additive error can be computed as

$$\begin{aligned}\mathcal{L}_{CM} &= \sum_{j=1}^{n} \varepsilon_A(e_j) \\ &= \sum_{j=1}^{n} f(e_j)\left(\frac{n}{w} - 1\right) = \left(\frac{n}{w} - 1\right)\sum_{j=1}^{n} f(e_j) \\ &= N\left(\frac{n}{w} - 1\right) \end{aligned} \tag{16}$$

Now, let us compute $\mathcal{L}_{FCM}$, the total additive error for the regular fair-count-min sketch.

$$\mathcal{L}_{FCM} = \sum_{j=1}^{n} \varepsilon_A(e_j) = \sum_{e_j \in \mathrm{g}_1} \varepsilon_A(e_j) + \cdots + \sum_{e_j \in \mathrm{g}_\ell} \varepsilon_A(e_j)$$

Following the same calculation as in Equation 16, for every group $\mathrm{g} \in \mathcal{G}$,

$$\sum_{e_j \in \mathrm{g}} \varepsilon_A(e_j) = N_{\mathrm{g}}\left(\frac{n_{\mathrm{g}}}{w_{\mathrm{g}}} - 1\right)$$

Hence,

$$\mathcal{L}_{FCM} = \sum_{e_j \in \mathrm{g}_1} \varepsilon_A(e_j) + \cdots + \sum_{e_j \in \mathrm{g}_\ell} \varepsilon_A(e_j) = \sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}}\left(\frac{n_{\mathrm{g}}}{w_{\mathrm{g}}} - 1\right)$$

From Theorem 1, we know, $w_{\mathrm{g}} = \frac{n_{\mathrm{g}}}{n} w, \forall \mathrm{g} \in \mathcal{G}$, while $\sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}} = N$. Therefore,

$$\begin{aligned}\mathcal{L}_{FCM} &= \sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}}\left(\frac{n_{\mathrm{g}}}{w_{\mathrm{g}}} - 1\right) \\ &= \sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}}\left(\frac{n_{\mathrm{g}}}{w_{\mathrm{g}}}\right) - \sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}} = \sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}}\left(\frac{n_{\mathrm{g}}}{w_{\mathrm{g}}}\right) - N \\ &= \left(\frac{n}{w}\right)\sum_{\mathrm{g} \in \mathcal{G}} N_{\mathrm{g}} - N \qquad \texttt{//since } \frac{n_{\mathrm{g}}}{w_{\mathrm{g}}} = \frac{n}{w}, \forall \mathrm{g} \in \mathcal{G} \\ &= N\left(\frac{n}{w}\right) - N = N\left(\frac{n}{w} - 1\right) \end{aligned} \tag{17}$$

Finally, combining Equations 12, 16, and 17, we get,

$$PoF = \mathcal{L}_{FCM} - \mathcal{L}_{CM} = N\left(\frac{n}{w} - 1\right) - N\left(\frac{n}{w} - 1\right) = 0 \tag{18}$$

That is, the PoF is zero for $d = 1$, when a uniform hashing scheme is used.

B.2 PoF $d = 1$ vs. $d > 1$ on Synthetic

C Additional Experiment Results

In this section, we present comprehensive experimental results across five datasets: 1) GOOGLE NGRAM, 2) CENSUS, 3) NYC BUS, 4) DDOS, 5)REDDIT-TWITTER. We also provide the absolute values of the approximation factors and additive errors for each group to facilitate a clearer understanding of the trends in unfairness and the associated price of fairness under each setting.

