

Toward a Multi-method Approach: Lossy Data Compression for Climate Simulation Data

Allison H. Baker, Haiying Xu, Dorit M. Hammerling, Shaomeng Li, and
John P. Clyne

The National Center for Atmospheric Research, Boulder, CO 80305, USA
{abaker, haiyingx, dorith, shaomeng, clyne}@ucar.edu

Abstract. Earth System Model (ESM) simulations are increasingly constrained by the amount of data that they generate rather than by computational resources. The use of lossy data compression on model output can reduce storage costs and data transmission overheads, but care must be taken to ensure that science results are not impacted. Choosing appropriate compression algorithms and parameters is not trivial given the diversity of data produced by ESMs and requires an understanding of both the attributes of the data and the properties of the chosen compression methods. Here we discuss the properties of two distinct approaches for lossy compression in the context of a well-known ESM, demonstrating the different strengths of each, to motivate the development of an automated multi-method approach for compression of climate model output.

1 Introduction

The Community Earth System Model (CESM) [9] is a popular earth system model code based at the National Center for Atmospheric Research (NCAR). A high-resolution CESM simulation can easily generate over a terabyte of data per compute day (e.g., [22]), outputting time slices of data for hundreds of variables at hourly, daily, and monthly sampling rates. In fact, the raw data requirements for CESM for the current Coupled Model Comparison Project (Phase 6) [19] are expected to exceed 10 petabytes [20]. The massive data volumes generated by CESM strain NCAR’s resources and motivated the work in [2], a first step in advocating for the use of lossy data compression on CESM output. In [2], errors in reconstructed CESM data (data that had undergone compression) resulting from multiple lossy compression methods were evaluated primarily in the context of an ensemble of simulations. The idea was that the effects of lossy compression on the original climate simulation should not, at a minimum, be statistically distinguishable from the natural variability of the climate system. Preliminary results indicated that this requirement could be met with a respectable compression rate with the *fpzip* compressor [18].

A more recent study in [1] applied *fpzip* lossy compression to a subset of the data from the CESM Large Ensemble (CESM-LE) Community Project [12], which was made available to climate researchers to examine features of the data relevant to their interests (e.g., extremes, variability patterns, mean climate characteristics). The results from several of these studies are discussed in [1], and

the authors conclude that while it is possible to detect compression effects in the data in some features, the effects are often unimportant or disappear in post-processing analyses. For this study, each CESM output variable was assessed individually to maximize compression such that the reconstructed data passed the ensemble-based quality metrics in [2]. This costly “brute force” approach required the generation of multiple ensembles and exhaustive testing of the compression algorithm’s parameter space.

Our goal is to simplify the process of determining appropriate compression for a given CESM dataset that both maximizes data reduction and preserves the scientific value of the data. Therefore, we must be able to detect problematic compression artifacts with metrics that do not require ensemble data. Further, because a single compression algorithm cannot obtain the best compression rate (and quality) on every CESM variable, we explore applying multiple types of compression methods to a CESM dataset. Once a particular method has been matched to a variable, then the amount of compression (i.e., parameters) must be chosen inexpensively as well. In this work, we progress toward identifying which type of compression method to use based on a variable’s characteristics and determining the strengths and weaknesses of different types of lossy compression algorithms in the context of CESM output. We also demonstrate the potential of a multi-method compression approach for CESM.

2 Challenges

Our ultimate goal is to develop an automated tool to integrate lossy compression into the CESM workflow. Given a CESM dataset, this tool must be able to *efficiently* determine which compression algorithm(s) to apply and evaluate the impact of the information loss. These two capabilities are particularly challenging for CESM simulation output due to the diversity of variables, and a variable’s characteristics determine how effectively it can be compressed. CESM variables may be smooth, constant, or contain abrupt changes. Variables may have large ranges of data values, artificial “fill” values, unpredictable missing values, or large numbers of zero values. Further, the same variable field may “look” different at different spatial and temporal resolutions.

The work in [2] customizes how aggressively each CESM variable is compressed by adjusting algorithm-specific parameters that control the amount of compression. However, here we further suggest using different compression algorithms on different variables. The benefit of a multi-method approach is that, for example, a compression method that does poorly on data with sharp boundaries but extremely well on smooth data would not be excluded from consideration, but simply applied only to smooth variables. The challenge of a multi-method approach is that determining the rules to automate the process of matching variables to appropriate lossy compression algorithms requires a thorough understanding of the features of each variable, the strengths and weaknesses of each compression method, and the evaluation metrics in the context of CESM

data. Further, once a lossy compression method has been chosen, method-specific parameters must be optimized as well.

