2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD) | 978-1-6654-6337-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/DRBSD56682.2022.00006

Understanding the Effects of Modern Compressors on the Community Earth Science Model

Robert Underwood*, Julie Bessac*, Sheng Di*, Franck Cappello*
* Argonne National Laboratory, Lemont, Illinois, USA
{runderwood, jbessac, sdi, cappello}@anl.gov

***Abstract*—The Community Earth Science Model (CESM) is an important tool in climate modeling that produces a large volume of data on each simulation. Researchers have increasingly been turning to both lossless and lossy compression as an approach to reduce the volume of data for the CESM climate applications. Choosing the best-qualified compressor is nontrivial, however, especially because of the advent of many modern lossless and lossy compressors and complicated scientific integrity assessment of climate data model. In this paper we evaluate 11 state-of-the-art compressors using the quality assessments developed by climate scientists to understand the effectiveness of the compressors on the CESM climate datasets with four different models. Our work also identifies the best compression ratio that can be reasonably achieved while meeting these strict quality requirements.**

***Index Terms*—lossy compression, lossless compression, sz, ZFP, MGARD, CESM**

I. Introduction

The Community Earth System Model (CESM) [1], [2]—a well-known climate research package—is a fully coupled global climate model to simulate the past, present, and future climate states of the Earth. These packages are used in large-scale simulations that produce extreme volumes of data, which are analyzed by scientists. For instance, nearly 2.5 PB of data were produced by the CESM for the Coupled Model Intercomparison Project, from which 170 TB were postprocessed and submitted to the Earth System Grid [3].

Error-bounded lossy compression has been considered as a solution to resolve big-data issues in climate simulations because it provides high compression ratios ($\approx 100\times$) and controls the data distortion based on user-specified error bounds. By comparison, lossless compressors such as zlib [4], zstd [5], and FPZIP [6] are not suitable for compressing enormous scientific datasets where large compression ratios are needed, because they systematically suffer from substantially lower compression ratios (one or two orders of magnitude lower in general) than do lossy compressors, although they can ensure the identical/lossless reconstructed dataset.

Although error-bounded lossy compressors allow users to control the data distortion with multiple different types of error bounds (such as absolute error bound, relative error bound, and peak signal-to-noise ratio), a significant gap remains for practical usage in that users often have specific requirements (e.g., particular quantity of interest or metric to preserve) regarding their post hoc analysis, and climate simulation is no exception. A number of studies have been done on the impact of applying lossy compressors on the post hoc analysis of the climate simulation datasets. The National Center for Atmospheric Research, for example, has been applying compression to CESM data for years. Specifically, Baker et al. [7] addressed the issue of striking a balance between meaningfully reducing data volume and preserving the integrity of the simulation data. They also identified specific challenges and concerns when compressing climate data from CESM. In addition, they performed a thoughtful spatiotemporal statistical analysis of CESM simulation output data affected by only ZFP compressor. Poppick et al. [8] analyzed the daily average surface temperature and daily average precipitation rate from a historical run of the CESM Atmosphere Model based on only two lossy compressors—SZ1.4 [9] and ZFP 0.5 [10].

In this paper we perform a comprehensive study to understand the effects of 11 modern compressors on CESM datasets including atmosphere, land, ocean, and ice models. We include both leading lossless compressors (such as zstd [5] and NDZip [11]) and lossy compressors (such as SZ [9], [12]–[16], SZ3 [17], ZFP [10], FPZIP [6], MGARD [18]–[20], MGARDx [21], Bit Grooming [22], and Digit Rounding [23]). Our evaluation also covers many quality assessments (such as structural similarity index measure (SSIM/d-SSIM), p-value of KS-test, Pearson's coefficient of determination, and spatial relative error), recommended by the climate researchers [7].

Our contributions are as follows.

1) We conduct an evaluation of 11 compressors, examining data from all four models of CESM, and we consider the performance of the compressor with regard to the state of the art in assessing compressor quality in climate science.
2) We discuss challenges with using the KS-test for compressor quality assessment for climate data and propose alternative approaches to be evaluated by the climate community.
3) We demonstrate a previously unevaluated compressor SZ3, which achieves the highest compression ratio on data from CESM.
4) We suggest a path forward for the design of lossy compressors on datasets containing many small buffers such as data from CESM's land and ice models.

The remainder of the paper is organized as follows. In Section II we describe the background of the research including

978-1-6654-6337-9/22/$31.00 ©2022 IEEE
DOI 10.1109/DRBSD56682.2022.00006

the CESM model, quality assessments for climate research, and compressors used in our study. In Section III we discuss related work. In Section IV we present the methodology we used in our investigation. In Section V we present and analyze the assessment results. In Section VI we conclude the paper with a brief summary and a short discussion of future work .

II. Research Background

In this section we present background knowledge about CESM, quality assessments, and compressors.

A. The Community Earth Systems Model

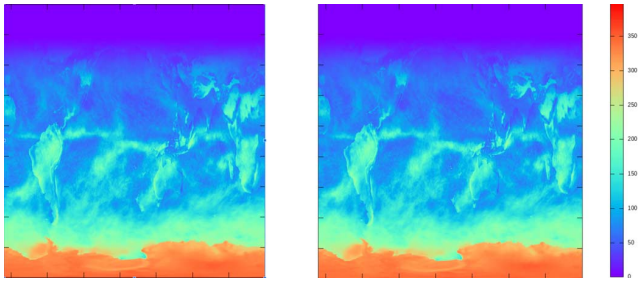

Fig. 1. FSUTOA from ATM. Upwelling solar flux at the top of the atmosphere: (left) original, (right) 21× small version compressed with SZ3 that meets all quality requirements from [24]; 91× smaller version is possible without the Kolmogorov–Smirnov test.

The Community Earth Systems Model was developed to provide a "core modeling system for studies of past and current climate, and projections of future climate change ... to address important scientific questions ... and [among other goals] support of U.S. national and international policy decisions" [1]. Therefore, accuracy of the model is of upmost importance. To this end, CESM also produces increasingly large volumes of data that can be used by scientists and policy makers to make informed inferences regarding climate.

Data for CESM is organized into multiple NetCDF files each representing a time-step in the simulation. Within each file, there are multiple variables (or fields) representing some physical or derived quantity from the simulation. We refer to an entire variable/field from within a single file as a buffer.

