

Error bounded compression for weather and climate applications

Langwen Huang¹, Luigi Fusco^{1, 4}, Florian Scheidl¹, Jan Zibell², Michael Armand Sprenger², Sebastian Schemm³, and Torsten Hoefler¹

¹Scalable Parallel Computing Laboratory, Department of Computer Science, ETH Zurich, Switzerland

²Institute for Atmospheric and Climate Science, Department of Environmental Systems Science, ETH Zurich, Switzerland

³Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences, University of Cambridge, Cambridge, UK

⁴Microsoft Ibérica, Madrid, Spain

Correspondence: Langwen Huang (langwen.huang@inf.ethz.ch)

Abstract. As the resolution of weather and climate simulations increases, the amount of data produced is growing rapidly from hundreds of terabytes to tens of petabytes. The huge size becomes a limiting factor for broader adoption, and its fast growth rate will soon exhaust all the available storage devices. To address these issues, we present EBCC (Error Bounded Climate-data Compressor). It follows a two-layer compression approach: a base compression layer using JPEG2000 to capture the bulk of the data with a high compression ratio, and a residual compression layer using wavelet transform and SPIHT (Set Partitioning In Hierarchical Trees) encoding to efficiently eliminate long-tail extreme errors introduced by the base compression layer. It incorporates a feedback rate-control mechanism for both layers that adjusts compression ratios to achieve the specified maximum error target. We evaluate EBCC alongside other established error-bounded compression methods on several benchmarks related to weather and climate science. The benchmarks include error statistics, a case study on primitive and derived variables near a tropical cyclone, evaluation of the closure of the global energy budget, and a Lagrangian air parcel trajectory simulation. This is the first time that trajectory simulation is used to benchmark compression methods. According to the histogram of the compression errors, our method concentrates most errors near zero, while other error-bounded methods tend to distribute errors uniformly within the error bound. EBCC outperforms other methods in the benchmarks at relative error targets ranging from 0.1% to 10% and achieves compression ratios from $15\times$ to more than $300\times$, respectively. In the energy budget closure and Lagrangian trajectory benchmarks, it can achieve more than $100\times$ compression while keeping errors within natural variability derived from ERA5 uncertainty members. This verifies the effectiveness of EBCC in creating heavily compressed weather and climate datasets suitable for downstream applications. The source code of EBCC is available in github.com/spcl/EBCC.

1 Introduction

Numerical weather and climate simulations produce a huge amount of data ranging from hundreds of terabytes to tens of petabytes (Bauer et al., 2021; Acosta et al., 2024; Govett et al., 2024). Analyzing these data requires significant storage and computing resources, which are only available to a few researchers. Moreover, the volume of data is growing rapidly as higher-resolution simulations are required to address the challenges posed by climate change, particularly extreme weather events. As the growth rate of the simulation data eclipses that of storage capacity, a shortage of storage space becomes inevitable (Hoefer et al., 2023; ECMWF, 2023). As a result, there is an urgent need for data compression to store high-resolution simulation data while ideally having fast data access.

The exploration of data compression for weather and climate data starts with the lossless compression method (Rice, 1996; Lindstrom, 2017; Huang et al., 2016) where data can be exactly restored after compression and decompression processes. However, they can hardly reach a high compression ratio above $6\times$ for common atmospheric variables (Mummadisetty et al., 2015; Huang et al., 2016) because they attempt to store the random noise component of the data, which is impossible to compress according to Shannon’s source coding theorem (Shannon, 1948). In contrast, lossy compression allows for an approximation of the original data in exchange for significantly higher compression ratios. The simplest form of lossy compression is quantization, which represents a block of data with linearly scaled integers with lower number of bits than the original format. It is often used in conjunction with lossless compression methods to achieve higher compression ratios (Silver and Zender, 2017). Due to its simplicity, it is widely supported in modern scientific data formats such as NetCDF-4 (Unidata, 2025), HDF5 (Folk et al., 1999), GRIB2 (WMO, 2025) and Zarr (Miles et al., 2025). Bit grooming (Zender, 2016) takes another approach of quantization by “shaving” the least significant bits of floating-point values to zeros. Bit rounding (Klöwer et al., 2021) extends this idea with the bit information criterion to determine the number of bits to shave off. Based on the quantization - lossless compression pipeline, transform-based methods such as JPEG (Wallace, 1991), JPEG2000 (ISO Central Secretary, 2024) and SPECK (Chao and Gray, 2006) add a reversible transform before the quantization step to further increase the compression ratio. The SPERR (Li et al., 2023) compressor extends SPECK with a outlier encoding layer to bound the maximum compression error. Variational autoencoders (Han et al., 2024; Mirowski et al., 2024) follow a similar pipeline as transform-based methods but replace the pre-defined transform with learned neural networks. Parallel to previous methods, prediction-based methods can achieve high compression ratios and max error bounded compression by estimating values from their neighbors and encoding the residuals. Examples include FPZIP (Lindstrom, 2017), and the SZ family of methods (Di and Cappello, 2016; Liang et al., 2022; Jian et al., 2024). In addition, it is also possible to compress multidimensional data via tensor decomposition (Ballester-Ripoll et al., 2019), or implicit neural representation (Huang and Hoefer, 2022).

