CliZ: Optimizing Lossy Compression for Climate Datasets with Adaptive Fine-tuned Data Prediction

Zizhe Jian*, Sheng Di†, Jinyang Liu*, Kai Zhao‡, Xin Liang§, Haiying Xu¶,
Robert Underwood†, Shixun Wu*, Jiajun Huang*, Zizhong Chen*, Franck Cappello†
*University of California, Riverside, CA, USA
†Argonne National Laboratory, Lemont, IL, USA
‡Florida State University, Tallahassee, FL, USA
§University of Kentucky, Lexington, KY, USA
¶National Center for Atmospheric Research, Boulder, CO, USA
zjian106@ucr.edu, sdi1@anl.gov, jliu447@ucr.edu, kai.zhao@fsu.edu, xliang@uky.edu, haiyingx@ucar.edu,
runderwood@anl.gov, swu264@ucr.edu, jhuan380@ucr.edu, chen@cs.ucr.edu, cappello@mcs.anl.gov

***Abstract*—Benefiting from the cutting-edge supercomputers that support extremely large-scale scientific simulations, climate research has advanced significantly over the past decades. However, new critical challenges have arisen regarding efficiently storing and transferring large-scale climate data among distributed repositories and databases for post hoc analysis. In this paper, we develop CliZ, an efficient online error-controlled lossy compression method with optimized data prediction and encoding methods for climate datasets across various climate models. On the one hand, we explored how to take advantage of particular properties of the climate datasets (such as mask-map information, dimension permutation/fusion, and data periodicity pattern) to improve the data prediction accuracy. On the other hand, CliZ features a novel multi-Huffman encoding method, which can significantly improve the encoding efficiency. Therefore significantly improving compression ratios. We evaluated CliZ versus many other state-of-the-art error-controlled lossy compressors (including SZ3, ZFP, SPERR, and QoZ) based on multiple real-world climate datasets with different models. Experiments show that CliZ outperforms the second-best compressor (SZ3, SPERR, or QoZ1.1) on climate datasets by 20%-200% in compression ratio. CliZ can significantly reduce the data transfer cost between the two remote Globus endpoints by 32%-38%.**

***Index Terms*—error-controlled lossy compression, climate datasets, distributed data repository/database**

I. Introduction

With the fast-increasing scale of climate simulations, exemplified by models like the Community Earth System Model (CESM) [1], [2], efficiently storing and transferring the vast amounts of climate data generated by these simulations has become a critical issue for scientists. For instance, recent literature [1] highlights that CESM can produce over 300 TB of data in the initial 30 ensemble simulations.

To resolve the big-data issues encountered by the large-scale parallel simulations, error-bounded lossy compressors [3]–[9] have been proposed and developed for years. While enabling users to control data distortion based on specified error bounds, the error-bounded lossy compressors can also significantly reduce the data volume.

Corresponding author: Sheng Di, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 Cass Avenue, Lemont, IL 60439, USA

Many existing studies demonstrated that a specific design with an optimized compression pipeline tailored for the specific datasets based on their data features can significantly improve the compression ratios as well as the execution performance compared with the generic-purpose lossy compressors such as SZ [3], [4], [10] and ZFP [5]. For instance, Pastri [11] is an error-bounded lossy compressor optimized for the Two-Electron Integrals dataset generated by quantum chemistry simulation GAMESS [12], leveraging the potential scaled-pattern feature observed in the dataset. MDZ [13] is a specific error-bounded lossy compressor developed for the Molecular Dynamics (MD) dataset, which exploits the spatial data pattern and high temporal smoothness observed in the MD datasets.

Regarding the aspect of climate research, many general-purpose lossy compression methods have been proposed and evaluated [14]–[17]. As demonstrated in recent studies, various error-bounded lossy compressors may exhibit largely different compression performances and qualities. For instance, Poppick et al. [16] evaluated the compression quality of two state-of-the-art lossy compressors – SZ and ZFP, and concluded that both of the two compressors exhibit high compression quality and fidelity, but they also suffer from prominent data distortion on some datasets with various error bounds. Robert et al. [17] performed a very comprehensive evaluation for climate datasets across different models such as atmosphere, ocean, and land, using many different state-of-the-art compressors including SZ3, ZFP, Zstd, MGARD, TTHRESH, BitGrooming, and Digit rounding. The evaluation shows that SZ3 exhibits the best compression quality in most test cases (such as d-SSIM [18], Pearson correlation, and Wasserstein distance).

In this paper, we aim to significantly optimize error-bounded lossy compression quality for climate datasets by taking advantage of the climate data features in particular, for which we need to address two grand challenges. On the one hand, climate simulations such as CESM [2] can involve different models such as the atmosphere model, land model, ocean model, and ice model, or a hybrid model involving multiple of the existing ones. Therefore, climate data may feature very diverse properties or patterns, which introduces a considerable

challenge for understanding/characterizing the data features. On the other hand, an error-bounded lossy compressor generally involves multiple stages, and we need to specifically revise each of them to optimize the compression of climate data. As such, developing an effective compressor by taking advantage of the specific climate data features/patterns is challenging.

In this paper, we develop *CliZ*, a novel error-bounded lossy compression algorithm tailored for climate simulation datasets. The key idea is to take full advantage of the potential properties and features of the climate data we explored to improve the data prediction accuracy, which can thus significantly improve the compression quality and ratio under the SZ3 framework. Our key contributions are summarized as follows:

- We carefully studied the properties and features of the climate datasets based on different simulation models and settings commonly used in the community, which is a fundamental work for our customized climate data compression design.
- We design a novel compression scheme by leveraging the climate data properties/features we explored, which involves two stages: offline training and online compression. The offline training stage aims to identify the property and feature, as well as selecting the best compression pipelines. The online compression stage will perform the compression operation on top of the input dataset based on the selected compression pipeline. The compression pipeline selected based on a few data fields or snapshots can be applied on all other datasets in the same climate model, because of fairly high consistency among the fields/snapshots in the same model.
- We perform comprehensive experiments based on different climate model datasets and compare CliZ with many other state-of-the-art related works, such as SZ3, QoZ, ZFP, and SPERR. Experiments show that CliZ can improve the compression ratios by up to 160%-870% over the best existing error-bounded lossy compressor (SZ3, SPERR, or QoZ) on climate datasets. Moreover, CliZ has comparable compression and decompression speeds based on our optimized implementation. Our experiments showed that using CliZ can significantly improve the Globus data transfer performance ($\sim$32%-38%) on WAN with the same reconstructed data quality than using other lossy compressors such as SZ3 and ZFP.

The remaining of the paper is organized as follows. In Section II, we discuss related work. In Section III, we formulate the research problem. In Section IV, we present the design overview. In Section V, we describe the key properties/features we explored in climate datasets. In Section VI, we detail our design and optimization strategies. In Section VII, we analyze the evaluation results. Finally, we provide a conclusion remark and discuss the future work in Section VIII.

II. Related Work

This section discusses the related works in two aspects: the error-bounded lossy compressors, and their usages in climate data research.

Error-bounded lossy compression is favored and tailored for various scientific data reduction use cases [19]. These compression tools fall into four primary categories: prediction-based, transform-based, dimension-reduction-based, and neural-network-based.

SZ2 [20], SZ3 [10], and QoZ [8] are notable examples of prediction-based compressors as they all employ data prediction strategies for compression. In contrast, transform-based compressors utilize data transformations to eliminate data correlations and then focus on compressing the resulting coefficients. ZFP [5] and SPERR [21] are typical instances implementing orthogonal discrete transform and wavelet transform, respectively. Dimension-reduction-focused compressors leverage techniques like (high-order) singular vector decomposition (SVD), where TTHRESH [22] is a prime example. There are also several compressors that use neural networks as their core design [23]–[26].

