2^β. Although the time complexity is still $O(n \log n)$, Figure A below shows that the average compression time over all datasets in Table III is significantly reduced by the improved BOS-B, compared to the original version, without losing compression ratio. We revise the aforesaid algorithm changes in Section V-B, Page 7. The corresponding compression time in Figure 11c is also updated.

(2) To compress all the values, it requires to scan them at least once, and thus the $O(n)$ time complexity of BOS-M is inevitable. Nevertheless, we replace QuickSelect [14] by a faster approximate median implementation [15] to improve the compression time of BOS-M. Again, Figure A illustrates the reduction of compression time by the improved BOS-M compared to the original version, without losing compression ratio. We update the algorithm details in Section VI-B, Page 8, and the corresponding compression time in Figure 11c as well.

Finally, we conduct an experiment on the average compression time over all datasets of improved BOS-V, BOS-B and BOS-M, by varying block size n, in Figure 16, Section VIII-G, Page 13. All the methods increase almost linearly owing to the existence of duplicate values in the datasets. The advanced BOS-B increases much slower than BOS-V, while the approximate BOS-M is the most efficient. (The corresponding decompression time is also presented for O4 below.)

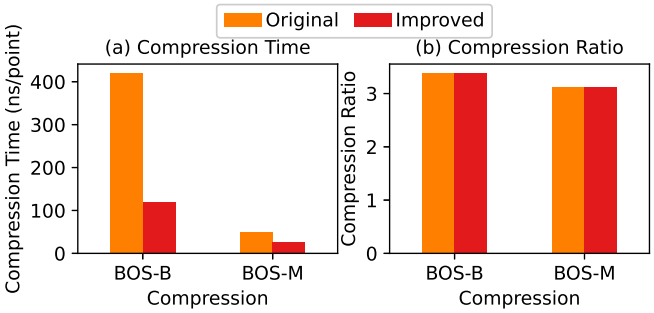

Fig.A: Improved compression time without losing compression ratio.

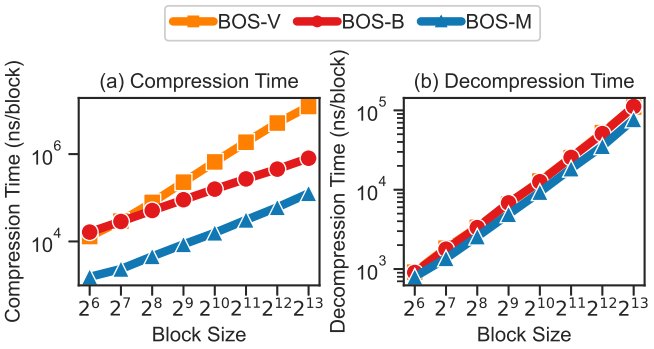

Fig.16: Compression and decompression time by varying block size n.

O3. *Generally, it is good to mention that BOS-V and BOS-B are optimal algorithms, while BOS-M is a heuristic. While BOS-M is quite fast, it does not have any theoretical guarantee on the quality of x_l and x_u.*

Reply: Thanks for your suggestions about the theoretical guarantee on the quality of x_l and x_u returned by the heuristic BOS-M. While it is difficult to bound the approximation ratio in general, given the various data distributions, we do obtain some theoretical guarantee for the special case of normal distribution, in Proposition **??** in Section VI-C, Page 8. The full proof procedure can be found in the appendix [**?**].

Let C_{opt} be the storage cost of the optimal solution for outlier separation problem, and C_{approx} be the storage cost of the solution x_l and x_u returned by the heuristic BOS-M. Since many real-world datasets follow the normal distribution as illustrated in Figure 9, we study the theoretical bound of approximation ratio $\rho = \frac{C_{\text{approx}}}{C_{\text{opt}}}$ under the normal distribution.

Proposition ??. For normal distribution $X \sim N(\mu, \sigma^2)$, the approximation ratio ρ of BOS-M satisfies

$$\rho \leq \begin{cases} 2 & \text{if } \sigma \leq \frac{5}{3}, \\ \lceil \log(3\sigma - 1) \rceil & \text{otherwise.} \end{cases}$$

Proof. (1) Firstly, we prove the upper bound of the storage cost C_{approx} for BOS-M,

$$C_{\text{approx}} \leq \begin{cases} \lceil \log(6\sigma + 1) \rceil n & \text{if } \sigma < \frac{1}{2}, \\ 2n & \text{if } \frac{1}{2} \leq \sigma \leq \frac{5}{3}, \\ \lceil \log(3\sigma - 1) \rceil n & \text{otherwise.} \end{cases}$$

For normal distribution $X \sim N(\mu, \sigma^2)$, the median is μ, the maximum value is approximately $\mu + 3\sigma$, and the minimum value is approximately $\mu - 3\sigma$, which correspond to the 99.7% confidence interval under the empirical rule (within three standard deviations from the mean).

