Towards Improving Rate-Distortion Performance of Transform-Based Lossy Compression for HPC Datasets

Jialing Zhang Aekyeung Moon Xiaoyan Zhuo Seung Woo Son
Department of Electrical and Computer Engineering
University of Massachusetts Lowell
Lowell, MA, USA

***Abstract*—As the size and amount of data produced by high-performance computing (HPC) applications grow exponentially, an effective data reduction technique is becoming critical to mitigating time and space burden. Lossy compression techniques, which have been widely used in image and video compression, hold promise to fulfill such data reduction need. However, they are seldom adopted in HPC datasets because of their difficulty in quantifying the amount of information loss and data reduction. In this paper, we explore a lossy compression strategy by revisiting the energy compaction properties of discrete transforms on HPC datasets. Specifically, we apply block-based transforms to HPC datasets, obtain the minimum number of coefficients containing the maximum energy (or information) compaction rate, and quantize remaining non-dominant coefficients using a binning mechanism to minimize information loss expressed in a distortion measure. We implement the proposed approach and evaluate it using six real-world HPC datasets. Our experimental results show that, on average, only 6.67 bits are required to preserve an optimal energy compaction rate on our evaluated datasets. Moreover, our knee detection algorithm improves the distortion in terms of peak signal-to-noise ratio by 2.46 dB on average.**

***Index Terms*—Discrete Transform, Lossy Compression, Energy Compaction, Rate-Distortion.**

I. Introduction

Today's high-performance computing (HPC) simulations and applications easily produce extremely large volumes of data. For example, Aeroacoustic CDF (Computational Fluid Dynamics) simulation [1] and Hardware/Hybrid Accelerated Cosmology simulation [2] generate terabytes of simulation results in each simulation time step. Transferring, analyzing and archiving such large amounts of data result in a massive burden on I/O and storage systems [3], which leads to pressing challenges for scalable HPC systems and frameworks. One approach to alleviate this problem is to reduce the amount of data through data compression techniques before storing to disk or transferring through network.

Data compression has two categories, lossless or lossy. Lossless compression techniques, such as GZIP [4] and FPC [5], preserve full precision, thus are more acceptable to domain scientists. However, as shown in many prior studies, they hardly achieve appreciable compression ratios, typically no more than two [6]. Lossy compression techniques, widely used in image and video data compression such as JPEG [7] and MPEG [8], can potentially obtain higher compression ratio by discarding a certain amount of fractional part in floating point numbers. However, they are less commonly used in scientific datasets because of their uncertainty in the amount of information loss.

Nevertheless, recent studies have reported that scientific data can actually tolerate a certain amount of accuracy loss [9]. For instance, Tao et al. [6] conducted a comprehensive study of understanding lossy compression on HPC datasets. They examined the impact of reduced accuracy on scientific data analysis frameworks using the state-of-the-art lossy compressors, including SZ [10], ZFP [11] and ISABELA [12]. Their results demonstrated that there is a trade-off between compression ratios and tolerable error rate as well as the complex interplay among compressor design, data features, and compression performance. Given the pressing challenges beyond many of HPC I/O system capabilities, an in-depth understanding of the benefits and pitfalls are needed to make lossy compression a promising candidate.

In this paper, we analyze a transform-based lossy compression strategy by exploiting the energy compaction, a well-established mathematical model, and evaluate rate-distortion performance using real-world HPC datasets. Specifically, we apply transforms on segmented block-data, find top coefficients based on optimal energy compaction rate, and quantize the remaining by applying binning. Our objective in this paper is to: 1) find an optimal energy compaction approach that retains the minimum data points for representing the most information; 2) design an adjustable parameter for finding the best trade-off solution; and 3) most importantly, minimize error in the reconstructed data.

II. Analysis of Transform-Based Lossy Compression

Discrete transforms, which have been widely used in image and video lossy compression systems, are considered an effective way to achieve higher compression ratios without losing much information [11], [13]. The main ideas behind transform techniques are energy compaction property [14] and the correlation within the signal. In this paper, we represent HPC data as a signal for better explaining the theory in signal processing. Motivated by these properties, we propose a transform-based lossy compression for HPC datasets. We first

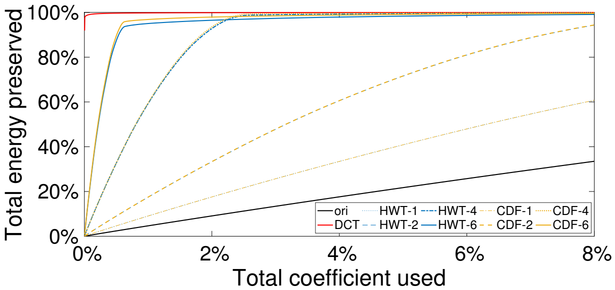


Fig. 1: Total energy attained by different transforms on dataset rlds. (ori: original time domain representation, DCT: DCT-II, HWT-2: 2-level HWT, CDF-4: 4-level CDF 9/7, etc.)

formulate the framework for evaluating energy compaction properties of several commonly used transforms in this section, then propose our compression strategy by presenting two algorithms in Section III.