Those aforementioned lossy compression methods have been investigated and leveraged on the climate science data. Baker et al. [14] demonstrated that climate data exhibit diverse patterns and correlations, so a single compressor with one specific compression mode cannot be adequate on all variables. The research shows that to achieve a high compression ratio and acceptable reconstructions of data, multiple compression methods need to be dynamically chosen for various climate variables. Poppick et al. [16] studied the compression error artifacts on climate data under the two state-of-the-art compressors ZFP [5] and SZ [3], [4], [7], [10]. Their research on two climate variables validated that compression metrics should involve multiple spatial and temporal scales other than the point-wise comparison of original and compressed output. Robert et al. [17] performed a very comprehensive evaluation of climate datasets across different models with many different state-of-the-art compressors including SZ3 [10], [27], ZFP [5], Zstd [28], MGARD [29], TTHRESH [22], BitGrooming [30], and Digit rounding. The evaluation shows that SZ3 exhibits the best compression quality in most of the test cases (such as SSIM [31], Pearson correlation, and Wasserstein distance).

Compared with the above existing related works, we develop a novel error-bounded lossy compressor for climate datasets by taking full advantage of the unique properties and features in the climate data. This work features a co-design regarding in-depth characterization and understanding of the climate data and efficient/advanced lossy compression techniques. Comprehensive experiments with different climate models (to be shown later) demonstrate that the compression ratio can be significantly improved under our optimized compression method compared with the state-of-the-art – SZ3, which has been validated by [17] as the best existing error-bounded lossy compressors in the community.

III. Problem Formulation

In this section, we present the problem formulation of climate data compression. Specifically, we focus on the climate datasets, each describing a meteorological quantity of a specific part of the earth at some timestamps. As such, each

dataset is composed of one or more rectangular-shaped 2D images. The first dimension of a 3D dataset could represent multiple layers of data (such as different atmosphere layers) or different timestamps/simulation steps. Our compression method aims to significantly improve the overall compression quality and ratio by leveraging the critical properties we explored based on the climate datasets.

We formulate the problem mathematically as follows. Consider a multi-dimensional climate dataset with an original data size of S. After compressing it with a lossy compressor, the compressed data size is denoted as S' in bytes. The compression ratio can be written as $R = S/S'$. Our research aims to develop a novel error-bounded lossy compressor specialized for climate simulation datasets, which can significantly improve the compression ratio and quality compared to the existing state-of-the-art general-purpose error-bounded compressors with comparable execution performance. Specifically, compared with other compressors, our compressor would either improve the compression ratios with the same level of data quality assessment or obtain much higher data quality with the same compression ratios. Multiple data assessment methods, including peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and visualization, are widely used to evaluate the quality of reconstructed data commonly in the climate community.

IV. Design Overview

We present the design overview in Fig. 1. CliZ is built upon the SZ3 [27] error-bounded lossy compression framework, so its compression stage comprises three essential steps: data prediction, linear-scale quantization, and Huffman+Zstd lossless compression.

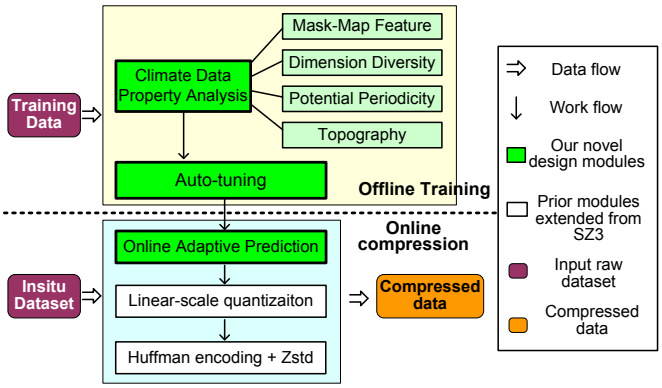


Fig. 1. Design Overview of CliZ

Before the online compression stage, we design an offline training stage, as shown in Fig. 1. This is because we observe that various climate datasets exhibit specific data properties/features, based on which we can substantially improve the data prediction accuracy, thus enhancing the compression ratio significantly. Based on our close observation, we note that each climate simulation dataset generally involves many data fields, such as potential temperature (TEMP), Salinity (SALT), Insitu Density (RHO), Sea Surface Height (SSH), and Solar Short-Wave Heat Flux (SHF_QSW). These fields often exhibit the same properties (such as periodicity and mask-map feature) within the same simulation model/setting. This motivates us to design such an offline training stage for each climate model or setting, and the users will perform the online compression based on the offline auto-tuned/optimized configuration settings.

We extracted four data properties/features, including the mask-map feature, dimension diversity, potential periodicity, and topography. The four critical climate data properties/features are used to improve the prediction accuracy and quantization efficiency during auto-tuning of the best compression strategy. In Fig. 2, we present how the four data properties/features are performed and correlated in the auto-tuning stage. As shown in the figure, given a training dataset, CliZ first executes a sampling method to significantly reduce the size of the training data to reduce the training cost with well-preserved accuracy (to be discussed later in detail). Then, CliZ checks various strategies with multiple combinations of potential data features, such as periodicity and diverse dimensions, to explore the best strategy for the dataset.

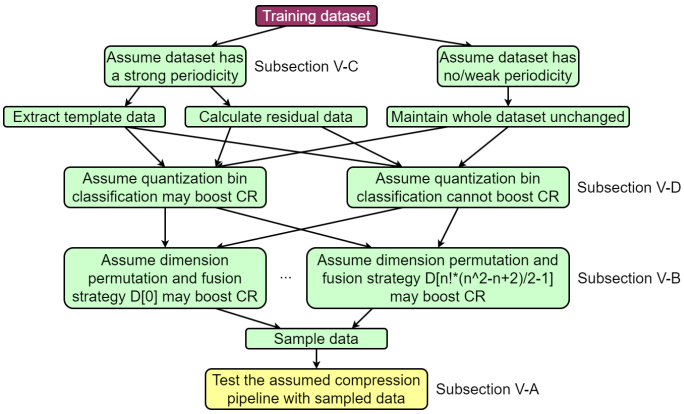


Fig. 2. Workflow of Offline Data Feature Analysis and Auto-tuning

In the online compression stage, CliZ performs an adaptive prediction method combining multiple prediction strategies, which can adapt to diverse data features in climate datasets, as shown in Fig. 1. Specifically, given testing data at runtime, CliZ first checks whether it should leverage the periodicity feature. Then, it performs data prediction, quantization, and encoding based on the optimized strategy (including the best dimension combination, denoted $D[i]$, mask-map-based prediction, and quantization classification). For example, data points near the edge of the missing areas would be predicted by available adjacent data points marked by the Mask map. This contrasts with the traditional dynamic spline interpolation used in SZ3, which performs prediction by following a strict neighborhood topology.

V. Exploring Key Properties in Climate Datasets

To fine-tune the CliZ design to optimize its compression for climate data, this section explores and analyzes four critical properties of climate datasets, including mask-map, diverse

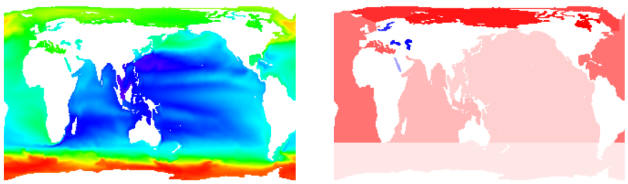
dimensions, periodicity, and topography. To the best of our knowledge, these analyses have a unique insight into the context of compression, especially compared to existing state-of-the-art general-purpose compression methods.

In the following text, we describe the fundamental concept and idea of the four features and provide detailed design and optimization strategies in the next section.

A. Mask-Map

The Mask-Map is the first important feature/property we explored for climate data prediction in CliZ.

In a climate dataset from the climate package (CESM), the data values of certain grid points may be missing or invalid, and these data values appear as tremendous data values (e.g., 2^{122}) in the dataset. These missing data points are used to mark uninteresting regions. For example, the ocean area may not be needed for the land model analysis. To address this, the dataset involves a mask map for indicating the valid values at the corresponding coordinates in various datasets. It may also provide information about the partition of valid data.