TABLE I
CESM DATASETS USED

Model	Datatype	Total Size	Per Buffer Size
Atmosphere	float32	1.5 TB	642 MB
Ocean	float32	235 GB	1.35 GB
Land	float32	41 GB	216 KB
Ice	float32	35 GB	480 KB

CESM has four major components: land, ice, ocean, and atmosphere. Each of these components produces differing volumes of floating-point data of distributions, as summarized in Table I. The largest of these is the atmosphere model, which has been studied extensively in the context of data reduction [7], [24]–[26], but the other fields are large as well, also warranting consideration for data reduction. Figure 1 provides a visualization of one field used in our experiments, the original field, and the corresponding lossy compressed field.

B. Quality Assessments for Climate Research

Baker et al. [7] proposed a series of quality assessment tools [1] for climate data. They proposed four assessments that have been refined in their later work: the structural similarity image metric (SSIM/d-SSIM), the p-value of the Kolmogorov–Smirnov (KS) test, the Pearson correlation coefficient of determination (R^2), and the spatial relative error (SRE). Each of these quality assessments can be viewed as often holding a value between 0 and 1. In their paper, Baker et al. proposed acceptable thresholds for these assessments.

The structural image similarity metric (SSIM) measures the degree of differences between original and decompressed data in two images [27]. For example, for two images x and $\hat{x}$ the following is computed:

$$SSIM(x,\hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)},$$

with $\sigma_{\cdot}$ the standard deviation of the studied data. Baker et al. later refined SSIM into data-SSIM or d-SSIM [25]. Their refined version attempts to better account for uses of the method for climate science by normalizing the inputs and changing certain constants used for perceptual corrections. Various studies have identified acceptable values of the SSIM: $\geq$ .98 [7], .99 [28], .99995 [26], or .995 (d-SSIM) [25].

The KS-test is a statistical test of the equality of continuous probability density functions. Its test statistics are based on the maximum distance between two cumulative distribution functions (CDFs) or their empirical forms:

$$D_{n,m} = \sup_u |F_{x,n}(u) - F_{\hat{x},m}(u)|.$$

Associated p-values are typically tabulated from the asymptotic distribution of $D_{n,m}$. The acceptable p-value of the KS-test is specified to be $\geq$ .05 in [7]. We further discuss the assumptions and uses of KS-test p-values in Sect. V-A1 as they relate to our results.

The R^2 test measures the strength of a linear relationship between the original data x and the corresponding elements in the decompressed data $\hat{x}$:

$$R^2(x,\hat{x}) = 1 - \frac{\sum_i^N (x_i - \hat{x}_i)^2}{\sum (x_i - \bar{x})^2}.$$

The acceptable value of R^2 is $\geq$ .99999 [26].

The SRE test measures the percentage of elements that differ by more than a given value-range relative threshold:

$$SRE\,(x,\hat{x}) = \frac{\sum_{i=0}^N \left(1 \text{ if } \left|\frac{x_i - \hat{x}_i}{x_i}\right| > \delta \text{ else } 0\right)}{N}.$$

[1]We use the term"assessments" here to mean the combination of a quality metric combined with a pass/fail threshold used for determining whether the quality is acceptable. We use the term "assessment" rather than "metric" because of the inclusion of the threshold and instead of "test" to avoid the ambiguity with frequentist statistical testing

The acceptable value of SRE is $\leq 5\%$ at $\delta \leq 1 \times 10^{-4}$ [7].

C. Modern Compressors

In Appendix A we describe the state-of-the-art modern compressors used by our study. Compared with prior works that evaluated lossy compressors relative to these assessments [7], [25], [26], we adopt newer versions where quality improvements have been made, and we include many previously unevaluated compressors such as MGARD, MGARDx, Digit Rounding, Bit Grooming, TThresh, and SZ3. We include FPZIP and Zstandard as leading lossless compressors.

We interface with the compressors via LibPressio [29] which provides a common interface for many compressors allowing us to write our experiments once instead of for each compressor. LibPressio also provides features to utilize equivalence relationships between different notions of error bounds such as the absolute point-wise error bound and the value-range relative point-wise error bound further simplifying implementation. Lastly LibPressio provides features for interfacing with NetCDF files (as well as other formats) which we use to read the data from its raw format.

We additionally utilize OptZConfig [30] which is built atop of LibPressio to identify which configurations will meet the quality assessment requirements from the previous section. The recommended configuration[2] of OptZConfig uses black-box optimization techniques to find the maximum compression ratio subject to some constraints on the quality. It accomplishes this via an iterative process which systematically tries different configurations retrieving the quality assessment results via APIs provided by LibPressio. Specific information on how we use OptZConfig can be found in Section IV.

III. Related Work

Cappello et al. [31] proposed the idea of classifying quality analysis tools into three levels. Level 1 analysis tools assess qualitative losses in quality and often leverage visualization to identify artifacts. Level 2 analysis tools assess quantitative losses in quality using general community measures such as PSNR. Level 3 analysis tools leverage the domain-specific quantitative quality measures such as those proposed in Section II-B.

Much of the work in the climate community has been a mix of levels 1, 2, and 3. The early work of Baker et al. [32] did not suggest thresholds but did suggest that the derivation relationships between climate variables should be accounted for when assessing quality requirements. The authors made the important observation that while some changes may be detectable, they may not be consequential. Baker et al. [7] used a mix of visualization (level 1) and quality assessments (levels 2 and 3) to assess what quality requirements may be amenable to the climate community for evaluating lossy compression. Poppick et al. [8] used visualization (level 1) to support a more detailed quantitative and qualitative analysis of the artifacts introduced by compression. Baker et al. [26] used a set of image quality assessments (level 2) and asked domain specialists to determine which images for them were diagnostically lossless (level 3). Pinard et al. [24] introduced a Python library LDCPY to compute these assessments for data stored in NetCDF format and focused on the four assessments described above. More recently, researchers have refined the classic SSIM metric into a specialized version for climate science promoting a level 2 metric to a level 3 one [25].

Similar work has been done outside the climate community to evaluate the effects of lossy compression (e.g., Laney et al. [33]), but we focus here on climate applications and their quality assessment.

In order to support identifying compressor configurations that preserve level 2 and 3 quality measures, a few approaches have been taken. One approach is to use bounded-linear functionals [20]. This approach works for many visualization tasks but is limiting for these assessments because they contain nonlinear terms. Another approach is to use black-box compression optimization [30], [34]. This approach is able to preserve these quality assessments but at a high overhead when used on individual buffers.

IV. Methodology

We downloaded CESM data from the Argonne National Laboratory Computing Resource Center (LCRC)[3]. We selected the runs for each component with the greatest volume of data. The datasets we used are summarized in Table I. These datasets are either 2D or 3D. We selected two fields at random for each dataset and considered several random timesteps and an additional field from the atmosphere because that model had the most data.