A. *Energy Compaction of Transform Compression*

It is well known that signals can be represented quite accurately using only a small portion of the transform coefficients [13]. Effective representation of a signal can be beneficial if the reconstructed one does not include much distortion. In other words, transforming an original signal to another domain (or basis) allows us to represent the signal in a more concise format. For example, Moon et al. [15] applied transform-based compression on IoT datasets and showed that, by keeping only 3.7% of the transform coefficients (discrete cosine transform coefficients in their case), they could represent 99.9% of the energy (information) attained by the original data.

In the theory of signal processing, the energy of a signal f, ε_f, is calculated as the sum of squares of individual values:

$$\varepsilon_{f}= \sum_{n=1}^{N} |f_n|^2 \, , n = 1, 2, \ldots, N, \tag{1}$$

where f denotes the transform coefficients of data x and N denotes the total number of transform coefficients.

Energy compaction can be measured as a function of preserved energy regarding the number of largest magnitude transform coefficients [13]. We formulate it as the energy portion contained in the first M of the entire N sorted transform coefficients. We refer to this as Energy Compaction Rate (ECR), which is denoted as:

$$ECR_f = \frac{\sum_{n=1}^{M} |f_n|^2}{\sum_{n=1}^{N} |f_n|^2}, n = 1, 2, \ldots, N, M \leq N. \tag{2}$$

B. *Estimation of Energy Compaction on Various Transforms*

A transform with high ECR is an indication of effective signal representation and has the potential for achieving high compression ratio with minimal information loss. Thus, selecting the proper transform for our lossy compressor is critical.

TABLE I: Energy compaction rate (%).

Transform	Threshold	rlds	mrsos	sedov	cellular	Eddy	Vortex
Original	1/32	6.03	21.65	27.27	6.53	25.91	44.28
	1/64	3.09	11.63	15.50	3.45	16.08	28.12
DCT-II	1/32	99.81	91.36	94.50	99.49	94.78	98.35
	1/64	99.69	88.17	92.06	99.13	89.29	96.93
HWT 5-level	1/32	96.94	33.22	65.91	92.86	36.64	36.01
HWT 6-level	1/64	93.63	17.60	47.87	86.67	18.12	20.19
CDF 9/7 5-level	1/32	98.08	39.17	62.78	91.82	24.76	27.07
CDF 9/7 6-level	1/64	95.83	21.58	44.46	84.47	11.97	15.47

In this section, we compare three commonly used discrete data transforms, which are known to be fast and efficient, by analyzing their ECR on real-world HPC datasets. Details about evaluated dataset are shown in Table II and will be discussed later in Section IV. Specifically, we evaluate Discrete Cosine Transform (DCT-II), Discrete Haar Wavelet Transform (HWT) and Cohen-Daubechies-Feauveau (CDF 9/7) wavelet, which were utilized in [16]–[18], respectively. The goal of our evaluation is to determine the most desirable transform that preserves the maximum energy using a fixed number of coefficients.

Figure 1 shows the energy-coefficient plot of three transforms on dataset rlds, where x-axis is the percentage of coefficient used and y-axis is the total percentage of energy preserved. It should be noted that HWT and CDF 9/7 require several recursive passes for a more concise decomposition. In other words, higher level of decomposition incurs higher computational cost but produces a more compact signal representation. For a fair comparison of computational cost, we used up to 6-level of HWT and CDF 9/7. As shown in Figure 1, DCT-II, HWT and CDF 9/7 preserve more energy than the original representation when the same number of coefficients are used. The reason is that transforms typically increase energy compaction during decorrelation by observing previous or future values in the current signal. Time domain representation, however, fails to reveal the correlation between different coefficients efficiently. We also observe that DCT-II brings the highest energy compaction rate on rlds.

Table I presents the ECR of different transforms on six evaluated datasets with a fixed amount of coefficients (i.e., threshold = M/N). Overall, we observe that DCT-II leads to very effective representation of the signal compared with HWT and CDF 9/7 on the evaluated datasets. Therefore, we use DCT-II as our main transform method in the remainder of the paper.

III. Energy Compaction Based Compression Algorithm

In this section, we present our lossy compression strategies based on discrete cosine transform (DCT-II), and develop two algorithms: compression with fixed energy compaction rate, and compression with optimal energy compaction rate using a knee-point detection mechanism. Both algorithms require quantization and encoding steps during compression to reduce error rates and improve compression ratios.