(a) SSH (b) Mask

Fig. 3. Sea Surface Height (SSH) at a Certain Time and its Mask Dataset

As illustrated in Fig. 3, the sea surface height (SSH) dataset focuses on only the ocean region. Hence, the data values in the land area are all invalid/missing (represented as white in the figure). As such, the mask map (as shown in Fig. 3(b)) shows three categories of the data values: 0 (i.e., invalid/missing values indicating non-water regions), positive integer values (shown as red, indicating different parts of the ocean), and negative integer values (shown as blue, indicating inland water bodies). Note that in the climate dataset, these invalid data are always represented as various huge numbers, which would significantly harm the lossy compression ratios because the invalid regions would substantially affect the accuracy of data prediction in prediction-based compressors such as SZ or the effectiveness of the near-orthogonal data transforms used in transform-based compressors such as ZFP.

B. Diverse Dimensions with Various Smoothness

In addition to the mask-map property, we explore the diverse smoothness features across different dimensions to improve the data prediction accuracy further.

The smoothness of datasets is one of the most critical factors for getting high compression ratios. The data values of climate datasets lie in structured grids representing geographical coordinates. It is worth noting that the physical distance between two adjacent data points (the distance unit) can correspond to different lengths. For example, in a dataset (temperature) about global atmosphere temperature with size 26(height)×1800(latitude)×3600(longitude), the data variation along the "height" dimension is 4.425 on average, while on the other two dimensions, the variations are 0.053 and 0.017, respectively. Obviously, the dimension with a smaller distance unit can present better smoothness and vice versa. Figure 4 demonstrates three slices along different dimensions at the center of the temperature dataset mentioned before. The data values rarely change along latitude and longitude but significantly along height. Our compression method features a dynamic auto-tuning strategy to select the best number of dimensions and orders according to the smoothness in different directions, which will be detailed later.

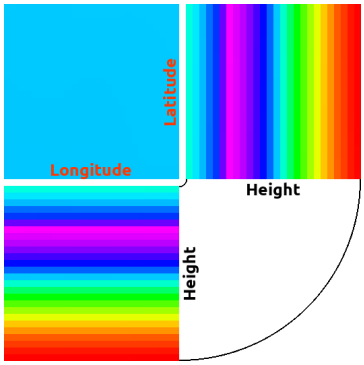


Fig. 4. Three slice (26 × 26) at the center of temperature dataset

C. Periodicity

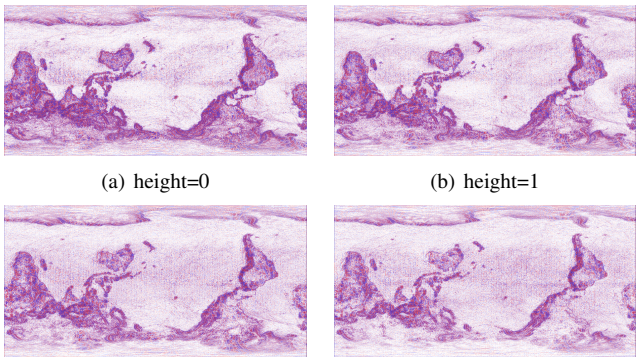
Periodicity is the third critical feature we explored in our optimized data prediction strategy. By close observation of many climate datasets/fields, we note that some climate datasets exhibit a strong periodicity across multiple snapshots/timesteps over time because meteorological phenomena typically follow an annual cycle. That is, the data snapshots at the same location in different periods (along time dimension) may have a higher similarity than their spatial neighbors. As such, our compression method CliZ also leverages such a periodicity feature to maximize the prediction accuracy, which will be detailed later.

D. Topography

In addition to the three essential climate data features mentioned above, we further explore the topograph feature to improve the quantization efficiency in CliZ. Topography is a crucial factor affecting climate meteorological phenomena. The atmosphere, for example, is influenced by the terrain of the continents, while the oceans are affected by the seabed topography. Such influence should be affected by latitude and longitude but not time or height. As such, there can be some hidden data characteristics related to topography, which can be helpful during compression. For example, Fig. 5 shows several horizontal slices of a log-scaled intermediate quantity "quantization bin"[1] at different heights in a global atmosphere temperature dataset. We can clearly observe that

[1]quantization bin here means the discretized values by a fixed-bin-size linear-scale quantization based on the original data.

different locations often exhibit largely different data values or variations, while the same locations or the areas with similar topography exhibit similar values even at different height slices. This motivates us to group the data values in different regions in our encoding stage, which can significantly improve the compression ratio, which will be shown later in detail.



(a) height=0 (b) height=1

(c) height=2 (d) height=3

Fig. 5. Quantization Bins at Different Heights

VI. Compression Design and Optimization

This section describes the offline auto-tuning strategy and the detailed designs of four critical feature explorations.

A. CliZ Auto-tuning

As mentioned in Section IV, CliZ comprises multiple compression pipelines to handle different datasets, so it needs to determine which one works best in every case. To this end, we adopt an efficient sampling and testing method for auto-tuning. The auto-tuning procedure is given in Fig. 2. To put it simply, auto-tuning will decide a compression pipeline that includes:

1) Sequence of dimensions and dimension fusion in subsection VI-C.
2) Whether the data is periodic and its periodicity in subsection VI-D.
3) Whether to use quantization bin classification in subsection VI-E.
4) Which fitting function to use. Fitting functions can be linear or cubic, which is introduced in SZ3 [10].

The generated compression pipeline will not include:

1) Whether to use the mask dataset in subsection VI-B. This is decided by the user.
2) The extracted periodic component if the dataset has periodicity in subsection VI-D. This is done during actual compression.
3) The classification of each horizontal position in subsection VI-E. This is done during actual compression.

The testing data used during auto-tuning is sampled by selecting some blocks of data at $\frac{1}{3}$ and $\frac{2}{3}$ along each dimension and connecting them together. For a dataset with n dimensions, 2^n blocks will be selected, and the approximate length of each side of every block is about $\frac{1}{2}sampling\_rate^{\frac{1}{n}}$ of the corresponding side length of the entire dataset, where sampling rate means the expected ratio between the volume of the test data and the whole dataset.

When decreasing the sampling rate, we can expect the time spent on auto-tuning to decrease. On the other hand, the difference between different compression pipelines can be less significant and even misleading, mainly due to the decrease in the selected block size. Smaller block size, for example, decreases the smoothness in all dimensions. So, the dimension permutation and fusion in subsection VI-C, which is highly related to the difference in smoothness between dimensions, may provide the second-best choices. Besides, petite block sizes can be a disadvantage for the cubic fitting predictor, as it refers from a broader range of data and is more likely to predict a data point while referring from other data points in different blocks. We provide a detailed accuracy and running speed analysis based on various sampling rates in section VII.

B. Mask-map based Data Prediction

The first optimization strategy is leveraging the mask-map, which can be found in each HDF5-format climate data file if the corresponding dataset includes the missing/invalid data points. The basic idea is predicting each data point based on the dynamic fitting method [10] yet with valid neighbors conveyed by the mask-map. This strategy increases the compression ratio by:

1) Reducing the amount of data to compress by not compressing the masked positions.
2) Improving the predicting accuracy of predictors by not inferring from masked values.

In the following, we first describe how the classic dynamic linear/cubic fitting is executed in the prediction, and then present our mask-map-based prediction method.

Fig. 6 illustrates the dynamic cubic fitting based prediction method used in SZ3 [10]. As shown in the figure, each prediction is conducted with four referenced data points at different certain strides with the target prediction data point. The strides (denoted by s in the following text) could be 1, 2, 4, 8, $\cdots$ in practice. The predicted value (denoted p) based on a cubic fitting can be written as the Formula (1):

$$p = -\frac{d_0}{16} + \frac{9d_1}{16} + \frac{9d_2}{16} - \frac{d_3}{16} \quad (1)$$

where d_0, d_1, d_2, and d_3 are the referenced data values at different strides (as shown in Fig. 6). For boundary cases, the cubic fitting predictor degrades to lower-degree fitting predictors through hard-coded adjustments. For linear fitting based prediction, the predictor refers from d_1 and d_2 in Fig. 6 and $p = \frac{1}{2}d_1 + \frac{1}{2}d_2$. For more details about the classic fitting method, we refer the readers to read [10].