Total Additive Error			n_{g_1} (n = 10,000)								
			9000	**8000**	**7000**	**6000**	**5000**	**4000**	**3000**	**2000**	**1000**
$d = 1$	theoretical	**CM**	19,047,019	28,054,816	37,081,001	46,092,289	55,121,050	64,147,002	73,186,882	82,323,243	91,265,426
		FCM	18,985,224	28,028,987	37,003,399	46,050,674	55,115,538	64,089,586	73,165,549	82,249,909	91,228,281
	empirical	**CM**	19,039,343	28,085,806	37,053,427	46,096,489	55,113,194	64,234,176	73,227,052	82,283,374	91,328,074
		FCM	18,991,509	27,982,573	37,002,246	46,071,394	55,086,890	64,133,815	73,168,189	82,230,649	91,276,271
		PoF	↓-47,834	↓-103,233	↓-51,181	↓-25,095	↓-26,304	↓-100,361	↓-58,863	↓-52,725	↓-51,803
$d = 5$	empirical	**CM**	7,964,348	11,572,548	16,458,026	22,023,181	28,305,699	34,442,308	40,959,703	47,230,290	54,257,770
		FCM	11,695,556	17,114,933	22,666,115	28,053,739	33,856,154	39,448,942	44,913,211	50,089,698	55,893,637
		PoF	↑3,731,208	↑5,542,385	↑6,208,089	↑6,030,558	↑5,550,455	↑5,006,634	↑3,953,508	↑2,859,408	↑1,635,867

Table 2: comparison of CM and FCM w.r.t additive errors and the price of fairness across different n_{g_1} values for $d = 1$ and $d = 5$.

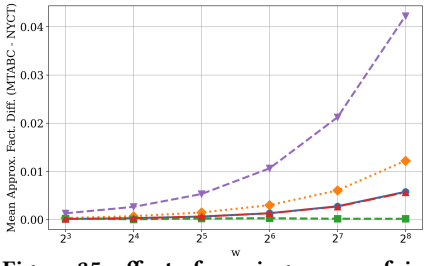

Figure 35: effect of varying w on unfairness, NYC BUS, $d = 5$.

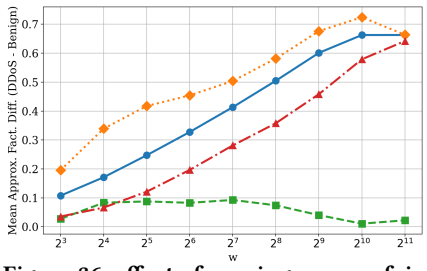

Figure 36: effect of varying w on unfairness, DDOS, $d = 5$.

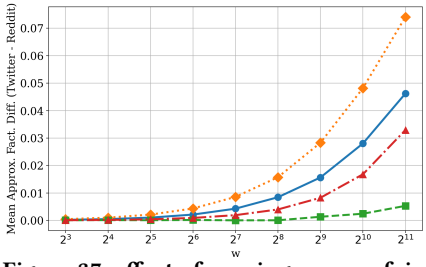

Figure 37: effect of varying w on unfairness, REDDIT-TWITTER, $d = 5$.

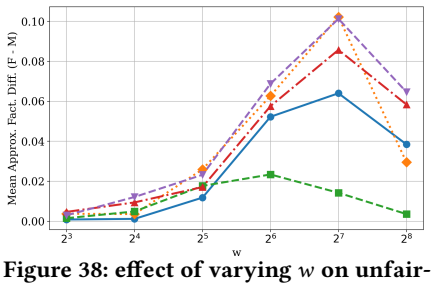

Figure 38: effect of varying w on unfairness, CENSUS, $d = 5$.

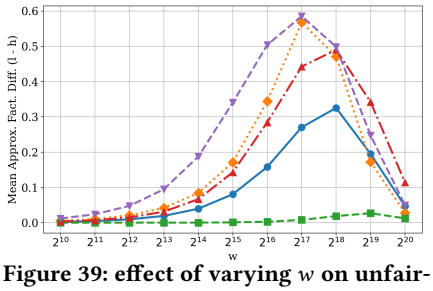

Figure 39: effect of varying w on unfairness, GOOGLE NGRAM, $d = 5$.

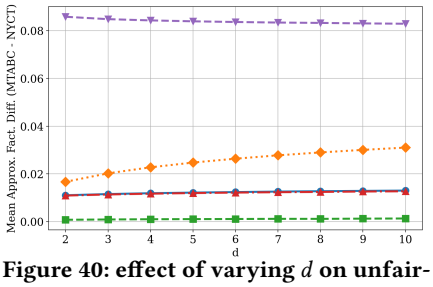

Figure 40: effect of varying d on unfairness, NYC BUS, $w = 512$.