A. Compression with Fixed Energy Compaction Rate

We first segment data into small fixed-size blocks and apply transform on them. The block-based transform design is applied due to its effectiveness in implementation and decorrelation efficiency explained in [13], [16], such as computationally less intensiveness, better decorrelation of the data if a proper block size is chosen (according to the data contents), and better characterization of local features than a global transform. We then save the top dominant block coefficients as is, based on a fixed ECR provided by users. To improve the fidelity of compression, we apply adjustable equal-width-binning quantization [19] on the remaining coefficients rather than rounding to zero, which is typically used in image and video compressions. After quantization, we use the Huffman encoding on bin indices to improve compression ratios.

Let us formulate our first algorithm based on fixed energy compaction rate. Suppose data is segmented into M blocks, each with the size of bz (i.e., each block has bz data point). For each block, top K coefficients are kept to preserve fixed $ECR\%$ of the energy contained. Since B bits can represent up to a maximum of 2^B different values, we equally assign the remaining coefficients into a maximum of $2^B - 1$ bins, where a coefficient falling in a certain bin is approximated as the bin center value.

B. Compression with an Optimal Energy Compaction Rate

Based on the definition described in Equations 1 and 2, we know that the energy compaction of transform coefficient (similar to cumulative energy distribution function) is concave (i.e., energy increases when more coefficient is added). Therefore, the system will reach a point at which the relative cost to increase energy preservation is no longer worth the corresponding performance benefit.

Motivated by this property, in our second algorithm, rather than using a fixed ECR, we extend it to find the optimal energy compaction point such that it can best balance inherent trade-offs between information loss and data reduction. We accomplish this by detecting the 'knee-point', similar to Kneedle Algorithm illustrated in [20]. The mathematical definition of 'knee' is given as a function of its first and second derivatives, which is calculated as:

$$K_f(x) = \frac{f''(x)}{(1 + f'(x)^2)^{1.5}}, \tag{3}$$

where $K_f(x)$ defines the curvature of f.

Given this definition, the overarching objective of our algorithm is to save top coefficients in each block that lead to an optimal energy compaction 'knee-point', which is summarized as follows. First, we fit the energy compaction of transform coefficient with a smoothing spline to preserve the shape. We then normalize the points of the smooth curve to the unit square. Next, we find the 'knee-point' in the normalized curve. This point indicates instances where the rate of increase in energy compaction rates begins to decrease. In other words, beyond the knee-point, there is a diminishing return in terms of data reduction and information loss.

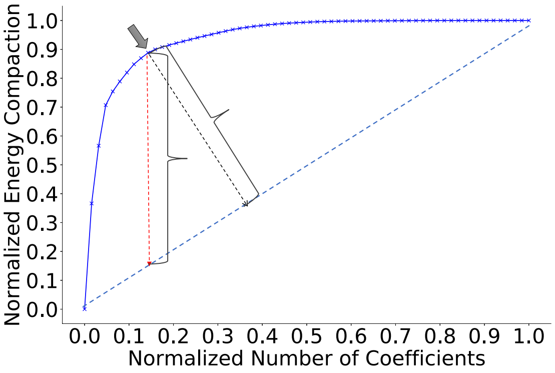
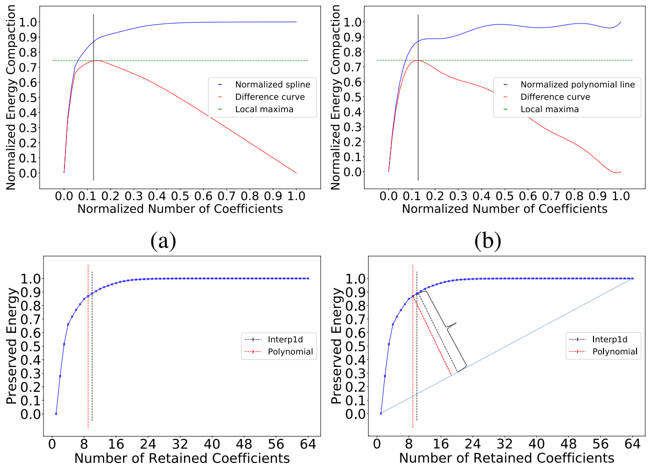


Fig. 2: Our Knee Detection Algorithm. Arrow symbol indicates the detected knee-point.



(a) (b) (c) (d)

Fig. 3: An example application of the spline fitting (interpolate a 1D function) and polynomial interpolation on block data of 'sedov'. (a) and (b) are normalized energy compaction, and (c) and (d) are preserved energy, for spline fitting and polynomial interpolation, respectively.

In general, the 'knee-point' is the point of maximum curvature in a fitted line, i.e., it is approximately the local maxima if the curve is rotated θ degrees clockwise about the minimum value of x and y through the line formed by the points (x_{min}, y_{min}) and (x_{max}, y_{max}) [20]. Figure 2 depicts the smoothed and normalized energy compaction, with the black dashed line indicating the maximum perpendicular distance from $y = x$. The red dashed line is the black dashed line after rotating 45 degrees clockwise.