Determining appropriate metrics to evaluate the impact of information loss is also challenging due to the diversity of data (e.g., smooth data may be easier to compress, but perhaps there is less tolerance for error). However, a second issue stems from not knowing in advance how a large publicly-available CESM dataset will be analyzed. Indeed, if we know how data will be analyzed, compression can be tailored to well preserve features of interest (e.g., top of the atmosphere surface radiation balance) in the reconstructed data. Finally, computational cost is a consideration. While the ensemble-based quality metrics that leverage the climate model systems’s variability were needed to establish the feasibility of applying lossy compression to CESM output in [2], an ensemble-based approach is expensive. On the other hand, simple metrics such as the root mean squared error (RMSE) or peak signal-to-noise ratio (PSNR) are insufficient for detecting features potentially relevant to climate scientists.

3 Lossy Compression Algorithms

Lossy compression algorithms for general floating-point scientific data have received attention recently (e.g., [3], [4], [6], [11], [14], [15], [16], [17], [18], [21]) due to their ability to compress much more aggressively than lossless approaches. A few studies have focused on applying lossy algorithms to climate simulation data in particular (e.g., [2] [25], [8]). Compression schemes can be described in terms of their modeling and encoding phases, and available compression algorithms differ in how these phases are executed. Predictive schemes and transform methods are common choices for the modeling phase in lossy compression algorithms. We focus on a representative algorithm of each type to explore how the two different types of compression algorithms differ in the context of CESM data. While not discussed in this work, note that algorithm performance and ease-of-use are important and desirable lossy compression method properties for CESM data are discussed in [2].

The *fpzip* compressor [18] models the floating-point numbers via predictive coding; as the data are traversed, values are predicted based on data already visited. The idea behind a predictive method is that the residual between the actual and predicted floating-point value is smaller than the original value and, therefore, can be encoded with fewer bits. The *fpzip* compressor [18] may be lossless or lossy depending on whether all bits are retained (or a number of least significant bits are truncated) before the floating-point values are converted to integers. Integer residual values are then encoded by a fast entropy encoder. In lossy mode, because discarding of bits effectively rounds toward zero, some introduction of bias is possible [15].

A tranform compression method aims to model the original data with a relatively small number of basis coefficients (i.e., those with the largest magnitudes) and then encode those coefficients. The compressor that we refer to as *SPECK* uses a discrete wavelet basis and encodes with the set partitioned embedded

block coder algorithm [10]. In this research we adopted the *SPECK* implementation from QccPack [7], with the CDF 9/7 wavelet transformation [5]. For 2D variables, a 2D transform was applied to each horizontal slice; for 3D variables, an additional 1D transform was applied along the Z axis. Normally a transform method cannot support a lossless option due to floating-point inaccuracies associated with the transform, and *SPECK* is no exception. The amount of compression with *SPECK* is controlled by specifying a target bit per voxel (i.e., a fixed rate). For example, for single precision data (32-bits), specifying a bit per voxel of 8 would yield a compression ratio (*CR*) of approximately 0.25, where (*CR*) is defined as the ratio of the size of the compressed file to that of the original file. Other examples of transform approaches include JPEG2000 (e.g. [25]) and *zfp* [17], the latter of which targets numerical simulation data.

4 Metrics

Three of the four metrics in [2] for evaluating information loss in CESM due to lossy compression are ensemble-based. We move away from ensemble-based metrics in this study largely due to cost considerations, though a second hindrance to automation with ensemble-based metrics is that variable properties across the ensemble cannot be known in advance to determine allowable error. For example, if a variable is constant across the ensemble, then there may be no tolerance for any error no matter how small. Therefore, for our comparison of the two lossy approaches here, we use the three metrics described next, as well as the Pearson correlation coefficient as in [2]. We do not claim that the following metrics (and tolerances) are comprehensive (notably absent are multivariate metrics and temporal considerations), but they reflect our evolution in terms of suitable metrics that measure different aspects of the data and illustrate the differences between the two lossy approaches that we compare in this work. Indeed, determining comprehensive and efficient metrics is a subject of on-going long-term research.

We consider a single temporal step for our analysis and denote the original spatial dataset X as $X = \{x_1, x_2, \dots, x_N\}$, with x_i a scalar and i the spatial index, and the reconstructed dataset $\tilde{X}$ by $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$. The range of X is denoted by R_X . The normalized maximum pointwise error (e_{nmax}) is the maximum norm, normalized by R_X , and the normalized RMSE ($nrmse$) is the RMSE between the original and reconstructed data, normalized by R_X .