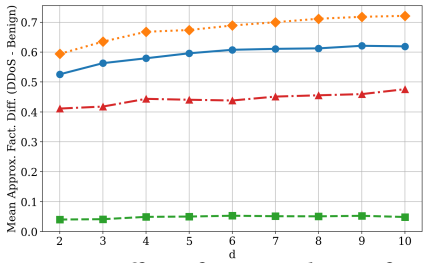

Figure 41: effect of varying d on unfairness, DDOS, $w = 64$.

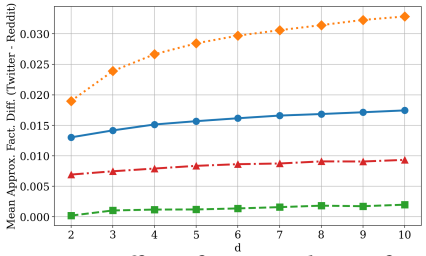

Figure 42: effect of varying d on unfairness, REDDIT-TWITTER, $w = 512$.

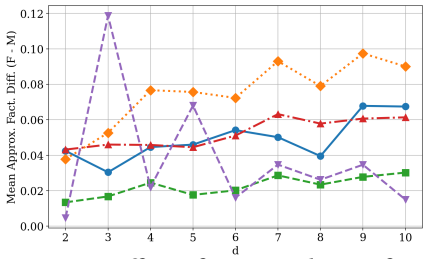

Figure 43: effect of varying d on unfairness, CENSUS, $w = 64$.

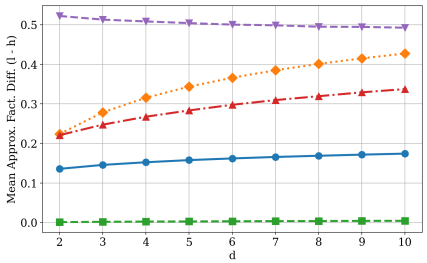

Figure 44: effect of varying d on unfairness, GOOGLE NGRAM, $w = 65536$.

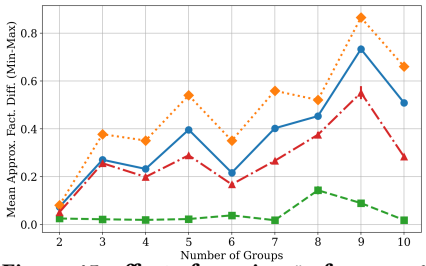

Figure 45: effect of varying # of groups ℓ on unfairness, CENSUS, $w = 64, d = 5$.

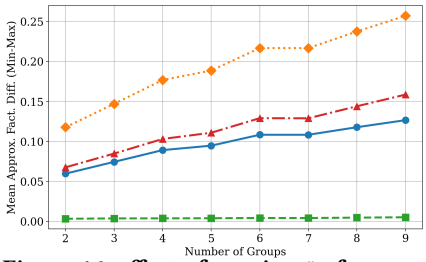

Figure 46: effect of varying # of groups ℓ on unfairness, GOOGLE NGRAM, $w = 65536, d = 5$.

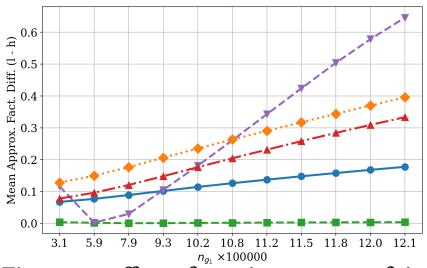

Figure 47: effect of varying n_{g_1} on unfairness, GOOGLE NGRAM, $w = 65536, d = 5$.

