DPZ: Improving Lossy Compression Ratio with Information Retrieval on Scientific Data

Jialing Zhang*, Jiaxi Chen*, Xiaoyan Zhuo*, Aekyeung Moon† and Seung Woo Son*
*University of Massachusetts Lowell, MA, USA, †ETRI, Daegu, South Korea
{Jialing_Zhang, Jiaxi_Chen, Xiaoyan_Zhuo}@student.uml.edu, akmoon@etri.re.kr, SeungWoo_Son@uml.edu

***Abstract*—Lossy compression on scientific data is coming into prominence as the scientific workflow is hampered significantly by large amounts of data produced by high-performance computing (HPC) applications. State-of-the-art lossy compressors, such as SZ and ZFP, show promising rate-distortion efficiency. However, as the data storage burden and need for feature-preserving compression continue to grow, relying on unitary or single-stage compression is becoming insufficient for obtaining desirable data reductions and feature preservation. This paper aims to improve the compression ratio by taking advantage of information retrieval (IR), a well-established topic but under-explored in lossy compression for scientific data. We propose our lossy compression technique, called DPZ, based on multi-stage feature extractions, a commonly employed step in IR. Unlike the prior works where the compression is either done by predicting or bit-plane encoding, this work focuses on preserving the key data content from each stage to the maximum extent, ultimately elevates the compression ratio. With the application of discrete cosine transform, principal component analysis, and quantization, DPZ obtains the dominant features with the least amount of bits possible. Specifically, a knee-point detection and an explained variance variation method are designed for finding optimal tradeoffs. DPZ also employs a sampling strategy to reduce computational overhead and estimate compressibility and parameters before compression. We evaluate the performance of DPZ using real-world scientific datasets. Experiments demonstrate that DPZ achieves superior compression ratios through multi-stage retrievals and outperforms SZ and ZFP at medium to high accuracy on most of the evaluated datasets.**

***Index Terms*—Lossy compression; information retrieval; PCA**

I. Introduction

Data compression is becoming crucial in HPC workflows, as even a single simulation run by modern HPC applications easily exceeds petabytes [1], [2]. For example, the data generated by the Large Hadron Collider (LHC) is expected to approach 150 PB/year by 2025 [3], potentially exabytes and beyond in cumulative storage volume. Lossless compression can alleviate such a burden but attaining an appreciable compression ratio is limited. As the simulations generate data far rapidly than the current scientific workflow can handle, lossy compression, on the other hand, is in an increasing need.

Error-controllable lossy compressors for scientific data generally can be classified into three categories. One is prediction-based, such as SZ [4]–[6], which relies on the prediction mechanism employed in the decorrelation stage. The second one is transform-based, such as ZFP [7], [8], TTHRESH [9], and DCTZ [10]–[12], which depends on efficient transforms to decorrelate original data. The last one is multigrid-based, such as MGARD [13]–[16], which decomposes data into multi-grid levels. Several state-of-the-art lossy data compressors, such as SZ and ZFP, have shown success in providing high compression ratios while bounding errors. However, the compression ratios achieved by unitary or single-stage lossy compressors are still far from being desired.

Luo et al. [17] recently proposed applying preconditioners on existing lossy compressors to improve the compression ratio. Specifically, they improved their prior work on [18] and implemented latent reduced models as a precondition on SZ and ZFP. Their approach showed high compression ratios but required a model selection strategy before compression as the reduced model does not apply to all datasets. Moreover, using preconditioners in conjunction with stand-alone compressors, such as SZ and ZFP, would show a lack of consistency in methodology (i.e., feature preservation), which might be insufficient for scientists to make production-level analyses.

As there is an increasing need for domain scientists to understand the "physics" or "features" in their simulations and analysis [1], a growing number of studies considered feature-relevant compression. For instance, a recent study by Fox et al. [19] explored the impact of lossy compression on the features of interest. The authors indicated that feature varies in distinct ways and suggested that implementing an algorithm to characterize data and select optimal compression parameters accordingly would be a rewarding path for the future of HPC simulations. Several studies [20], [21] also discuss the potential need for developing a feature-relevant compressor to improve the compression ratio compared to the growth of the data volumes. However, their analyses are based on the compression results of existing compression schemes, which are not purposely designed for feature-relevant reduction.

In this work, complementary to the current lossy compression techniques and the feature preservation idea, we propose our lossy compressor, called DPZ. We implement our mechanism that preserves features by retrieving the highest information with the minimum data content, thereby achieving high compression ratios. Our key contributions are as follows:

- We study the properties of efficient retrieval methods and formulate their information preservation over the features of interest. We explore the viability and effectiveness of a different combination of retrievals and propose our framework with multi-stage feature extractions.
- We design a block decomposition strategy such that the compressor applies to arbitrary dimensional data.

By preserving the locality of the original data during decomposition and adopting the optimized block size, it improves compressibility through feature selection.

- We apply discrete cosine transform (DCT) on the decomposed data and implement principal component analysis (PCA) to transform coefficients into eigenspace. To preserve the most information, we introduce our k-PCA method using knee-point detection and explained variance variation, thereby containing maximum variations through k selection while discarding less informative data contents. We also design a quantization scheme for the selected components to improve the compression ratio further. Lastly, we develop a sampling strategy to estimate preliminary reduction ratios and provide proper parameters before compression. More importantly, it would reduce computational overhead.
- Our experimental results show that, compared with SZ and ZFP, our proposed compressor DPZ, although simple, achieves superior compression ratios on tested real-world datasets at medium to high accuracy.

II. Motivation

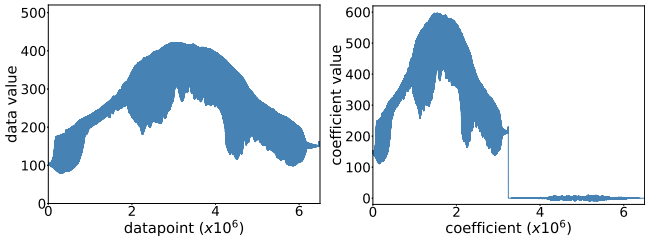
A. Information Retrieval

The intuition behind our compression strategies arises from information retrieval (IR), a system designed for finding information archives and search on written data. The basic idea of IR, while its definition can be rather broad, is to find data content of an unstructured nature, where data does not have a clear structure that satisfies the information needed from within large collections of data [22]–[25]. One of the commonly-used retrieval methods in IR [26] is feature extraction that typically utilizes techniques such as data transform, dimensionality reduction, and autoencoder [27]–[30]. Also, to obtain the desired information, the IR system usually includes several extraction stages, each of which involves a specific model for its data representation purposes, to acquire key content from different features (e.g., vector, index).

B. IR Methods in Lossy Data Compression

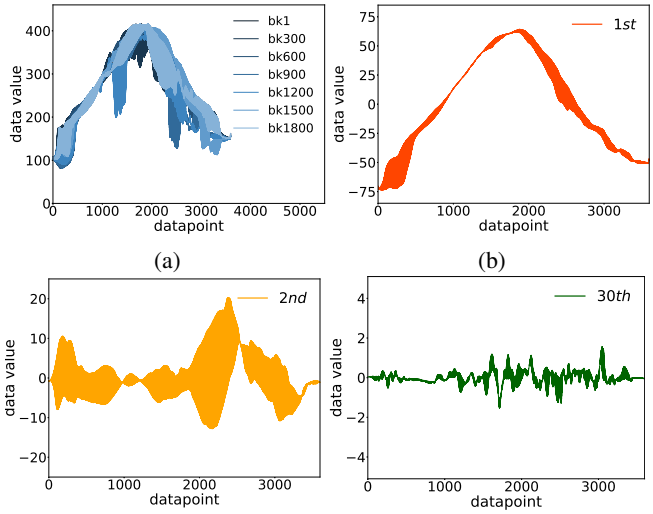
In the same way as IR systems, lossy data compression retrieves information through transforming data representations such that minimal bits preserve the dominant information of the original data. Simple arithmetic transforms such as fixed-point conversion could preserve meaningful data information through the truncation of unnecessary bits. Quantization, a local approach that transforms several with-in-range points into some approximations, can also be used for information retrieval. However, the compression ratios obtained by these transforms could be limited when data itself has less bit representation or has high dispersion.