Pearson correlation coefficient: The Pearson correlation coefficient (PCC) indicates the strength of the linear relationship between the original and reconstructed data, and a value of one indicates a perfect (positive) correlation. Lossy compression should not degrade this relationship, and as such, we require that $PCC \geq .99999$ as the acceptance threshold for this test [2]. The PCC is useful as it is sensitive to outliers in the data (but is invariant to mean shifts).

Kolmogorov-Smirnov Test: The two-sample Kolmogorov-Smirnov (KS) test detects a potential shift in the distribution. The KS test is a nonparametric

hypothesis test for evaluating whether two datasets are drawn from the same probability distribution (the null hypothesis) and is based on the supremum distance between two empirical cumulative density functions (CDFs). We use the SciPy statistical functions package two-sample KS test at the 5% confidence level. Note that the KS test benefits from large sample size, which we have here, making it more accurate/sensitive. This test should detect smoothing, skew, or other distribution-changing features in the reconstructed data. For example, if many points with the same value in the distribution are systematically under- or overestimated (by even a tiny amount), this test will fail even if the discrepancy is undetectable in the sample mean and standard deviation.

Spatial relative error: While checking the maximum norm of the error gives a minimum guarantee of precision, the error may only be large at one single point. On the other hand, a measure of average error (e.g., RMSE) can hide an error at a single or a few point(s). To better describe the spatial extent of the error, we determine the percentage of spatial grid locations at which the relative error is greater than a specified tolerance δ . In particular, for each variable X at each grid point, we calculate the relative error: $re_{x_i} = (x_i - \tilde{x}_i)/x_i$ (if $x_i = 0$, then we calculate the absolute error). If percentage of grid points with $re_{x_i} > \delta$ exceeds 5%, this test will fail. We are compressing single-precision (32-bit) data in CESM, and for our experiments we use $\delta = 1e^{-4}$.

Structural similarity index: The structural similarity Index (SSIM) was developed to measure the perceived change in structural information between two images, as the commonly used RMSE is typically not well suited to such a task [23]. Data visualization is a key component in many climate simulation post-processing analyses, as evidenced by the popularity of the Atmosphere Working Group Diagnostics Package (AMWG-DP). Clearly, visual evidence of information loss due to compression in post-processing image analysis would be problematic, particularly if scientific conclusions are affected. Computing the SSIM for 2D slices of the original and reconstructed data provides an indication as to whether the difference is noticeable. An SSIM score of one indicates that two images are identical, while lower scores indicate some degree of difference. Most threshold values for minimum allowable SSIM for compression in the medical imaging research field, which focuses on “diagnostically lossless” [13], range from .95 to .99. While an appropriate SSIM threshold is clearly application dependent (and requires further research for CESM), we use .98 in this study as it is commonly cited as the level of visual indistinguishability (e.g., [24]).

5 Multi-method Comparison

We limit our investigation to output from the atmospheric model component of CESM, the Community Atmosphere Model (CAM), evaluating the same data as in [2], which were annual averages obtained from the 1.1 release version of CESM, using a spectral element (SE) dynamical core on a cubed-sphere

Table 1. Representative CESM variable characteristics

variable name	description	dim.	x_{min}	x_{max}	% zeros
H2O2	H ₂ O ₂ concentration	3D	9.44e-13	3.55e-9	0.1
FSNTC	Clearsky net solar flux (top of model)	2D	4.57e1	3.80e2	0.0
TS	surface temperature	2D	2.15e2	3.04e2	0.0
TAUY	zonal surface stress	2D	-2.66e-1	2.44e-1	0.0
CLOUD	Cloud fraction	3D	0.0	8.95e-1	22.3
PRECSC	Convective snow rate	2D	0.0	6.80e-9	75.8
TOT_ICLD_VISTAU	Total in-cloud visible sw optical depth	3D	0.0	6.75e1	27.3
PRECCDZM	Convective precipitation rate (ZM deep)	2D	0.0	2.39e-7	4.6
OMEGAT	Vertical heat flux	3D	-2.74e2	2.01e2	0.0
FLNS	Net longwave flux at surface	2D	1.14e1	1.50e2	0.0
VQ	Meridional water transport	3D	-9.21e-2	1.07e-1	0.0
NUMLIQ	Grid box averaged cloud liquid num.	3D	1.00e-12	1.10e8	43.1
WSUB	Diagnostic sub-grid vertical velocity	3D	2.00e-1	1.30e0	0.0