Although lossy compression methods can achieve higher compression ratios than lossless ones, they introduce an additional complexity to the user due to the rate-distortion trade-off: users must decide a compression level where the resulting error metrics are acceptable for their applications. However, there is no consensus on the error metrics for weather and climate data. Point-wise metrics such as root-mean-square error (RMSE), peak signal-to-noise ratio (PSNR), and maximum absolute error are often used as the default metrics to evaluate compression errors. In addition, these metrics can be directly used as

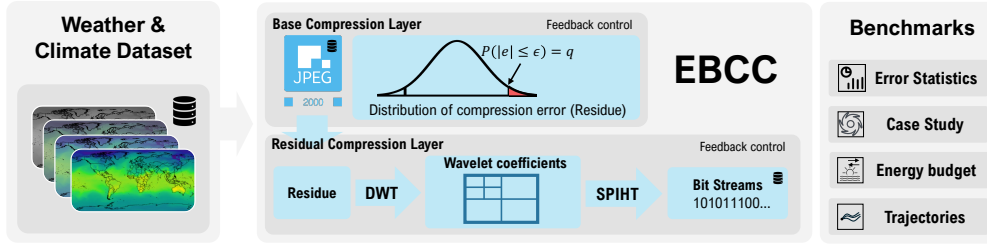


Figure 1. Diagram of EBCC (Error Bounded Climate Compressor) method. It compresses weather & climate datasets using a two-layer approach (Section 2): the base compression layer uses JPEG2000 for bulk compression, while the residual compression layer encodes excessive errors in the residue using discrete wavelet transform (DWT) and the set partitioning in hierarchical tree (SPIHT) algorithm. Both layers incorporate feedback control to achieve the target error bound. EBCC is evaluated on a suite of benchmarks and outperforms existing compression methods (Section 3)

the error target in the compression methods. However, point-wise metrics oversimplify the compression errors as they ignore the correlation between neighboring data points. Baker et al. (2019) suggests using the structural similarity index (SSIM) to evaluate the visual quality of the compressed climate data, and proposes a SSIM target of 0.99995 indicating visually indistinguishable data according to field experts. More statistical and visual metrics are proposed in Pinard et al. (2020); Tintó Prims et al. (2024). These metrics are useful when the compressed data is directly inspected by scientists or used as an end product, but they do not indicate the impact of downstream applications when the compressed data is used as input. Han et al. (2024) evaluates the impact of compression on the forecast skills of data-driven weather forecasting models. However, only one case study can hardly cover the whole field of weather and climate applications, and data-driven weather forecasting models are shown to be insensitive to small errors in the input data (Selz and Craig, 2023). We identify a gap in existing works that there lacks a comprehensive benchmark to evaluate compressed weather and climate data on representative applications.

Therefore, we create a benchmark suite for testing weather and climate applications on compressed datasets. It collects a wide range of test cases including error statistics, hurricane case studies, energy budget closure calculations, and Lagrangian air parcel trajectory simulation. The Lagrangian benchmark, which is to trace the radioactive cloud of the Fukushima incident, is a novelty and proves to be a error-sensitive benchmark. We consider it to be a relevant test for compression methods in safety-critical applications.

With the benchmarks in mind, we introduce a new error-bounded compression method EBCC (Figure 1). It utilizes JPEG2000 in the base compression layer to optimize RMSE, a residual compression layer to eliminate long-tail extreme errors introduced by the base layer, and a feedback rate-control mechanism in both layers to achieve the specified maximum error target. We test our method alongside other established methods including SZ (Di and Cappello, 2016), SZ3 (Liang et al., 2022), SPERR (Li et al., 2023), and Bitrounding (Klöwer et al., 2021). Our method outperforms all other methods across a wide range of compression ratios from $15\times$ to more than $300\times$. It concentrates most errors near zero rather than distributing them uniformly within the error bound, resulting in less distortion in derived variables and simulated trajectories. It is capable of creating more than $100\times$ compressed data while having downstream errors less than natural variability. Our contributions include:

- We create a novel error-bounded compressor that outperforms existing methods on weather and climate data.
- We create a comprehensive benchmark suite to test compression methods on weather and climate applications, including basic statistics, case studies on hurricane, energy budget closure, and Lagrangian trajectory simulations.
- We introduce ERA5 uncertainty ensembles as an indicator of natural variability for choosing error targets in practice, and show EBCC can achieve over $100\times$ compression while keeping downstream errors within the natural variability.

2 Method

To compress weather data with EBCC, we split the four-dimensional input array (time, level, latitude, longitude) into multiple chunks of the same shape and encode each chunk separately. Note that the implementation of EBCC supports arbitrary multidimensional arrays. We adapt the four-dimensional-array interpretation for the sake of the benchmark. Next, each chunk is flattened into a two-dimensional array. For example, an array of shape (T, P, H, W) is flattened into $(T \times P \times H, W)$. Then the 2D array is normalized according to its minimum and maximum values and fed to the EBCC compressor following the algorithm 1. Inside, the data is firstly compressed by the base compression layer using a JPEG2000 codec. JPEG2000 can compress data with a near-optimal RMSE under a given compression ratio, but it can result in long-tail errors which are problematic for weather data, in particular extreme weather events. Therefore, we add a second residual compression layer to keep overall compression error within a specific error bound. Unlike SPERR which directly encodes in the spatial domain, our method transforms the residue into wavelet coefficients and encodes the coefficients using the SPIHT algorithm (Said and Pearlman, 1996). The SPIHT algorithm goes through the bit planes of the wavelet coefficients from the most significant to the least significant plane and encodes significant bits in each bit plane. The residues related to excessive errors are likely encoded earlier, and therefore can be compressed with a high compression ratio. In fact, it creates an embedded code sequence where any prefix is also a valid sequence. Thanks to this property, we can easily control the compression rate by truncating the generated sequence.