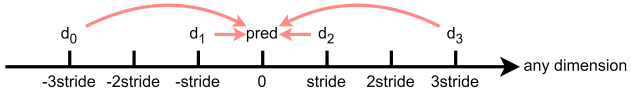


Fig. 6. Cubic Fitting Prediction

Now, we describe our mask-map-compatible fitting particularly designed for climate datasets with missing data values. As an example, we focus on cubic fitting predictors with no

boundary cases. Specifically, if some of the four referenced data points are invalid (masked), CliZ will alter the coefficients of the valid referenced data so that the prediction method is still an effective polynomial function fitting. For simplicity of description, based on the four *referenced data points* d_i (where i=0,1,2,3), we denote *validity of each referenced data point* (affected by the mask-map) as $v_i \in \{0, 1\}$ and *coefficient of each referenced data point* as p_i, respectively. That is, the prediction value can be generalized as $p = \sum_{i=0}^{3} p_i d_i$. Table I shows the specific validity values and coefficients of the classic fitting prediction method where all data points are valid (i.e., Formula (1)).

TABLE I
VALIDITY AND COEFFICIENTS WHEN ALL DATA POINTS ARE VALID

v_0	v_1	v_2	v_3	p_0	p_1	p_2	p_3
1	1	1	1	$-\frac{1}{16}$	$\frac{9}{16}$	$\frac{9}{16}$	$-\frac{1}{16}$

Theorem 1. *The mask-map-based dynamic fitting predictor with optimal coefficients is written as Formula (2).*

$$p_i = \prod_{j=1}^{4} (v_j M_{i,j} + (1 - v_j) B_{i,j})$$
$$\textit{where}\ \ M = \begin{pmatrix} 1 & -0.5 & 0.25 & 0.5 \\ 1.5 & 1 & 0.5 & 0.75 \\ 0.75 & 0.5 & 1 & 1.5 \\ 0.5 & 0.25 & -0.5 & 1 \end{pmatrix} \qquad (2)$$
$$B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Proof. Similar to the derivation of the classic cubic fitting (Formula (1)), we can derive the optimal coefficients for the situation with one invalid/missing referenced data point as Table II. In fact, if one of the four data points is masked, the cubic fitting prediction method degrades into a quadratic fitting. The detailed derivation is omitted here because of space limitations.

TABLE II
VALIDITY AND PARAMETER WHEN THREE DATA POINTS ARE VALID

v_0	v_1	v_2	v_3	p_0	p_1	p_2	p_3
0	1	1	1	0	$\frac{3}{8}$	$\frac{3}{4}$	$-\frac{1}{8}$
1	0	1	1	$\frac{1}{8}$	0	$\frac{9}{8}$	$-\frac{1}{4}$
1	1	0	1	$-\frac{1}{4}$	$\frac{9}{8}$	0	$\frac{1}{8}$
1	1	1	0	$-\frac{1}{8}$	$\frac{3}{4}$	$\frac{3}{8}$	0

As for the situation with 0∼2 valid referenced data points (according to the mask-map), we set the prediction value as zero, apply constant fitting, and apply linear fitting, respectively.

Combining all the five situations with 0-4 valid reference data points (such as Formula (1) and Table II), we can get the general Formula (2). □

Similar to SZ3 [10], the use of lower-degree predictors for boundary cases is hard-coded, but at the same time, the parameters of these predictors dynamically change based on whether the data points are masked.

Last but not least, we discard quantization bins at the masked positions when generating the sequence containing quantization bins in CliZ, so the masked data would not affect Huffman encoding and lossless post-processing.

C. Dimension Permutation and Fusion

In this subsection, we discuss how to improve the prediction accuracy by dimension permutation and fusion strategy. This strategy increases the compression ratio by changing the referenced data of some predictions to increase the prediction accuracy.

Note that the traditional design of dynamic fitting predictor in SZ3 (i.e., Formula (1)) is executed along all different dimensions iteratively in the divide-conquer sequence, which means in each round all predictions are along the same dimension, and every predicted data point will be used to predict one new data point on average. So, predictions made alone in the next dimension will be roughly twice those made along the current dimension. In all, $\frac{2^{i-1}}{2^n-1}$ of the predictions occur along the $i \in 1, 2, ..., n$th dimension given n dimensions in all, which is not the best choice for climate dataset in most cases. In fact, the accuracy of predictions can be enhanced when made along dimensions that exhibit high smoothness. To achieve a more accurate prediction, we should make predictions on the smoother dimensions more frequently. Our solution would permute the order of dimensions for prediction and fuse some of them to find the best case with the highest prediction accuracy. Permutation can increase the number of predictions on smooth dimensions. Fusion, which means considering several adjacent dimensions as one dimension without affecting the data storage sequence, can significantly increase the distance between adjacent points in all fused dimensions except the last one, and there is no chance that short-distance predictions will happen along these dimensions.

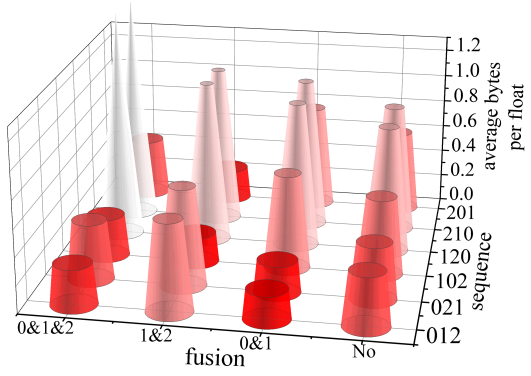


Fig. 7. Diverse Bit-rates of Different Dimension Permutation/Future Cases, based on Global Atmosphere Temperature Dataset)

In Fig. 7, we demonstrate the diversity of compression ratios among multiple cases with various dimension permutations and fusions when compressing the global atmosphere temperature dataset by CliZ. The Z-axis refers to the bit rate (i.e., the average required number of bits per data point after the

compression): the lower the bit rate, the higher the compression ratio. Red conical frustums correspond to the compression methods with low bit rates (i.e., high compression ratios). Note that the fusion is conveyed after order swapping. For example, sequence "021" and fusion "0&1" means exchanging the second and third dimensions and then fusing the original first and third dimensions. We can find multiple red frustums as short as each other. The shortest guy uses sequence "102" and fusion "1&2", and the approximate result is sequence "012" and fusion "0&1", projecting only 0.065% difference in-between.

D. Periodic Component Extraction

As mentioned previously, climate datasets with dimensions representing time information may exhibit a strong periodicity since meteorological phenomena typically follow an annual cycle, based on which we can further improve the prediction accuracy for some climate datasets. Specifically, according to the HDF5 metadata or dataset name, if the physical meaning of a dimension is identified as "time with periodicity", CliZ will separate the data into "template" and "residual". Template data is the average data among all periods, and its length along the time dimension shrinks to the length of a period. Residual data is the difference between the template and source data, and its size remains the same as the input. CliZ will then test other optimization strategies separately on the template and residual data. This strategy increases the compression ratio because of significantly improved smoothness in all directions.

We use FFTW [32] to estimate the shortest period based on randomly sampled data rows along the time dimension. We illustrate this idea using the sea surface height (SSH) dataset as an example. Its dimensions are 320(longitude)×384(latitude)×1032(time). Without loss of generality, we suppose ten rows of data are used, and FFTW would generate ten groups of data indicating the amplitude of basis functions with frequency 0∼516, as shown in Fig. 8. B∼K means ten rows of data. It is clearly observed that all rows of data reach a high peak when frequency equals 86. There are also other high peaks at multiples of 86, but we only adopt the peak with the smallest frequency, which means the largest period. So, the length of a period in our design is set to $1032/86 = 12$.