We selected: QI (Ice, 2D, 384×320), AICE (Ice, 2D, 384×320), DISPVEGC (Land, 2D, 288×192), QOVER(Land, 2D, 288×192), KE(Ocean, 3D, $3600 \times 2400 \times 42$), TEMP(Ocean, 3D, $3600 \times 2400 \times 42$), PRECT(Atmosphere, 2D, 3600×1800), T(Atmosphere $3600 \times 1800 \times 26$), and FSUTOA(Atmosphere, 2D 3600×1800). For Ice we chose time-steps 1909.09, 1921.11, 1926.07, 1994.08. For Land we chose time-steps 1978.11,1979.03, 1985.01 1987.02, 1991.08. For Ocean we chose time-steps 0147.01 and 0147.12 [4]. For Atmosphere we choose timesteps 0001.03, 0001.11, 0001.11, 0002.10, 0004.12, and 0005.04.

We ran the experiments on Bebop at the LCRC. We selected the machine for its availability of CPU nodes. Because of lack of time, space, maturity of the GPU-based compressor implementations [30], and lack of support for GPU compressors in I/O used in climate science libraries and codes [29], we leave the evaluation of GPU compressors to future work.

Software was chosen to be the latest available versions on Spack [35] and an additional repository for compressors [36] at the time of experimentation maintained by the LibPressio developers. Exact versions are listed in Table II. We use LibPressio [29] to provide a consistent interface between

[2]We used `opt:search=fraz`, `opt:max_iterations=50`, `fraz:search_threads=$(ncpus)`

[3]https://trac.mcs.anl.gov/projects/parvis/wiki/Datasets

[4]there were many fewer time-steps from Ocean models available so we choose fewer of them

TABLE II
HARDWARE AND SOFTWARE VERSIONS

Component	Version	Component	Version
CPU	Intel Xeon E5-2695v4	MGARDx	0.0.1
Memory	128 GB DDR4	NDZip	0.0.1
Network	Intel OmniPath	SZ	2.1.12.2
GCC	10.2.0	SZ3	3.1.5.4
Bit Grooming	2.1.9	TThresh	0.0.5
Digit Rounding	2.1.9	ZFP	0.5.5
FPZIP	1.3.1	Zstandard	1.21.1
MGARD	1.0.0	LibPressio	0.86.5
OptZConfig	0.13.3		

different compressors, abstracting away differences in the compressor interface, such as the order of dimensions, which fields are mutated, how datatype information is passed, and how compressors refer to the absolute pointwise error bound if it is supported.

We read the data directly from the netCDF files created by CESM using the LibPressio support for netCDF. We conducted compression and decompression operations on an in-memory copy of each buffer.

We wrote a LibPressio external metric to compute the d-SSIM and KS-test. The KS-test was brought in from Scipy, which is implemented in C, and the d-SSIM was ported to Julia from the implementation in LDCPY [24]. We chose to port d-SSIM because it was not easy to call just this function from LDCPY and because Julia provides additional performance. An improved implementation would embed this computation. We used the Pearson's correlation coefficient and spatial relative error implementations from LibPressio's core metrics implementations in C++.

With LibPressio-compatible metrics implementations for each quality metric, we can use OptZConfig [30] to identify the compression configuration that resulted in the highest compression ratio while satisfying compressor constraints. We chose the value-range relative error bound mode as the only input for compressors that support absolute error bounds and used lower_bound = 1×10^{-155}, upper_bound=1×10^{-1}, max iterations=50, objective_mode=max. For compressors that support other notions of error bounds (i.e. Bit Grooming), we used the largest and smallest possible values of those bounds in OptZConfig.

Rather than the penalty functions used in [30] that mark outcomes that don't meet quality objectives with a large negative value, we instead used -1 $\times$ the Manhattan distance from the assessments to the closest configuration that met all the quality requirements. This approach enabled us to search a path toward a feasible configuration more easily than simply returning $-\infty$.

Methods like OptZConfig can have high overhead when used for a single buffer [30], but they often find reusable configurations that can be used for many timesteps [34]. The bottleneck in these operations for most compressors (except TThresh and MGARD) is the computation of the assessments, which may benefit from acceleration.

[5]IEEE 32 bit floating point values have 23 mantissa bits and have varying precision over their domain. 10^-15 is intended to be a value so small as to intuitively be unnecessary for most uses

V. RESULTS AND TAKEAWAYS

A. Compressors That Do Best on CESM-Atmosphere?

We first consider CESM-Atmosphere since it has the most data and is the most explored in the literature. We present results for the "upwelling solar flux at top of atmosphere" (FSUTOA) field from CESM-Atmosphere (Figure 1)[6]. Other fields from CESM-Atmosphere follow similar patterns with varying compression ratios depending on the field.

1) KS-test: We first observe that different absolute error bounds are needed to preserve these assessments with different compressors. The reason is that different compressors produce different distributions of compression error. Considering the p-value of the KS-test in that context presents a special challenge. For example, to meet all compression requirements from [24], SZ needs a value-range relative error bound of $\ll 1 \times 10^{-15}$, whereas ZFP needs only a value-range relative error bound of $< 6 \times 10^{-5}$, which affects substantially the corresponding compression ratios [7]. This difference in the error-bound requirements suggests that the KS-test may be both too sensitive to what may be inconsequential errors with value-range relative error magnitudes smaller than 1×10^-15 and, as we will show, too insensitive to larger pointwise errors.

Other properties of the KS-test lead to nonintuitive results when applied as a measure of quality. The KS-test has been shown to have a low power (probability of correctly rejecting the null hypothesis) when testing normality [37], [38] in comparison with other commonly used tests such as the Anderson–Darling test. This lack of power is also illustrated in the following numerical experiments. Additionally, the power and p-value of a statistical test are known to depend on the number of datapoints. In particular, the KS-test is known to be overly sensitive to small differences in large datasets, hence overrejecting the null hypothesis for large datasets. One of the preferred alternatives to the KS-test is the Anderson–Darling test because it considers the difference between the two entire empirical CDFs and not only the maximum difference between CDFs as does the KS-test. However, the Anderson–Darling test is computationally expensive.