Table 2. A list of the lowest *CR* variants of *SPECK* and *fpzip* for each representative CESM variable

variable name	<i>SPECK</i>				<i>fpzip</i>				DWT→IDWT max. abs. error
	variant	e_{nmax}	$nrmse$	<i>CR</i>	variant	e_{nmax}	$nrmse$	<i>CR</i>	
H2O2	speck_2	2.47e-4	2.47e-5	.06	fpzip_20	2.56e-4	2.05e-5	.23	0.0
FSNTC	speck_8	1.75e-4	2.08e-5	.26	fpzip_24	2.33e-5	1.18e-5	.36	0.0
TS	speck_4	1.46e-3	1.95e-4	.13	fpzip_24	8.71e-5	4.95e-5	.28	0.0
TAUY	speck_12	8.04e-6	1.24e-6	.38	fpzip_24	1.41e-5	7.92e-7	.54	0.0
CLOUD	—	—	—	—	fpzip_24	1.70e-5	2.42e-6	.36	8.88e-16
PRECSC	—	—	—	—	fpzip_16	4.01e-3	1.97e-4	.12	2.53e-24
TOT_ICLD_VISTAU	—	—	—	—	fpzip_24	2.68e-5	5.84e-7	.38	8.88e-15
PRECCDZM	speck_24	7.44e-9	1.61e-9	.77	fpzip_16	3.89e-3	4.16e-4	.24	5.29e-23
OMEGAT	speck_16	2.24e-8	3.11e-9	.51	fpzip_24	1.09e-5	2.04e-7	.52	0.0
FLNS	speck_12	1.38e-5	2.72e-6	.38	fpzip_24	2.81e-5	5.19e-6	.42	0.0
VQ	speck_16	3.50e-9	3.82e-10	.51	fpzip_24	9.53e-6	6.52e-7	.48	0.0
NUMLIQ	—	—	—	—	fpzip_32	0.0	0.0	.46	5.96e-8
WSUB	—	—	—	—	fpzip_32	0.0	0.0	.43	0.0

1-degree global grid (48,602 horizontal grid-points and 30 vertical levels). CESM data are written to single-precision (truncated from double-precision), and we use all 198 default output variables, 101 of which are two-dimensional (2D) and 97 three-dimensional (3D). We define *fpzip*_Y as *fpzip* where *Y* indicates the number of bits to retain before quantization, and we evaluate with $Y = \{8, 12, 16, 20, 24, 28, 32\}$. Therefore, *fpzip*₈ is the most aggressive and *fpzip*₃₂ is lossless. We define *speck*_M as *SPECK* where *M* indicates the bit target rate and evaluate with $M = \{1, 2, 4, 8, 12, 16, 24, 32\}$. Therefore, *speck*₁ is the most aggressive and *speck*₃₂ is the least (closest to lossless). Note that because the CAM SE data is output as a 1D array for each horizontal level (space-filling curve ordering), we reorder the CAM data to be spatially coherent data before applying the transform method. In particular, the original 48,602 horizontal grid points were mapped to the six cubed-sphere faces (91x91x1), and *SPECK* is applied to each face independently (3D variables have 91x91x30 input arrays). *SPECK* also takes two additional parameters related to the wavelet transform levels; given that the wavelet transform kernel size is 9, we set XY-level to 4 ($\log_2(91/9) + 1$) and Z-level to 2 ($\log_2(30/9) + 1$).

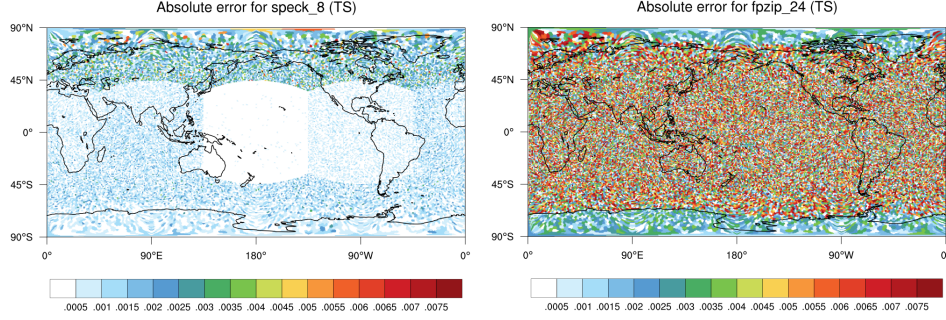


Fig. 1. Absolute error between the original and reconstructed data with speck_8 (left) and fpzip_24 (right) for variable TS. Both methods shown attain a similar CR .