Two types of discrete transforms are known to be effective for feature extraction: deterministic and statistical [31]. Deterministic transform, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT) having invariant basis vectors independent of the datasets, can simplify the data representation. For example, Figure 1a and Figure 1b show



(a) (b)

Fig. 1: The distribution of the FLDSC dataset in different forms. (a) flatten original data, (b) after discrete transform.



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

Fig. 2: (a) The overlay distribution of selected blocks (e.g., bk1: the 1st block-data) of FLDSC. The distribution of (b) the 1st, (c) the 2nd, and (d) the 30th components after PCA.

the distribution of CESM-ATM FLDSC and its re-arranged coefficients after applying a deterministic transform (DCT in this example), respectively. From Figure 1b, we can see that the trend exhibited in the left part of coefficients reproduces the shape of the original data. As a result, we can keep the information close to the original data by preserving only these coefficients and discarding the remaining ones.

Statistical transform, such as PCA and linear discriminant analysis (LDA) that have different basis vectors depending on the statistical specification of the datasets, is another effective feature extraction method. The unsupervised PCA is suitable for data compression. In PCA, data is projected from a high-dimension space to a low-dimensional space such that the most efficient data representation can be extracted. Figure 2a depicts the overlay of seven selected feature vectors (block-data) for PCA of 2D FLDSC (a total of 1800 blocks, and each includes 3600 datapoints). Figure 2b–2d show the distribution of PCA components after the projection. As we can see, the 1st PCA component (depicted in Figure 2b) captures an overall trend of the original overlay, and the remaining PCA components (e.g., 2nd and 30th) are less representative of the original. Thus, we need to retrieve the major information of the data by selecting the largest primary components.

C. Challenges on Scientific Data

Though utilizing feature extraction methods to retrieve important information in lossy data compression shows its effectiveness, it is still not straightforward to achieve promising performance on floating-point scientific data due to several challenges. First, data compression needs to achieve not only high compression ratios and but also high precision. Striking a balance between these two is not an easy task. Second, the complexity of data content varies across different applications and would require special knowledge to define or formulate features of interest associated with compression performance. Last, as no compressor can be designed best for all datasets, a preliminary compressibility estimation before compression is necessary for reference. These challenges motivate us to design a lossy data compressor with efficient rate-distortion performance and preliminary reduction estimation. Our compressor, which takes advantage of multi-stage feature extractions in IR, finds the desired data contents systematically and preserves the information required.

III. Exploring IR framework & Problem Formulation

In this section, we first present the unique properties of effective retrieval methods and formulate our problem in the context of feature representation and information preservation. Then, we explore the viability and effectiveness of different combinations of retrievals and identify the potential framework of multi-stage feature extractions.

A. Feature Representation & Information Preservation

1) Features: How to select a suitable feature and formulate its performance on information preservation is crucial. As in each retrieval method, a feature of interest has a different representation. For deterministic transform, coefficients could be the feature as they represent the datapoints in the transformed space. For statistical transform like PCA, components in the lower dimensional space could be the feature. To exploit these properties in our context, rather than finding an inherent data feature (i.e., application relevant) or a domain-specific one, which is beyond the scope of this work, we focus on transform-based features (i.e., feature vectors) that have high information preservation and can be quantified in terms of the numeric information loss over the selected number of features (to be illustrated later in Section III-A3). Accordingly, in our design, we aim at obtaining the least number of features that contribute to the most information required, thereby achieving high compression ratios and low errors. To achieve this goal, we identify suitable retrieval methods and useful metrics and formulate them on information preservation.

2) Feature Extraction Methods: As demonstrated in Section II-B, both deterministic and statistical approaches show desirable information retrieval, so we want to employ both methods in our design. Regarding deterministic transforms, prior studies [7], [10], [32]–[34] employed several transforms, such as DCT, DFT, DWT, etc. In this work, we use DCT (i.e., DCT-II) as our first desired retrieval method as it has demonstrated its effectiveness in lossy compression on real-world datasets. Also, some unique properties of DCT make it a powerful retrieval method. For example, the most crucial characteristic of DCT is that it takes correlated input data and concentrates its energy in just the first few transformed coefficients. DCT expresses a finite sequence of data points through a sum of the cosine function and can be written in a vector form as $z = A^T x$, where x is dataset and matrix A is an orthogonal matrix, i.e., $A^T = A^{-1}$.

Regarding statistical transform, we use PCA as a retrieval method in this category. Compared to other reductions such as singular value decomposition (SVD) and NMF, PCA is provenly efficient in generating the optimal linear transformation that projects data into space that is preferably close to its intrinsic dimension [35], [36]. Moreover, it supports inverse transformation, which is essential for reconstruction. Specifically, it performs an orthogonal transformation to the basis of correlation eigenvectors, and projects onto a subspace of s-dimension spanned by those eigenvectors $p_1, p_2, p_3 ... p_s$, which correspond to the largest eigenvalues $\lambda_1, \lambda_2, \lambda_3 ... \lambda_s$. The s principal components of an t dimensional original feature x can be written in a vector form as $y = D^T x$, where D is an $t \times s$ matrix ($s < t$).

As the computational overhead is also critical to the compression performance, we consider combining these two discrete transforms only in our multi-stage framework.

3) Information Preservation: To see how information is retrieved (a.k.a., preserved) over the selected (a.k.a, extracted) features in each method of our choice, we formulate it via the number of selected features versus *information* preserved by those. In general, information (or energy) can be measured using metrics such as gain or entropy. However, as the target information obtained by each method is independent, we formulate them separately. We start by representing the dataset as x. The total number of features in the transformed dataset is m, and k is the selected number of features ($k \leq m$).

In DCT, conventional methods like zigzag, zonal masking, and discrimination power analysis (DPA) are often used in image compression for selecting features. But as for scientific data, the energy compaction rate (ECR) [32], a function of preserved energy regarding the number of largest magnitude transform coefficients, is more appropriate. The energy is calculated as the sum of squares of individual values, and the ECR is denoted as:

$$ECR = \frac{\sum_{i=1}^{k} |f_i|^2}{\sum_{i=1}^{m} |f_i|^2}, \tag{1}$$

where f is the transformed format of dataset x.

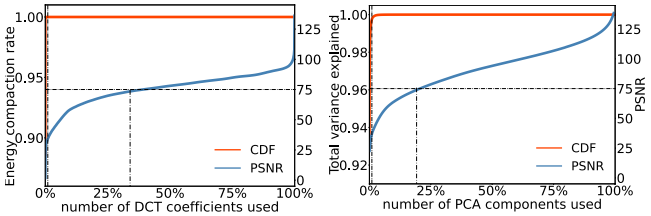
In PCA, on the other hand, principal components represent the directions of the data that explain the most variance (i.e., the degree of spread). In other words, each principal component represents the line that captures most information of the data. Therefore, the greater the variance kept by a line, the larger the dispersion of the data points associated with it, thus the more information it has. The total variance explained (TVE) by the number of largest primary components (also

called cumulative proportion of variance explained) can be used to see how much information is retrieved over the selected features. It is denoted as:

$$TVE = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i}, \tag{2}$$

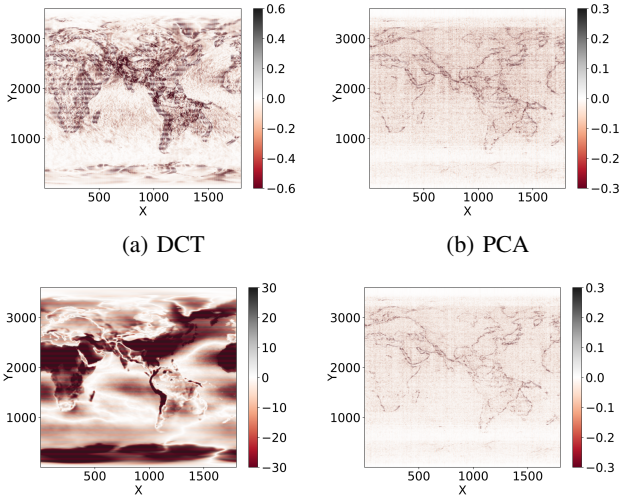
where λ_i is the eigenvalues that calculate how much variance (the average of the squared differences from the mean) can be explained by its associated eigenvector p_i.