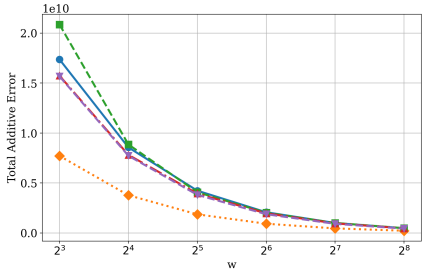

Figure 48: effect of varying w on price of fairness, NYC BUS, $d = 5$.

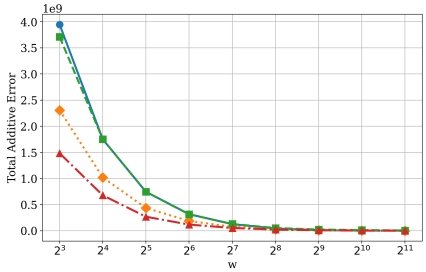

Figure 49: effect of varying w on price of fairness, DDOS, $d = 5$.

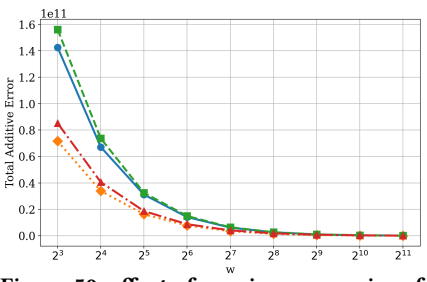

Figure 50: effect of varying w on price of fairness, REDDIT-TWITTER, $d = 5$.

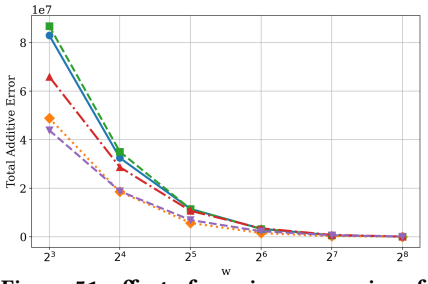

Figure 51: effect of varying w on price of fairness, CENSUS, $d = 5$.

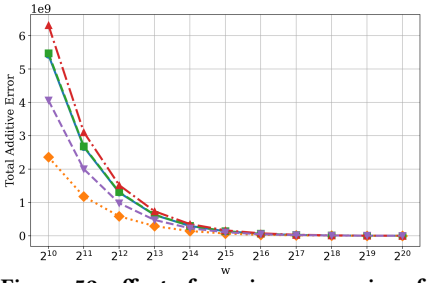

Figure 52: effect of varying w on price of fairness, GOOGLE NGRAM, $d = 5$.

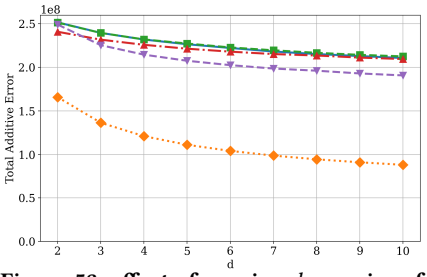

Figure 53: effect of varying d on price of fairness, NYC BUS, $w = 512$.

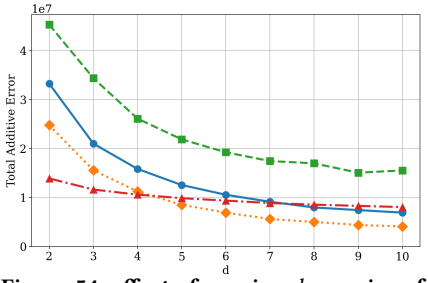

Figure 54: effect of varying d on price of fairness, DDOS, $w = 64$.

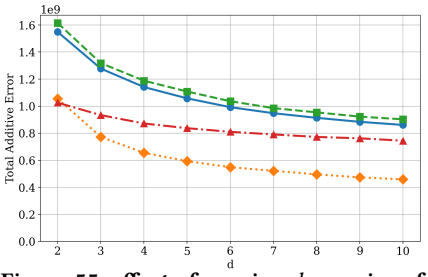

Figure 55: effect of varying d on price of fairness, REDDIT-TWITTER, $w = 512$.