In addition to the one-dimensional (1D) interpolation method, we also fit the energy compaction with polynomial interpolation (which generates more smooth curve). Figure 3 shows an example of using both fitting methods on a block data of 'sedov'. As shown in Figure 3a and 3b, the intersections of vertical black lines and fitting lines (blue curve lines) are the knee-points (local maxima). Figure 3c and 3d show the number of retained (used) coefficients and the perpendicular distance based on two fittings in our knee detection algorithms.

TABLE II: Dataset and its characteristics.

Code	Dataset	Value Range	Avg Value	Entropy	Dimension
FLASH [21]	sedov	4.2385	1.0000	4.9702	31040*154
	cellular	$2.6482E^7$	$2.2083E^7$	4.1190	32768*295
CMIP5 [22]	rlds	361.2303	285.8844	7.2106	12960*100
	mrsos	44.5000	7.6916	4.4864	12960*100
Nek5000 [23]	eddy	4.8345	$3.2366E^{-8}$	7.6047	16384*999
	vortex	0.0550	0.0017	7.5797	37024*99

IV. Experimental Evaluation

A. Datasets

We conduct our experiments on the Massachusetts Green High Performance Computing Cluster (MGHPCC) for running real HPC applications at various scales to generate datasets. To evaluate our proposed lossy compression, we use six generated datasets, all in double-precision floating-point. The detailed description of the datasets are summarized in Table II.

B. Evaluation Schemes and Metrics

We analyze our compression strategy using the following six schemes:

- **A1**: compression with fixed energy compaction rate.
- **A2**: compression with optimal energy compaction rate.
- **A2_interp1d**: A2 using 1D interpolation.
- **A2_polynomial**: A2 using polynomial interpolation.
- **A1_B**: A1 with equal-width-binning.
- **A2_interp1d_B** & **A2_polynomial_B**: A2_interp1d and A2_polynomial with equal-width-binning, respectively.

We choose rate-distortion, a critical metric used in evaluating the quality of compressed data, to assess the overall compression quality. As for rate-distortion, rate (or bit-rate) refers to the average number of bits used to represent a data point after the compression. It is equal to the number of full bits (i.e., 64-bit for double precision) used to represent each original data point divided by the overall compression ratio. On the other hand, the Compression Ratio (CR) is defined as:

$$CR = \frac{D}{D'}, \tag{4}$$

where D is the original size and D' is the compressed size.

Distortion is assessed using Peak Signal-to-Noise Ratio (PSNR) to measure the overall distortion between the original data and the reconstructed (decompressed) data, which can be expressed in terms of logarithmic decibel scale:

$$PSNR = 20 * log_{10}(data\_range) - 10 * log_{10}(MSE), \tag{5}$$

where $data\_range$ and MSE refer to data value range and mean squared compression error, respectively.

Therefore, higher PSNR represents less error, and smaller bit-rate represents higher compression ratio.

C. Evaluation Results

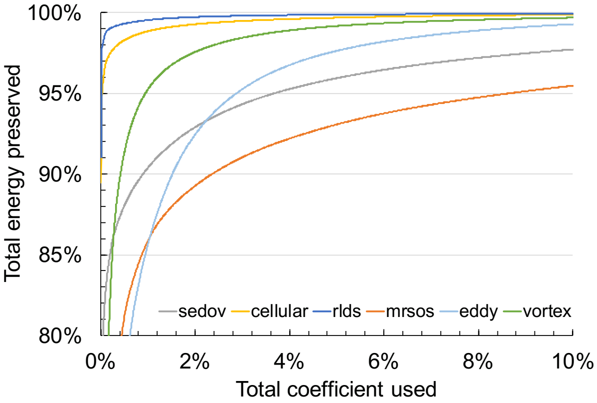


Fig. 4: Energy compaction on evaluated datasets.

