What can real information content tell us about compressing climate model data?

Hayden Sather
Department of Computer Science
Colorado School of Mines
Golden, CO, USA
hrsather@mines.edu

Alexander Pinard
Applied Mathematics and Statistics
Colorado School of Mines
Golden, CO, USA
apinard@mines.edu

Allison H. Baker
Computational and Information Systems Lab
National Center for Atmospheric Research
Boulder, CO, USA
abaker@ucar.edu

Dorit M. Hammerling
Applied Mathematics and Statistics
Colorado School of Mines
Golden, CO, USA
hammerling@mines.edu

***Abstract*—The massive data volumes produced by climate simulation models create an urgent need for data reduction. Lossy compression in particular is a solution that can significantly reduce storage requirements, however, a tradeoff must be made between the amount of compression applied and the scientific integrity of the data. Determining how much compression can be applied is therefore vital for applying lossy compression. One particular metric for gauging the quality of compression is the percentage of real information present in the original data that is preserved in the compressed data. We compute bitwise real information content for several climate variables from the popular Community Earth System Model, and we investigate the amount of compression that can be applied to each of these climate variables using two popular compression algorithms designed for floating-point data while preserving 99% of the real information content. The analysis of the real information content of data after lossy compression has been applied shows a helpful visualization of how compression artifacts have been introduced to the data. Finally, we demonstrate how this real information content can be used in a straightforward manner to determine compressor settings for our data.**

***Index Terms*—Real Information Content, Entropy, Bit Grooming, ZFP, Lossy Compression, Climate Model Data**

I. Introduction

Output data from climate models is typically stored as 32- or 64-bit floating-point values. Due to factors such as numerical discretization error and round-off error, the last bits of these floating-point values are often essentially "noise". Removing the noisy bits is of great interest as these simulation data sets are typically massive and costly to store, and data compression methods are not effective on noisy (i.e., essentially random) data. However, it is a difficult task to determine an appropriate cutoff point between the bits that contain important information and the bits that do not. Furthermore, climate model data contains a variety of different variables (e.g., surface temperature, precipitation, heat flux, etc.), which often have different distributional characteristics, and might hence have different cutoff points. Klöwer et al. [1] propose a method to determine, for floating-point data, which bits contain noise and which contain valuable information by introducing the concept of real information content. In short, real information content is measured by the amount of dependence the average bit in a given bit position has with adjacent bits in the same bit position. Here, adjacency means spatially adjacent, though it could also be applied to, e.g., temporally adjacent values for time-series data. The underlying idea is that noise bits will be independent and uniformly distributed, meaning that their values are 0's and 1's with equal probability, and independent of values in adjacent grid points. Conversely, detecting dependence between spatially adjacent data bits indicates that the bits contain useful or "real" information. In other words, these are the bits that we want to preserve. Our work specifically uses the real information content approach for the following two broader tasks:

1) What does the real information content look like for popular atmospheric variables in our data set? What does the real information content look like for the same variables after lossy compression has been applied? And can we detect potential artifacts introduced by compression via the real information content. The latter is a novel contribution to the field.
2) Can we use the real information content of a variable to choose which bit position should be used as the cutoff for bit-grooming (BG) types of compression algorithms (i.e., [2], [3])? In other words, by finding the location in the data where the bits become spatially independent, can we specify the parameter for the BG approach that preserves the real information after compression?

II. Test Data and Compressors

The simulation data used for our experiments were generated by the Community Earth System Model (CESM) [4] at the National Center for Atmospheric Research (NCAR). In particular, we use a subset of atmospheric data from the publicly available CESM Large Ensemble Project (CESM-LENS) [5]. We look at two years of ten popular daily 2D

TABLE I: Set of 2D daily CAM variables investigated.

variable name	description	units
FLNS	net longwave flux (surface)	W/m^2
ICEFRAC	fraction of surface area covered by sea-ice	(fraction)
LHFLX	surface latent heat flux	W/m^2
PRECT	precipitation rate	m/s
PSL	sea-level pressure	Pa
Q200	specific humidity: 200 mbar pressure surface	kg/kg
TAUX	zonal surface stress	N/m^2
TS	surface temperature	K
WSPDSRFAV	horizontal total wind speed average (surface)	m/s
Z500	geopotential: 500 mbar pressure surface	m

TABLE II: Selected characteristics of the investigated variables for the first time slice. Note that the absolute nonzero minimum is marked "–" when equivalent to the minimum.

variable name	min min	abs min (nonzero)	max	cutoff bit
FLNS	-32.92	3.91e-3	190.45	11
ICEFRAC	0	1.87e-12	1.0	None
LHFLX	-48.89	1.91e-18	772.58	9
PRECT	-1.16e-20	8.60e-32	4.33e-6	21
PSL	9.50e4	–	1.07e5	17
Q200	2.41e-6	–	1.60e-4	9
TAUX	-2.04	6.63e-8	1.01	16
TS	218.12	–	321.47	11
WSPDSRFAV	3.03e-2	–	33.12	10
Z500	4.88e3	–	5.91e3	17

atmospheric variables from ensemble member 31, starting with the year 2006, shown in Table II. For each variable, we list the description, units, maximum and minimum values, and the "cutoff bit", which is further explained in Section IV. These test data are single-precision, or 32-bit, data. Recall that for single-precision data, each 32-bit value consists of a single sign bit, followed by 8 exponent bits, followed by the 23-bit significand (or mantissa).