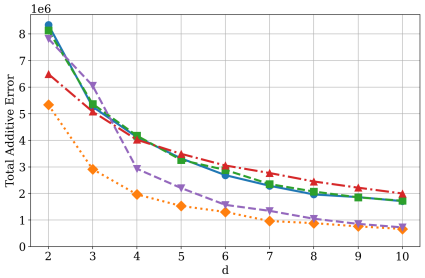

Figure 56: effect of varying d on price of fairness, CENSUS, $w = 64$.

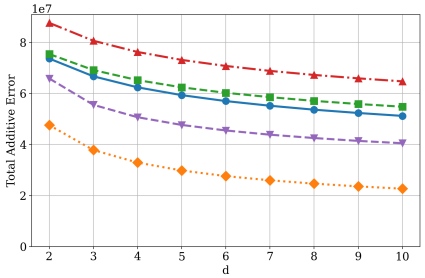

Figure 57: effect of varying d on price of fairness, GOOGLE NGRAM, $w = 65536$.

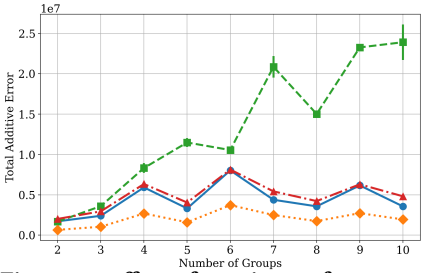

Figure 58: effect of varying # of groups ℓ on price of fairness, CENSUS, $w = 64$, $d = 5$.

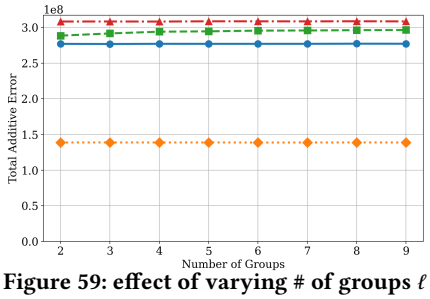

Figure 59: effect of varying # of groups ℓ on price of fairness, GOOGLE NGRAM, $w = 65536$, $d = 5$.

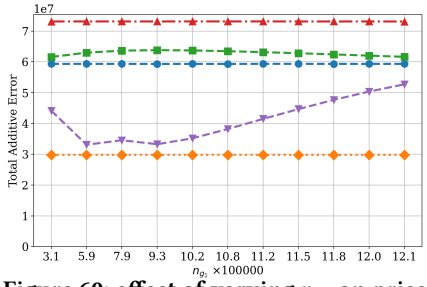

Figure 60: effect of varying n_{g_1} on price of fairness, GOOGLE NGRAM, $w = 65536$, $d = 5$.

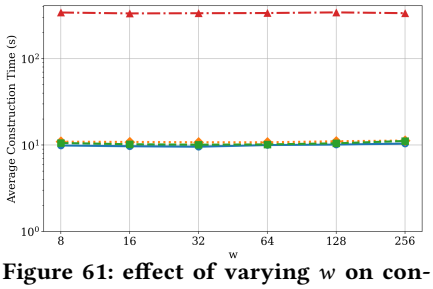

Figure 61: effect of varying w on construction time, NYC BUS, $d = 5$.

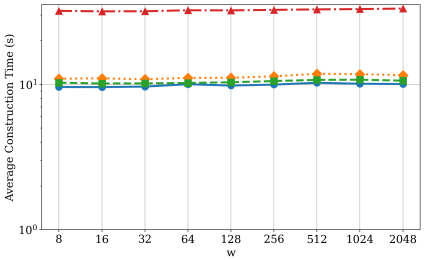

Figure 62: effect of varying w on construction time, DDOS, $d = 5$.

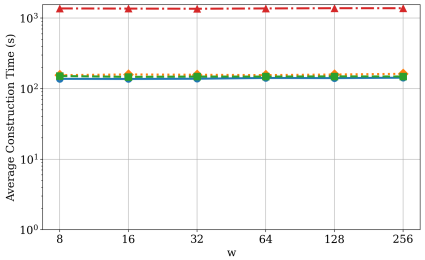

Figure 63: effect of varying w on construction time, REDDIT-TWITTER, $d = 5$.