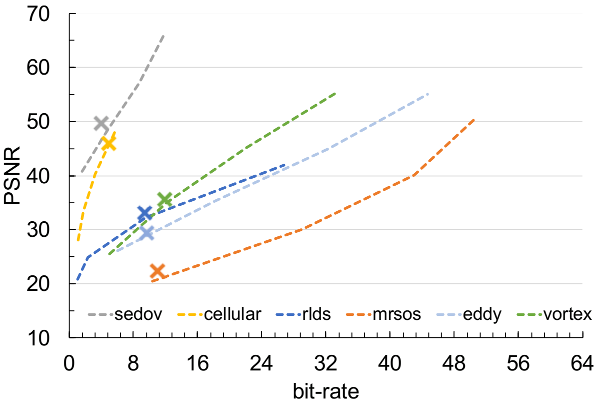
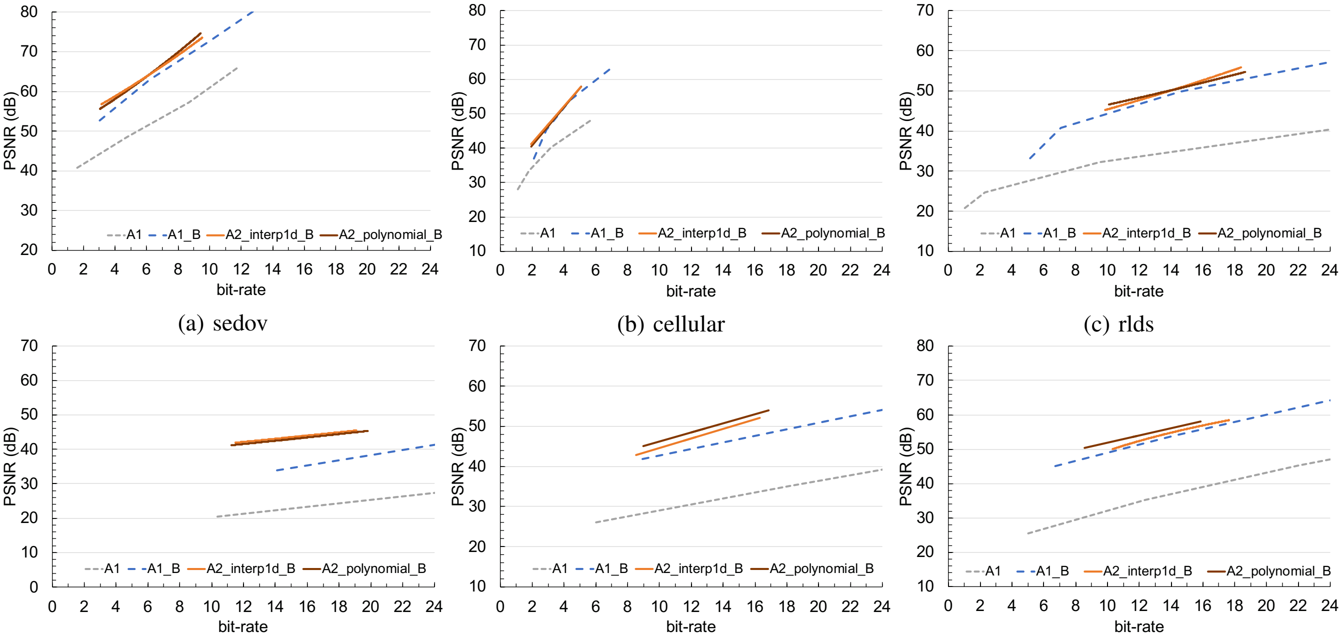


Fig. 5: Rate-distortion on A1 and A2_interp1d (dashed line: rate-distortion on A1 with bz of 64; cross sign: rate-distortion on A2_interp1d with bz of 64).

a) Energy Compaction and Datasets: The characteristics of HPC data vary as the data is generated from different applications or solvers. Thus, the inherent compressibility also varies from application to application. Figure 4 shows the energy compaction rate of our evaluated datasets with different energy compaction properties. For example, to preserve 95% of the energy, mrsos needs more than 8% of the coefficients, which is relatively higher than other datasets. Hence, mrsos is considered as a dataset with low energy compact property. It also shows that sedov preserves more energy than eddy when less than 2% of coefficients are used. However, beyond that, eddy preserves more energy than sedov. Based on these observations, it is critical to find the 'knee-point' of each dataset rather than using a fixed energy compaction rate.

b) Comparison between Fixed and Optimal Energy Compaction Rates: Figure 5 shows the rate-distortion of A1 and A2_interp1d. As shown in the figure, we observe that A2_interp1d overall achieves higher PSNRs than the A1 counterpart when the same bit-rates are used. The PSNR is especially higher on mrsos, which means our knee-point detection algorithm is effective for datasets with low energy compaction property. Figure 6, on the other hand, shows the rate-distortion of A1, A1_B, A2_interp1d_B, and A2_polynomial_B. We can see that from A1 to A1_B, PSNR increases with a great extent, showing an average increment of 15.8 dB on all six datasets. A2_interp1d_B, which is an improvement from A1_B, improves the PSNR further, with an average increment



(a) sedov (b) cellular (c) rlds

(d) mrsos (e) eddy (f) vortex

Fig. 6: Rate-distortion charts per each dataset for different algorithms.

TABLE III: Total number of coefficients, on average, used in each block.

Algorithm	sedov	cellular	rlds	mrsos	eddy	vortex
Interp1d	3.929	6.844	9.356	11.172	9.597	11.925
Polynomial	3.853	6.449	9.557	10.956	9.182	10.122

TABLE IV: Average energy compaction rate (%).