We conduct our experiments using Bit Grooming Quantization [2], [3] and ZFP [6]. Bit Grooming reduces data to a specified precision via quantization and then compresses with a standard lossless compressor. We use the Granular BitRound (GBR) quantization variant and the label BG_K, where K indicates the number of significant digits (NSD) to preserve. ZFP is a popular transform-based lossy compressor that we use in fixed-precision mode. The data are labeled ZFP_P, where the fixed-precision mode parameter (P) specifies how many uncompressed bits per transform coefficient to store. See Appendices A and B for details on these compressors. Note that the techniques described in this paper could be applied to any type of lossy compression algorithm.

III. Overview of Real Information Content

This section provides more detail on the mutual information and real information content methodology described in Klöwer et al. [1]. Note that the Earth's atmosphere is typically simulated over its surface in a discretized fashion that roughly resembles a rectangular grid. The grid can be 2-D for surface data or 3-D if there is a vertical component. Model data also typically contains a temporal component because the model outputs values at discrete points in time. In this setting, the bitwise real information is defined as the mutual information of adjacent points in the same bit-position and assessed using statistical dependence between neighboring grid points. Bits that have little to no statistical dependence on bits in the same bit-position from adjacent grid points are considered noise (or "false" information) based on the assumption that climate variables vary smoothly in space and hence bits that contain real information should be statistically dependent in space.

Therefore, the real information provides a measure of how much information each bit contains on average and can be expressed as a percentage of the sum of the real information of all the bits. Importantly, the last few percents of real information are often spread across many mantissa bits. This behavior indicates that many of the least significant bits contain little real information, and the assumption is that their removal has a negligible effect on the scientific information contained in the data.

For our purposes, we use two-dimensional grids that only contain latitude and longitude at a fixed vertical level. However, as done with the ECMWF data in [1], we flatten the two dimensions into a single dimension in row order (e.g., adjacent points in the flattened array are adjacent in latitude. Some redundancy is lost by only considering one adjacent point by flattening the data. But we found that the loss of redundancy is worth the simplicity of the algorithms and improvement of performance. Perhaps if the amount of data were a limitation, it would be better to consider more adjacent points instead of a single one like we currently do.) Then the adjacent grid points are considered the current and the previous (or prior) grid point in the 1-D array. Any spatially or temporally adjacent point could be chosen, but here we use the previous point as the adjacent point matching Klöwer et al.'s [1] choice. The way the real information equation is set up in this case requires two input vectors, a vector of current grid points and a single additional vector of adjacent grid points. It is possible to have multiple neighbors depending on how adjacency is defined, and our setup can be generalized to a setting with multiple adjacent points.

Here we define several bitstreams. We let B^m represent the bitstream of all bits in the size N grid at the m-th bit positions. B^m_{adj} is the bitstream representing the value of the first $N-1$ grid points of the N total grid points at bit position m, $B^m_{adj} = (B^m_1, B^m_2 ... B^m_{N-1})$. We also let B^m_{cur} be the bitstream representing the value of the last $N-1$ bits at bit position m, $B^m_{cur} = (B^m_2, B^m_3 ... B^m_N)$. Then from eq. (2) in [1], recall that the unconditional entropy of a bitstream $B^m = (B^m_1, B^m_2, ..., B^m_k, ..., B^m_l)$ is defined as

$$H = -p^m_0 \log_2(p^m_0) - p^m_1 \log_2(p^m_1), \qquad (1)$$

where p^m_0 is the unconditional probability of a bit B^m_k in B^m being 0 and p^m_1 is the unconditional probability of a bit B^m_k in B^m being 1.

Let p^m_{ij} be the probability that the bits are in the state $B^m_{adj_k} = j$ and $B^m_{cur_k} = i$ (simultaneously), and let $p^m_{adj_j}$ and $p^m_{cur_i}$ be the unconditional probabilities that $B^m_{adj_k} = j$

and $B^m_{cur_k} = i$, respectively, for an arbitrary $k \in [1, N-1]$ that denotes the index in the vector B^m_{adj}. Then, as stated in equation (4) in [1], the mutual information M of bitstreams B^m_{cur} and B^m_{adj} is calculated as

$$M(B^m_{cur}, B^m_{adj}) = \sum_{i=0}^{1}\sum_{j=0}^{1} p^m_{ij} log_2(\frac{p^m_{ij}}{p^m_{adj_j} p^m_{cur_i}}). \qquad (2)$$

We use the spatial dependence of *spatially adjacent* bits in each 32-bit floating-point value to derive the real information content. We think of bits being spatially adjacent to one another if they are in the same position in the 32-bit binary representation of a pair of values in neighboring grid points (i.e. each subsequent pair of values in the flattened array of grid points). Computing the mutual information between B^m_{adj} and B^m_{cur} is equivalent to computing the mutual information on all spatially adjacent bits in the mth bit-position. This calculation is completed for each l bit positions of a bitstream, which for our case is $l = 32$ bits.