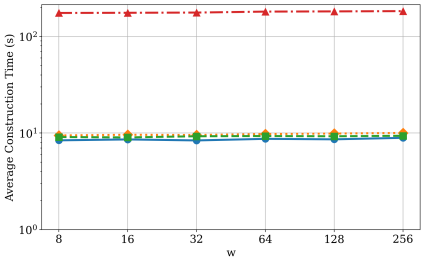

Figure 64: effect of varying w on construction time, CENSUS, $d = 5$.

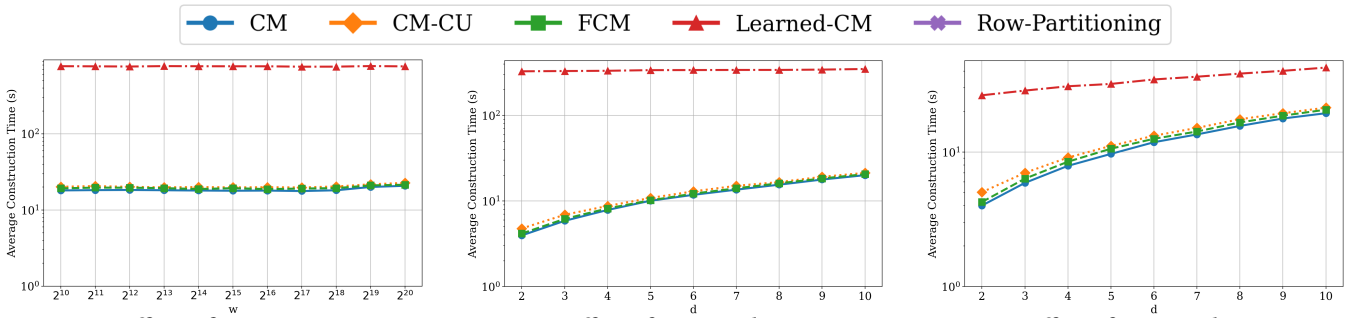

Figure 65: effect of varying w on construction time, GOOGLE NGRAM, $d = 5$.

Figure 66: effect of varying d on construction time, NYC BUS, $w = 512$.

Figure 67: effect of varying d on construction time, DDOS, $w = 64$.

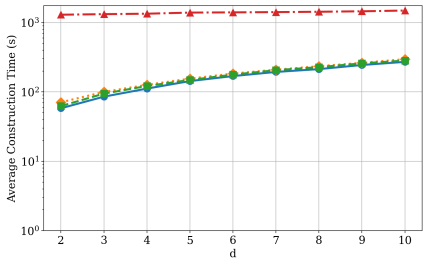

Figure 68: effect of varying d on construction time, REDDIT-TWITTER, $w = 512$.

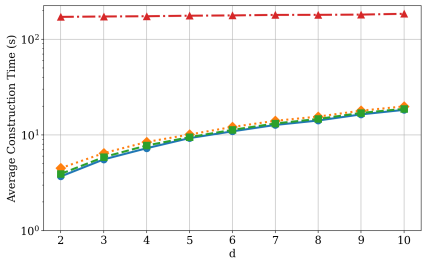

Figure 69: effect of varying d on construction time, CENSUS, $w = 64$.

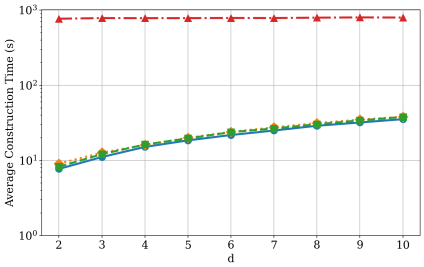

Figure 70: effect of varying d on construction time, GOOGLE NGRAM, $w = 65536$.