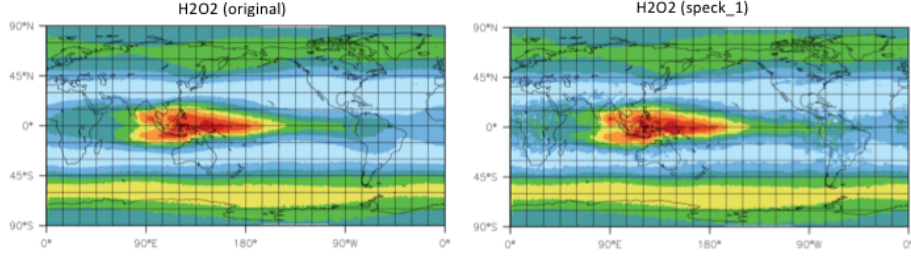


Fig. 2. Variable H2O2 (level 7) in original data (left) and after speck_1 compression (right). The SSIM index for the images is below the .98 threshold. Colorbars for these two plots have been omitted as they are identical and do not contribute information.

5.1 Detailed Investigation of Representative Variables

We examine a subset of the variables in detail (Table 1). For each variable, we determine the most aggressive (i.e., lowest CR) variant of *SPECK* and *fpzip* that pass the four tests described in Section 4 (see Table 2). Comparing *SPECK* to *fpzip* is complicated by the fact that *SPECK* uses a fixed-rate specification and *fpzip* does not. Values for e_{nmax} and $nrmse$ are listed in Table 2, but are not used as selection metrics, and the rightmost column is discussed in Sect. 6.

The top section in Table 2 lists four variables, H2O2, FSNTC, TS, and TAU, which have a lower CR with *SPECK* than with *fpzip*. These variables all have either very few or no zeroes. Each variable is also quite smooth (intuitive for surface temperature, TS). For H2O2, while the range is a bit larger overall, the range within each horizontal level is smaller. Note that *fpzip* does not do poorly on these four variables, but *SPECK* compresses more aggressively. Figure 1 illustrates the difference in the two methods via the absolute error for TS with speck_8 (left) and fpzip_24 (right), which achieve a similar CR of .26 and .28, respectively. The error with *SPECK* is uniformly smaller at this same compression ratio, which makes sense given that more aggressive compression via speck_4 is acceptable (Table 2). Note that the cubed-sphere faces are evident in Figure 1.

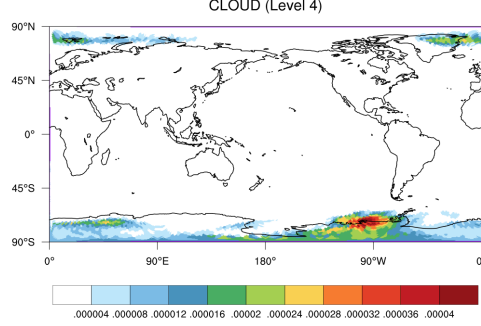


Fig. 3. Variable CLOUD (level 4) in the original data.

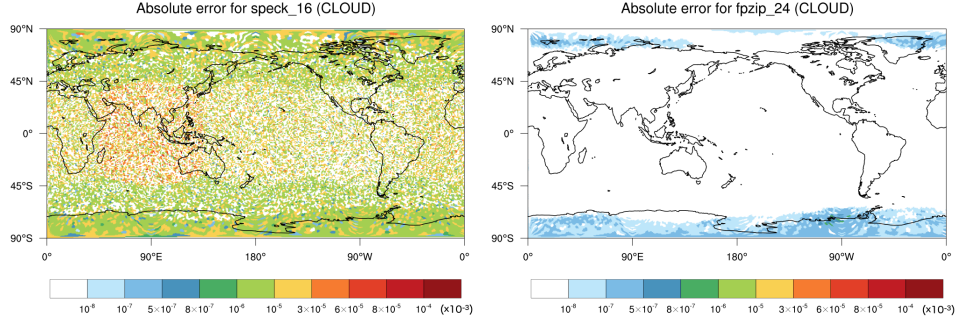


Fig. 4. Absolute error between the original and reconstructed data with speck_16 (left) and fpzip_24 (right) for variable CLOUD (level 4).

For FNSTC, TS, and TAU_Y, more aggressive variants of the two compressors fail the spatial relative error test. For H202, though, the more aggressive variant of *SPECK* fails the SSIM test, which can be visually confirmed by the noticeable difference in Figure 2 along the contour between blue and light blue.