While the two layer compression approach provides a flexible way to compress data with a near-optimal RMSE while maintaining a maximum absolute error bound, there is no direct way to determine the compression ratio as the compression parameter a priori given an error bound. Instead, we use feedback loops to find a suitable compression ratio iteratively. Since the base compression layer should compress the bulk part of the data while only leaving a small portion of excessive values to the sparse wavelet compression layer, we set a parameter q to quantify the "small portion", and control the base compression ratio so that the portion of non-excessive errors is q . In other words, the error bound is close to the q -quantile of the compression error. Similarly, with base compression fixed, we perform a single SPIHT encoding of the wavelet-transformed residue, and binary search the lowest truncation point that satisfies the error bound. We use binary search in feedback loops because it requires approximately 4 iterations to converge in practice, and replacing it with more complex searching algorithms cannot help much. Since the residual compression layer is designed for sparse excessive errors, the two-layer approach is suboptimal when the base compression alone can compress well without introducing excessive errors. In this case, we create a fallback mechanism at the end of algorithm 1 using pure base compression ($q = 1$), if it creates a higher compression ratio. It ensures EBCC can always have a higher or equal compression ratio compared with pure base compression for any $0 < q < 1$.

Algorithm 1: EBCC Compression Process

Input: 2D array A , max error target ϵ , initial base compression ratio r_0 , proportion q that base compression errors do not exceed ϵ .

Output: Compressed byte sequence e , can be decompressed with error less than ϵ .

Auxiliary functions: Encoding and decoding function of base compression method enc_{base} , dec_{base} , discrete wavelet transformation DWT and its inverse IDWT, and encoding and decoding function of SPIHT $\text{enc}_{\text{SPIHT}}$, $\text{dec}_{\text{SPIHT}}$.

```

1 Function  $\text{enc}_{\text{base}}(A, r, \epsilon)$ 
2    $e_A \leftarrow$  Compressed data of applying base compression
   method to  $A$  with a compression ratio  $r$ ;
3    $q_A \leftarrow$  Proportion of compression errors that do not exceed  $\epsilon$ ;
4   return  $e_A, q_A$ ;
5 end

   // Choose valid CR range for binary search
6  $e_A, q_{A0} \leftarrow \text{enc}_{\text{base}}(A, r_0, \epsilon)$ ;
7  $r_{\text{low}} \leftarrow r_0, r_{\text{high}} \leftarrow r_0$ ;
8  $q_A \leftarrow q_{A0}$ ;
9 while  $q_A < q$  do                                // Initial CR too high
10   $r_{\text{low}} \leftarrow r_{\text{low}}/2$ ;
11   $e_A, q_A \leftarrow \text{enc}_{\text{base}}(A, r_{\text{low}}, \epsilon)$ ;
12 end
13  $e_f \leftarrow e_A$ ;
14  $q_A \leftarrow q_{A0}$ ;
15 while  $q_A > q$  do                                // Initial CR too low
16   $r_{\text{high}} \leftarrow r_{\text{high}} * 2$ ;
17   $e_A, q_A \leftarrow \text{enc}_{\text{base}}(A, r_{\text{high}}, \epsilon)$ ;
18 end

   // Binary search for best feasible CR
19  $\text{tol} \leftarrow 10^{-3}$ ;
20 while  $r_{\text{high}} - r_{\text{low}} > \text{tol}$  do
21    $r_{\text{mid}} \leftarrow (r_{\text{low}} + r_{\text{high}})/2$ ;
22    $e_A, q_A \leftarrow \text{enc}_{\text{base}}(A, r_{\text{mid}}, \epsilon)$ ;
23   if  $q_A < q$  then
24      $r_{\text{high}} \leftarrow r_{\text{mid}}$ ;
25   else
26      $r_{\text{low}} \leftarrow r_{\text{mid}}$ ;
27      $e_f \leftarrow e_A$ ;
28   end
29 end

30  $e_{\text{base}} \leftarrow e_f$ ;
31  $A_{\text{base}} \leftarrow \text{dec}_{\text{base}}(e_{\text{base}})$ ;
32  $A_{\text{residue}} \leftarrow A - A_{\text{base}}$ ;
33  $W_A \leftarrow \text{DWT}(A_{\text{residue}})$ ;
34  $e_{\text{SPIHT}} \leftarrow \text{enc}_{\text{SPIHT}}(W_A)$ ;
35  $t_{\text{low}} \leftarrow 0, t_{\text{high}} \leftarrow \text{len}(e_{\text{SPIHT}})$ ;

   // Binary search for best feasible truncation point
36 while  $t_{\text{high}} - t_{\text{low}} > \text{tol}$  do
37    $t_{\text{mid}} \leftarrow (t_{\text{low}} + t_{\text{high}})/2$ ;
38    $A_{\text{rec}} \leftarrow \text{IDWT}(\text{dec}_{\text{SPIHT}}(e_{\text{SPIHT}}[t_{\text{mid}}])) + A_{\text{base}}$ ;
39   if  $\max(|A - A_{\text{rec}}|) > \epsilon$  then
40      $t_{\text{low}} \leftarrow t_{\text{mid}}$ ;
41   else
42      $t_{\text{high}} \leftarrow t_{\text{mid}}$ ;
43   end
44 end

45  $e_{\text{residue}} \leftarrow e_{\text{SPIHT}}[t_{\text{high}}]$ ;
46  $e \leftarrow \text{Concat}(e_{\text{base}}, e_{\text{residue}})$ ;
47 return  $e$                                 // Compressed byte sequence