4) Compression Evaluation: To assess the potential impact on the compression performance of each method, we formulate the correlation between the number of selected features and their compression performance. We use Peak Signal-to-Noise Ratio (PSNR), a commonly used metric on lossy data compression, to assess the compression quality. PSNR is expressed in terms of the logarithmic decibel scale and is equal to $20 \times log_{10}(\text{data range}) - 10 \times log_{10}$ (Mean Squared Error). Figure 3 shows the relationship between the number of selected features versus its preserving information and versus PSNRs by applying DCT and PCA separately on the FLDSC dataset. As shown in the figure, only 1% of features can contain more than 90% of the information, measured in cumulative ECR and TVE in both methods. Moreover, around 35% and 20% of features of DCT and PCA, respectively, could achieve a PSNR of 75 dB. This result demonstrates the effectiveness of both methods for attaining high information compaction (and ultimately high compression ratios) and indicates that a mechanism to find the optimal tradeoff between compression ratios and compression quality is needed. The question then arises if the combination of DCT and PCA could construct a better feature extraction approach with both advantages as in IR systems where a combination of retrieval methods is usually employed for desired information.



(a) (b)

Fig. 3: Comparison of two retrieval methods: DCT and PCA. The primary y-axis shows the CDF of information preservation, and the secondary y-axis shows the PSNR.



(a) DCT (b) PCA

(c) DCT on PCA components (d) PCA on DCT coefficients

Fig. 4: Visualization of errors with different transforms on 2D FLDSC dataset with the compression ratio of 5X.

B. Retrieval Framework

1) Combination of Methods: To evaluate the feasibility of combined transforms, we separately apply DCT, PCA, DCT on PCA components (DCT on PCA), and PCA on DCT coefficients (PCA on DCT) on the FLDSC dataset. For a fair comparison, we use the fixed compression ratio (the original size divided by the compressed size) of 5X. In other words, we keep 20% of major features and discard the remaining, and compare the errors between the original and the reconstructed one. While combining other retrieval methods afterward (e.g., quantization or bit reordering) than these transforms alone generate higher compression ratios than 5X, this comparison result serves to emphasize the impact of various combinations of transforms on extracting features on scientific datasets. Figure 4 visualizes the absolute error introduced by different combinations of transforms. Specifically, Figure 4a–4b show the error introduced by single-stage feature extraction, while Figure 4c–4d display the one from two-stage. Surprisingly, given the same approximation compression ratio (5X), DCT on PCA components generates the most errors, while PCA on DCT proves the least. This result illustrates the combination of retrieval methods (i.e., PCA on DCT) somehow enhances the compression quality, but how we can elevate the performance further with such a combination needs an investigation.

2) PCA in DCT domain: We start by discussing why PCA on DCT is more effective than others. First, both DCT and PCA are orthogonal linear transformations where the transformed data is symmetric [37]. Compared to DCT, PCA is more effective as it is the optimal linear transformation (operates over the correlation matrix) for normally distributed data [38] (normality assumption). As proved in the study [39] that the coefficients of block DCT are normal distribution, applying PCA on energy concentrated DCT coefficients is more effective than data in its original domain. This property is also proved in [38], [40] that adopting DCT before PCA yields the same results as those obtained from the spatial domain. Second, compared to DCT on PCA, PCA on DCT is more viable as proved in face recognition and image compression [41], [42]. This feasibility is because the transformation matrix of DCT is universal such that it can be used to the following stage, while the fixed set of eigenvectors obtained from the original data in PCA could not approximate data well in the other domain [43]. Lastly, PCA can be mathematically proved to be directly implemented in the DCT domain, as follows.

Suppose the original data $X = [x_1, x_2, ..., x_n]^T$ can be de-

noted as an n-dimensional random vector, the PCA projection matrix $D = [d_1, d_2, ..., d_m]$ can be acquired by eigenanalysis of the covariance matrix of X, V_X, which is denoted as:

$$V_X d_i = \lambda_i d_i, i = 1, 2, ..., m, (m < n), \tag{3}$$

where $V_X = E[(X - \bar{X})(X - \bar{X})^T]$ and $\bar{X} = E[X]$.

Suppose further that the original data is transformed by a DCT orthogonal matrix A, then the covariance matrix V_Z of the transformed random vector Z can be obtained by:

$$\begin{aligned} V_Z &= E[(Z - \bar{Z})(Z - \bar{Z})^T] \\ &= E[(A^T X - A^T \bar{X})(A^T X - A^T \bar{X})^T] \\ &= A^T E[(X - \bar{X})(X - \bar{X})^T] A \\ &= A^T V_X A. \end{aligned} \tag{4}$$

Therefore, the PCA projection matrix $\tilde{D} = [\tilde{d_1}, \tilde{d_2}, ..., \tilde{d_m}]$ for the DCT orthogonal transformed data can be acquired by eigenanalysis of the covariance matrix V_Z:

$$V_Z \tilde{d_i} = \lambda_i \tilde{d_i}, i = 1, 2, ..., m. \tag{5}$$

By substituting Equation 4 into Equation 5 and using $A^T = A^{-1}$, we can obtain the relationship of the eigenvector of the covariance matrices V_X and V_Z,

$$d_i = A\tilde{d_i}. \tag{6}$$

Based on Equation 6, we conclude that the PCA projection matrices D and $\tilde{D}$ (PCA in DCT domain) satisfy $\tilde{D} = A^T D$, which proves that we can directly implement PCA in the DCT domain. We can also prove a similar projection result by using 2D DCT conversion, where the DCT on 2D matrix $M \times N$ can be computed using separable 1D row and column transformations, i.e., $Z = A_M^T X A_N$, $X = A_M Z A_N^T$.

It is noteworthy that, while we use DCT as an input to PCA, PCA in other transform domains (e.g., wavelet transforms) should also work if the coefficients show normality, high information preservation, and can be mathematically proved for direct implementation.

3) Multi-stage Framework: As we proved that PCA could be directly implemented on DCT coefficients, a direct benefit is that we can skip the inverse DCT transform and therefore decrease the computational cost (DCT-II in this work is lossless and reversible). The second advantage is that the feature selection step only occurs in one stage (PCA in DCT domain) rather than two, thereby potentially reducing the computation complexity of the multi-staged method. Moreover, the compression accuracy could be improved as shown in Figure 4. Since the combination projection result is still orthogonal, adding additional stages could elevate the performance further. However, there needs to be a mechanism that can select the optimal compression parameters.

IV. Proposed Lossy Compressor

In this section, we present the design of our lossy compressor DPZ as illustrated in Figure 5. The framework consists of three stages of retrievals: data decomposition and transformation, k-PCA selection, and quantization and encoding. The

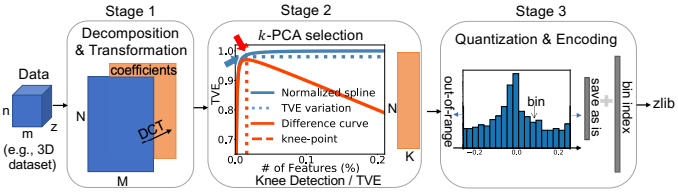


Fig. 5: An overview of our proposed compression framework.

decompression is in reverse order. Moreover, DPZ incorporates a lossless compression add-on and a sampling strategy for compression improvements.