The second section in Table 2 lists four variables that achieve a lower *CR* with *fpzip*. The first three of these variables (CLOUD, PRECSC, TOT_ICLD_VISTAU) contain sizable percentages of zeros, which *SPECK* typically does not exactly preserve. Even the least aggressive *SPECK* variant, speck_32 (which does not reduce the file size) cannot pass the KS test, which detects the shift in distribution caused by reconstructing zero values in the original data as very small values (positive and negative). These fields also contain more abrupt jumps in the data, which are not favorable to a transform method. For example, the 3D variable CLOUD contains many zero values, very small numbers, and large ranges. Some levels have ranges of eight orders of magnitude, half of the levels have no zeros, the surface level (level 0) has all zeros, and level 4 (Fig. 3) is 95% zeros with a range of five orders of magnitude. Figure 4 shows the absolute error for CLOUD on level 4 with fpzip_24 (*CR* = .36 and passes all metrics) and speck_16 (*CR* = .51 and does not pass), and it is clear that this variable is challenging for a transform method. The fourth variable, PRECCDZM, has fewer zeros but a large range, and can be compressed with *SPECK*, but not as aggressively as

Table 3. All variables (198) categorized by method with lowest CR .

Category	Number of variables	<i>SPECK</i> averages			<i>fpzip</i> averages		
		e_{nmax}	$nrmse$	CR	e_{nmax}	$nrmse$	CR
<i>SPECK</i> better	87	4.23e-4	5.62e-5	.16	1.35e-4	2.25e-5	.30
<i>fpzip</i> better	12	4.26e-7	5.91e-8	.45	7.76e-4	1.09e-4	.37
<i>fpzip</i> / <i>SPECK</i> similar	24	5.31e-6	9.58e-7	.32	2.59e-4	1.18e-5	.32
<i>fpzip</i> (<i>SPECK</i> fails)	63	—	—	—	1.38e-3	6.71e-5	.29
lossless (<i>fpzip</i> .32)	12	—	—	—	0.0	0.0	.50

fpzip. More aggressive variants of *SPECK* and *fpzip* on PRECCDZM fail the KS test and correlation coefficient test, respectively.

The third section in Table 2 contains variables for which both approaches achieve a similar CR . These variables all fail the spatial relative error test if compressed more aggressively with either method. Note that while the CR is similar, both $nrmse$ and e_{nmax} are notably smaller with *SPECK* for variables OMEGAT and VQ. Finally, the bottom of Table 2 lists two variables for which only lossless compression can pass the metrics. Lossy compression of NUMLIQ (which has both a huge range and a high percentage of zeros) resulted in KS and SSIM test failures for both *SPECK* and *fpzip*. In contrast, WSUB does not have a large range, but it does have a large number of non-zero constants (29% of the data values are equal to 0.2). Neither lossy approach preserved this prevalent constant, resulting in KS test failures indicating a shifted distribution. We note that *fpzip* lossless compression is slightly better than NetCDF4 lossless compression (essentially *gzip*), which results in CR of .48 for both variables. Lossless compression achieves a respectable CR on these two variables due to their large numbers of constant values.

5.2 Full Set of Variables

Now we look at all 198 variables and divide them into five categories according to which lossy compression approach passes the Sect. 4 metrics with a lower CR (Table 3). We find that 87 variables do better with *SPECK* than *fpzip*, and the average CR and error measurements for that subset of variables is given in the first row of Table 3. The average CR is approximately a factor of two smaller with *SPECK* for these variables. The second and third row of Table 3 show that *fpzip* outperforms *SPECK* on only 12 variables and that they perform similarly in terms of CR on 24 variables. However, in both of these cases the *SPECK* average errors are a couple of orders of magnitude smaller, indicating that traditional error metrics may be insufficient for identifying certain problematic features. Finally, for the remaining 75 variables (rows 4 and 5), *SPECK* is not an option as it fails the metrics even with its least aggressive variant. Of these, twelve of the variables cannot pass with *fpzip* in lossy mode either and require the lossless variant of *fpzip* (*fpzip*.32).

6 Characterizing Data

The two lossy compression approaches that we evaluated have different strengths. Unsurprisingly, transform methods are challenged by CESM datasets with abrupt

changes and large ranges of values. They can also be problematic when zeros or constants must be preserved for post-processing analyses. However, for smooth CESM data, our results indicate that transform methods can compress more aggressively and more accurately than a predictive method like *fpzip*. On the other hand, *fpzip*'s general utility and effectiveness is valuable; it can be applied successfully to every CESM variable and its lossless option is a necessity.