```

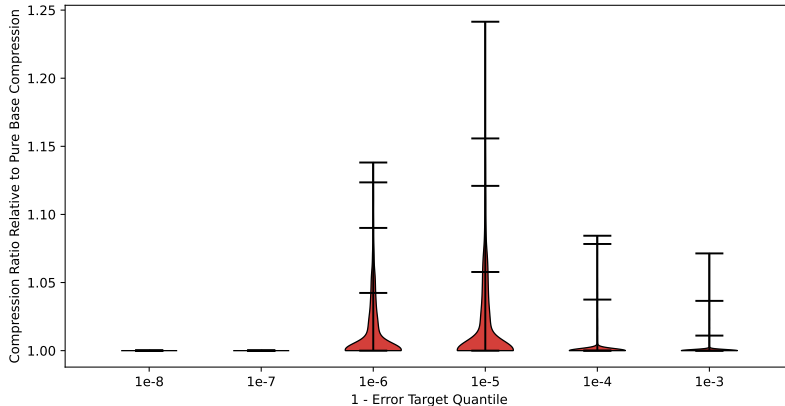


Figure 2. Distribution of compression ratios for different parameter q . The compression ratios are presented relative to using pure base compression (JPEG2000) only ($q = 1$). Horizontal ticks in the violin plot represent maximum value, 0.999, 0.99, and 0.9 quantiles of compression ratios.

2.1 Implementation Details

We implement EBCC as a C library using OpenJPEG as the JPEG2000 codec and a custom SPHIT encoder following Puchinger (2017). It currently supports single-precision floating-point input values and can be easily extended to other number formats. We also wrap the C library into an HDF5 filter plugin (Folk et al., 1999) to support compression and decompression of HDF5 datasets. In addition, it enables transparent decompression for tools supporting the HDF5 format, including NetCDF (Unidata, 2025), NCO (Zender, 2008), CDO (Schulzweida, 2023), or xarray (Hoyer and Hamman, 2017).

In the benchmark, we utilize the SZ, SZ3, and SPERR compression plugins from hdf5plugin (Vincent et al., 2025) to produce compressed NetCDF files and use enstools-compression (Tintó Prims et al., 2024) to automate the compression and decompression process.

2.2 Ablation Study

We introduce a parameter q in EBCC to control the amount of excessive errors left for the residue compression layer. The optimal value of q is dependent on the input data. If the input data is very smooth, we can set $q = 1$ since the base compression layer alone can compress it without introducing excessive errors. On the other hand, if the input data contains some ‘spikes’ where the base compression layer tends to smooth them out, we can set q to a smaller value than 1 to allow the residual compression layer to encode the ‘spikes’. Therefore, we perform an ablation study to investigate the effect of q on compressing weather and climate data. In the ablation study, we compress a sample ERA5 dataset with the same configuration as Section 3.1 while varying q from $1 - 10^{-3}$ to 1 and range relative max error bound from 0.1% to 10%. We compress each 2D slice of the sample dataset and calculate the relative compression ratio compared with $q = 1$.

In Figure 2, we collect the relative compression ratios for different q values, and plot their distribution. In every case, the relative compression ratios are larger than or equal to 1, thanks to the pure base compression fallback mechanism. The residual compression layer is most effective when q ranges from $1 - 10^{-5}$ to $1 - 10^{-6}$. It can improve the compression ratio by up to 24% compared with pure base compression. However, when q is outside of the range, the relative compression ratios converge to 1 because the residual compression layer either encodes too much data that it becomes less efficient than the base compression layer, resulting in a pure compression fallback, or it does not receive enough data as the base compression layer simply compresses all the data when q is too close to 1. As a result, we set $q = 1 - 10^{-5}$ in the following benchmarks. q might need to be tuned when having a different dataset to get an optimal compression ratio.

3 Benchmarks

We design a suite of benchmarks varying from compressing single-time-step sample data and collecting error statistics to using compressed data in complex applications inspired by real-world scenarios. In the first group of benchmarks, we apply compression on a small subset of the ERA5 dataset (Hersbach et al., 2020), and collect error statistics of the compression results including structural similarity (SSIM), histogram of point-wise errors, and power spectra of the reconstructed data. We further examine the visual quality of compressed data with a case study in a tropical cyclone scenario. In addition to directly compressed variables, we compute horizontal divergence from compressed primitive variables to test the compressors’ abilities of preserving relative values of neighboring locations and correlated variables. In the end, we apply the compressed data to a global energy budget closure analysis and Lagrangian trajectory simulation as representative real-world applications. We also identify thresholds of compression errors suitable for practical use.