A. Data Decomposition and Transformation

We first decompose the multidimensional data into a block-based 2D matrix (i.e., M 1D blocks with N datapoints). We achieve this by converting the data into 1D and rearrange them into 2D while maintaining the original data order, where the value of $N \times M$ is equal to the size of the flattened data. Then, we apply DCT transform to each block. This block-based method reduces the 2D DCT transformation time (using parallelism on blocks) and makes our compressor applicable to arbitrary dimensional data. We note that in PCA feature selection, the dimensionality (i.e., the number of blocks) should be smaller than the number of samples (i.e., the number of datapoints). Therefore, we set M smaller than N. We also note that the number of blocks would affect compressibility as well as parallelism. And our empirical analysis shows that under the condition that $M < N$, the larger the M is, the higher the compression ratios. Therefore, $\frac{N}{M}$ is set to the smallest common divisor (greater than 1). For example, for a 3D data size of $128 \times 128 \times 128$ ($n \times m \times z$), M is calculated as $2^{(log_2 \sqrt{n \times m \times z})//2} = 1024$ and N is equal to $n \times m \times z / 1024 = 2048$, where the smallest common divisor is 2 (which is equal to N/M). Another advantage of our decomposition is that the sequence order of each block preserves the locality of data, which could improve the compression ratios further during feature extraction, especially on the smooth dataset. The reason is that PCA is based on Pearson correlation coefficients, so the higher the linearity between features, the larger the dimension is reduced.

B. k-PCA Selection

Next, we implement PCA on the transformed DCT block data. We note that a normalization or standardization step is often needed before transforming data into its eigenspace as the range of each feature's values varies largely. However, such scaling would redistribute the *weight* of the variance in our case, as the features (i.e., the decomposed block-data in the DCT domain) have the same unit norm. Therefore, we only apply it to low linearity data (determined based on sampling, which will be explained in Section IV-D).

As demonstrated in Section III, variance explained measures how much information is retrieved over the selected components. In this stage, we introduce our k-PCA selection

approach to preserve information to the maximum extent. We propose two methods: knee-point detection and explained variance variation, as illustrated in Algorithm 1.

1) Method 1: Knee-point Detection: We define knee-point as the optimal information retrieval point that can best balance the tradeoffs between compression ratio and compression accuracy. Specifically, it is the point of maximum curvature of our fitted cumulative total variance explained (TVE) curve. Depending on the precision requirement, the curve can be fitted either through a one-dimensional (1D) interpolation or polynomial (polyn) interpolation, the latter of which generates a smoother curve [32]. Mathematically, the knee-point is a local maximum that can be calculated as a function of the first and second derivatives of the spline curve and states situations where the degree of increase in the cumulative proportion of variance explained starts to decline [44]. By detecting the knee-point, we can obtain the number of k components to preserve. This method provides us an aggressive option for achieving the highest compression ratio (i.e., obtaining the least features) while retrieving the most worthy information. That is, beyond k, there would be a diminishing return in terms of compression ratio and accuracy. We note that this method does not require extra parameter tuning, meaning it finds an optimal solution automatically.

2) Method 2: Explained Variance Variation: Our second method to meet different accuracy required by different applications is to use the total variance explained (TVE) as a fine-tuning parameter. A 95% of TVE is the most commonly used threshold to select k (the first couple of largest components) in the general case. In our design, we start with the threshold of 99% ("two-nine") and alter it from 99.9% ("three-nine") to "eight-nine" to adjust the compression quality. Based on our empirical analysis, "eight-nine" is strict enough for generating high compression quality. But we need to tighten the threshold at this stage, as the selected k features can be compressed further in the following stage (i.e., quantization). This method provides a dynamic option of tuning compression accuracy, which supplements our knee-point detection mechanism.

We note that these two methods perform slightly differently in terms of compression ratio and accuracy, which allow domain scientists to choose the one best suited for their application. We also note that both methods can be applied to the compression performance curve (i.e., PSNR curve in Figure 3) to generate compression results with high accuracy, but it requires a time-consuming reconstruction step. To improve the speed of this stage, we propose k selection in sampling (to be illustrated later in Section IV-D).

Algorithm 1 k-PCA Selection.

Input: Dataset X with M number of features.
Method 1: knee-point detection; Method 2: explained variance variation.
tve: user-defined total variance explained.
sf: spline fitting method (1D or polynomial interpolation).
Output: k: selected components.
1: Apply PCA on dataset X and generate the cumulative TVE curve f.
2: **if** choose Method 1 **then**
3: fit f with selected fitting method sf to preserve the shape (denoted as s_f).
4: normalize s_f to the unit square.
5: determine the value k from $K_{s_f}(x) = \frac{s_f''(x)}{(1+s_f'(x)^2)^{1.5}}$.
6: return k (first detected local maxima).
7: **else if** choose Method 2 **then**
8: **for** $k = 1, 2, \ldots, M$ **do**
9: **if** $TVE \geq tve$ **then** return k
10: **end if**
11: **end for**
12: **end if**

C. Quantization and Encoding

As proved in Section III-B, PCA on DCT follows a normal distribution where the values of our k-PCA are symmetric around zero. This property plays a critical role as it makes quantization effective for selected k-PCA, ultimately improving the compression ratio further. Specifically, we design a uniform quantizer where the with-in-range datapoints of the k component will be saved as its corresponding code value (determined by its represented bin index and its center value), while the out-of-range ones will be saved as is. The bin index is mapped to an integer encoded as either 1-byte unsigned char or 2-byte unsigned short integer depends on the needs. We define the bounding range symmetric about zero with each half equals to $P \times B$ and set the width of each bin equals to $2P$, where B is the number of bins and P is the defined error bound (i.e., 1E-3, 1E-4, etc.). The difference between the approximated value and the original one is bounded within P. We note that this error bound is designed only for approximation on k-PCA. Taking advantage of such a symmetric and equal-width quantization, we apply zlib [45] (a lossless compressor) to compress the indexing and out-of-range datapoints further. The combination with zlib is efficient in terms of speed and indexing and does not involve any computationally intensive tasks.

It is worth mentioning that, unlike studies that apply preconditioners on existing lossy compressors, our algorithm is information-oriented, where the reconstruction at any level shows consistency of the original data itself. In other words, we not only avoid computing "delta" (the differences between reconstruction on the reduced model and original data) and applying inverse transformation in compression, but also consistently extract dominant features without creating extra redundancy, thus potentially improves the compression ratio.

D. Sampling Strategy

We note that PCA can be computationally expensive (with time complexity of $O(min(M^3, N^3))$ of searching the directions of maximum variance), especially when the number of features (M) and the number of datapoints (N) are large. However, if we have a way of selecting k based on a "priori" (i.e., sample set), then the complexity of PCA can be significantly reduced. To accomplish this goal, we develop a sampling strategy, as shown in Algorithm 2. Our sampling strategy reduces the variance searching time, provides proper compression parameters (i.e., k selection), and estimates the preliminary data compressibility.

1) Sampling Algorithm and Parameter Selection: Our sampling algorithm is composed of the following steps. We first divide the block-data Y into S (10 by default) subsets. Next,

Algorithm 2 Our proposed sampling Strategy.

Input: Dataset Y with M number of features.
SR: sampling rate.
S: the number of subsets.
T: the number of random pick subsets.
tve: user-defined total variance explained.
Output: k_s: estimated k, CR_p: estimated preliminary compression ratio, VIF: variance inflation factor.
1: Generate sampling data S_y based on SR for compressibility estimation.
2: Calculate VIF of S_y. If $VIF < 5$, apply standardization in Algorithm 1.
3: Divide dataset Y into S subsets, randomly pick T subsets as sample data.
4: Compute the variances of sample data (S_{R1},S_{R2},...,S_{RT}) and choose their corresponding k ($k_{S_{R1}}$, $k_{S_{R2}}$,..., $k_{S_{RT}}$) based on tve.
5: Estimate k_e as ($k_{S_{R1}}$+$k_{S_{R2}}$+...+$k_{S_{RT}}$)//T.
6: Estimate preliminary CR as CR_p=$CR_{stage1\&2} \times CR'_{stage3} \times CR'_{zlib}$, where $CR_{stage1\&2}$=k_e/M.

we randomly pick T (3 by default) subsets as sample data (S_{R1},S_{R2},...,S_{RT}), compute their variances, and select their corresponding k based on defined tve. We then obtain the estimate k_e by averaging the value of $k_{S_{R1}}$, $k_{S_{R2}}$,...,$k_{S_{RT}}$. Our empirical observation suggests that the computation over the first S_f, the middle S_m, and the last subsets S_l as sample data usually gives a more proper estimation on high linearity block-data due to its locality and our decomposition mechanism. Therefore, if S is set to 10, k_e equals to the average value of k_{S_1}, k_{S_5} and $k_{S_{10}}$. This algorithm provides a way of selecting leading principal components (k_e) based *only* on the sample data, such that it reduces the compression time (Stage 2) for the remaining. Specifically, when $k \ll min(M, N)$, the time complexity of k-PCA can be reduced to $O(k^3)$. Though the estimated k_e is not strict on tve for the remaining subsets, it would not affect much of their compression performance in terms of rate-distortion (to be described in Section V). Overall, selecting k based on sampling provides a constant compression ratio performance, while selecting k based on tve offers us a stable compression quality.