We conducted the following numerical experiments with simulated samples and altered data to highlight some limitations of the KS-test:

(1) **Pointwise error** To study the effects of high pointwise

[6]FSUTOA was chosen randomly from the atmosphere variables considered because atmosphere is the most well studied

[7]The KS test measures changes in the empirical CDF function. SZ introduces different distributions of error depending on the error bound and value range of the data [9] which likely induces the failure of the KS test; however with value range relative error much less than 10^{-15} we argue these differences will be found to be inconsequential. ZFP on the other hand consistently introduces errors more faithful to the original dataset's distribution; at least as measured by the KS test (there still are errors; an example of systematic artifacts from ZFP in [8]; these are improved, but still present in the latest version). Lastly ZFP tends to over-preserve with respect to the error bound which it may benefit from here as well

errors, we generated 50 repeats of Gaussian samples from $\mathcal{N}(0,1)$ with increasing size $n = 10^i$, for $i = 1, ..., 7$. One sample has 1% of pointwise corruption consisting of 1% of the values assigned randomly and uniformly between 10^{10} and 10^{15}. Corresponding distributions of the KS-test p-values are shown in Figure 2, top panel. We observe that for these samples, the KS-test fails to detect the pointwise error on datasets with less than 10^5 datapoints. This result highlights the high sensitivity of the test to data size but also its insensitivity to substantial pointwise error.

(2) **Noise and bias** Over 50 repeats we generated Gaussian samples of increasing size from $\mathcal{N}(0,1)$, and we added a bias $\epsilon \sim \mathcal{N}(0.01, 0.2)$ to one sample. This type of error can be a typical compression error; in practice, the error range of value is smaller. We report the corresponding KS-test p-value distributions in the bottom panel of Figure 2. Once again, the test fails to detect the bias in the mean and variance for smaller datasets, highlighting the limitations of the KS-test.

(3) **Sample size** Another KS-test experiment was run between two Gaussian samples of the same distribution $\mathcal{N}(0,1)$ with increasing sample size $n = 10^i$, for $i = 1, ..., 9$. The KS-test fails to accept the null hypothesis when the sample size is over 10^8, highlighting the hypersensitivity of the test for large datasets. This result creates doubts about whether the KS-test properly detected the corruption and bias or was sensitive to any variations in large datasets.

These issues may be caught by other assessments proposed by Pinard et al. [24], and because the proposed assessments are designed to be used in concert are not going to be accepted; however, requiring the KS-test punishes the performance of some compressors over others for differences that may not be consequential to the user. We propose that the climate community consider alternative methods that may more accurately reflect the desired ability to ensure that distributions do not meaningfully change.

TABLE III
COMPARING MEASURES OF DISTRIBUTION DIFFERENCES ON ATMOSPHERE DATA WITH SMALL INJECTED NOISE: $\mathcal{U}$ MEANS UNIFORM NOISE, $\mathcal{N}$ MEANS GAUSSIAN NOISE

Dataset Name	$p(KS-test)$	Wasserstein	Hellinger	J-S
FSUTOA($\mathcal{U}$)	$< 2 \times 10^{-16}$	3×10^{-17}	0	10338
FSUTOA($\mathcal{N}$)	$< 2 \times 10^{-16}$	1×10^{-17}	0	10338
T($\mathcal{U}$)	1	0	0	0
T($\mathcal{N}$)	1	0	0	0
PRECT($\mathcal{U}$)	0.0067	2×10^{-16}	0	1.71
PRECT($\mathcal{N}$)	0.0065	8×10^{-17}	0	1.67

As pointed out earlier, since statistical tests may require users to adapt the error bounds to meet the p-value requirements, alternatives to statistical tests should also be considered. In particular, many metrics for probability densities have been developed. For instance, the Hellinger distance[8] is commonly used because it provides an intuitive global measure of distance between two distributions [39]. The Wasserstein distance, also known as the Earth's mover distance [40], [41], is commonly used in climate studies [42]. In one-dimensional settings, the Wasserstein distance corresponds to an L^p-norm between the quantile functions of the distributions at stake, providing an intuitive and exhaustive comparison of distributions. Another commonly used divergence is the Jensen–Shannon divergence[9], seen as a symmetrized version of the Kullback–Leibler divergence. Following the common practices, these metrics are calculated with the empirical estimates of the pdfs and cdfs at stake.

In Table IV we gather the values of these three metrics for the two examples shown in Figure 2 (experiments 1 and 2 from above). The three metrics equal 0 when the compared distributions are equal. We observe that these metrics unmistakably discriminate the corrupted and biased samples from the original samples.

We also consider two cases of adding noise with distribution $\mathcal{N}(0, 1 \times 10^{-16})$ and with distribution $\mathcal{U}(-4 \times 10^{-16}, 4 \times 10^{-16})$ to the three atmosphere datasets FSUTOA, precipitation rate (PRECT), and temperature (T). We summarize these results in Table III. We find that the KS-test determines a significant difference for the FSUTOA and PRECT cases but does not find a difference for T (p=1), while FSUTOA and T have similar ranges of values. Further investigations will be pursued on the temperature field. We also compute the Wasserstein distance, the Jensen–Shannon divergence, and the Hellinger distance for each of these results. Both Wasserstein and Hellinger distances show intuitive small value results since the added errors are very small. In addition, the Wasserstein distance discriminates between the two types of errors while showing very small values. On the other hand, the Jensen–Shannon lacks comprehensive interpretation on this example.

Key findings: The p-value of the KS-test may be too strict and unreliable a requirement prohibiting the adoption of compressors that may be otherwise acceptable. The Wasserstein distance may be a good candidate for a replacement.

2) Other quality assessments: In this section we run the search for each quality metric independently to show which quality metric is the limiting factor for achieving high compression ratios for each compressor. Because of the reasons outlined in the preceding subsection, we exclude the KS-test and focus on the other three quality assessments from [24]. Table V shows which quality metric is the limiting factor, indicated by which compression ratio is the smallest for each compressor. Two compressors, MGARD and MGARDx, fail to preserve the d-SSIM satisfactorily. The best lossy compressor on this dataset (SZ3) gets a compression ratio of 59.81, surpassing each of the three quality assessments. For comparison, the best lossless compressor FPZIP gets a compression ratio of 1.95. The other compressors either get a

[8] $H^2(p,q) = \frac{1}{2}\int_{\mathcal{X}} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx$ with p and q the probability density functions to be compared

[9] $JSD(p,q) = \frac{1}{2}D(p,m) + \frac{1}{2}D(q,m)$ where p and q are the probability density functions to be compared and m is the uniform mixture of p and q, $D(p,q) = \int_{\mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$