Based on the benchmark suite, we test our method against other established compression methods: SZ, SZ3, SPERR, and Bitrounding. For error-bounded methods including SZ, SZ3, and SPERR, we vary the range-relative max absolute error tolerance from 0.1% to 10% to get a series of compressed data with varying compression ratios. Following Liang et al. (2022), the range-relative max error is defined as the ratio of max absolute error and data range in a data chunk (see Appendix A for details). For Bitrounding, we vary the kept bit-wise information content from 80% to 99.9%. In energy balance analysis and Lagrangian trajectory simulation, we only benchmark EBCC and SZ3 since SZ3 has been leading in previous benchmarks.

Table 1. Variables used in the benchmarks and sizes of related data stored in 32-bit floating-point format. u850 refers to zonal wind u at pressure level 850hPa, similar for v850 and u1000.

Section	Benchmark	Variables in Short Name	Raw Data Size
3.1	Error Statistics	$z, t, q, u, v, w, u10, v10, t2m, msl$	1.8 GB
3.2	Case Study	$u850, v850, u1000, t2m$	15.8 MB
3.3	Energy Budget Analysis	$z, t, q, v, rsnt, rlnt, rlms, rsns, hfss, hf1s$	111.0 GB
3.4	Lagrangian Trajectory Simulation	u, v, w, ps	6.1 GB

Table 2. Abbreviations of variables used in the benchmarks. TOA refers to top of atmosphere.

Short Name	Long Name	Unit	Short Name	Long Name	Unit
Pressure-level variables					
z	Geopotential	m^2s^{-2}	t	Temperature	K
q	Specific humidity	kg kg^{-1}	u	Zonal wind	m s^{-1}
v	Meridional wind	m s^{-1}	w	Vertical velocity	Pa s^{-1}
Surface variables					
u10	10m zonal wind	m s^{-1}	v10	10m meridional wind	m s^{-1}
t2m	2m temperature	K	msl	Mean sea level pressure	Pa
ps	Surface pressure	Pa	rsnt	Net shortwave radiation at TOA	W m^{-2}
r1nt	Net longwave radiation at TOA	W m^{-2}	r1ns	Net longwave radiation at surface	W m^{-2}
rsns	Net shortwave radiation at surface	W m^{-2}	hfss	Surface sensible heat flux	W m^{-2}
hfls	Surface latent heat flux	W m^{-2}			

3.1 Error Statistics

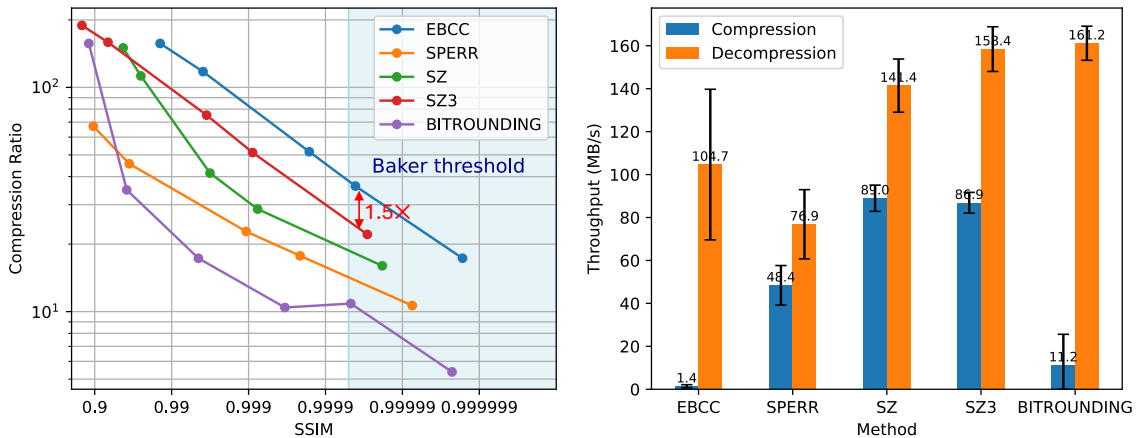


Figure 3. Left: Line plot of compression ratio (the higher the better) and structural similarity (SSIM, the higher the better) for each compression method (rate-distortion plot). The 0.99995 SSIM threshold proposed by Baker et al. (2019) is marked on the plot. The numbers are grouped according to the compression method and error target (0.1%, 0.5%, 1%, 5%, 10%), and aggregated using harmonic mean for compression ratios and arithmetic mean for structural similarities over all variables, pressure levels, and timesteps. The values for the Bitrounding method are accumulated according to SSIM bins at [0.95, 0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999]. **Right:** Compression and decompression throughput for each compression methods. The higher the better. The numbers are aggregated using arithmetic mean over all variables, pressure levels, timesteps and error targets. Error bars represent standard deviations.