2) Compressibility and Compression Ratio Estimation: The compression ratio achievable by our k-PCA algorithm is highly related to the linearity between features. So we introduce the variance inflation factor (VIF), which allows detecting the collinearity between features, as a compressibility indicator in our compressor. Specifically, we randomly generate sampling data with a sampling rate of SR on block-data and calculate its VIF value. VIF is calculated as $1/(1-R^2)$, where R^2 is a statistical measure of how well a single feature can be described by others. Unlike the Shannon entropy, which estimates the inherent data information level, VIF quantifies how much the variance is increased due to collinearity which is more suitable for compressibility prediction on DPZ. And as higher VIFs produce high compression ratios in Stage 2 (k-PCA), smaller ks are then needed for obtaining high TVEs. Furthermore, based on our empirical results, DPZ can estimate the overall compression ratio by multiplying the approximated reduction factors of each stage: CR_p= $CR_{stage1\&2} \times CR'_{stage3} \times CR'_{zlib}$, where $CR_{stage1\&2}$ equals to k_e/M, the approximate CR'_{stage3} ranges between 1.9X to 2.5X and the approximate CR'_{zlib} is around 1.25X in general based on tested datasets (to be evaluated in Section V-C6).

V. Evaluation

A. Experiment Setup and Evaluated Schemes

We conduct our experiments on a dual-core Intel i5 CPU with 8GB RAM running at 2GHz. We evaluate our proposed compressor DPZ against two state-of-the-art lossy compressors, SZ v2.0 and ZFP v0.5.5. We assess our DPZ using the following two schemes:

- **DPZ-l**: DPZ (loose) with P of 1E-3 and 1-byte indexing.
- **DPZ-s**: DPZ (strict) with P of 1E-4 and 2-byte indexing.

Based on different performance demands, both schemes can be used in conjunction with either knee-point detection (a high compression ratio oriented compression by finding the optimal point) or explained variance variation (an error-aware compression by satisfying required variance).

B. Datasets and Metrics

We evaluate the compression performance of nine scientific datasets generated from three real HPC applications [49] summarized in Table I. The comparison is based on rate-distortion, compression accuracy (i.e., PSNR) versus bit-rate (inversely proportional to compression ratio (CR)), a critical metric used in evaluating the overall compression quality. 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., 32-bit for single precision) to express each original data point divided by the overall compression ratio.

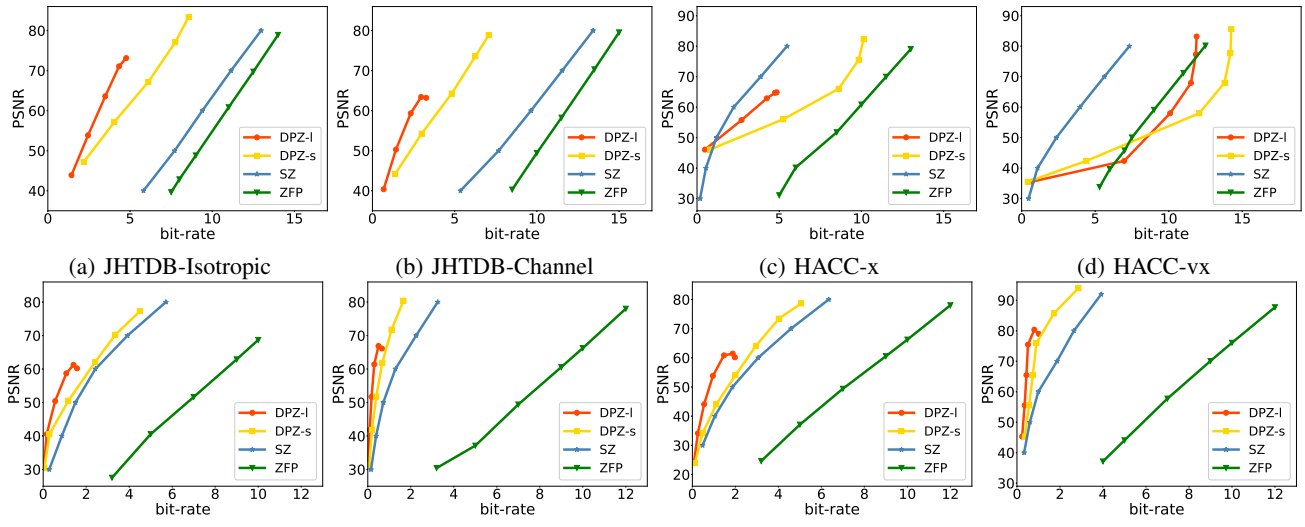
C. Evaluation Results

1) Rate-distortion Comparison: Figure 6 presents the rate-distortion of different compressors on nine datasets (CLDLOW shows a similar result to CLDHGH, thus not presented here). Specifically, we vary the TVE from "three-nine" to "eight-nine" (explained variance variation) on DPZ and evaluate SZ and ZFP based on their configurations to achieve similar PSNR for a fair comparison. By the definition of rate-distortion, the curve on the upper left part with a higher positive slope has better compression performance than the ones on the lower right with a lower positive slope. Therefore, we observe that DPZ-l outperforms DPZ-s on most evaluated datasets but exhibits a limitation of achieving high PSNR when TVE is approaching "eight-nine". On the other hand, DPZ-s shows a steady and competitive performance, particularly on the JHTDB (3D) and CESM (2D) datasets compared with SZ and ZFP. This result is compelling as HACC (1D) proves to be less compressible (VIF lower than cutoff value) using DPZ discussed later in Section V-C6. Overall, DPZ achieves superior compression ratios (preferable on high dimensional datasets) compared with SZ and ZFP at medium to high compression accuracy (PSNR in between 30 dB and 90 dB).

2) Compression based on Knee-point Detection: To find the best tradeoff points of DPZ, we show the compression performance of both schemes based on knee-point detection in Table II. We present only six datasets here due to space limitation, and the performance of the remaining datasets is consistent with the evaluation results shown in Figure 6. As

TABLE I: Scientific datasets and their descriptions.

Source	Dataset Name	Type	Dimension	Size	Format
JHTDB [46]	"Isotropic1024-coarse","Channel"	Turbulence simulation	128 × 128 × 128	5.04GB	32-bit float
CESM-ATM-Taylor [47]	"CLDHGH","CLDLOW","PHIS","FREQSH","FLDSC"	Climate simulation	1800 × 3600	1.47GB	32-bit float
HACC [48]	"x","vx"	Cosmology particle simulation	2097152	496MB	32-bit float



(a) JHTDB-Isotropic (b) JHTDB-Channel (c) HACC-x (d) HACC-vx

(e) CESM-CLDHGH (f) CESM-FLDSC (g) CESM-FREQSH (h) CESM-PHIS

Fig. 6: Comparison of rate-distortion using different lossy compression methods on selected datasets.