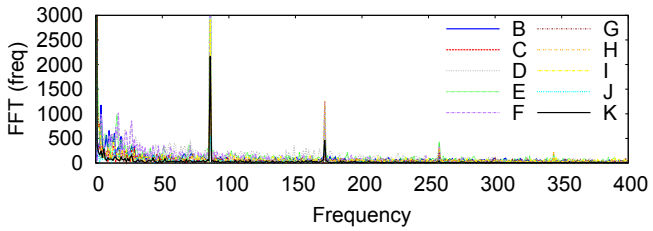
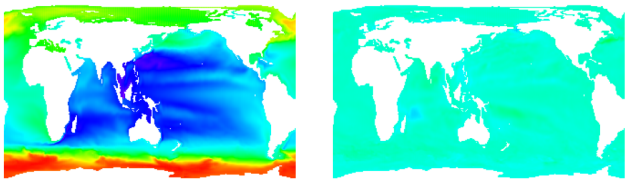


Fig. 8. Result of FFTW on 10 Rows of Data (space too much)

We can also compare the original horizontal slice and the slice in residual data. The original version is shown in Fig. 3(a). The version in the residual data is shown in Fig. 9. We can find that the data achieves a higher continuity along two spatial dimensions after processing. This is because when the most significant periodic component (and the constant component) is removed, the periodicity in the difference along spatial dimensions is also removed such that the residual parts are pretty close to zero for most of the data points.



(a) Original (b) Residual

Fig. 9. A Slice of Original and Residual Data

E. Quantization Bin Classification

As mentioned in Section V-D, some topography patterns may appear on the quantization bins, other than the above-mentioned temporal periodicity feature. We note that these patterns can be classified as shifting and dispersion, and mixing the two patterns may significantly affect the effectiveness of Huffman encoding. As such, we develop a quantization bin classification method to mitigate the negative impact. Briefly speaking, this strategy may generate multiple Huffman trees as an encoding procedure. Compared to SZ3, which uses classical Huffman encoding and generates one Huffman tree, this strategy leverages the patterns of quantization bins to enhance encoding efficiency.

The fundamental concept involves adjusting the quantization bins through a method of bin shifting and subsequently classifying all quantization bins into multiple groups, each of which will be encoded by different Huffman trees. In general, the frequency curve of quantization bins features a relatively sharp symmetric distribution with quantization bin zero being the unique peak. However, we observe two major cases against the sharp symmetric distribution situation: quantization bin shifting and quantization bin dispersion.

- The quantization bin shifting means that most of the quantization bins at a specific coordinate location across all different snapshots or heights have either negative or positive values. For this case, we mark the corresponding coordinate location and shift the quantization bin values towards 0, such that the highest-frequency quantization bin would turn 0.
- The quantization bin dispersion refers to the absence of any quantization bins projecting a very high frequency based on a threshold λ (the optimal setting is 0.4, as demonstrated in Theorem 2). To cluster as many high-frequency quantization bins as possible, we build one Huffman tree for the quantization bins at the coordinate locations with the highest frequency greater than λ. The quantization bins at other coordinates would be encoded separately by a different Huffman tree.

Combining the above two steps can significantly improve Huffman encoding efficiency (thus achieving a higher compression ratio). Quantization bin shifting merges the most

frequent quantization bins into a common new quantization bin 0, and quantization bin classification groups all quantization bins whose values are close to each other, which will be encoded by the same Huffman tree.

In practice, we classify the quantization bins into two groups for two reasons. On the one hand, it is worth noting that marking coordinate locations with shifting and dispersion requires additional space. Suppose there are $2j+1$ types of shifting (peaking at 0, ±1, ..., $\pm j$) and $k+1$ types of dispersion, each coordinate location will occupy about $log_2((2j+1)(k+1))$ bits. Based on our careful experiments, we observe that the compression ratio cannot be significantly increased when j or k is greater than 1. Therefore, we set $j=k=1$ in our implementation.

We prove the optimal threshold setting is 0.4 as follows.

Theorem 2. *The optimal threshold setting (denoted by λ) for the above-mentioned quantization bin classification is 0.4.*

Proof. According to the definition of the quantization bin classification threshold, when the frequency of the local peak $f_0 > \lambda$ and when only three leaf nodes are remaining in the Huffman tree to merge, the other two leaf nodes (supposing their frequencies are f_1 and f_2) will be merged.

There are two common situations to discuss:

1) *Two other leaf nodes are generated by merging.*
   We have some constraints according to the situation.
   $$f_1 < 2f_2$$
   $$f_2 < 2f_1$$
   We also have some constant constraints.
   $$0 < f_0 < 1$$
   $$0 < f_1 < 1$$
   $$0 < f_2 < 1$$
   $$f0 + f1 + f2 = 1.$$
   We want to find λ which meets the requirement that for any $f_0 > \lambda$ and any $f1$ and $f2$ satisfying the above-mentioned constraints. As such, the following two inequalities hold:
   $$f_0 > f_1$$
   $$f_0 > f_2$$
   Hence, we get $\lambda > 0.4$.
2) *One of the leaf nodes (such as $f1$) has never been merged.*
   We have a constraint according to this situation.
   $$f_1 > \lambda(f1 + f2)$$
   $$f0 > f1$$
   Other constraints and the goal are the same as in the first situation. Then, we have $\lambda > \frac{3-\sqrt{5}}{2}$.

To guarantee that λ works fine on both common conditions, we need to choose $\lambda = 0.4$. □

VII. Performance Evaluation

This section presents systematic performance evaluations and analyses for CliZ, comparing it to several other state-of-the-art error-bounded lossy compressors.

A. Experimental Setup

In the following, we describe the platform and the datasets used in our experiments.

Platform: The performance evaluation is conducted on a medium-scale cluster – Argonne Bebop [33]. Each computing node in the cluster has two Intel(R) Xeon(R) E5-2695 v4 CPUs and 128GB DRAM.

Datasets: We mainly focus on global climate datasets related to the ocean, atmosphere, land, and ice. All the data are stored in single-precision data format, and detailed information, including dimensions and the existence of corresponding mask datasets and periodicity, is listed in Table III. The description of the benchmarked datasets is as follows:

1) SSH: Sea surface height collected once a month.
2) CESM-T: Atmosphere temperature at a certain time.
3) RELHUM: Atmosphere relative humidity at a certain time.
4) SOILLIQ: Liquid water content in the soil collected once a month.
5) Tsfc: Surface temperature of snow or ice collected once a month.
6) Hurricane-T: Atmosphere temperature around Hurricane Isabel at a certain time.

TABLE III
Information about Tested Datasets

Name	Lat	Long	Time	Height	Mask	Period
SSH	384	320	1032	No	Yes	Yes
CESM-T	1800	3600	No	26	No	No
RELHUM	1800	3600	No	26	No	No
SOILLIQ	96	144	360	15	Yes	Yes
Tsfc	384	320	360	No	Yes	Yes
Hurricane-T	500	500	No	100	No	No

B. Metrics of Evaluating Data Distortion

For the CliZ compressor, we use the widely used rate-distortion graph to assess the compression quality, consistent with prior works. The x-axis in the graph is bit-rate, defined as the average number of bits per data in the compressed format and can be computed by 32 over compression ratios for single-precision floating-point data. The y-axis is the distortion metric, where we adopt the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) as the distortion metrics. PSNR is a well-recognized measurement to quantify the quality of lossy compression, which is defined as:

$$PSNR = 20log_{10}\frac{d_{max} - d_{min}}{RMSE}, \tag{3}$$

where d_{max} and d_{min} denote the maximal and minimal values in the original data, and $RMSE$ represents the root of the mean squared error between the original and reconstructed data.

SSIM is a popular perceptual metric broadly used in the image-processing community. The overall SSIM for input data x and reconstructed data y can be computed by:

$$SSIM = \frac{1}{N}\sum_{i=0}^{N} SSIM_i(x, y), \tag{4}$$

, where $SSIM_i(x, y)$ is the SSIM value for the i-th sliding window defined as follows:

$$SSIM_i(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (5)$$

where μ_x and μ_y are means of x and y, respectively, σ_x and σ_y are their respective variances, σ_{xy} is the covariance of x and y, and c_1 and c_2 are two small constants used to stabilize the division. Generally speaking, higher PSNR and SSIM indicate less distortion in the reconstructed data, thus better compression quality.