We compress one-day sample data (1st March, 2024) with 10 pressure-level and surface-level variables (Table 1, Table 2). Those variables represent the basic state of the atmosphere and are commonly used for weather forecasting (Rasp et al., 2020). The left plot in Figure 3 shows the plot of compression ratio versus structural similarity (SSIM) for each compression method with relative error targets ranging from 0.1% to 10%, and annotates the 0.99995 SSIM threshold indicating the visually indistinguishable regime according to field experts (Baker et al., 2019). EBCC consistently outperforms other methods by having higher compression ratios at the same SSIM level. It can achieve a compression ratio of $35\times$ and maintain the SSIM above Baker’s threshold at a 0.5% relative error target, while other methods can only achieve $10\times$ to $25\times$.

We measure the compression and decompression throughput for each method as shown in Figure 3 (right). The throughput values are measured in single-thread on a desktop computer with an Intel Core i7-4770 CPU @ 3.4GHz and $2 \times 8\text{GB}$ DDR3 memory @ 1600 MHz. EBCC achieves a decompression throughput of 104.7 MB/s which is comparable to other methods (76.9 MB/s to 161.2 MB/s). It has a much lower compression throughput of 1.4 MB/s due to the overhead of double feedback loops. However, we can easily compress multiple data chunks in parallel to achieve a higher throughput. Also, the compression throughput is not a critical factor in real-world applications since the weather data is usually compressed offline once and decompressed multiple times when accessed.

Furthermore, we examine the compression errors in histogram plots (Figure 4), and power spectra (Figure 5). In these plots, we select compressed data with compression ratios close to $10\times$ and $100\times$ by binary searching the error target (for EBCC, SZ, SZ3, and SPERR) and bit information (for Bitrounding). According to the error histogram of compressed 2m temperature in Figure 4, EBCC concentrates most errors towards 0 and has far fewer large errors compared with others. EBCC distributes errors close to a Gaussian distribution at a $10\times$ compression ratio, while compression errors of other methods are more dispersed than the Gaussian distribution. SZ and SZ3 distribute errors close to a uniform distribution. As the compression ratio increases to $100\times$, the distributions of errors converge to Gaussian distributions with the exception of Bitrounding, which has more negative errors than positive errors. Histogram plots of compression errors for other variables are available in the Appendix, which show a similar trend as in Figure 4.

In the spectra plots shown in Figure 5, we calculate the power of the spherical harmonic signal in each degree for both original and compressed geopotential data at 500hPa. We choose this variable because differences among methods are more distinctive, and features observed in Figure 5 also apply to other variables (available in the Appendix). In the high accuracy regime with a compression ratio of $10\times$ (left), all the compression methods perform well by having almost identical spectra as the reference one. When zooming in the high-frequency region, EBCC and Bitrounding are closest to the reference, and other methods add artificial high-frequency components to the compressed data. In the high compression ratio regime with a compression ratio of $100\times$ (right), EBCC and SZ3 still conform well to the reference while others add significant medium-to-high frequency artifacts. EBCC slightly underestimates high-frequency details which effectively smoothens the data, but it does not introduce high frequency artifacts in any case unlike SZ3 (see in Appendix). We will further inspect compressed data in the following section.

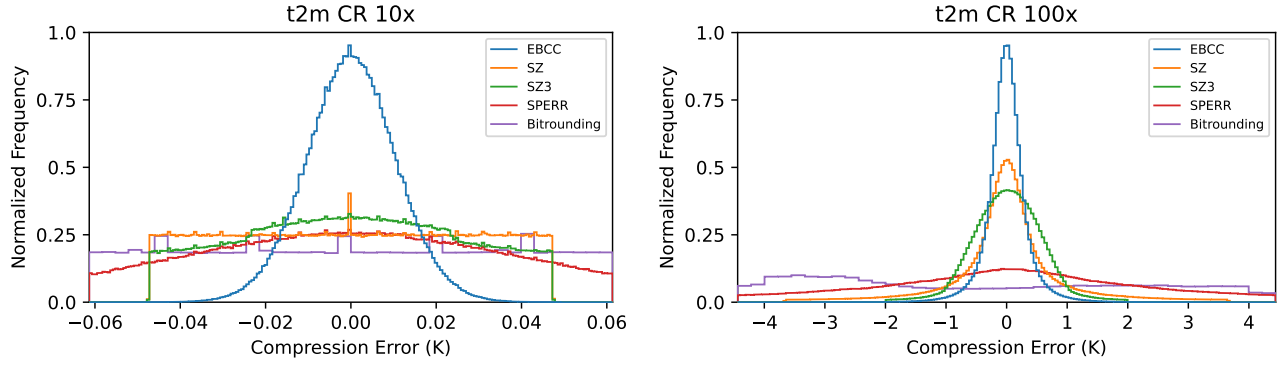


Figure 4. Histogram of compression errors for 2m temperature (t_{2m}) with compression ratios around $10\times$ (left) and $100\times$ (right). X-axis is cropped at the maximum compression error of EBCC. SPERR and Bitrounding have excessive errors in the cropped regions. They result in higher maximum errors while reaching a similar compression ratio as other methods.