TABLE II: Compression performance based on knee-point detection with different interpolations on selected datasets.

	metric	Isotropic		Channel		CLDHGH		PHIS		HACC-x		HACC-vx	
		1D	polyn	1D	polyn	1D	polyn	1D	polyn	1D	polyn	1D	polyn
DPZ-l	CR	55.75	22.73	**110.38**	26.13	**186.89**	38.87	**155.62**	54.27	42.64	28.08	14.88	9.00
	PSNR	12.34	12.79	8.08	8.66	37.20	48.09	42.04	76.09	47.33	49.38	37.75	39.15
	mean θ	1.94E-1	1.84E-1	2.76E-3	2.60E-3	9.83E-3	2.69E-3	4.48E-3	9.06E-5	3.15E-3	2.48E-3	8.83E-3	7.78E-3
DPZ-s	CR	65.75	23.89	**127.44**	25.51	**160.71**	29.62	**113.87**	34.52	47.34	49.29	25.11	15.00
	PSNR	12.34	12.85	8.08	8.66	37.20	48.09	42.04	76.70	47.34	49.29	37.74	39.15
	mean θ	1.94E-1	1.83E-1	2.51E-3	2.60E-3	9.83E-3	2.69E-3	4.48E-3	8.38E-5	3.15E-3	2.50E-3	8.83E-3	7.77E-3

TABLE III: Breakdown of compression ratio on selected datasets.

Stage	TVE	Isotropic		Channel		CLDHGH		PHIS		HACC-x		HACC-vx	
		DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s
Stage 1&2	99.9%	8.192		13.128		34.615		30.508		16.126		1.196	
	99.999%	2.107		2.473		3.704		13.953		1.218		1.005	
	99.99999%	1.260		1.316		1.727		4.687		1.006		1.001	
Stage 3	99.9%	2.335	1.989	2.494	1.991	2.872	1.985	2.836	1.963	3.667	1.997	2.000	1.991
	99.999%	3.251	1.997	3.387	1.998	3.697	1.998	3.128	1.983	3.972	2.000	2.000	1.992
	99.99999%	3.515	1.998	3.649	1.999	3.853	1.999	3.625	1.994	3.977	2.000	2.000	1.992
zlib	99.9%	1.078	1.168	1.407	1.258	1.979	2.588	1.543	1.458	1.132	1.269	1.326	1.113
	99.999%	1.255	1.313	1.590	1.452	2.174	3.329	1.667	1.571	1.559	1.526	1.343	1.125
	99.99999%	1.453	1.440	1.994	1.664	3.057	4.835	2.337	1.993	1.650	1.569	1.344	1.126

TABLE IV: Accuracy loss between stages in terms of Δ PSNR (dB) on selected datasets.

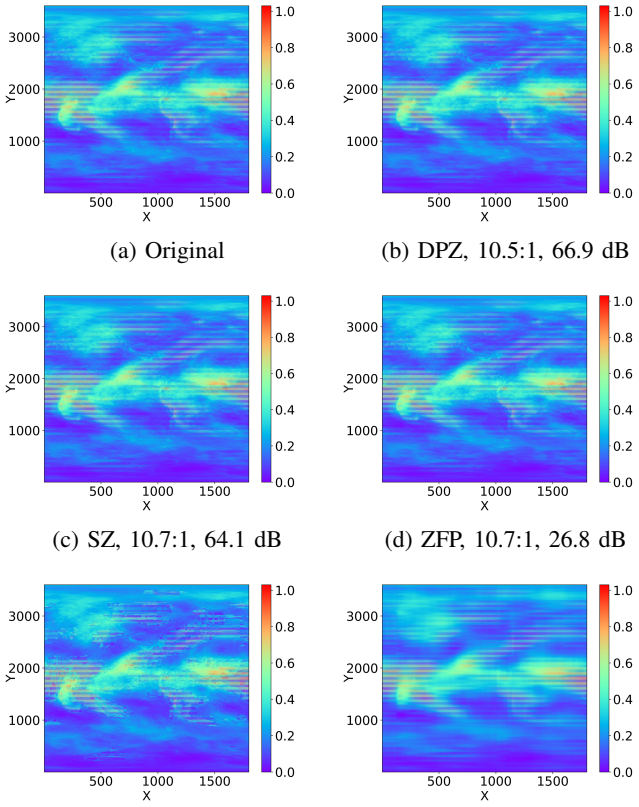
TVE	Isotropic		Channel		CLDHGH		PHIS		HACC-x		HACC-vx	
	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s	DPZ-l	DPZ-s
99.9%	0.001	0.001	0.688	0.673	0.062	0.059	0.001	0.001	0.003	0.001	0.005	0.001
99.999%	0.321	0.002	1.009	0.016	1.756	0.026	0.037	0.001	3.201	0.047	0.518	0.006
99.99999%	11.462	0.546	16.758	1.386	20.309	3.287	5.461	0.110	20.892	3.460	2.522	0.330

expected, DPZ with knee-point detection produces aggressive CRs. In particular, DPZ could achieve CRs above 100X on Channel, CLDHGH, and PHIS while obtaining a reasonable PSNR and average relative error θ (data-range based error). We observe that the compression accuracy of both schemes is similar, but for JHTDB (Isotropic and Channel) and HACC, DPZ-s shows higher CRs than DPZ-l, while for CESM, DPZ-l shows higher CRs than DPZ-s. This result illustrates that there could be potential for improving the compression ratio further with proper parameter settings in quantization and encoding. We also notice that the polynomial interpolation (polyn) curve fitting improves the compression accuracy but reduces the CR to a certain degree (between 1.5X and 5X lower).

3) Performance Breakdown: The compression ratio of DPZ is calculated by multiplying each stage's reduction factor. To investigate each stage's contribution to the compression performance, we break down the change of CR and PSNR as shown in Table III and Table IV, respectively. We can make several observations from these results. First, at Stage 1&2 (data decomposition with DCT & k-PCA), CLDHGH and PHIS are more compressible than other datasets, while HACC-vx is the hardest to compress. Second, the CR changes significantly with varying TVE. We notice that at Stage 3 (quantization and encoding) and the zlib stage, the CR improves when TVE increases as more components are selected and thus quantized and encoded accordingly. The CR of DPZ-l at Stage 3 is higher than 2X, but no more than 4X on most datasets (except HACC-vx), and the CR of DPZ-s is close to 2X. The CR by zlib on both schemes ranges from 1X to 5X. On an average, the CR by zlib is 1.42X, 2.38X, and 1.2X on JHTDB, CESM, and HACC, respectively. Overall, when TVE varies from "three-nine" to "seven-nine", the compression ratio reduces at Stage 1&2 but increases at Stage 3 and the zlib (lossless) stage.

Table IV, on the other hand, presents the accuracy loss in terms of Δ PSNR (dB) between Stage 1&2 and Stage 3. We notice that the accuracy drops when TVE increases, especially on DPZ-l (as it has higher CR than DPZ-s as shown in Table III). These observations indicate that DPZ could achieve high CR from Stage 3 and zlib without much information loss in Stage 1&2 when TVE gets tight. DPZ could also achieve high CR from Stage 1&2 without much information loss in Stage 3 when TVE gets loose. Overall, DPZ achieves the best trade-off between high accuracy by selecting DPZ-s and high CR (or more aggressively in conjunction with knee-point detection) by selecting DPZ-l.

4) Visualization: Figure 7 displays the visualization results of the original data and the decompressed data from different compressors on dataset CLDHGH. Figure 7b-7d visualize the decompressed data when CR is around 10.5X. According to the figures, all visualization results look similar to the original data shown in Figure 7a. However, significant differences indeed exist in much smaller regions (hard to visualize as a whole because human eyes are less sensitive to minor changes) in terms of compression accuracy. We note that the visualization presents here is for an overall picture of the compression result, and the numeric compression result (i.e., rate-distortion) is still our primary concern. Specifically, in this case, when CR is around 10.5X, DPZ (i.e., DPZ-s) achieves a PSNR of 66.9 dB, SZ achieves 64.1 dB, and ZFP achieves 26.8 dB. Figures 7d-7f visualize the decompressed data when PSNR is around 26 dB. As we can see in the figure, ZFP displays the most accurate visualization result, while SZ and DPZ show the fidelity loss at certain degrees. However, SZ achieves a CR of 154.5X, and DPZ achieves a CR of 489.1X, 14.4X and 45.7X higher than ZFP in the presented case, respectively. Overall, the visualization effect shows that DPZ preserves the data information (e.g., edges, trends) well and shows the smooth details of the original data.