Algorithm	sedov	cellular	rlds	mrsos	eddy	vortex
Interp1d	98.44	99.59	99.99	99.99	89.83	93.55
Polynomial	98.44	99.59	99.99	99.99	90.12	91.08

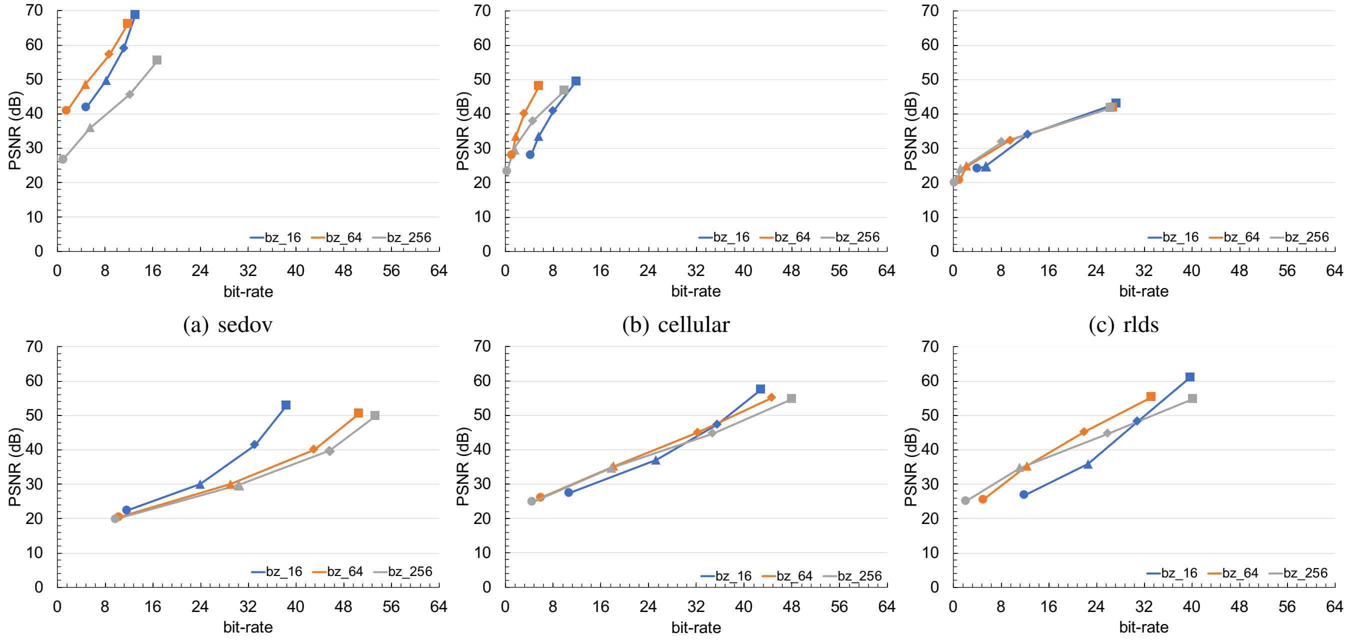
of 2.46 dB on all six datasets. Furthermore, for mrsos, which originally exhibited less energy compaction property, PSNR improved by 6.49 dB. We also find that the optimal algorithm does not show a significant improvement on cellular, which inherently has a high energy compaction property. Because of this property, the chances of our knee algorithm for further optimization are relatively low.

c) Rate-distortion and Block Size: We next evaluate how block size would affect the performance of our algorithm. Figure 7 presents the rate-distortion of A1 with block size (bz) of 16, 64 and 256. The energy compaction rates (ECR) were set to 90%, 99%, 99.9% and 99.99%. While we observe an overall trend that higher PSNRs need higher bit-rates, A1 on dataset sedov, cellular and rlds shows higher PSNRs with less increase in bit-rates. We also observe that more bit-rate is needed to preserve 99.9% or higher ECR on the majority of the evaluated datasets. This indicates that the system has reached a point where increasing energy no longer benefits the performance. It is also shown that varying bz does not affect the rate-distortion much on datasets such as rlds and eddy, while sedov, cellular, mrsos and vortex have higher PSNRs when bz is set to 16 and 64 (smaller block size).

d) Spline Fitting: We also observe that the differences between A2_interp1d_B and A2_polynomial_B are relatively small on most evaluated datasets. Table III and IV show the average coefficients used and energy compaction rates in each block of A2_interp1d_B and A2_polynomial_B, with bz of 64. As we can see, the optimal energy compaction rates of rlds and mrsos are higher than other datasets, which need more coefficients as a result. Unlike rlds and mrsos (both from CMIP5), the optimal energy compaction rates of eddy and vortex (both from Nek5000) are much smaller, but still need to preserve the same number of coefficients. Datasets sedov and cellular (both from FLASH) show the best energy compaction property where minimum coefficients are needed for representing the most amount of information. The result also shows that, when block size is set to 64, the average bit-rate of FLASH application to preserve an average energy of 99.015% is 5.5 (with compression ratio of 11.64), and the overall average on six datasets is energy of 96.915% and bit-rate of 6.67 (with compression ratio of 7.11).

V. Related Work

Numerous efforts have been made by researchers to apply data compression on scientific datasets for various purposes such as reducing checkpoint overheads. Lakshminarasimhan et al. [24] proposed a technique called ISABELA where the B-Spline transformation is applied on sorted data to make scientific dataset amenable to data compression. The fitting function used in ISABELA can guarantee a 0.99 correlation with the original data.