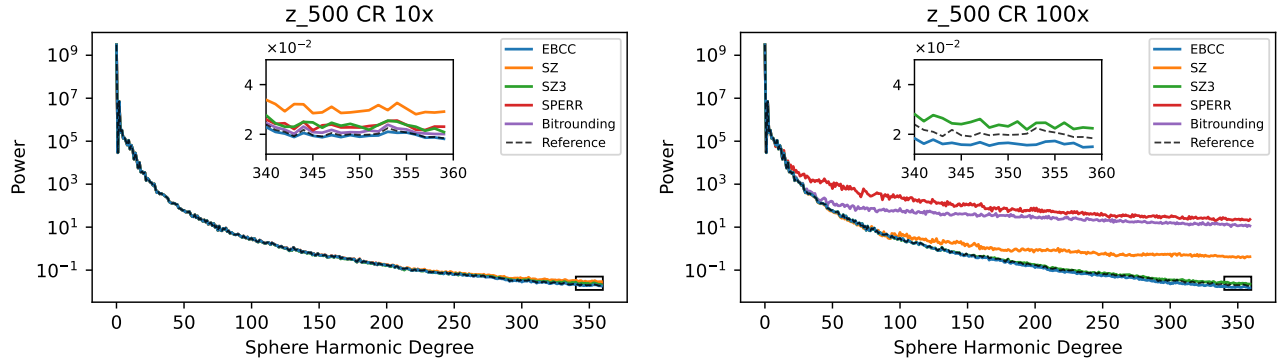


Figure 5. Spherical harmonics power spectra of compressed and original data for geopotential (z) at 500hPa with compression ratios around $10\times$ (left) and $100\times$ (right).

3.2 Case Study of Hurricane Iota

Weather and climate data are often used for visual examination, thus it is important for compressors to preserve the visual fidelity of the compressed data. To test this, we select the data containing the record-breaking hurricane Iota on 16th Nov 2020. We use the same binary searching technique as in Section 3.1 to find the right compression parameters that produce $10\times$ and $100\times$ compressed data. Due to the constraint of the Bitrounding method, it cannot achieve a high compression ratio larger than $17\times$. In Figure 6, all of the compressed data are perceptually similar to the reference with a compression ratio of $10\times$. When relaxing to $100\times$, SZ produces obvious blocked artifacts, SPERR reduces the data into several regions with distinct values, SZ3 introduces some mild bumps. EBCC smoothens some details around the hurricane, but it does not add any artifacts. The residue

plot in Figure 6 bottom verifies this observation. In addition, it shows EBCC has an edge over SZ3 because the compression errors of EBCC are lower in magnitude and have lower low-frequency component.