C. Evaluation Results and Analysis

1) Rate-distortion:

Fig. 10 shows the rate-distortion of five climate datasets compressed by five lossy compressors. We can find that in most cases, CliZ reaches a much better PSNR and SSIM with a fixed bit rate, or it obtains a much higher compression ratio with the same data distortion level. With the same reconstructed data quality measured by PSNR, CliZ can significantly improve the compression ratio by 2-9× over the state-of-the-art compressor QoZ1.1 in most cases. It also outperforms the second-best compressor (SZ3, SPERR, or QoZ1.1) by $160\% \sim 870\%$ in compression ratio in general. With the same reconstructed data quality measured by SSIM, CliZ can get 7× compression ratios in general compared with the second-best choice (SZ3, QoZ1.1, or SPERR). In particular, when compressing the SOILLIQ dataset with a relative error boundary of 10%, the compression ratio of CliZ is about 1955× as high as the compression ratio of QoZ1.1, which is the second best.

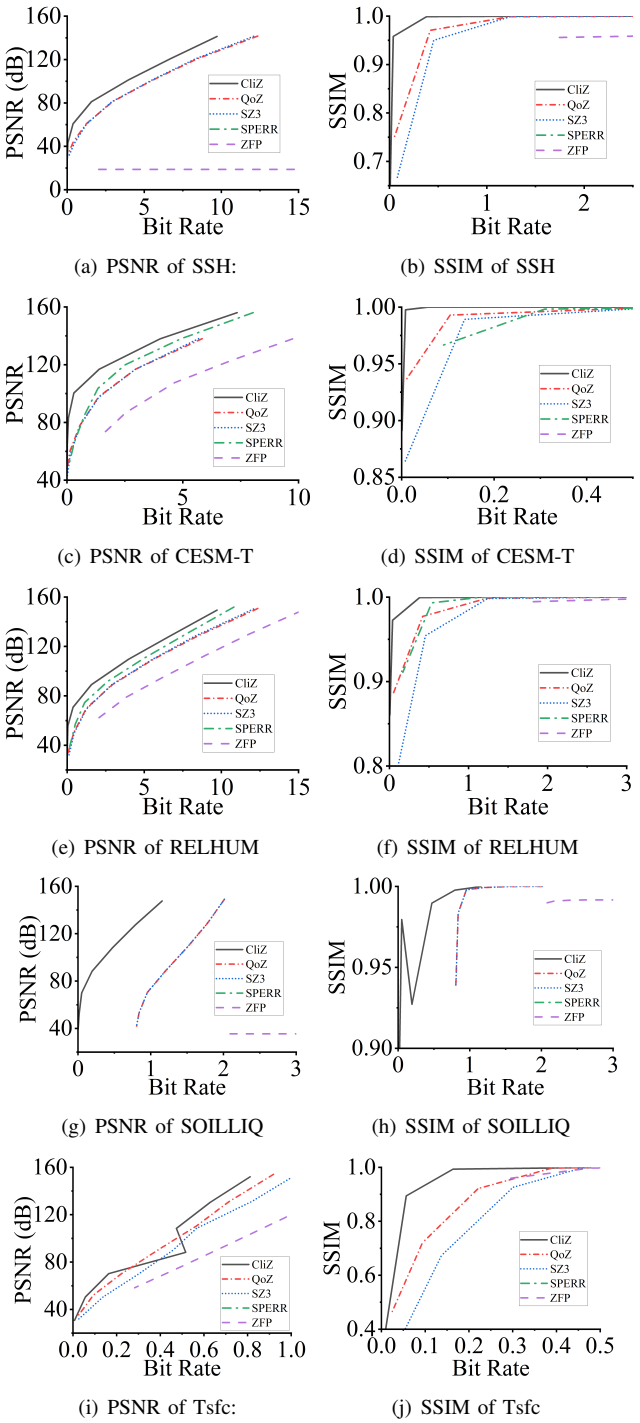
The key reason CliZ can get such a significant improvement in compression ratio is four-fold:

1) CliZ does not compress invalid data or let them affect the prediction.
2) CliZ makes better predictions with dimension permutation and fusion.
3) CliZ extracts periodic components so that the remaining data is much smoother than the raw data.
4) CliZ features a more effective encoding with quantization bin classification.

2) Sampling Cost, Running Time and Compression Ratio:

For CliZ auto-tuning, we mainly focus on (1) the time cost of compressing sampled datasets compared with the time cost of the full data compression and (2) the compression ratio of the full datasets based on the configuration that is optimized in terms of the sampled datasets. According to the sampling strategy in section VI-A, sampling rate = 1 means that all possible compression pipelines are tested on the whole dataset. In this situation, the testing result is 100% precise and can lead to the highest compression ratio.

In principle, the time cost of sampled-data compression increases linearly with the number of possible compression pipelines (except the pipelines for extracted periodic components) and sampling rate. Fig. 11 demonstrates this using two example datasets – SSH and CESM-T, both of which



(a) PSNR of SSH: (b) SSIM of SSH

(c) PSNR of CESM-T (d) SSIM of CESM-T

(e) PSNR of RELHUM (f) SSIM of RELHUM

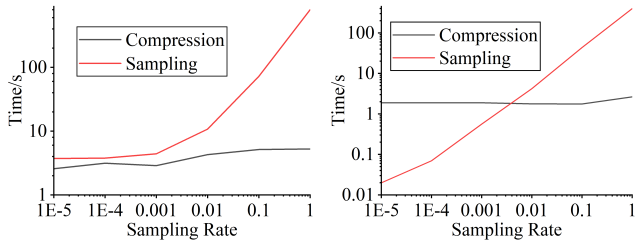
(g) PSNR of SOILLIQ (h) SSIM of SOILLIQ

(i) PSNR of Tsfc: (j) SSIM of Tsfc

Fig. 10. Rate-distortion graphs

have three dimensions. The difference is that the SSH dataset contains some periodicity while the CESM-T dataset does not. For the SSH dataset, all possible compression pipelines include two possible conditions of periodic component extraction (periodic or not periodic), two possible conditions of quantization bin classification (can or cannot improve compression ratio), six possible sequences of dimensions, four possible dimension fusion (no fusion, 0&1, 1&2, 0&1&2) and two possible fitting based predictors, which means 192 compression pipelines in

all. The CESM-T dataset has no periodicity and only has 96 pipelines.



(a) Running Time on SSH Dataset (b) Running Time on CESM-T Dataset

Fig. 11. Sampling and Compression Time Given Different Sampling Rate

From Fig. 11 we can find that introducing periodic components causes a constant increase in sampling time. Otherwise, the relationship between sampling time and sampling rate is almost linear. Sampling time is acceptable when the sampling rate is lower than 1%, which is also the sampling rate we use in subsection VII-C1.

In case there is a strict requirement on the total running time, the sampling rate of 0.1% can be used without too much loss in the compression ratio. Here, we still use the SSH dataset as an example. Fig. 12 shows the relationship between sampling rate and estimated compression ratio with different compression pipelines.

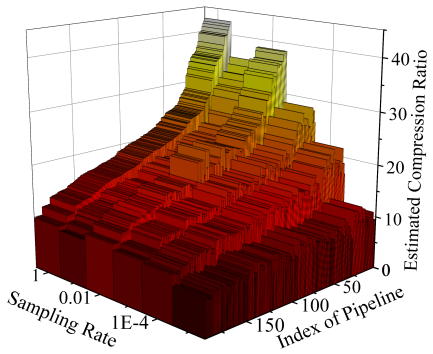


Fig. 12. Estimated Compression Ratios with Different Sampling Rates

As we look for the best compression pipeline, an ideal auto-tuning strategy should be able to provide the same order of estimated compression ratio for different pipelines given any sampling rate. This is almost achieved as shown in Fig. 12, where pipelines are sorted based on the estimated (but precise) compression ratio when tested on the whole dataset. With a sampling rate greater or equal to 0.1%, the estimated optimal pipeline reaches a similar compression ratio compared with the precise optimal pipeline. Numerical results about how much compression ratio is lost with a low sampling rate are given in Table IV. In this table, "Compression Ratio" means the actual compression ratio when using the estimated optimal compression pipeline, but not the estimated compression ratio. Notice that when the sampling rate is 100% or 10%, the compression pipelines are the same but yield different compression ratios. This is because the compression pipelines for template data are different, which is not shown in the table.