The storage cost C_β of BOS-M with bit-width β is

$$\begin{aligned} C_\beta &= C(\mu - 2^\beta, \mu + 2^\beta) \\ &= n_l \lceil \log(3\sigma - 2^\beta + 1) \rceil + n_u \lceil \log(3\sigma - 2^\beta + 1) \rceil \\ &\quad + (n - n_l - n_u)(\beta + 1) \\ &= (n_l + n_u) \lceil \log(3\sigma - 2^\beta + 1) \rceil \\ &\quad + (n - n_l - n_u)(\beta + 1). \end{aligned}$$

According to β from $\lceil \log(6\sigma + 1) \rceil$ to 1, C_β first decreases and then increases. The upper bound of C_{approx} is thus

$$C_{\text{approx}} \leq \min\{C_{\lceil \log(6\sigma+1) \rceil}, C_1\},$$

where

$$C_{\lceil \log(6\sigma+1) \rceil} = \lceil \log(6\sigma + 1) \rceil n,$$

and

$$C_1 = (n_l + n_u) \lceil \log(3\sigma - 1) \rceil + 2(n - n_l - n_u).$$

To discuss 4 different cases of C_1 as follows, we could convert C_1 to

$$C_1 = 2n + (\lceil \log(3\sigma - 1) \rceil - 2)(n_l + n_u).$$

a) When $\sigma \leq \frac{1}{2}$ ($\mu - 2 < \mu - 3\sigma$), i.e., there are no upper

and lower outliers with $\beta = 1$, we have $n_l + n_u = 0$ and $C_1 = 2n \geq \lceil \log(6\sigma+1) \rceil n = C_{\lceil \log(6\sigma+1) \rceil}$.

b) When $\frac{1}{2} \leq \sigma < \frac{2}{3}$ ($\mu - 2 < \mu - 3\sigma$), i.e., there are no upper and lower outliers with $\beta = 1$, we have $n_l + n_u = 0$ and $C_1 = 2n < C_{\lceil \log(6\sigma+1) \rceil}$.

c) When $\frac{2}{3} \leq \sigma \leq \frac{5}{3}$, we have $0 \leq \lceil \log(3\sigma - 1) \rceil \leq 2$ and $2(n - n_l - n_u) \leq C_1 \leq 2n < C_{\lceil \log(6\sigma+1) \rceil}$.

d) When $\sigma > \frac{5}{3}$, we have $\lceil \log(3\sigma - 1) \rceil \geq 2$ and C_1 increases with σ growing. Then, when σ tends to positive infinity, we have there are no center values and $C_1 = \lceil \log(3\sigma - 1) \rceil n < C_{\lceil \log(6\sigma+1) \rceil}$.

Therefore, we conclude that

$$C_{\text{approx}} \leq \begin{cases} \lceil \log(6\sigma+1) \rceil n & \text{if } \sigma < \frac{1}{2}, \\ 2n & \text{if } \frac{1}{2} \leq \sigma \leq \frac{5}{3}, \\ \lceil \log(3\sigma - 1) \rceil n & \text{otherwise.} \end{cases}$$

(2) Moreover, we can prove that $C_{\text{opt}} \geq n$.

The storage cost is larger than the sum of bit-width for each value, thus the optimal cost C_{opt} has

$$C_{\text{opt}} = \sum_{i=1}^{n} b_i,$$

where

$$b_i = \begin{cases} 1 & \text{if } x_i = x_{\min}, \\ \lceil \log(x_i - x_{\min} + 1) \rceil & \text{otherwise.} \end{cases}$$

Thus, we have $C_{\text{opt}} \geq n$, even when all values are the same.