(a) sedov (b) cellular (c) rlds

(d) mrsos (e) eddy (f) vortex

Fig. 7: Rate-distortion of Algorithm A1 with different block size (bz) and energy compaction rate (ECR). x-axis: bit-rate, y-axis: PSNR (dB). (circle marker: ECR = 90%, triangle marker: ECR = 99%, diamond marker: ECR = 99.9%, and square marker: ECR = 99.99%).

Chen et al. [25] applied a lossy compression method, called NUMARCK, on the change ratios between consecutive checkpoints. More recently, Yuan et al. [26] presented a parallelized version of NUMARCK. The rationale behind NUMARCK is that data values change smoothly in most scientific simulations, thus storing the difference makes sense rather than storing data as is. One drawback of this approach is the higher memory requirement, which is not suitable for future extreme-scale systems where memory will be scarce due to the increasing number of cores. Sasaki et al. [27] also applied similar lossy compression, which is based on wavelet transformation. Their approach, however, is limited to multi-dimensional datasets, particularly a climate application called NICAM. Baker et al. [28] also evaluated how data compression impacted on climate data.

A recent study by Di and Cappello [29] used several curve-fitting techniques to predict successive data points, and represented those predictable data as the corresponding fitting models. For unpredictable datasets, they applied lossy compression using a binary representation analysis. Tao et al. [30] extended Di and Cappello's approach by employing an adaptive quantization mechanism to improve accuracy of their prediction-based compression algorithm.

Our proposed approach is different from these prior studies. Our mechanism is based on solutions used in image and video data compressions. Moreover, our approach of applying transform and quantization on datasets with any dimensionality is different from prior studies that require linearization of data before applying compression [29]. Unlike our recent work that uses a fixed coefficient selection mechanism [15], [16], the proposed approach selects coefficients dynamically based on our knee-point calculation mechanism. We also provide a mechanism to tune several key factors in lossy compression such as amount of error introduced by lossy compression, compression time, compression ratio, etc.

VI. Conclusion

In this paper, we analyze a lossy compression strategy by exploiting the energy compaction rate within our compression framework and evaluate rate-distortion performance using six real-world HPC datasets. Specifically, we apply block-based transforms and choose top dominant coefficients with the maximum energy compaction rates. Remaining coefficients are quantized using equal-width binning. Our experimental results show that our technique requires only 6.67 bits on average to preserve an optimal energy compaction rate for evaluated datasets. We also show that our optimization algorithm improves the distortion rate (in terms of PSNR) by 2.46 dB on average.

In our future work, we plan to expand the proposed compression technique in several ways. First, we plan to apply multiple transforms in our compression mechanism and find the optimal one for each data block that generates the highest energy compaction rate. We also plan to improve the rate-distortion of our technique by optimizing the quantization model. Lastly, we plan to extend our method to single-precision datasets and incorporate into various layers in our DCTZ framework [16] and HPC I/O software stack.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No.1751143.

REFERENCES

[1] S. M. Najmabadi, P. Offenhuser, M. Hamann, G. Jajnabalkya, F. Hempert, C. W. Glass, and S. Simon, "Analyzing the effect and performance of lossy compression on aeroacoustic simulation of gas injector," *Computation*, vol. 5, no. 2, 2017. [Online]. Available: http://www.mdpi.com/2079-3197/5/2/24

[2] M. Zeyen, J. Ahrens, H. Hagen, K. Heitmann, and S. Habib, "Cosmological particle data compression in practice," in *Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization*, ser. ISAV'17. New York, NY, USA: ACM, 2017, pp. 12–16. [Online]. Available: http://doi.acm.org/10.1145/3144769.3144776

[3] R. Lucas, J. Ang, K. Bergman, S. Borkar, W. Carlson, L. Carrington, G. Chiu, R. Colwell, W. Dally, J. Dongarra, A. Geist, R. Haring, J. Hittinger, A. Hoisie, D. M. Klein, P. Kogge, R. Lethin, V. Sarkar, R. Schreiber, J. Shalf, T. Sterling, R. Stevens, J. Bashor, R. Brightwell, P. Coteus, E. Debenedictus, J. Hiller, K. H. Kim, H. Langston, R. M. Murphy, C. Webster, S. Wild, G. Grider, R. Ross, S. Leyffer, and J. Laros III, "Doe advanced scientific computing advisory subcommittee (ascac) report: Top ten exascale research challenges," 2014.

[4] P. Deutsch, "Gzip file format specification version 4.3," United States, 1996.

[5] M. Burtscher and P. Ratanaworabhan, "Fpc: A high-speed compressor for double-precision floating-point data," *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 18–31, Jan 2009.