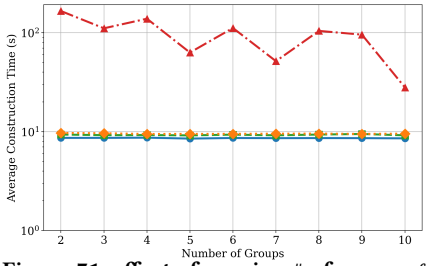

Figure 71: effect of varying # of groups ℓ on construction time, CENSUS, $w = 64$, $d = 5$.

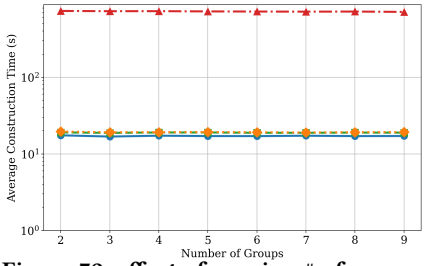

Figure 72: effect of varying # of groups ℓ on construction time, GOOGLE NGRAM, $w = 65536$, $d = 5$.

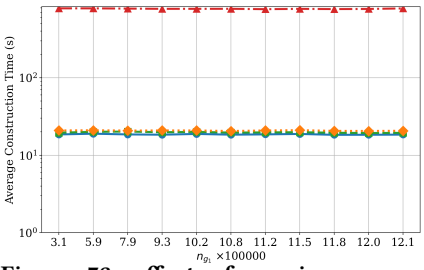

Figure 73: effect of varying n_{g_i} on construction time, GOOGLE NGRAM, $w = 65536$, $d = 5$.

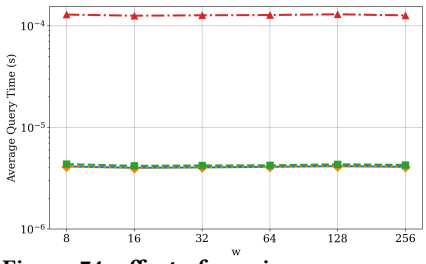

Figure 74: effect of varying w on query time, NYC BUS, $d = 5$.

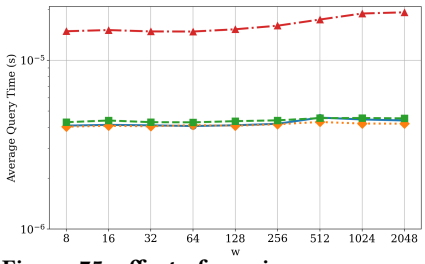

Figure 75: effect of varying w on query time, DDOS, $d = 5$.

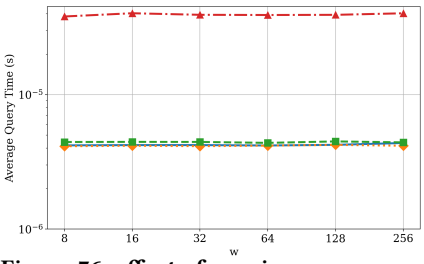

Figure 76: effect of varying w on query time, REDDIT-TWITTER, $d = 5$.

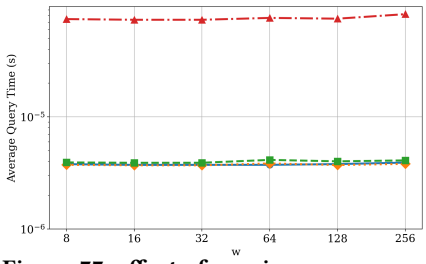

Figure 77: effect of varying w on query time, CENSUS, $d = 5$.

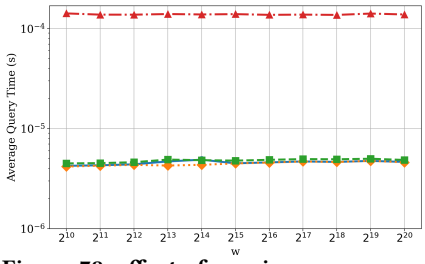

Figure 78: effect of varying w on query time, GOOGLE NGRAM, $d = 5$.