(3) Finally, we derive that

$$\rho = \frac{C_{\text{approx}}}{C_{\text{opt}}} \leq \begin{cases} \lceil \log(6\sigma+1) \rceil & \text{if } \sigma < \frac{1}{2}, \\ 2 & \text{if } \frac{1}{2} \leq \sigma \leq \frac{5}{3}, \\ \lceil \log(3\sigma - 1) \rceil & \text{otherwise.} \end{cases}$$

a) When $\sigma < \frac{1}{2}$, we have $\rho \leq \lceil \log(6\sigma+1) \rceil \leq 2$.

b) When $\frac{1}{2} \leq \sigma \leq \frac{5}{3}$, we have $\rho \leq 2$.

c) When $\sigma > \frac{5}{3}$, we have $\rho \leq \lceil \log(3\sigma - 1) \rceil$.

To sum up, we conclude that

$$\rho = \frac{C_{\text{approx}}}{C_{\text{opt}}} \leq \begin{cases} 2 & \text{if } \sigma \leq \frac{5}{3}, \\ \lceil \log(3\sigma - 1) \rceil & \text{otherwise.} \end{cases}$$

□

O4. *The decompression time of BOS is generally larger than the decompression time without using BOS.*

Reply: We acknowledge that the decompression time of BOS is generally larger than the decompression time without using BOS, since the outliers need to be processed separately. Nevertheless, we can improve the decompression time by arranging the storage layout in Figure 7, so that the data only needs to be scanned once. In the previous submission, the center values, lower outliers and upper outliers are stored separately. They need to be merged in decompression, by scanning all the values twice. In the revised layout, the lower outliers, center values and upper outliers are stored together in the original data order. Their corresponding bit-widths, α, β, γ, are marked by a bitmap. Consequently, the decompression process only needs to scan the data once. We revise the aforesaid storage layout in Section VII-A, Page 8.

As shown in Figure B below, the average decompression time over all datasets for BOS-V, BOS-B and BOS-M is improved by the new layout, compared to the original version, without losing compression ratio. We also update the decompression time of BOS in Figure 11c. It is not surprising that the decompression time increases linearly with the block size n, as illustrated in Figure 16b. BOS-M has less decompression time, since it separates fewer outliers. We explain the results in Section VIII-B2, Page 12.

double check

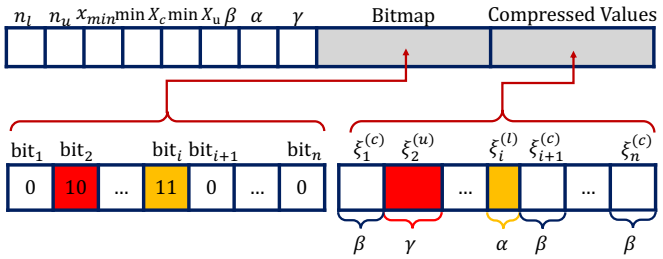

Fig.7: (Improved) storage layout of bit-packing with outlier separation (BOS).

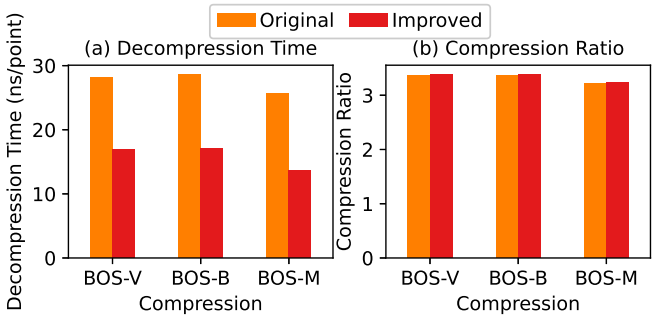

Fig.B: Improved decompression time without losing compression ratio.

C. Reviewer 2

Thank you very much for reading our paper carefully and the helpful suggestions. Below is our response to your comments (in blue).

O1. *The issue is a little bit special or the generality needs to be clearly pointed out.*

Reply: Following the suggestion, we clarify the generality of the outlier issue in Section VIII-A2, Page 10. As the data distribution illustrated in Figure 9, outliers commonly exist in real datasets. We count the corresponding number of lower and upper outliers separated by BOS-V in each dataset in Figure 10. As discussed in O3 below, even for those datasets with a relatively small proportion of outliers, by separating them, the compression ratio could still be significantly improved. Therefore, the outlier issue is general and worthwhile to address in compression.

O2. *To better motivate the issue, the advantage of employing the operator should be explicitly pointed out such as the*