An automated tool for a multi-method approach must be able to assess easily measurable properties of a variable's data to determine which type of compression approach will be most effective. Our experimental results with *SPECK* and *fpzip* indicate that if the least aggressive variant of *SPECK* (*speck_32*) is able to pass the metrics, then the "best" (i.e., lowest *CR* that passes metrics) *SPECK* variant is likely to be as good or better than that of the best *fpzip* variant. When *speck_32* fails on CESM variables, the reason is a KS test failure. CESM variables with many zero values (or many constants in general) are particularly problematic as zeros are frequently reconstructed as small (positive or negative) values, causing the underlying distribution to shift and the KS test to fail.

In an attempt to predict *SPECK* effectiveness, we looked at a variety of variable properties (range, gradient, number of zeros, etc.) and investigated the cause of the *speck_32* failures. We found that the key to the failures was *SPECK*'s CDF 9/7 wavelet transformation (DWT), which can suffer from floating-point computation induced-error. We refer to the process of applying DWT followed immediately by an inverse DWT (IDWT) as DWT→IDWT, which is lossless in infinite precision. In practice, DWT→IDWT was lossless for some CESM variables and lossy for others, as indicated by the maximum absolute error between the original data and the data after DWT→IDWT in the rightmost column in Table 2 (e.g., a zero value indicates lossless). For all but 4 of the 198 total variables, we found that variables with non-zero absolute errors after DWT→IDWT indicate that *SPECK* is not appropriate for these variables. Note that WSUB in Table 2 is an exception as it requires lossless despite its zero DWT→IDWT error (and is a target of future study). Therefore, applying a standalone DWT→IDWT test (e.g., via QccPack) is promising method for automating the decision as to whether to use a wavelet transform method such as *SPECK*.

7 Concluding Remarks

Transform methods are enticing due to their ability to compress both aggressively and accurately. Unfortunately, *SPECK* was unsuitable for 38% of the variables in our test CESM dataset (though issues with preserving zeros or other constants could conceivably be addressed by a pre-processing step). However, the 2x improvement of *SPECK* over *fpzip* indicates that an automated multi-method approach is worth pursuing. Indeed, large climate simulations commonly produce data volumes measured in hundreds of terabytes or even petabytes, and even a modest reduction in *CR* is quite significant in terms of data reduction and impact on storage costs. Future work includes more research on appropriate metrics, as the selection of the most appropriate type of compression scheme must

now be followed by a specification of the parameters that control the amount of compression. Further, we note that we chose rather conservative tolerances for our metrics that, if relaxed, would likely be more favorable to a transform method.

References

1. Baker, A.H., Hammerling, D.M., Mickleson, S.A., Xu, H., Stolpe, M.B., Naveau, P., Sanderson, B., Ebert-Uphoff, I., Samarasinghe, S., De Simone, F., Carbone, F., Gencarelli, C.N., Dennis, J.M., Kay, J.E., Lindstrom, P.: Evaluating lossy data compression on climate simulation data within a large ensemble. *Geoscientific Model Development* 9(12), 4381–4403 (2016), <http://www.geosci-model-dev.net/9/4381/2016/>
2. Baker, A., Xu, H., Dennis, J., Levy, M., Nychka, D., Mickelson, S., Edwards, J., Vertenstein, M., Wegener, A.: 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*. pp. 203–214. HPDC '14 (2014)
3. Bicer, T., Yin, J., Chiu, D., Agrawal, G., Schuchardt, K.: Integrating online compression to accelerate large-scale data analytics applications. *Parallel and Distributed Processing Symposium, International* pp. 1205–1216 (2013)
4. Burtscher, M., Ratanaworabhan, P.: FPC: A high-speed compressor for double-precision floating-point data. In: *IEEE Transactions on Computers*. vol. 58, pp. 18–31 (January 2009)
5. Cohen, A., Daubechies, I., Feauveau, J.C.: Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.* 45, 485–560 (1992)
6. Di, S., Cappello, F.: Fast error-bounded lossy HPC data compression with SZ. In: *2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2016, Chicago, IL, USA, May 23-27, 2016*. pp. 730–739 (2016), <http://dx.doi.org/10.1109/IPDPS.2016.11>
7. Fowler, J.E.: Qccpack: An open-source software library for quantization, compression, and coding. In: *International Symposium on Optical Science and Technology*. pp. 294–301. International Society for Optics and Photonics (2000)
8. Hübbe, N., Wegener, A., Kunkel, J.M., Ling, Y., Ludwig, T.: Evaluating lossy compression on climate data. In: *Proceedings of the International Supercomputing Conference (ISC '13)*. pp. 343–356 (2013)
9. Hurrell, J., Holland, M., Gent, P., Ghan, S., Kay, J., Kushner, P., Lamarque, J.F., Large, W., Lawrence, D., Lindsay, K., Lipscomb, W., Long, M., Mahowald, N., Marsh, D., Neale, R., Rasch, P., Vavrus, S., Vertenstein, M., Bader, D., Collins, W., Hack, J., Kiehl, J., Marshall, S.: The Community Earth System Model: a framework for collaborative research. *Bulletin of the American Meteorological Society* 94, 1339–1360 (2013)
10. Islam, A., Pearlman, W.A.: Embedded and efficient low-complexity hierarchical image coder. In: *Electronic Imaging'99*. pp. 294–305. International Society for Optics and Photonics (1998)
11. Iverson, J., Kamath, C., Karypis, G.: Fast and effective lossy compression algorithms for scientific datasets. In: *Proceedings of the 18th International Conference on Parallel Processing*. pp. 843–856. Euro-Par'12, Berlin, Heidelberg (2012)