[6] T. Lu, Q. Liu, X. He, H. Luo, E. Suchyta, J. Choi, N. Podhorszki, S. Klasky, M. Wolf, T. Liu, and Z. Qiao, "Understanding and modeling lossy compression schemes on hpc scientific data," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2018, pp. 348–357.

[7] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer Publishing Company, Incorporated, 2013.

[8] D. Le Gall, "Mpeg: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991. [Online]. Available: http://doi.acm.org/10.1145/103085.103090

[9] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W. keng Liao, and A. Choudhary, "Data Compression for the Exascale Computing Era - Survey," *Supercomputing Frontiers and Innovations*, vol. 1, no. 2, 2014. [Online]. Available: http://superfri.org/superfri/article/view/13

[10] S. Di and F. Cappello, "Fast error-bounded lossy hpc data compression with sz," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, pp. 730–739.

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

[12] S. Lakshminarasimhan, N. Shah, S. Ethier, S.-H. Ku, C. S. Chang, S. Klasky, R. Latham, R. B. Ross, and N. F. Samatova, "Isabela for effective in situ compression of scientific data," *Concurrency and Computation: Practice and Experience*, vol. 25, pp. 524–540, 2013.

[13] X. Cai and J. S. Lim, "Algorithms for transform selection in multiple-transform video compression," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5395–5407, Dec 2013.

[14] J. S. Lim, *Two-dimensional Signal and Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

[15] A. Moon, J. Kim, J. Zhang, and S. W. Son, "Lossy compression on iot big data by exploiting spatiotemporal correlation," in *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2017, pp. 1–7.

[16] J. Zhang, X. Zhuo, A. Moon, H. Liu, and S. W. Son, "Efficient Encoding and Reconstruction of HPC Datasets for Checkpoint/Restart," in *35th International Conference on Massive Storage Systems and Technology*, 2019.

[17] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of lossy compression for application-level checkpoint/restart," in *2015 IEEE International Parallel and Distributed Processing Symposium*, May 2015, pp. 914–922.

[18] S. Li, S. Sane, L. Orf, P. Mininni, J. Clyne, and H. Childs, "Spatiotemporal wavelet compression for visualization of scientific simulation data," in *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2017, pp. 216–227.

[19] Z. Chen, S. W. Son, W. Hendrix, A. Agrawal, W. keng Liao, and A. N. Choudhary, "Numarck: Machine learning algorithm for resiliency and checkpointing," *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 733–744, 2014.

[20] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, June 2011, pp. 166–171.

[21] Flash Center for Computational Science, "FLASH User's Guide: Version 4.4," 2016.

[22] G. A. Meehl, C. Covey, B. McAvaney, M. Latif, and R. J. Stouffer, "Overview of the Coupled Model Intercomparison Project," *Bulletin of the American Meteorological Society*, vol. 86, no. 1, pp. 89–93, 2005.

[23] P. Fischer, J. Lottes, S. Kerkemeier, O. Marin, K. Heisey, A. Obabko, E. Merzari, and Y. Peet, "Nek5000 User Documentation," Argonne National Laboratory, Tech. Rep. ANL/MCS-TM-351, 2015.

[24] S. Lakshminarasimhan, N. Shah, S. Ethier, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, "Compressing the Incompressible with ISABELA: In-situ Reduction of Spatio-temporal Data," in *Proceedings of the 17th International Conference on Parallel Processing - Volume Part I*, ser. Euro-Par'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 366–379. [Online]. Available: http://dl.acm.org/citation.cfm?id=2033345.2033384

[25] Z. Chen, S. W. Son, W. Hendrix, A. Agrawal, W.-k. Liao, and A. Choudhary, "NUMARCK: Machine Learning Algorithm for Resiliency and Checkpointing," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 733–744. [Online]. Available: https://doi.org/10.1109/SC.2014.65

[26] Z. Yuan, W. Hendrix, S. W. Son, C. Federrath, A. Agrawal, W. Liao, and A. N. Choudhary, "Parallel Implementation of Lossy Data Compression for Temporal Data Sets," in *23rd IEEE International Conference on High Performance Computing, HiPC 2016, Hyderabad, India, December 19-22, 2016*, 2016, pp. 62–71. [Online]. Available: http://dx.doi.org/10.1109/HiPC.2016.017

[27] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of Lossy Compression for Application-Level Checkpoint/Restart," in *Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium*, ser. IPDPS '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 914–922. [Online]. Available: http://dx.doi.org/10.1109/IPDPS.2015.67

[28] A. H. Baker, H. Xu, J. M. Dennis, M. N. Levy, D. Nychka, S. A. Mickelson, J. Edwards, M. Vertenstein, and A. Wegener, "A Methodology for Evaluating the Impact of Data Compression on Climate Simulation Data," in *Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing (HPDC)*, 2014, pp. 203–214.

[29] S. Di and F. Cappello, "Fast Error-Bounded Lossy HPC Data Compression with SZ," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, pp. 730–739.

[30] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization," in *Proceedings of the 31th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society, 2017.