Klöwer et al. [1] also show that we can think about mutual information in terms of the conditional and unconditional entropies, and this perhaps more intuitive approach is detailed in Appendix C. In summary, by defining conditional entropies, H_0 and H_1, at a particular grid location that are conditioned on the state of the preceding bit in the bitstream B^m being 0 or 1, respectively, we can show that the mutual information in (2) can also be written as

$$M = H - p_{adj_0} H_0 - p_{adj_1} H_1, \qquad (3)$$

where p_{adj_0} and p_{adj_1} are the unconditional probabilities that the adjacent bit is 0 or 1, respectively. The expression of mutual information in this form is referred to by Klöwer et al. [1] as the real information content and indicates that it is the difference between the bitwise unconditional entropy and the bitwise conditional entropies.

IV. Cutoff Bit Selection

We use the idea of real information content to find an appropriate cutoff for the number of bits that carry real information content by adapting Klöwer et al.'s [1] method to analyze the real information content for each bit position for our data. We follow the convention that the first bit is indexed with a zero. Hence, a cutoff bit of 8 means we keep the first 9 bits, and consider the remaining bits noise. The algorithm to find the cutoff bit for a given variable is described below.

1. Perform real information content analysis on the original (i.e. uncompressed) data for each bit position
2. Sum over all bit positions to find the total real information.
3. Iteratively add bits, left-to-right, and calculate the respective real information after each is added to compare to the total.
4. Once enough bits are added and the information content is at or above a set threshold percentage of total information.

The rightmost column in Table II lists the cutoff bit determined by the algorithm above for each variable for a threshold of 99%, meaning we preserve 99% of the real information content. This threshold is adjustable depending on the application. We note that some variables do not have a cutoff point, and our algorithm is able to recognize this and output that there is no appropriate cutoff. For example, ICEFRAC has no natural bit position where the left of the position contains 99% of the real information and the right contains the remainder. Instead, every bit position carries a non-negligible amount of information. In our test data, this is the only variable with this behavior.

V. Compressor Setting Selection

Another application of real information content analysis is to determine the best setting for a particular compression algorithm. To find the "best" setting for a given compressor, we want to find the setting that results in the most data reduction while satisfying a measure of quality. In this case, we use the information content to evaluate the quality of the compressed data using the following steps:

1. Compress and reconstruct the original data with all of the specified compression parameters.
2. Perform real information content analysis on all of the original data and the compressed (then reconstructed) data.
3. Iterate through each of the real information content data arrays generated by the compression algorithm, starting with the least compression to the most compression.
4. Compare each compressed real information data array to the original real information data array via a distance metric (described below).
5. Choose the compression algorithm that provides the most compression without exceeding a specified distance metric.

Table III shows the settings for each variable and compressor determined using the method described above. The column, M, is the total real information in the original data. The column, K for BG_K is the parameter, K, for the BG compression algorithm, and the column, P for ZFP_P is the parameter, P, for the ZFP compression algorithm, both found by the algorithm described in Section 4.1. Note that when a variable can not be compressed well enough by any setting for a given compressor, the algorithm will return that there is no appropriate setting (indicated by "N/A" in the table). This situation occurs for ICEFRAC for both compressors and for PRECT for ZFP, shown in Figures 2 and 4, respectively.

A. Difference Metric

We introduce a difference metric, Algorithm 1, to assess the difference in real information content between the compressed and original data. We note that because this metric computes an average distance between the original and compressed data, it is theoretically possible for compression artifacts to both increase and decrease the real information in different regions of the data, causing the average real information over the entire compressed dataset to remain consistent with the original. However, we have not encountered this behavior in practice, and this metric has been useful.

TABLE III: Information content and parameters for BG and ZFP that preserve information content.

Variable	M	K for BG_K	P for ZFP_P
FLNS	0.87	2	8
ICEFRAC	6.06	N/A	N/A
LHFLX	3.09	2	8
PRECT	0.61	2	N/A
PSL	2.91	4	12
TAUX	1.33	2	8
TS	3.81	3	12
Q200	2.62	2	8
WSPDSRFAV	1.58	2	8
Z500	3.00	3	12

Algorithm 1 Difference Metric

```
1. At bit positions where the original information content is
   below a low threshold (we chose 0.005), set the correspond-
   ing information of the compressed data to zero at that bit.
2. Determine the sum of the real information content over all
   bit positions for the original data.
3. Determine the absolute difference in real information
   between the compressed and original data for each position
   and sum them over all.
4. Divide the value from step 3 by the total real information
   content from step 2 to obtain the distance metric.
```

B. Artifact assessment

For the purpose of the particular comparison described in Algorithm 1, note that we only consider bit positions which have real information in the original data. However, we can use the difference in real information content in bit positions that correspond to noise in the original data to detect potential artifacts introduced by the compressors. In other words, the difference metric can indicate whether the compressor has introduced artifacts that manifest themselves by introducing structure in the data that was not present in the original data, particularly in the bit positions corresponding to noise. This artifact distance calculation is done by taking the sum of the absolute value of differences between the real information content of the original data and the compressed/reconstructed data over all bit positions, including those that correspond to noise in the original data. Note that this value is not normalized, so it is not suitable for comparing the effectiveness of compression algorithms between different variables, but can be used to compare compressors for the same variable. We will see examples of this artifact distance metric in the figures for each variable in the following section.