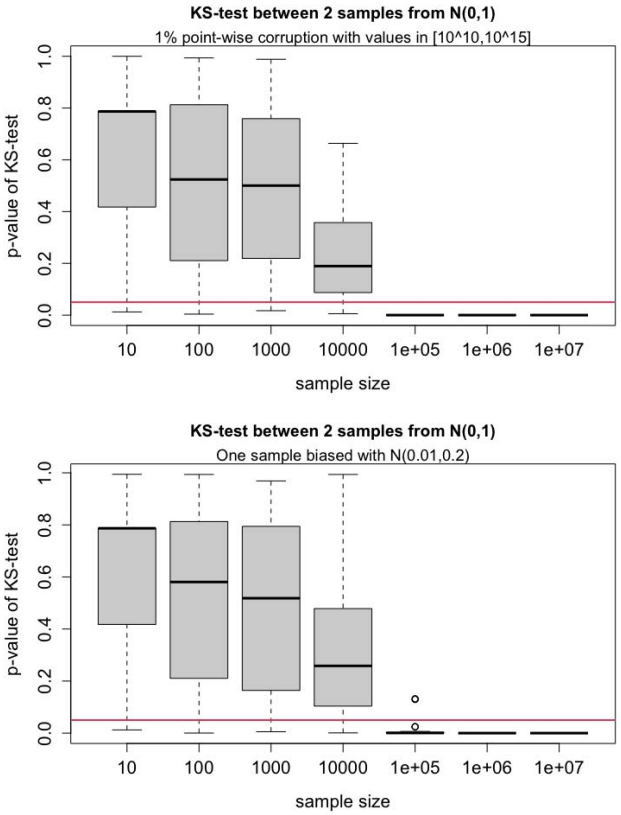

Fig. 2. Distributions of p-values for the KS-test over 50 times when testing whether two samples with increasing sample size $n = 10^i$, for $i = 1, ..., 7$, can be considered from the same distribution via the KS-test. The red line corresponds to the nominal value 0.05 below which the test typically rejects the null hypothesis. Top panel: two samples from $\mathcal{N}(0, 1)$ with one sample corrupted with 1% of the values assigned randomly and uniformly between 10^{-10} and 10^{-15}. Bottom panel: two samples from $\mathcal{N}(0, 1)$ with one sample being added a bias in the mean and variance via an added random variable $\epsilon \sim \mathcal{N}(0.01, 0.2)$. The results suggest that the p-value of the KS-test is not a good fit at large data sizes. The top boxplots highlights the over sensitivity of the KS-test when sample size increases, in the meantime the bottom boxplots highlights the lack of discrimination of samples tained by errors by the KS-test.

worse compression ratio or cannot pass the assessments. We observe similar results for other CESM-Atmosphere fields.

For some compressors, we understand why particular compressors perform better than others. For example, TTHRESH does not perform well on the datasets (including FSUTOA) that are 2D datasets [43]. Likewise, SZ3 generally outperforms Bit Grooming and Digit Rounding because of its more sophisticated prediction scheme and better lossless encoder (Zstd vs gzip) [17], [44], [45]. Between leading compressors such are SZ3 and ZFP, however, why a compressor is going to perform the best on a given dataset at a given error bound is unclear, requiring greater work to measure and understand lossy compression ability [46].

Finally, in Table VI we provide the proposed Wasserstein and Jensen-Shannon metrics computed on original and decompressed FSUTOA data. The Hellinger distance is not shown as it was degenerate due to the strong similarity between original

TABLE IV
MEDIAN VALUES OF THE PROPOSED METRICS OVER THE 50 REPEATS OF THE TWO EXAMPLES PRESENTED IN FIGURE 2 (EXPERIMENTS 1 AND 2 FROM ABOVE). FROM TOP TO BOTTOM, THE WASSERSTEIN DISTANCE, THE HELLINGER DISTANCE, AND THE JENSEN–SHANNON (J-S) DIVERGENCE. THE THREE METRICS EQUAL 0 WHEN THE COMPARED DISTRIBUTIONS ARE EQUAL. THE SYMBOL "X" MEANS THAT THE DISTANCE COULD NOT BE CALCULATED BECAUSE QUANTITIES ARE NOT INTEGRABLE.

Sample Size	10	10^2	10^3	10^4	10^5	10^6	10^7
Wasserstein ($\times 10^{13}$)	5	0.9	0.6	0.5	0.5	0.5	0.5
Hellinger	x	x	x	x	x	x	x
J-S	0.6	7.5	73.	738	7385	73767	737803
Wasserstein	0.50	0.18	0.06	0.03	0.02	0.02	0.02
Hellinger	0.29	0.12	0.05	0.02	0.01	0.01	0.01
J-S	0.7	7.1	74	736	7383	73782	737834

TABLE V
MAXIMUM COMPRESSION RATIO THAT MEETS EACH QUALITY METRIC FROM [24]. X INDICATES A FAILURE TO RUN TO A SOLUTION.

Compressor	Pearson R^2 2	Spatial Error	d-SSIM
SZ	30.65	31.49	39.86
SZ3	93.00	93.00	59.81
ZFP	13.27	13.27	18.87
Zstd	1.35	1.35	1.35
FPZIP	1.95	1.95	1.95
MGARD	27.10	4.69	X
NDZip	1.64	1.64	1.64
MGARDx	14.70	6.49	X
TTHRESH	16.10	16.10	2.98
Bit Grooming	1.51	1.51	1.51
Digit Rounding	1.86	1.86	1.86

and decompressed data. As discussed earlier, the Wasserstein and Jensen-Shannon metrics provide more gradual comparison of the compressors than the KS-test. In particular, MGARD, SZ and TTHRESH are assigned the same 0 p-value for the KS-test; however, their Wasserstein and Jensen-Shannon metrics provide more nuanced information about their compression errors and enables a ranking between these compressors.

Key findings: The best lossy compressor, SZ3, gets a $30.7\times$ improvement over lossless compressors and $3.2\times$ improvement over the next compressor while passing all three quality assessments.

B. Improving Performance For Land And Ice models In CESM

When we expand our analysis to other fields from other models, we find that depending on the field either SZ3 or ZFP has the best compression ratios that satisfy all three assessments on the atmosphere and ocean models. Currently, however, specialized encoding techniques such as FPZIP have the greatest compression ratios satisfying all three assessments on land and ice in the cases we tested. When examining the overhead costs that make lossy compressors perform worse, the metadata (i.e., settings used) and entropy data (i.e., Huffman trees) appear to generate too much overhead for small buffers. Some of Z-standard's more exotic features may point a way forward for lossy improvements, namely Common

TABLE VI
ILLUSTRATION OF PROPOSED METRICS ON DECOMPRESSED FSUTOA DATA. THE HELLINGER DISTANCE IS NOT SHOWN AS IT WAS DEGENERATE DUE TO THE STRONG SIMILARITY BETWEEN ORIGINAL AND DECOMPRESSED DATA.