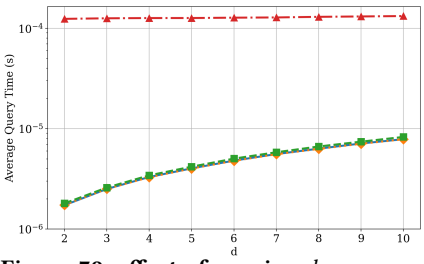

Figure 79: effect of varying d on query time, NYC BUS, $w = 512$.

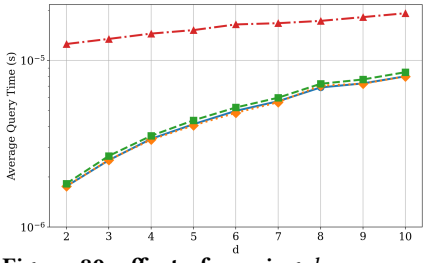

Figure 80: effect of varying d on query time, DDOS, $w = 64$.

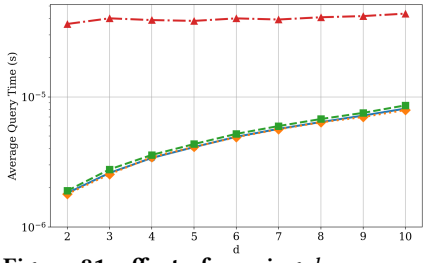

Figure 81: effect of varying d on query time, REDDIT-TWITTER, $w = 512$.

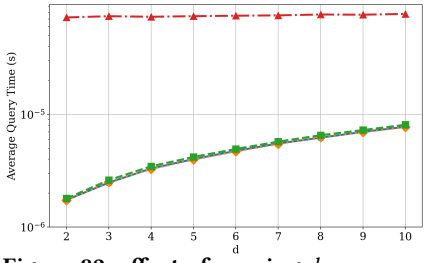

Figure 82: effect of varying d on query time, CENSUS, $w = 64$.

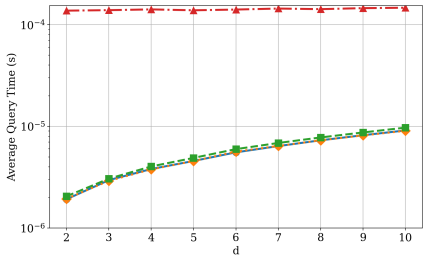

Figure 83: effect of varying d on query time, GOOGLE NGRAM, $w = 65536$.

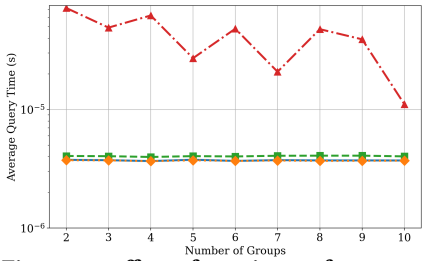

Figure 84: effect of varying # of groups ℓ on query time, CENSUS, $w = 64, d = 5$.

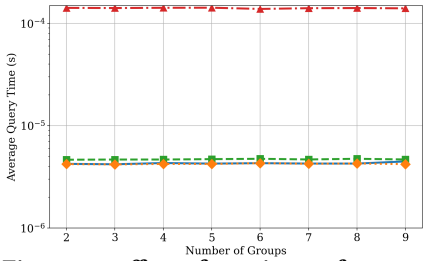

Figure 85: effect of varying # of groups ℓ on query time, GOOGLE NGRAM, $w = 65536, d = 5$.

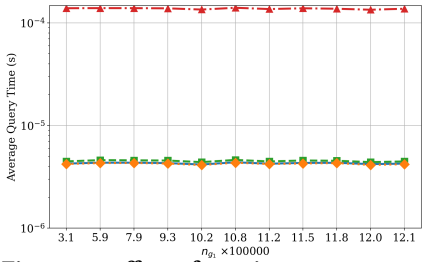

Figure 86: effect of varying n_{g_1} on query time, GOOGLE NGRAM, $w = 65536, d = 5$.