VI. Real Information Content Experiments

In this section, we compute the real information content for each bit position for multiple CESM atmospheric variables in our previously described test set, both with and without lossy compression. Each plot features the real information content curve of the original uncompressed data, along with the real information content curve of the data after being compressed and reconstructed with their corresponding compression scheme and parameters. The real information content for the original data is given by a solid blue line, while the lines for the compressed data are dashed. Figures 1 through 10 contain the following supplemental information:

1) The cutoff bit, shown by the solid red line and the number to its right (as described in Section IV).
2) The start of the mantissa bits, shown by the dashed black line. (This location is the same for all values and is specified by the IEEE format.)
3) The compression parameter found by our algorithm in Section V that retains 99% of the real information content, shown in the upper right corner of each plot (and by an asterisks in the legend).
4) The difference metric (Section V-A) for each compression level, given by the first 3 decimal-point value next to the compression level in the legends. A value of zero indicates there is no difference between the original data and the compressed data.
5) The size of the artifact of each compression level (Section V-B), given by the second 3 decimal-point value next to the compression level in the legends. A value of zero indicates there are no artifacts in the data.

As shown in Figures 1 through 10, the pattern of real information content is strongly bimodal with two distinct peaks for the compressed data, while the original data does not have this second peak. This second peak is likely an artifact introduced by the compressors, which is undesired as the original data shows that there should be no spatial structure at those low-significance bits. After identifying these peaks, in practice we could likely take additional steps to remove these artifacts if desired. However, here we simply highlight that these artifacts are being introduced. Also, an interesting pattern emerges when the compression parameters for ZFP and BG are decreased, corresponding to progressively more compression (i.e., more data reduction). In particular, the real information content of the compressed data contains a spike that was not present in the original data, and as the compressor parameters are changed to cause increasingly aggressive data reduction, this second spike moves closer and closer to the first peak in the real information. This pattern is quite clear in Figure 1 which shows the real information content for FLNS compressed with the ZFP and BG compression algorithms (left and right panels, respectively). Figure 5 also exhibits a similar pattern for the PSL variable. Notice that the PSL data has compression artifacts that eventually reach the real information peak of the original data for the most aggressive compressor parameters. This interaction is undesired because this means that real information will be lost, indicating that the spatial structure of the data is not preserved.

The ICEFRAC variable shown in Figure 2 contains interesting results as well. For ICEFRAC, its original data doesn't have a single peak. Notice that even in this graph, as the compression level rises, the curves of the real information of the compressed data grow further away from the information content curve of the original data. This pattern indicates that the more compressed the ICEFRAC data is, the greater the compression artifacts modify the real information of the data. Even a small amount of lossy compression creates too many artifacts. Since this effect is undesired, it appears that ICEFRAC is not a good candidate to be compressed by any of the tested compression parameters for ZFP or BG.

Fig. 1: The Real Information Content of FLNS.

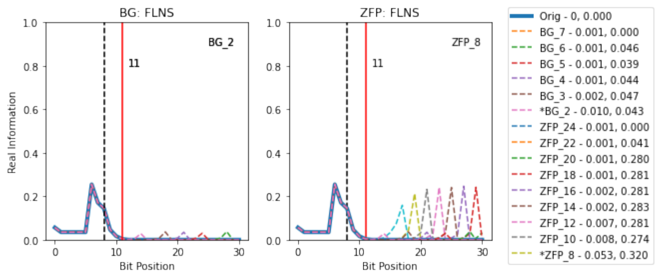


Fig. 2: The real information content of ICEFRAC. Note that there is no acceptable bit position that will serve as a cutoff.

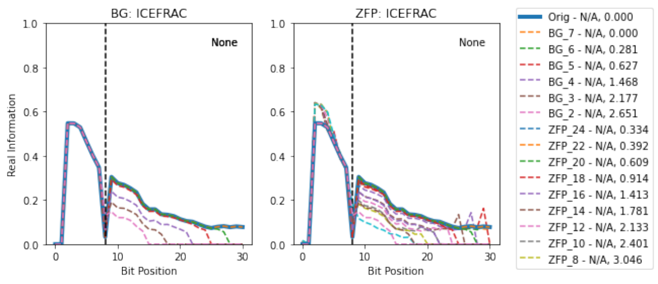


Fig. 3: The real information content of LHFLX.

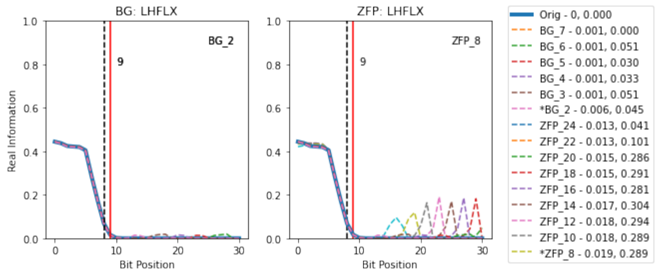


Fig. 4: The real information content of PRECT.