	MGARD	SZ	TTHRESH	ZFP	MGARD	SZ	TTHRESH	ZFP
Bound	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-10}	10^{-10}	10^{-10}	10^{-10}
KS-test p-value	0.00	0.00	0.00	0.98	0.00	0.00	1.00	0.98
Wasserstein	125.261	7.46×10^{-4}	6.36×10^{-1}	3.56×10^{-5}	125.261	7.46×10^{-4}	0.00	3.56×10^{-5}
JS	583393.701	908.029	3981.525	97.793	583393.701	908.029	0.000	97.793

Dictionary" and "External Metadata." "Common Dictionary" allows a common Huffman tree provided by the user to be used for encoding for multiple buffers. "External Metadata" allows common metadata to be excluded from the compressed stream and stored in an auxiliary location and passed in separately to decompression. Together, these features allow the user to store only one copy of the metadata for all the buffers that share a common configuration, thus dramatically improving storage for compressors such as SZ/SZ3/MGARD/MGARDx/Bit Grooming/Digit Rounding/Zstandard/GZip (as well as compressors like SPERR [47] when Zstandard is used) which all feature entropy encoding stages and store this kind of metadata in their compressed byte streams.

For climate codes, however, leveraging these features would require improvements of HDF5. Currently, HDF5 stores configurations for compressors per dataset and must be encodable as a series of contiguous bytes. In order to achieve benefits from using compressors, the ability to de-duplicate the serialized compressor configurations between datasets is needed to actually improve the overhead incurred by application codes. For every timestep of the quantity of ice (QI) field from the ice application, the Huffman tree alone accounts for 52% of the output of SZ prior to the final lossless encoding stage. If externalized, there would be a significant increase in compression performance.

Additionally, in order to better support GPU-based compressors, better support for divergent compression and decompression configurations (i.e., the compression system had a datacenter-grade GPU, and the decompression system has a laptop-grade GPU with fewer resources or no GPU at all) is required, as well as better support for GPU primitives such as `cudaStream_t` that are not serializable, are consistent between uses, and are required for resource sharing.

Moreover, in order to see larger improvements for ice and land, more data needs to be passed to the compressors at a single time. Doing so would require applications codes to modify their usage of NetCDF/HDF5 to consider larger chunk sizes and to place related data into the same datasets instead of separate HDF5/NetCDF files. Not only will this improve the compression performance, it also improves runtime performance. As shown in Figure 3, it is faster to read larger data segments and decompress them than to read and decompress individual small slices for moderate increases in chunk sizes.

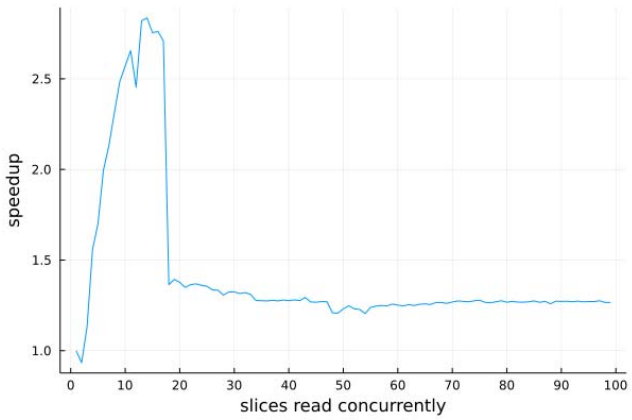

Fig. 3. Median speedup from reading larger chunks vs. serial independent reading of slices when using SZ $\epsilon_{abs} = 1\times10^{-4}$ on CLOUD from Hurricane on SDRBench [48], The drop in speedup corresponds to a systematic change in the read and compression performance that occurs when decompression no longer fits in the 45 MB L2+L3 cache but is still 1.5× faster than serial reads of individual slices. Tools such as OptZConfig [30] can automatically identify optimal configurations for number of slices read simultaneously.

Key findings: For models such as land and ice that have many small buffers, improvements to share overhead between buffers both for compressors and I/O libraries will be key to achieving high performance.

VI. CONCLUSIONS AND FUTURE WORK

We have evaluated 11 different compressors on data from all four models of CESM. We have highlighted how well each compressor is able to compress data under the quality assessments developed by the climate community. We further have identified challenges posed by the KS-test and proposed alternative methods for use by the climate community. Moreover, we propose a path forward for the design of lossy compressors for the land and ice models.

We see two areas for future work. In climate science, more work is needed to identify quality assessments that evaluate compressors that both meet the stringent needs of the climate community and identify meaningful changes between original and lossy compressed data. In compressor design, further work is needed to improve the ability of climate codes to adopt lossy compression. While this includes improvements to user-facing features such as packaging compression libraries for climate researchers to easily download and use, it also requires

improvements to I/O libraries and compressors to handle the unique challenges posed by climate data.

ACKNOWLEDGMENTS

This research was supported by the ECP, Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations -– the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nation's exascale computing imperative. The material 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 and OAC-2104023. We acknowledge the computing resources provided on Bebop (operated by Laboratory Computing Resource Center at Argonne).

REFERENCES

[1] 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, W. H. Lipscomb, M. C. Long, N. Mahowald, D. R. Marsh, R. B. Neale, P. Rasch, S. Vavrus, M. Vertenstein, D. Bader, W. D. Collins, J. J. Hack, J. Kiehl, and S. Marshall, "The Community Earth System Model: A Framework for Collaborative Research," *Bulletin of the American Meteorological Society*, vol. 94, no. 9, pp. 1339–1360, Sep. 2013.

[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] L. Cinquini, D. Crichton, C. Mattmann, J. Harney, G. Shipman, F. Wang, R. Ananthakrishnan, N. Miller, S. Denvil, M. Morgan, Z. Pobre, G. M. Bell, C. Doutriaux, R. Drach, D. Williams, P. Kershaw, S. Pascoe, E. Gonzalez, S. Fiore, and R. Schweitzer, "The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data," *Future Generation Computer Systems*, vol. 36, pp. 400–417, Jul. 2014.

[4] L. P. Deutsch, "RFC 1952 GZIP File Format Specification version 4.3," IETF, Tech. Rep., 1996.

[5] Facebook, Inc., "Zstandard - Real-time data compression algorithm," https://facebook.github.io/zstd/, Jun. 2019.

[6] P. Lindstrom and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1245–1250, Sep. 2006.

[7] 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*, J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, Eds. Cham: Springer International Publishing, 2017, vol. 10524, pp. 30–42.