(a) Original (b) DPZ, 10.5:1, 66.9 dB

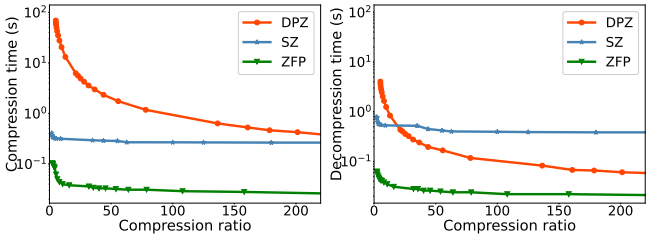
(c) SZ, 10.7:1, 64.1 dB (d) ZFP, 10.7:1, 26.8 dB

(e) SZ, 154.5:1, 26.5 dB (f) DPZ, 489.1:1, 27.8 dB

Fig. 7: Visualization of the CLDHGH dataset.(a) original, (b)-(f) decompressed from different compressors.

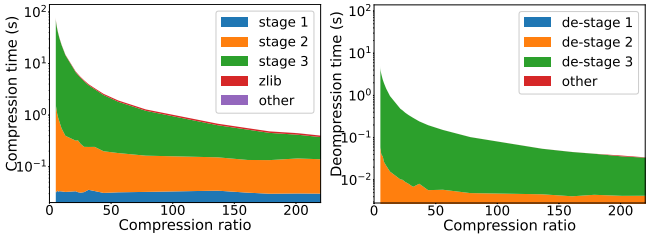
5) Compression Throughput: While the goal of DPZ is to extract the minimal data that contains the highest information, it incurs some computation overhead. To understand overhead incurred by DPZ in detail, we plot the compression and decompression time versus compression ratio of three compressors shown in Figure 8 (Isotropic datasets as an example). DPZ (a similar trend in both schemes) is slower than SZ and ZFP in compression throughput but narrows the gap in decompression throughput, particularly when CR (shown in x-axis) increases. Our experiment results show that DPZ in conjunction with our sampling strategy improves the overall compression speed by 1.23X, on average, compared with the non-sampling DPZ, on the evaluated datasets. To give more insight into the computational overhead incurred by DPZ, we break down the compression time. As shown in Figure 9, Stage 2 and Stage 3 contribute most of the time cost as PCA and quantization are highly dependent on the dimension of the coefficients. We note that parallelization can be applied to DPZ to reduce the computational cost, as our compression mechanism is block-based. Specifically, our quantization and encoding mechanism (Stage 3) can be easily parallelizable without any communication among the distributed blocks.

6) Evaluation of Sampling Strategy: Figure 10 shows the VIF distribution of HACC-vx, Isotropic, and PHIS. We set



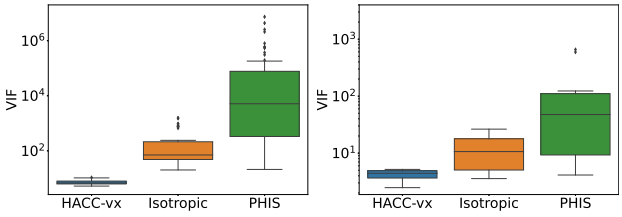
(a) Compression (b) Decompression

Fig. 8: Comparison of compression time.



(a) Compression (b) Decompression

Fig. 9: Breakdown of compression time of DPZ.



(a) SR=2.5%. (b) SR=1%.

Fig. 10: VIF of sampling datasets.

SR as 2.5% and 1% to estimate the data compressibility. As shown in the figures, the average VIFs (middle line of the boxplot) of HACC-vx are smaller than those of Isotropic and PHIS, which is consistent with the results presented in Figure 6. HACC-vx shows a relatively low VIF (lower than the commonly used cutoff value of 5) when only 1% of data is selected. Since both figures show a clear distinction in the VIF distribution, we consider that 1% is fair enough to estimate the potential compressibility of data in DPZ.

We then test our parameter selection algorithm by setting S (the number of subsets) to 5 and 10 (with corresponding TVE of "five-nine" to "seven-nine") and estimate CR_p based on k_e. For example, to obtain a TVE of "five-nine", Stage 2 needs a k_e of 12 (S of 10) from a total M of 1800, thus gives a CR_p ranges from 35.63X to 46.88X. By using k_e, DPZ achieves the final compression ratio of 45.4X, which falls within the range of CR_p. Our evaluation result shows that there is a 76.6% of chance that the overall compression ratio falls into the CR_p range when S is set to 10, whereas a 63.3% of chance when S is set to 5. In other words, higher S produces a higher prediction accuracy on compressibility. This result demonstrates the effectiveness of our sampling strategy on parameter selection with preliminary compressibility estimation.

VI. Related Work

Lossy compression for scientific data has recently received much attention due to its promising compression results and tolerance of minor numerical errors in scientific applications. SZ and ZFP are the most well-known compressors. SZ [4]–[6] estimates the compression value using a linear scale to quantize the difference into user-set error bound. It has implementations on CPU and GPU, which is now a modular parametrizable compression framework. ZFP [7], [8] fixes the compression rate based on an embedded coding scheme and is designed for datasets with a dimension of up to 4D. DCTZ [10]–[12] has an adaptive quantization with two specific tasks and can achieve high compression ratios on double-precision data. It is the predecessor of DPZ. MGARD [13]–[16] provides different norms to control data distortion and offers a high degree of compression flexibility. TTHRESH [9] is a tensor decomposition-based compressor that is designed for high dimensional visual data, which could achieve a high compression rate with smooth visual degradation. Unlike these studies, DPZ is information-oriented in different stages and provides optimal parameters accordingly. The implementation of multi-stage feature extractions in DPZ exhibits its superiority in terms of information retrieval. In particular, the mechanism of PCA in the DCT domain and the k selection approach play an important role in achieving high compression ratios, thanks to the linearity property of block-data.

VII. Conclusion

In this work, we propose a lossy compression technique, called DPZ, based on information retrieval. Specifically, we identify metrics for evaluating feature retrieval and information preservation, develop a multi-stage feature extractions (DCT, PCA, and quantization) framework, and implement an algorithm that selects optimal compression parameters. Moreover, we propose a sampling strategy that estimates compressibility and improves the compression speed. Our experimental results show that DPZ achieves competitive performance compared with SZ and ZFP on real-world datasets. We believe that our method is thus a good choice for compression on applications with reasonable error tolerance and high compression ratio achievement. In our future work, we plan to expand the DPZ algorithm to exploit parallelism for better scalability. We also plan to explore the PCA-type reduction model for speed improvement and analyze the effect of DCT coefficients truncation before applying PCA. Lastly, we plan to evaluate DPZ on more diverse datasets (e.g., non-linearly correlated ones).

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No.1751143 and the NVIDIA hardware grant. The authors acknowledge the MIT SuperCloud, MGHPCC and Lincoln Laboratory Supercomputing Center for providing (HPC, database, consultation) resources that have contributed to the research results reported within this paper.

REFERENCES

[1] R. Ross, L. Ward, P. Carns, G. Grider, S. Klasky, Q. Koziol, G. K. Lockwood, K. Mohror, B. Settlemyer, and M. Wolf, "Storage Systems and Input/Output: Organizing, Storing, and Accessing Data for Scientific Discovery. [Full Workshop Report]," 2019.

[2] C. Grelck, E. Niewiadomska-Szynkiewicz, M. Aldinucci, A. Bracciali, and E. Larsson, *Why High-Performance Modelling and Simulation for Big Data Applications Matters*. Springer, 2019, pp. 1–35.

[3] S. Habib, R. Roser, R. Gerber, K. Antypas, E. Dart, S. Dosanjh *et al.*, "High Energy Physics Exascale Requirements Review. An Office of Science review sponsored jointly by Advanced Scientific Computing Research and High Energy Physics," 11 2016.