TABLE IV
ESTIMATED OPTIMAL PIPELINE AND LOSS IN COMPRESSION RATIO

Sampling Rate	100%	10%	1%	0.1%	0.01%	0.001%
Periodicity	12	12	12	12	12	12
Classification	Yes	Yes	Yes	Yes	Yes	No
Permutation	201	201	201	021	021	021
Fusion	1&2	1&2	1&2	0&1	No	No
Fitting	Cubic	Cubic	Linear	Linear	Linear	Linear
Compression Ratio	28.394	28.336	28.196	27.458	24.918	23.414
Loss	0%	0.20%	0.70%	3.30%	15.23%	17.54%

3) Performance Breakdown:

Auto-tuning is critical because the four optimization strategies may provide uncertain improvement in compression ratio. Besides, the extra time cost may also vary with datasets. Here, we use the SSH dataset and another dataset called "Hurricane-T" as two examples to illustrate this situation.

In Table V and VI, we compare the time and compression ratio of the estimated optimal pipeline when sampling rate = 1% with the condition when the status of each optimization strategy is toggled. "CR Improvement" means the percentage increase in the compression ratio of the estimated optimal pipeline compared with the other conditions. Ideally, "CR Improvement" should be positive since optimization strategies are expected to have a positive effect. "Time Increment" means the percentage time increment of the estimated optimal pipeline compared with the other conditions. It is usually positive since more strategies mean higher computation cost, as demonstrated in table V.

TABLE V
TIME AND COMPRESSION RATIO OF THE OPTIMAL PIPELINE AND WHEN EACH STRATEGY IS CANCELED (SSH)

Periodicity	12	12	12	12	No
Mask	Yes	No	Yes	Yes	Yes
Classification	Yes	Yes	Yes	No	Yes
Permutation	201	201	012	201	201
Fusion	1&2	1&2	No	1&2	1&2
Fitting	Linear	Linear	Linear	Linear	Linear
Compression Ratio	28.196	12.117	24.012	27.011	20.988
CR Improvement	0%	132.70%	17.42%	4.39%	34.34%
Time/s	4.734	4.730	2.572	4.007	4.575
Time Increment	0%	0.08%	84.06%	18.14%	3.48%

TABLE VI
TIME AND COMPRESSION RATIO OF THE OPTIMAL PIPELINE AND WHEN EACH STRATEGY IS CANCELED (HURRICANE-T)

Classification	Yes	No	Yes
Permutation	012	012	021
Fusion	No	No	0&1
Fitting	Linear	Linear	Linear
Compression Ratio	55.401	55.592	54.058
CR Improvement	0%	−0.34%	2.48%
Time/s	0.459	0.325	0.606
Time Increment	0%	41.23%	−24.26%

For the Hurricane-T dataset (as shown in Table VI, we note that the estimated optimal pipeline may not be the best among all the three conditions. Turning off quantization bin classification increases compression ratios and decreases the time cost. We choose a random configuration about dimension permutation and fusion in Column 3 for a comparison,

which shows the random setting may cause a much lower performance. Besides, this dataset has no mask information or periodicity, so only four features (Classification, Permutation, Fusion, and Fitting) are involved in each configuration, as shown in Table VI.

Mask-map-based data prediction plays a vital role in the compression of the SSH dataset, and it is easy to understand that the effect of mask-map-based data prediction is highly related to how much invalid data exists. The hurricane-T dataset has no mask data, so this strategy cannot work. On the contrary, SOILLIQ shows an extremely high improvement in compression ratio when compressed by CliZ, because about 70% of the earth's surface is covered with water and is regarded as invalid.

Quantization bin classification can slightly improve compression ratio in many cases unless the relative error boundary is lower than 10^{-3}. It is suitable for conditions when around half of the positions are classified to be encoded by each Huffman tree. Besides, the dimensions other than latitude and longitude are expected to be the time since the topological pattern may not be evident at high altitudes. Severe convective weather, such as a hurricane, will further decrease the height topological patterns can still exist. On the other hand, since it only relates to the terrain and physical meaning of the data, sliced datasets along time or height dimensions can usually share the same classification map file.

4) Scaled Performance:

Finally, we evaluate the compression-enabled climate data transfer performance across two remote sites on Globus over WAN to address the emerging need for lossy compression to accelerate data sharing performance geographically. We run the tests with 256∼1024 processes/cores: each core handles an individual data compression and transfers the compressed file through Globus. We compare the performance of three lossy compressors, CliZ, SZ3, and ZFP. For CliZ, we evaluate it based on the datasets with shared configuration files and marking-bit-array of quantization bin classification.

In the aspect of data transferring, the error-bounded lossy compression has been effective in significantly reducing the overall transfer time cost over Globus, according to a recent study [34]. In that work, the authors utilized only SZ3 to compress different scientific simulation datasets including climate data, and then transferred the compressed data between geographically distributed data servers. Experiments showed that the total time cost can be reduced by 76% compared with the data transfer time without the compression technique.

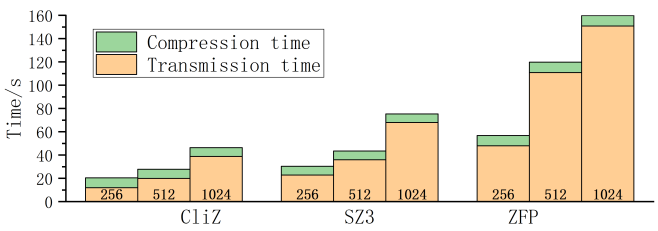
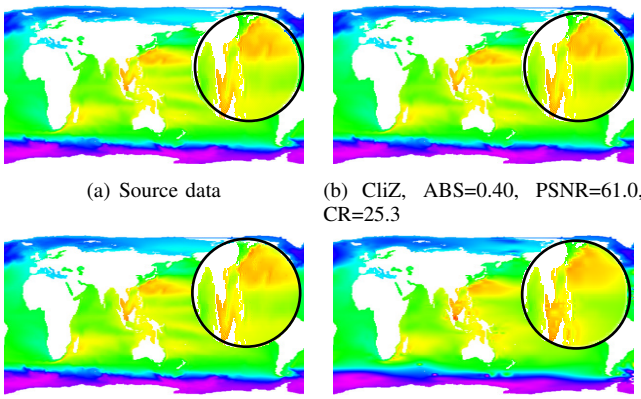


Fig. 13. Compression and Transmission Time on Globus (256, 512, and 1024 refer to the number of cores)

In our experiment, we tune the three compressors we use to have the same PSNR data distortion level, which is about 117 dB. As shown in Fig. 13, all three compressors have similar compression time costs. More specifically, when using 1024 cores, CliZ, SZ3, and ZFP took 7.37 seconds, 7.38 seconds, and 8.82 seconds, respectively. In contrast, it took much less time to transfer the CliZ-compressed data (only about 68 seconds) versus the solution with SZ3 (114 seconds) or ZFP (311 seconds). The overall data transfer between the two sites (ANL Bebop to Purdue Anvil) can be reduced by $32\% \sim 38\%$. This is because CliZ can lead to much higher compression ratios and thus much smaller data sizes to transfer.

We also evaluate the visual quality of the different lossy compressors when they are tuned to be with the same compression ratios on the climate datasets, especially because climate researchers often need to analyze the climate simulation data by visualization of the corresponding datasets [15], [17]. As shown in Fig. 14, the lossy reconstructed data under CliZ has a very high visual quality compared with the original dataset. At the same time, SZ3 and QoZ both suffer from obvious data distortions in visualization at the same (or lower) compression ratio of 25.