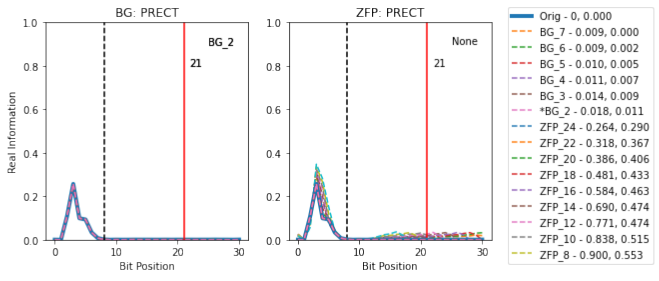


Fig. 5: The real information Content of PSL.

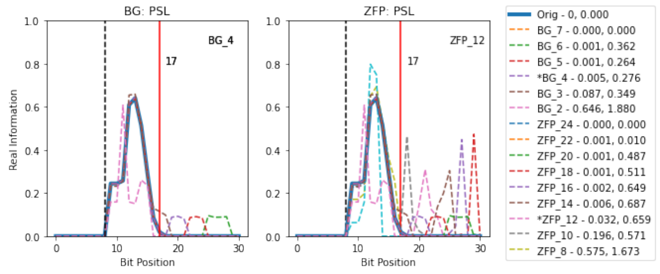


Fig. 6: The real information Content of TAUX.

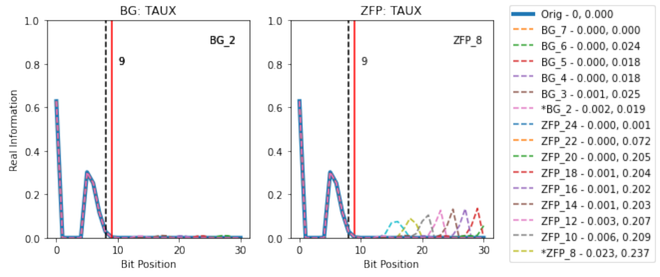


Fig. 7: The real information Content of TS.

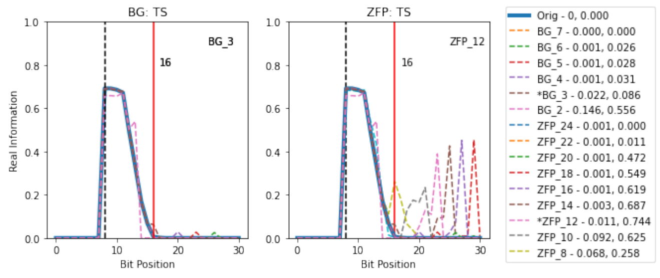


Fig. 8: The real information content of Q200.

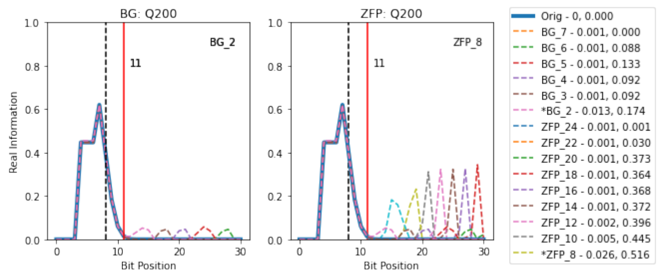


Fig. 9: The real information content of WSPDSRFAV.

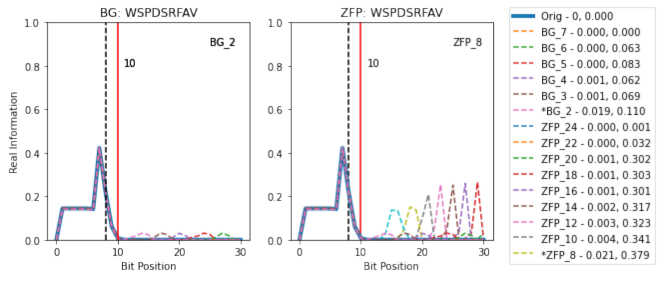
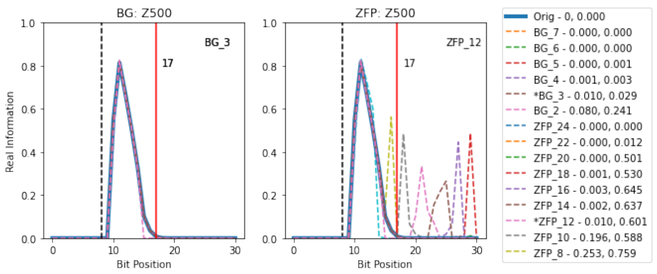


Fig. 10: The real information content of Z500.



PRECT in Figure 4 is another variable whose behavior differs from others. Variables such as PRECT with large ranges of values (e.g., see Table II) are well known to be challenging for lossy compressors, particularly methods such as ZFP which finds a common exponent for each block after the initial partition. (Note the data values for ICEFRAC span a large range of magnitudes as well.) In Figure 4, it rather appears that the cutoff bit (indicated by the red line) is in the incorrect position as compared to the other variable plots. However, interesting enough, PRECT contains some real information in nearly all of the 32 bits positions, likely due to the distribution of exponent order in the data.