12. Kay, J., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., Arblaster, J., Bates, S., Danabasoglu, G., Edwards, J., Holland, M., Kushner, P., Lamarque, J.F., Lawrence, D., Lindsay, K., Middleton, A., Munoz, E., Neale, R., Oleson, K., Polvani, L., Vertenstein, M.: 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* 96 (2015)
13. Kowalik-Urbaniak, I., Brunet, D., Wang, J., Koff, D., Smolarski-Koff, N., Vrscaj, E.R., Wallace, B., Wang, Z.: The quest for 'diagnostically lossless' medical image compression: a comparative study of objective quality metrics for compressed medical images (2014)
14. Lakshminarasimhan, S., Shah, N., Ethier, S., Klasky, S., Latham, R., Ross, R., Samatova, N.F.: Compressing the incompressible with ISABELA: In-situ reduction of spatio-temporal data. In: *Proceedings of the 17th International Conference on Parallel Processing. Euro-Par'11, Bordeaux, France (Aug 29 - Sep 2 2011)*
15. Laney, D., Langer, S., Weber, C., Lindstrom, P., Wegener, A.: Assessing the effects of data compression in simulations using physically motivated metrics. In: *Supercomputing 2013 (SC'13)* (2013)
16. Li, S., Gruchalla, K., Potter, K., Clyne, J., Childs, H.: Evaluating the Efficacy of Wavelet Configurations on Turbulent-Flow Data. In: *Proceedings of IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 81–89. Chicago, IL (Oct 2015)
17. Lindstrom, P.: Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics* 20(12), 2674–2683 (2014)
18. Lindstrom, P., Isenburg, M.: Fast and efficient compression of floating-point data. *IEEE Transactions on Visualization and Computer Graphics* 12, 1245–1250 (2006)
19. Meehl, G., Moss, R., Taylor, K., Eyring, V., Stouffer, R., Bony, S., Stevens, B.: Climate model intercomparisons: Preparing for the next phase. *Eos, Transactions American Geophysical Union* 95(9), 77–78 (2014)
20. Paul, K., Mickelson, S., Xu, H., Dennis, J.M., Brown, D.: Light-weight parallel Python tools for earth system modeling workflows. In: *IEEE International Conference on Big Data*. pp. 1985–1994 (October 2015)
21. Sasaki, N., Sato, K., Endo, T., Matsuoka, S.: Exploration of lossy compression for application-level checkpoint/restart. In: *Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium*. pp. 914–922. IPDPS '15 (2015)
22. Small, R.J., Bacmeister, J., Bailey, D., Baker, A., Bishop, S., Bryan, F., Caron, J., Dennis, J., Gent, P., Hsu, H.m., Jochum, M., Lawrence, D., Muoz, E., diNezio, P., Scheitlin, T., Tomas, R., Tribbia, J., Tseng, Y.h., Vertenstein, M.: A new synoptic scale resolving global climate simulation using the community earth system model. *Journal of Advances in Modeling Earth Systems* 6(4), 1065–1094 (2014)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4), 600–612 (2004)
24. Wegener, A.: Compression of medical sensor data. *IEEE Signal Processing Magazine* 27(4), 125–130 (July 2010)
25. Woodring, J., Mniszewski, S.M., Brislawn, C.M., DeMarle, D.E., Ahrens, J.P.: Revisiting wavelet compression for large-scale climate data using JPEG2000 and ensuring data precision. In: Rogers, D., Silva, C.T. (eds.) *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 31–38. IEEE (2011)