[8] A. Poppick, J. Nardi, N. Feldman, A. H. Baker, and D. M. Hammerling, "A Statistical Analysis of Compressed Climate Model Data," *The 4th International Workshop on Data Reduction for Big Scientific Data (DRBSD-4)*, 2018.

[9] 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 (IPDPS)*, May 2017, pp. 1129–1139.

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

[11] F. Knorr, P. Thoman, and T. Fahringer, "Ndzip: A High-Throughput Parallel Lossless Compressor for Scientific Data," in *2021 Data Compression Conference (DCC)*, Mar. 2021, pp. 103–112.

[12] 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.

[13] 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, Mar. 2019.

[14] D. Tao, S. Di, X. Liang, Z. Chen, and F. Cappello, "Fixed-PSNR Lossy Compression for Scientific Data," in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2018, pp. 314–318.

[15] X. Liang, S. Di, D. Tao, Z. Chen, and F. Cappello, "An Efficient Transformation Scheme for Lossy Data Compression with Point-Wise Relative Error Bound," in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2018, pp. 179–189.

[16] 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)*. Seattle, WA, USA: IEEE, Dec. 2018, pp. 438–447.

[17] 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)*, Apr. 2021, pp. 1643–1654.

[18] 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. 5-6, pp. 65–76, Dec. 2018.

[19] ——, "Multilevel Techniques for Compression and Reduction of Scientific Data—The Multivariate Case," *SIAM Journal on Scientific Computing*, vol. 41, no. 2, pp. A1278–A1303, Jan. 2019.

[20] ——, "Multilevel Techniques for Compression and Reduction of Scientific Data-Quantitative Control of Accuracy in Derived Quantities," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2146–A2171, Jan. 2019.

[21] X. Liang, B. Whitney, J. Chen, L. Wan, Q. Liu, D. Tao, J. Kress, D. Pugmire, M. Wolf, N. Podhorszki, and S. Klasky, "MGARD+: Optimizing Multilevel Methods for Error-Bounded Scientific Data Reduction," *IEEE Transactions on Computers*, vol. 71, no. 7, pp. 1522–1536, Jul. 2022.

[22] 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, Sep. 2016.

[23] X. Delaunay, A. Courtois, and F. Gouillon, "Evaluation of lossless and lossy algorithms for the compression of scientific datasets in NetCDF-4 or HDF5 formatted files," Numerical Methods, Preprint, Nov. 2018.

[24] A. Pinard, D. M. Hammerling, and A. H. Baker, "Assessing Differences in Large Spatio-temporal Climate Datasets with a New Python package," in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 2699–2707.

[25] A. H. Baker, A. Pinard, and D. M. Hammerling, "DSSIM: A structural similarity index for floating-point data," *arXiv:2202.02616 [cs, stat]*, Feb. 2022.

[26] A. H. Baker, D. M. Hammerling, and T. L. Turton, "Evaluating image quality measures to assess the impact of lossy data compression applied to climate simulation data," *Computer Graphics Forum*, vol. 38, no. 3, pp. 517–528, Jun. 2019.

[27] Z. Wang, A. C. Bovick, H. R. Sheikh, and E. P. Simoncelli, "The SSIM Index for Image Quality Assessment," https://www.cns.nyu.edu/~lcv/ssim/, Feb. 2011.

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

[29] R. Underwood, V. Malvoso, J. C. Calhoun, S. Di, and F. Cappello, "Productive and Performant Generic Lossy Data Compression with LibPressio," in *2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7)*, Nov. 2021, pp. 1–10.

[30] R. Underwood, J. C. Calhoun, S. Di, A. Apon, and F. Cappello, "OptZConfig: Efficient Parallel Optimization of Lossy Compression Configuration," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–1, 2022.

[31] F. Cappello, S. Di, and A. M. Gok, "Fulfilling the promises of lossy compression for scientific applications," in *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and*

AI, J. Nichols, B. Verastegui, A. B. Maccabe, O. Hernandez, S. Parete-Koon, and T. Ahearn, Eds. Cham: Springer International Publishing, 2020, pp. 99–116.

[32] A. H. Baker, D. M. Hammerling, S. A. Mickelson, H. Xu, M. B. Stolpe, P. Naveau, B. Sanderson, I. Ebert-Uphoff, S. Samarasinghe, F. De Simone, F. Carbone, C. N. Gencarelli, J. M. Dennis, J. E. Kay, and P. Lindstrom, "Evaluating lossy data compression on climate simulation data within a large ensemble," *Geoscientific Model Development*, vol. 9, no. 12, pp. 4381–4403, Dec. 2016.

[33] D. Laney, S. Langer, C. Weber, P. Lindstrom, and A. Wegener, "Assessing the effects of data compression in simulations using physically motivated metrics," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. Denver Colorado: ACM, Nov. 2013, pp. 1–12.

[34] R. Underwood, S. Di, J. C. Calhoun, and F. Cappello, "FRaZ: A Generic High-Fidelity Fixed-Ratio Lossy Compression Framework for Scientific Floating-point Data," in *34th IEEE International Parallel and Distributed Processing Symposium*. New Orleans: IEEE, May 2020.

[35] T. Gamblin, M. LeGendre, M. R. Collette, G. L. Lee, A. Moody, B. R. de Supinski, and S. Futral, "The Spack package manager: Bringing order to HPC software chaos," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Austin Texas: ACM, Nov. 2015, pp. 1–12.

[36] R. Underwood, "Robertu94/spack_packages," Jan. 2022.

[37] M. A. Stephens, "EDF statistics for goodness of fit and some comparisons," *Journal of the American statistical Association*, vol. 69, no. 347, pp. 730–737, 1974.

[38] N. M. Razali, Y. B. Wah *et al.*, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests," *Journal of statistical modeling and analytics*, vol. 2, no. 1, pp. 21–33, 2011.

[39] D. Pollard, *A user's guide to measure theoretic probability*. Cambridge University Press, 2002, no. 8.

[40] M. Muskulus and S. Verduyn-Lunel, "Wasserstein distances in the analysis of time series and dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 1, pp. 45–58, 2011.

[41] F. Santambrogio, "Optimal transport for applied mathematicians," *Birkäuser, NY*, vol. 55, no. 58-63, p. 94, 2015.

[42] Y. Robin, P. Yiou, and P. Naveau, "Detecting changes in forced climate attractors with Wasserstein distance," *Nonlinear Processes in Geophysics*, vol. 24, no. 3, pp. 393–405, 2017.

[43] 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, Sep. 2020.

[44] "Bitgroomingz," https://github.com/disheng222/BitGroomingZ, online.

[45] "Digit rounding z," https://github.com/disheng222/digitroundingZ, online.