The presence of visible compression artifacts in the real information content is a surprising outcome of these experiments. Furthermore, it is interesting to see that the Bit Grooming method consistently had artifacts that were smaller in magnitude in comparison to the ZFP compression algorithm. As shown in the legends of the figures above, as the amount of compression increases, the size of the artifact also increases. Also shown is that when the amount of compression increases, the location of the artifact moves from bits of lower significance to bits of higher significance. For many variables, when compression is aggressive enough, the artifacts are located in the bits that contain real information content in the original data. This occurrence indicates that artifacts are affecting the real information that we care about. The presence of these artifacts also indicates that compression algorithms may introduce structure into the data. If artifacts are introduced into the bit positions where the original data has real information content, this indicates that the real information of the data is modified, which is undesirable. Even if the artifacts are in the bits that did not contain real information in the original data, it is still important to be aware of these artifacts because they could potentially introduce problems in the future.

VII. Determining a Bit Grooming parameter

As explained in Section II, for the Granular BitRound (GBR) quantization approach, the data is labeled BG_K, where K is number of significant digits to preserve. One of the goals of this research was to verify whether it is possible to find the appropriate bit grooming parameter K from the analysis of the real information content of the original data. (That is, without having to compress and evaluate the data first.)

The first step is to convert the cutoff bit position to the correct GBR parameter. Let M indicate the position of the cutoff bit as determined by the real information content, and then $0 \leq M < 32$ for single-precision data. Recall that bit position 0 is the sign bit and the next eight positions contain the exponent bits. So to determine how many of the significand bits contain real information (M_S), we simply subtract off the sign and exponent bit positions such that $M_S = max(M - 9, 0)$. We want M_S positive as NSD bit-grooming approaches affect the significand. Note that the NSD approaches assume that the number of bits necessary to represent a single base-10 digit is $ln(10)/ln(2) = 3.32$. Therefore, the number of significand bits to preserve for a single precision number given parameter K is $nsb = ceil(3.32 * K) + 1$. Because the size of the significand for single precision is 23, then any $K > 6$ does not affect the data. So for a given M_s calculated from the real information content(for example, as described in Section V), in practice, one can calculate the corresponding parameter K by $K = floor(\frac{M_S+1}{3.32})$. Because of the $ceil$ operation, each K corresponds to a range of bit values as shown in Table IV. In Appendix D, Calculating K in this manner is compared against using another data quality metric, the DSSIM.

TABLE IV: Determining K for Granular BitRound given the information content in the significand (M_S).

number of bits (M_s)	K
0-2	0
3-5	1
6-8	2
9-12	3
13-15	4
16-18	5
19-22	6
23	7

Finally, we note that a new quantization option called BitRound has recently been released. This quantization allows the user to specify the number of significant bits (NSB) to keep, instead of the NSD in GBR, which allows for more fine grain specifications. Then we can simply use $K = M_S + 1$ as the parameter for BitRound. In future work, we will experiment with BitRound as is has recently been made available on the NCAR HPC system.

VIII. Conclusion and Future Work

In this paper, we demonstrate that using real information content can be quite useful in the context of lossy data compression for climate simulation output data. First, we show that the real information content measure can help identify the cutoff bit such that a large percentage of the real information is retained even after, for example, performing truncation beyond this bit position. Second, recalling that most compressors require a parameter to control the amount of data reduction, we find that Klöwer's technique is useful for making an informed compressor parameter choice. In particular, we can identify the compressor setting that maximally compresses the data without introducing compression artifacts into the part of the data that contains real information. This parameter selection is most straightforward for approaches such as Bit Grooming that allow the specification of significant digits or bits to retain, but is in principle applicable to wide range of compressors. Additionally, visualizing the real information content 1) helps to better understand the structure of data in terms of which bits may be most important to an end-user, 2) provides insight on the level of similarity between the original and the reconstructed data, and 3) indicates which bits are affected by the compression algorithm. We further use the presented analysis to detect whether a lossy compressor introduces any artifacts into the reconstructed data, which could potentially affect science results derived from the data. Overall, we find the real information content approach to be a powerful tool in our effort to determine how to best apply lossy compression to climate simulation data such that maximal data reduction is balanced with minimal information loss.

This paper lays the groundwork for future investigation into using the real information content to determine compression parameters for climate datasets. For example, this paper shows a connection between the fixed-precision compression parameters and the real information content of a dataset. Future work in this direction will involve expanding the investigation to include lossy compressors that do not use a fixed-precision mode, and whose ideal compression settings are likely to have a more complicated relationship with the real information content. In addition, a more regional, compared to the current global, comparison of the spatial artifacts introduced by several compressors using real information content as a metric could also be of interest. Also, further analysis will be done to establish the connection between the real-information content of a dataset and quantities or visualization characteristics that are of interest to researchers.