[4] 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.

[5] 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)*, 2017.

[6] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello, "Error-Controlled Lossy Compression Optimized for High Compression Ratios of Scientific Datasets," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 438–447.

[7] P. Lindstrom, "Fixed-Rate Compressed Floating-Point Arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec 2014.

[8] J. Diffenderfer, A. Fox, J. Hittinger, G. Sanders, and P. Lindstrom, "Error analysis of zfp compression for floating-point data," 05 2018.

[9] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, "TTHRESH: Tensor Compression for Multidimensional Visual Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, Sept. 2020.

[10] J. Zhang, X. Zhuo, A. Moon, H. Liu, and S. W. Son, "Efficient Encoding and Reconstruction of HPC Datasets for Checkpoint/Restart," in *35th Symposium on Mass Storage Systems and Technologies (MSST)*, 2019, pp. 79–91.

[11] J. Zhang, J. Chen, A. Moon, X. Zhuo, and S. W. Son, "Bit-Error Aware Quantization for DCT-based Lossy Compression," in *IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–7.

[12] "DCTZ," https://github.com/swson/DCTZ, 2019.

[13] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel techniques for compression and reduction of scientific data—the unstructured case," *in preparation*, dec 2018.

[14] X. Liang, B. Whitney, J. Chen, L. Wan, Q. Liu, D. Tao *et al.*, "MGARD+: optimizing multi-grid based reduction for efficient scientific data management," *CoRR*, vol. abs/2010.05872, 2020. [Online]. Available: https://arxiv.org/abs/2010.05872

[15] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel techniques for compression and reduction of scientific data—the multivariate case," *SIAM Journal on Scientific Computing*, vol. 41, no. 2, pp. A1278–A1303, 2019. [Online]. Available: https://doi.org/10.1137/18M1166651

[16] ——, "Multilevel techniques for compression and reduction of scientific data-quantitative control of accuracy in derived quantities," *SIAM J. Sci. Comput.*, vol. 41, no. 4, pp. A2146–A2171, 2019. [Online]. Available: https://doi.org/10.1137/18M1208885

[17] H. Luo, D. Huang, Q. Liu, Z. Qiao, H. Jiang, J. Bi, H. Yuan, M. Zhou, J. Wang, and Z. Qin, "Identifying Latent Reduced Models to Precondition Lossy Compression," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 293–302, 2019.

[18] H. Luo, Q. Liu, Z. Qiao, J. Wang, M. Wang, and H. Jiang, "DuoModel: Leveraging Reduced Model for Data Reduction and Re-Computation on HPC Storage," *IEEE Letters of the Computer Society*, pp. 5–8, 2018.

[19] W. Fox, M. Wolf, J. Logan, J. Y. Choi, S. Klasky, and T. Kurc, "Feature-Relevant Data Reduction for In Situ Workflows," in *The 4th International Workshop on Data Reduction for Big Scientific Data (DRBSD-4)*, 2018.

[20] I. Yakushin, K. Mehta, J. Chen, M. Wolf, I. T. Foster, S. Klasky, and T. Munson, "Feature-preserving Lossy Compression for In Situ Data Analysis," in *International workshop on Performance modelling, Runtime System and Applications at the Exascale (EXA_PMRA20)*, 2020, pp. 10:1–10:9.

[21] J. Chen, D. Pugmire, M. Wolf, N. Thompson, J. Logan, K. Mehta, L. Wan, J. Y. Choi, B. Whitney, and S. Klasky, "Understanding performance-quality trade-offs in scientific visualization workflows with lossy compression," in *2019 IEEE/ACM 5th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-5)*, 2019, pp. 1–7.

[22] A. Singhal, "Modern information retrieval: A brief overview." *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001. [Online]. Available: http://dblp.uni-trier.de/db/journals/debu/debu24.html#Singhal01

[23] M. Sanderson and W. B. Croft, "The history of information retrieval research," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1444–1451, May 2012.

[24] S. T. Klein, "16 - data compression in information retrieval systems," in *Database and Data Communication Network Systems*, C. T. Leondes, Ed. San Diego: Academic Press, 2002, pp. 573–633. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124438958500180

[25] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[26] D.-A. Manolescu, "Feature Extraction - A Pattern for Information Retrieval," 1998.

[27] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, 2014, pp. 372–378.

[28] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56 – 70, May 2020. [Online]. Available: https://jastt.org/index.php/jasttpath/article/view/24

[29] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1048–1060, 2020.

[30] D. Birvinskas, V. Jusas, I. Martisius, and R. Damasevicius, "EEG Dataset Reduction and Feature Extraction Using Discrete Cosine Transform," in *Proceedings of the 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*. USA: IEEE Computer Society, 2012, p. 199–204. [Online]. Available: https://doi.org/10.1109/EMS.2012.88

[31] S. Dabbaghchian, M. P. Ghaemmaghami, and A. Aghagolzadeh, "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology," *Pattern Recognition*, vol. 43, no. 4, pp. 1431–1440, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320309004142

[32] J. Zhang, A. Moon, X. Zhuo, and S. W. Son, "Towards Improving Rate-Distortion Performance of Transform-Based Lossy Compression for HPC Datasets," in *IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–7.

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

[34] T. Mantoro and F. Alfiah, "Comparison methods of DCT, DWT and FFT techniques approach on lossy image compression," in *International Conference on Computing, Engineering, and Design (ICCED)*, 2017, pp. 1–4.

[35] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-Norm Principal-Component Analysis via Bit Flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, 2017.

[36] M. Fan, N. Gu, H. Qiao, and B. Zhang, "Intrinsic dimension estimation of data by principal component analysis," *CoRR*, vol. abs/1002.2050, 2010. [Online]. Available: http://arxiv.org/abs/1002.2050

[37] J. Shlens, "A tutorial on principal component analysis," *CoRR*, vol. abs/1404.1100, 2014. [Online]. Available: http://arxiv.org/abs/1404.1100

[38] W. Chen, M. J. Er, and S. Wu, "Pca and lda in dct domain," *Pattern Recogn. Lett.*, vol. 26, no. 15, p. 2474–2482, Nov. 2005. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.05.004

[39] G. T. Narayanan, "A study of probability distributions of DCT coefficients in JPEG compression," Master's thesis, New Jersey Institute of Technology, 2010.

[40] A. Pnevmatikakis and L. Polymenakos, "Comparison of eigenface-based feature vectors under different impairments," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, 2004, pp. 296–299 Vol.1.

[41] R. J. Yadav and M. S. Nagmode, "Compression of hyperspectral image using pca–dct technology," in *Innovations in Electronics and Communication Engineering*, H. S. Saini, R. K. Singh, and K. S. Reddy, Eds. Singapore: Springer Singapore, 2018, pp. 269–277.

[42] M. Sharkas, "Application of DCT Blocks with Principal Component Analysis for Face Recognition," in *Proceedings of the 5th WSEAS International Conference on Signal, Speech and Image Processing*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2005, p. 107–111.

[43] C. Lv and Q. Zhao, "A Universal PCA for Image Compression," in *Proceedings of the 2005 International Conference on Embedded and Ubiquitous Computing*. Springer-Verlag, 2005, p. 910–919.

[44] 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.

[45] M. A. Greg Roelofs, "zlib," https://github.com/madler/zlib, 2017.

[46] "Johns Hopkins Turbulence Database." [Online]. Available: http://turbulence.pha.jhu.edu

[47] S. Li, S. Di, K. Zhao, X. Liang, Z. Chen, and F. Cappello, "SDC Resilient Error-bounded Lossy Compressor," 2020.

[48] S. Habib, V. Morozov, N. Frontiere, H. Finkel, A. Pope, K. Heitmann *et al.*, "HACC: Extreme Scaling and Performance across Diverse Architectures," *Commun. ACM*, vol. 60, no. 1, p. 97–104, Dec. 2016. [Online]. Available: https://doi.org/10.1145/3015569

[49] K. Zhao, S. Di, X. Liang, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello, "SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors," 2021.