[46] D. Krasowska, J. Bessac, R. Underwood, J. C. Calhoun, S. Di, and F. Cappello, "Exploring Lossy Compressibility through Statistical Correlations of Scientific Datasets," in *2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7)*, Nov. 2021, pp. 47–53.

[47] "Sperr." [Online]. Available: https://github.com/NCAR/SPERR

[48] K. Zhao, S. Di, X. Lian, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello, "SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors," in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 2716–2724.

[49] "Sz2 github repo," http://github.com/szcompressor/SZ, online.

[50] "Zfp github repo," https://github.com/LLNL/zfp, online.

[51] "Mgard github repo," https://github.com/CODARcode/MGARD, online.

[52] "Mgardx github repo," https://github.com/lxAltria/MGARDx, online.

[53] "Nco with bit grooming," https://github.com/nco/nco, online.

[54] "Digit rounding code," https://github.com/CNES/Digit_Rounding, online.

[55] "Tthresh code," https://github.com/rballester/tthresh, online.

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

[57] "Sz3 modular composable compression framework," https://github.com/szcompressor/SZ3, online.

[58] "Fpzip," https://github.com/LLNL/fpzip, online.

[59] "Ndzip code," https://github.com/celerity/ndzip, online.

[60] J. Liu, S. Li, S. Di, X. Liang, K. Zhao, D. Tao, Z. Chen, and F. Cappello, "Improving lossy compression for sz by exploring the best-fit lossless compression techniques," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 2986–2991.

[61] "Zlib," https://zlib.net/, online.

APPENDIX

A. Description of Modern Compressors

Version of the compressors used are found in Table II.

- SZ [9], [12]–[16]. SZ (a.k.a. SZ2) is an error-bounded lossy compressor based on the classic prediction-based compression model. It splits each dataset into many subblocks (16x16 for 2D and 6x6x6 for 3D) and adopts a hybrid data prediction method that combines the Lorenzo predictor and linear-regression-based predictor in each block. Then, SZ2 uses a linear-scaling quantization to control the compression errors with the user-specified error bound, which is followed by a customized Huffman encoder and dictionary encoder (Zstd). The detailed design and code can be found in [16] and [49], respectively.
- ZFP [10]. ZFP is an error-bounded lossy compressor designed based on the data transform model. Unlike SZ, ZFP performs the exponent alignment and near-orthogonal transform on each small block (block size is 4x4 for 2D and 4x4x4 for 3D), followed by an embedded encoding to significantly reduce the data size. In general, ZFP has a better compression quality (e.g., higher visual quality with the same compression ratio) than does SZ2 on smooth 3D datasets but relatively low quality on 2D and 1D datasets according to prior studies [16], [17]. The ZFP code can be found in [50].
- MGARD/MGARDx [18]–[21]. MGARD [18]–[20], short for "MultiGrid Adaptive Reduction of Data," is also an error-controlled lossy compressor for scientific datasets. It supports not only error-bound control but also bounded-linear quantity of interest (QoI). In particular, MGARD provides optimization for the compression of unstructured datasets in addition to the structured mesh datasets. MGARDx (a.k.a., MGARD+) [21] is an improved version of MGARD, which can possess higher compression ratios and much higher throughput, because of a levelwise coefficient quantization method and an adaptive decomposition method, as well as a series of algorithmic optimization strategies. MGARD and MGARDx can be downloaded from [51] and [52], respectively.
- Bit Grooming [22]. Bit Grooming is an error-bounded lossy compressor designed by analyzing the significant bit-planes that need to be kept in terms of the user-specified error bound and leveraging the DEFLATE algorithm to reduce the data size. Since Bit Grooming code was tightly coupled with the NetCDF Operators (NCO) software stack [53], it could be used only to compress the datasets stored in the NetCDF format. We carefully extracted the Bit Grooming code into a standalone version (called Bit_GroomingZ), such that it can be used on a generic binary-format data file. The Bit Grooming code and its standalone version can be found in the NCO website [53] and Bit GroomingZ website [44], respectively.
- Digit Rounding [23]. Digit Rounding is also an error-bounded lossy compressor designed by calculating/manipulating the number of significant bits according to the user-specified error bounds. Digit Rounding was originally developed with the HDF5 library because it needs to compress the truncated datasets by the HDF5 gzip filter, so it could be applied only on HDF5 data files. Similar to Bit Grooming, we also extracted a standalone version (called digit_roundingZ) for Digit Rounding, especially for our characterization work. The original Digital Rounding code and our standalone version can be downloaded from [54] and [45], respectively.
- TTHRESH [43]. TTHRESH is an error-controlled lossy compressor designed based on higher-order singular vector decomposition (HOSVD)—a generalization of the SVD to three and more dimensions. Because of the particularly efficient data decorrelation step, HOSVD, TTHRESH can obtain a much higher compression ratios than can other non-SVD compressors such as SZ and ZFP, but it may suffer from substantially lower throughput (e.g., 1 order of magnitude slower). Thus, TTHRESH is suitable mainly for the offline use case that requires extremely high compression ratios yet does not require the compression/decompression speed. The TTHRESH code is downloadable from [55].
- SZ3 [17]. SZ3 is an error-bounded lossy compressor that can get a significantly improved compression ratio and quality over SZ2 with negligible or slightly higher execution overhead. The key reason for SZ3's higher quality is that it adopts a more efficient dynamic cubic-spline interpolation-based prediction method in comparison with the traditional Lorenzo+Linear-regression predictor used in SZ2. In addition, SZ3 [56] is a composable framework allowing users to customize their own compression pipeline to adapt to various datasets and use cases. SZ3 code can be downloaded from [57].
- FPZIP [6]. FPZIP is a prediction-based compressor supporting both lossless and lossy compression of 2D and 3D floating-point data arrays. It includes four steps: Lorenzo prediction, mapping to integer, computing residuals, and fast entropy encoding. The FPZIP code can be downloaded from [58].
- NDZip [11] NDZip is a lossless compressor designed for compressing the multidimensional univariate floating-point datasets. In particular, ndzip optimizes the parallel compression performance by leveraging a data-parallel Integer Lorenzo transform for small hypercubes and a hardware-friendly residual coding scheme. The NDZip code can be found here [59].
- Zstandard (Zstd) [5]. Zstd is an outstanding lossless compressor that has been widely used in different tools, libraries, or environments. According to prior studies [60], Zstd generally obtains compression ratios comparable to those of other lossless compressors such as zlib [61] and gzip [4], while Zstd is generally much faster than them (about 2–3$\times$ in throughtput). Zstd can be found in [5].