Finally, it is important also to note that using the real information content to compress data rests on certain assumptions about the dataset. In particular, it assumes spatial continuity of the variable being modeled. In practice, we would need to verify this assumption in order to ensure that we are not inadvertently removing "false" information from that data that is not truly white noise. Evaluation of these assumptions could be useful when the spatial continuity of a climate variable is not a given such as heterogeneously distributed land fluxes.

Acknowledgment

We would like to acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation.

References

[1] M. Klöwer, M. Razinger, J. J. Dominguez, P. D. Düben, and T. N. Palmer, "Compressing atmospheric data into its real information content," *Nature Computational Science*, vol. 1, no. 11, pp. 713–724, 2021.

[2] C. S. Zender, "Bit grooming: statistically accurate precision-preserving quantization with compression, evaluated in the netcdf operators (nco, v4.4.8+)," *Geoscientific Model Development*, vol. 9, no. 9, pp. 3199–3211, 2016. [Online]. Available: http://www.geosci-model-dev.net/9/3199/2016/

[3] X. Delaunay, A. Courtois, and F. Gouillon, "Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netCDF-4 or HDF5 files," *Geoscientific Model Development*, vol. 12, no. 9, pp. 4099–4113, 2019. [Online]. Available: https://gmd.copernicus.org/articles/12/4099/2019/

[4] J. Hurrell, M. Holland, P. Gent, S. Ghan, J. Kay, P. Kushner, J.-F. Lamarque, W. Large, D. Lawrence, K. Lindsay, W. Lipscomb, M. Long, N. Mahowald, D. Marsh, R. Neale, P. Rasch, S. Vavrus, M. Vertenstein, D. Bader, W. Collins, J. Hack, J. Kiehl, and S. Marshall, "The Community Earth System Model: a framework for collaborative research," *Bulletin of the American Meteorological Society*, vol. 94, pp. 1339–1360, 2013.

[5] J. Kay, C. Deser, A. Phillips, A. Mai, C. Hannay, G. Strand, J. Arblaster, S. Bates, G. Danabasoglu, J. Edwards, M. Holland, P. Kushner, J.-F. Lamarque, D. Lawrence, K. Lindsay, A. Middleton, E. Munoz, R. Neale, K. Oleson, L. Polvani, and M. Vertenstein, "The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability," *Bulletin of the American Meteorological Society*, vol. 96, 2015.

[6] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.

[7] A. H. Baker, A. Pinard, and D. M. Hammerling, "DSSIM: a structural similarity index for floating-point data," 2022, https://arxiv.org/abs/2202.02616.

Appendix

A. Bit Grooming Quantization

This lossy compression approach reduces data storage in two stages. First, the data is reduced to a specified precision via quantization. The idea is to remove the noise bits that do not contain meaningful information and simply waste storage space. Removing the noise bits by reducing the precision means that all bits beyond a selected bit position are essentially set to zero, though this zeroing is done in a more sophisticated manner that mitigates bias. Second, the quantized data is then compressed with a standard lossless compression algorithm. Because the quantization step reduces the noise in the data, the lossless approach can compress the data much more that if it were applied to the original data. Our bit-groomed test data was generated with netCDF Operators (NCO) 5.0.3 and uses the Granular BitRound (GBR) quantization variant. The data is labeled BG_K, where K indicates the input to the `--ppc` argument, which is number of significant digits (NSD) to preserve[1].For example, `ncks -L 1 --ppc <var>=<K> in.nc out.nc`. Note that the lower the K value, the more aggressive the compression. See [2], [3] for more information.

Note that as mentioned in Section VII more recent versions of NCO, beginning with 5.0.5, support BitRound, which expects a user-specified number of significant bits (NSB, or "keepbits") to retain, instead of the NSD that GBR requires. This alternative option allows more fine-grain specification, as the options for NSB are 1–23 for single precision. Note thatBitRound requires the additional NCO parameter `--baa=8` (so, for example, `ncks -L 1 --baa=8 --ppc <var>=<K> in.nc out.nc`).

B. ZFP

This lossy compression technique, at a high level, partitions the data into blocks, decorrelates the blocks, then compresses the blocks. In particular, ZFP partitions D-dimensional arrays into D-dimensional blocks with 4 values in each dimension (e.g., 2D data is partitioned into 4x4 blocks). Each block is compressed independently via a floating-point representation

[1]http://nco.sourceforge.net/nco.html#Compression

with a single common exponent per block, an orthogonal block transform, and embedded encoding.

We use ZFP 0.5.5 in fixed-precision mode, meaning the precision encoded for the transform coefficients is fixed[2], which is indirectly related to the relative error. The data is labeled ZFP_P, where the fixed-precision mode parameter (P) specifies how many uncompressed bits per coefficient to store, so the smaller the value of P, the more aggressive the compression. We also use a newer ZFP feature that addresses biased error (i.e., we enable the pre-rounding mode by configuring ZFP with `cmake -DZFP ROUNDING MODE=ZFP ROUND FIRST -DZFP WITH TIGHT ERROR=ON`). See [6] for more information on ZFP.