(a) Source data

(b) CliZ, ABS=0.40, PSNR=61.0, CR=25.3

(c) SZ3, ABS=1.87, PSNR=48.0, CR=25.3

(d) QoZ, ABS=200, PSNR=27.8, CR=15.6

Fig. 14. Visualization of Source and Compressed Data

VIII. Conclusion and Future Work

We propose a novel error-bounded lossy compressor – CliZ, which is optimized for climate datasets. To improve the data prediction accuracy, we explore the usage of mask-map, dimension permutation/fusion method, and leveraging periodical patterns across multiple snapshots. We also develop a novel quantization bin classification method, which can effectively improve the Huffman encoding efficiency. Key insights based on our experiments are summarized as follows.

- With the same PSNR, CliZ can significantly improve the compression ratio by 2-8× over SZ3 in most cases. It also outperforms the second-best compressor (SZ3, SPERR, or QoZ1.1) by 20%-200% in compression ratio.
- With the same SSIM, CliZ can get 2× compression ratios in general compared with the second-best choice (SZ3, QoZ1.1, or SPERR).

- CliZ has very similar compression and decompression time cost with SZ3 and ZFP on the climate data compression, and it is substantially faster than SPERR.
- The experiments with data transfer between two remote sites on WAN (ANL Bebop to Purdue Anvil) show that the overall Globus data transfer cost can be reduced by $32\% \sim 38\%$ especially because of the substantially higher compression ratio.

In the future, we plan to integrate the CliZ into HDF5 [35] and NetCDF [36] to service as many climate users as possible in practice.

ACKNOWLEDGMENTS

This research was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357, and supported by the National Science Foundation under Grant OAC-2003709, OAC-2104023, OAC-2311875, OAC-2330367, OAC-2311756, and OAC-2313122. We acknowledge the computing resources provided on Bebop (operated by Laboratory Computing Resource Center at Argonne) and ACCESS Anvil machine operated by Purdue University.

REFERENCES

[1] J. E. Kay and et al., "The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability," *Bulletin of the American Meteorological Society*, vol. 96, no. 8, pp. 1333–1349, 2015.

[2] J. W. Hurrell, M. M. Holland, P. R. Gent, S. Ghan, J. E. Kay, P. J. Kushner, J.-F. Lamarque, W. G. Large, D. Lawrence, K. Lindsay *et al.*, "The community earth system model: a framework for collaborative research," *Bulletin of the American Meteorological Society*, vol. 94, no. 9, pp. 1339–1360, 2013.

[3] S. Di and F. Cappello, "Fast error-bounded lossy HPC data compression with SZ," in *IEEE International Parallel and Distributed Processing Symposium*, 2016, pp. 730–739.

[4] 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 *2017 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2017, pp. 1129–1139.

[5] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.

[6] K. Zhao, S. Di, X. Liang, S. Li, D. Tao, Z. Chen, and F. Cappello, "Significantly improving lossy compression for hpc datasets with second-order prediction and parameter optimization," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 89–100. [Online]. Available: https://doi.org/10.1145/3369583.3392688

[7] X. Yu, S. Di, K. Zhao, jiannan Tian, D. Tao, X. Liang, and F. Cappello, "Szx: an ultra-fast error-bounded lossy compressor for scientific datasets," *arXiv preprint arXiv:2201.13020*, 2022.

[8] J. Liu, S. Di, K. Zhao, X. Liang, Z. Chen, and F. Cappello, "Dynamic quality metric oriented error bounded lossy compression for scientific datasets," in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2022, pp. 1–15.

[9] ——, "Faz: A flexible auto-tuned modular error-bounded compression framework for scientific data," in *Proceedings of the 37th International Conference on Supercomputing*, ser. ICS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1–13. [Online]. Available: https://doi.org/10.1145/3577193.3593721

[10] K. Zhao, S. Di, M. Dmitriev, T.-L. D. Tonellot, Z. Chen, and F. Cappello, "Optimizing error-bounded lossy compression for scientific data by dynamic spline interpolation," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 1643–1654.

[11] A. M. Gok, S. Di, Y. Alexeev, D. Tao, V. Mironov, X. Liang, and F. Cappello, "PaSTRI: Error-bounded lossy compression for two-electron integrals in quantum chemistry," in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2018, pp. 1–11.

[12] "GAMESS: Enabling GAMESS for exascale computing in chemistry and materials," https://www.exascaleproject.org/wp-content/uploads/2019/10/GAMESS.pdf, 2020, online.

[13] K. Zhao, S. Di, P. Danny, Z. Chen, and F. Cappello, "Mdz: An efficient error-bounded lossy compressor for molecular dynamics simulations," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022.

[14] 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*, ser. HPDC '14. NY, USA: ACM, 2014, pp. 203–214.

[15] A. H. Baker, H. Xu, D. M. Hammerling, S. Li, and J. P. Clyne, "Toward a multi-method approach: Lossy data compression for climate simulation data," in *High Performance Computing*. Springer International Publishing, 2017, pp. 30–42.

[16] A. Poppick, J. Nardi, N. Feldman, A. H. Baker, A. Pinard, and D. M. Hammerling, "A statistical analysis of lossily compressed climate model data," *Computers & Geosciences*, vol. 145, p. 104599, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S009830042030580X

[17] R. Underwood, J. Bessac, S. Di, and F. Cappello, "Understanding the effects of modern compressors on the community earth science model," in *2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2022, pp. 1–10. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/DRBSD56682.2022.00006

[18] D. Tao, S. Di, H. Guo, Z. Chen, and F. Cappello, "Z-checker: A framework for assessing lossy compression of scientific data," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 285–303, 2019.

[19] F. Cappello, S. Di, S. Li, X. Liang, G. M. Ali, D. Tao, C. Yoon Hong, X.-c. Wu, Y. Alexeev, and T. F. Chong, "Use cases of lossy compression for floating-point data in scientific datasets," *International Journal of High Performance Computing Applications (IJHPCA)*, vol. 33, pp. 1201–1220, 2019.

[20] 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*. IEEE, 2018.

[21] "Sperr," https://github.com/NCAR/SPERR.

[22] 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, pp. 2891–2903, 2019.

[23] J. Liu, S. Di, K. Zhao, S. Jin, D. Tao, X. Liang, Z. Chen, and F. Cappello, "Exploring autoencoder-based error-bounded compression for scientific data," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021, pp. 294–306.

[24] A. Glaws, R. King, and M. Sprague, "Deep learning for in situ data compression of large turbulent flow simulations," *Physical Review Fluids*, vol. 5, no. 11, p. 114602, 2020.

[25] T. Liu, J. Wang, Q. Liu, S. Alibhai, T. Lu, and X. He, "High-ratio lossy compression: Exploring the autoencoder to compress scientific data," *IEEE Transactions on Big Data*, 2021.

[26] L. Hayne, J. Clyne, and S. Li, "Using neural networks for two dimensional scientific data compression," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 2956–2965.

[27] X. Liang *et al.*, "SZ3: A modular framework for composing prediction-based error-bounded lossy compressors," https://arxiv.org/abs/2111.02925, 2021, online.

[28] Y. Collet, "Zstandard – real-time data compression algorithm," *http://facebook.github.io/zstd/*, 2015.

[29] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel techniques for compression and reduction of scientific data—the univariate case," *Computing and Visualization in Science*, vol. 19, no. 5, pp. 65–76, 2018.

[30] C. S. Zender, "Bit grooming: statistically accurate precision-preserving quantization with compression, evaluated in the netCDF Operators

(NCO, v4. 4.8+)," *Geoscientific Model Development*, vol. 9, no. 9, pp. 3199–3211, 2016.
[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[32] M. Frigo and S. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216 –231, Feb. 2005.
[33] "Bebop," https://www.lcrc.anl.gov/systems/resources/bebop/.
[34] Y. Liu, S. Di, K. Chard, I. Foster, and F. Cappello, "Optimizing scientific data transfer on globus with error-bounded lossy compression," in *43rd IEEE International Conference on Distributed Computing Systems (IEEE ICDCS)*. IEEE, 2023.
[35] HDF5. [Online]. Available: http://www.hdfgroup.org/HDF5
[36] R. Rew and G. Davis, "NetCDF: an interface for scientific data access," *IEEE Computer Graphics and Applications*, vol. 10, pp. 76–82, 1990.