C. Real Information Content

Klöwer et al. [1] show that we can think about mutual information in a more intuitive way: in terms of the conditional and unconditional entropies that together give the bit information content of each bitstream B for each of the 32 bit positions. Conditional entropy is similar to unconditional entropy in that it returns an entropy value for all bit positions of a bitstream. However, conditional entropy depends on the weighted probability of a bit adjacent to it being a 0 or a 1. We define conditional probability that the current bit in a bitstream's value is i given that the adjacent bit's value is j as

$$p^m_{i|j} = p^m_{ij}/p^m_{adj_j}, \tag{4}$$

where, as stated above, $p^m_{adj_j}$ refers to the probability that $B^m_{adj_k} = j$, and p^m_{ij} is computed as the probability that $B^m_{adj_k} = j$ and $B^m_{cur_k} = i$. We have four conditional probabilities for our bitstream:

$$\begin{aligned} p^m_{0|0} = p^m_{00}/p^m_{adj_0}, \quad p^m_{0|1} = p^m_{01}/p^m_{adj_1}, \\ p^m_{1|0} = p^m_{10}/p^m_{adj_0}, \quad p^m_{1|1} = p^m_{11}/p^m_{adj_1}. \end{aligned} \tag{5}$$

With this notation, we can rewrite (2) and expand it as follows:

$$\begin{aligned} M(B^m_{cur}, B^m_{adj}) = p^m_{00} log_2(\frac{p^m_{00}}{p_{cur_0} p_{adj_0}}) + p^m_{01} log_2(\frac{p^m_{01}}{p_{cur_0} p_{adj_1}}) \\ + p^m_{10} log_2(\frac{p^m_{10}}{p_{cur_1} p_{adj_0}}) + p^m_{11} log_2(\frac{p^m_{11}}{p_{cur_1} p_{adj_1}}), \end{aligned} \tag{6}$$

Then for each bit position m ($m < 32$), we calculate $M(B^m_{cur}, B^m_{adj})$ using (6) for adjacent pairs of bit values at adjacent grid locations B^m_{cur} and B^m_{adj} in our grid and take the average over all grid locations to find the total mutual information for that bit position.

To further simplify, we define the conditional entropies, H_0 and H_1, at a particular grid location that are conditioned on the state of the preceding bit in the bitstream B^m being 0 or 1, respectively. Here we define these slightly differently than Klöwer's work (c.f. eq. (5) in [1]) to be more precise about the fact that we are using conditional probabilities:

$$H_0 = -p^m_{0|0} \log_2 p^m_{0|0} - p^m_{0|1} \log_2 p^m_{0|1}, \tag{7}$$

$$H_1 = -p^m_{1|0} \log_2 p^m_{1|0} - p^m_{1|1} \log_2 p^m_{1|1}, \tag{8}$$

[2]https://zfp.readthedocs.io/en/release0.5.5/

TABLE V: Comparison of K values for BG_K as determined by real information and DSSIM (multiple thresholds).

Varable Name	Real Information Content	DSSIM (.95)	(.995)	D (.9995)
FLNS	3	2	3	3
ICEFRAC	N/A	2	4	6
LHFLX	3	2	3	5
PRECT	2	2	3	5
PSL	5	5	6	7
TAUX	2	2	3	4
TS	5	4	5	7
Q200	3	2	4	5
WSPDSRFAV	3	2	3	4
Z500	5	4	5	7

Now if we substitute the conditional probability definitions as given in (5) into (6), then do some regrouping and substitution using the definitions of the unconditional entropy in (1) and conditional entropies in (7) and (8), we can show that

$$M = H - p_{adj_0} H_0 - p_{adj_1} H_1, \tag{9}$$

which matches what Klöwer et al. [1] refer to as the real information content.

This expression for the real information content indicates that it is the difference between the bitwise unconditional entropy and the bitwise conditional entropies. This value should be non-negative, and it can only be equal to zero in the case when there is no real information at a bit position. There are two situations in which there is no real information at a bit position: 1) if the value at that position is completely random, meaning $p^m_0 = p^m_1 = \frac{1}{2}$ where p^m_i is again defined as the unconditional probability that bit $B^m_k = i$ for $k \in (1...N)$ and the value of each bit is independent; 2) the value at that position is the same (i.e. all zeros).

D. Comparing parameters determined by the DSSIM

A natural question is whether the parameter K determined for GBR via the 99% real information content threshold is similar to that chosen by the Data Structural Similarity Index, or DSSIM, measure [7] that we have used in other work to evaluate quality and choose an appropriate compression parameter. The DSSIM indicates the similarity between the compressed and original data, where a value of 1.0 indicates that the data sets are equivalent. Therefore, the closer the DSSIM tolerance is set to 1.0, the more conservative the compression has to be to achieve that tolerance. In practice, we typically choose a tolerance of either .95 or .995. Table V lists the parameter K that is optimal for GBR based on the real information content as well as based on the DSSIM for three different DSSIM tolerances . In particular, the second column, "Real Information Content" lists the K parameter determined for GBR based on the real information content method as described in Section V. The last three columns contain K as determined using the listed DSSIM tolerance to evaluate quality. With the exception of the ICEFRAC variable, which requires further investigation, the K values determined based on the information content (column 2) agree nicely with the K values suggested by the DSSIM for tolerances .95 and .995.