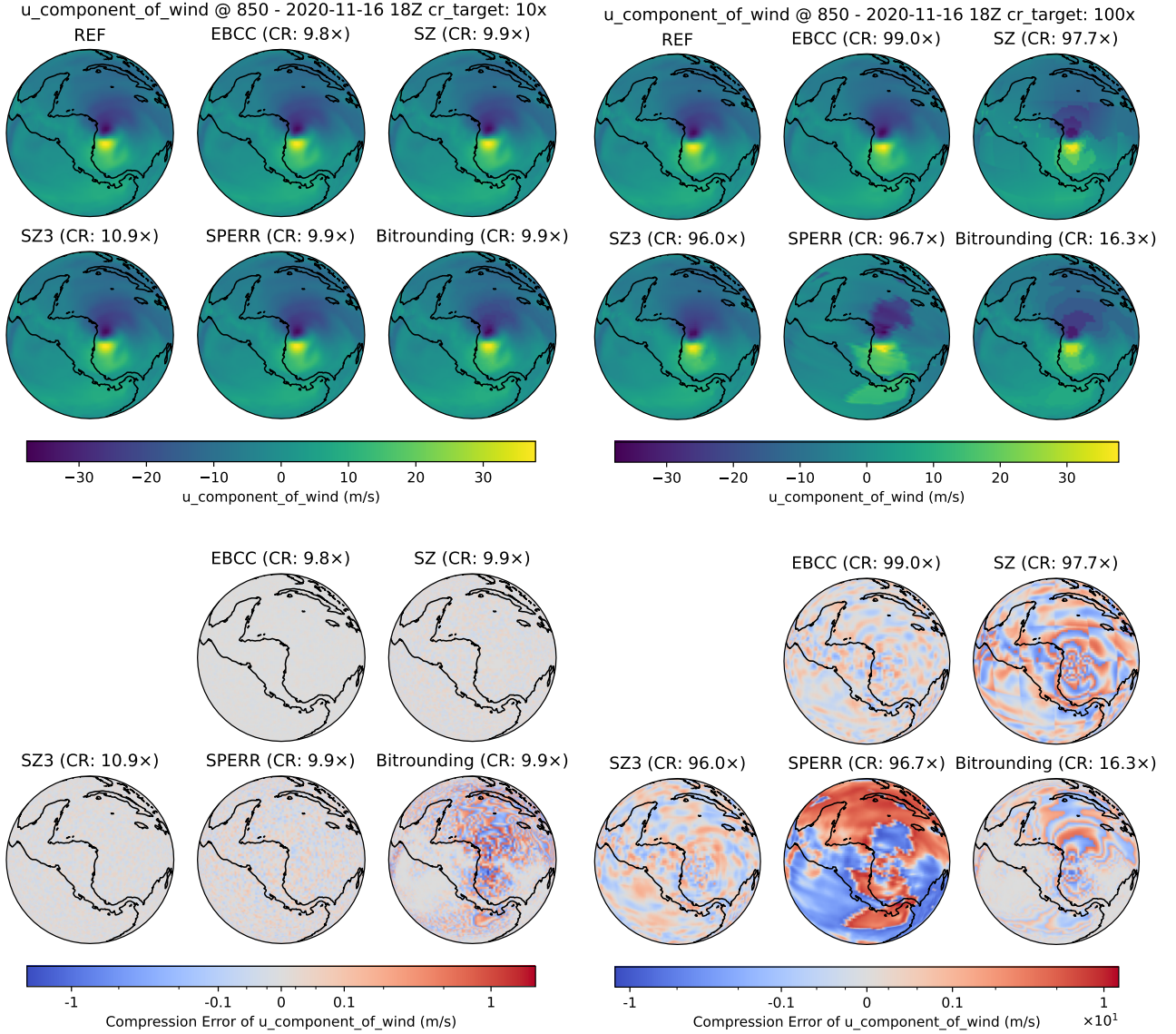


Figure 6. Top: Plots of compressed 850hPa zonal wind speed (u) of hurricane Iota with compression ratios around 10 $\times$ (left) and 100 $\times$ (right). **Bottom:** Plots of compression errors with corresponding compression ratios as top plots.

While compression methods including EBCC, SZ, and SZ3 can preserve visual features of compressed data even at a high compression ratio of 100 $\times$, this is less likely when examining derived variables computed from compressed primitive variables. Figure 7 shows the derived 850hPa horizontal wind divergence at the hurricane location. Horizontal divergence is calculated

from the spatial derivatives of two components of horizontal wind speed. Compressors have to reconstruct the first order spatial derivatives accurately to maintain a good fidelity of computed divergence. At a compression ratio of $10\times$, there are noticeable degradations for Bitrounding compressed data. Others are close to the reference with EBCC producing the least errors. At a compression ratio of $100\times$, EBCC results in a slightly blurred image, SZ3 introduces high-frequency dot-like artifacts, while SZ and SPERR produce near incomprehensible results. Bitrounding produces low-quality horizontal divergence albeit at a $14.5\times$ compression ratio.

3.3 Energy Budget Closure Analysis

It is important to maintain the physical consistency of the compressed data. In section 3.2, we have tested a simple case of calculating wind divergence. In this section, we perform a more complex benchmark by testing the closure of the global atmospheric energy budget using compressed data. In essence, this benchmark examines the deviation in the sum of compressed data in an attempt to reproduce atmospheric heat transport derived from 3D atmospheric fields. Ideally, we expect compression errors to be zero-centered, thus not adding bias to statistics such as spatial or temporal mean. The zonal mean moist static energy (MSE) framework is a commonly used approximation of the atmospheric energy budget which has been used to study the Arctic climate (Graversen and Burtu, 2016) or storm track shifts (Shaw et al., 2018). At a given latitude band,

$$\partial_t \langle [\bar{h}] \rangle + \partial_y \langle [\bar{v}\bar{m}] \rangle - [\overline{F_{NE}}] = R \approx 0 \quad (1)$$

(Neelin and Held, 1987). The first term on the left-hand side corresponds to the tendency of the thermal energy within the atmospheric column, $h = c_p T + Lq$, with c_p the specific heat capacity of dry air and L the latent heat of evaporation, which are both taken as constant here. T and q are air temperature and specific humidity, respectively. The second term denotes the meridional divergence of the flux of MSE that is the product of the meridional wind, v , and $m = c_p T + Lq + \Phi$ with Φ the geopotential. The third term represents the net energy (NE) input into the column by surface and top of atmosphere radiative and turbulent heat fluxes. Square brackets denote zonal averages and angle brackets the mass-weighted vertical integration using climatological mean surface pressure (Boer and Sargent, 1985). The compression algorithm is evaluated on monthly mean data denoted by overbars. This comprehensive combination of operations and variables helps us better understand the effect of compression in a real-world scenario. We make use of 111 gigabytes of 1° -resolution ERA5 data from August to October 2016 to calculate the terms in Equation 1 for September 2019. Then we compress the data with relative error targets of 0.1%, 0.5%, 1%, 5%, and 10% respectively, and compute the zonal mean energy budget residual R for the original and compressed data. Perfect budget closure would correspond to a vanishing residual $R \equiv 0$. The computed residual has the order of $\approx 15 \text{ Wm}^{-1}$ (black line in Figure 8) which is in line with other previous estimates (Trenberth et al., 2011; Mayer et al., 2017). A considerable fraction of this residual can be linked to uncertainties in assimilated surface heat fluxes (Robertson et al., 2020). Simplifications in the method following (Mayer et al., 2017; Cox et al., 2024) further introduce discrepancies, particularly over the steep Antarctic topography near 70°S . Turning to the evaluation of the compression algorithms, this reference residual serves as a benchmark for the performance of the compression algorithms. In other words, deviations from the reference should ideally be small compared to the reference residual itself.

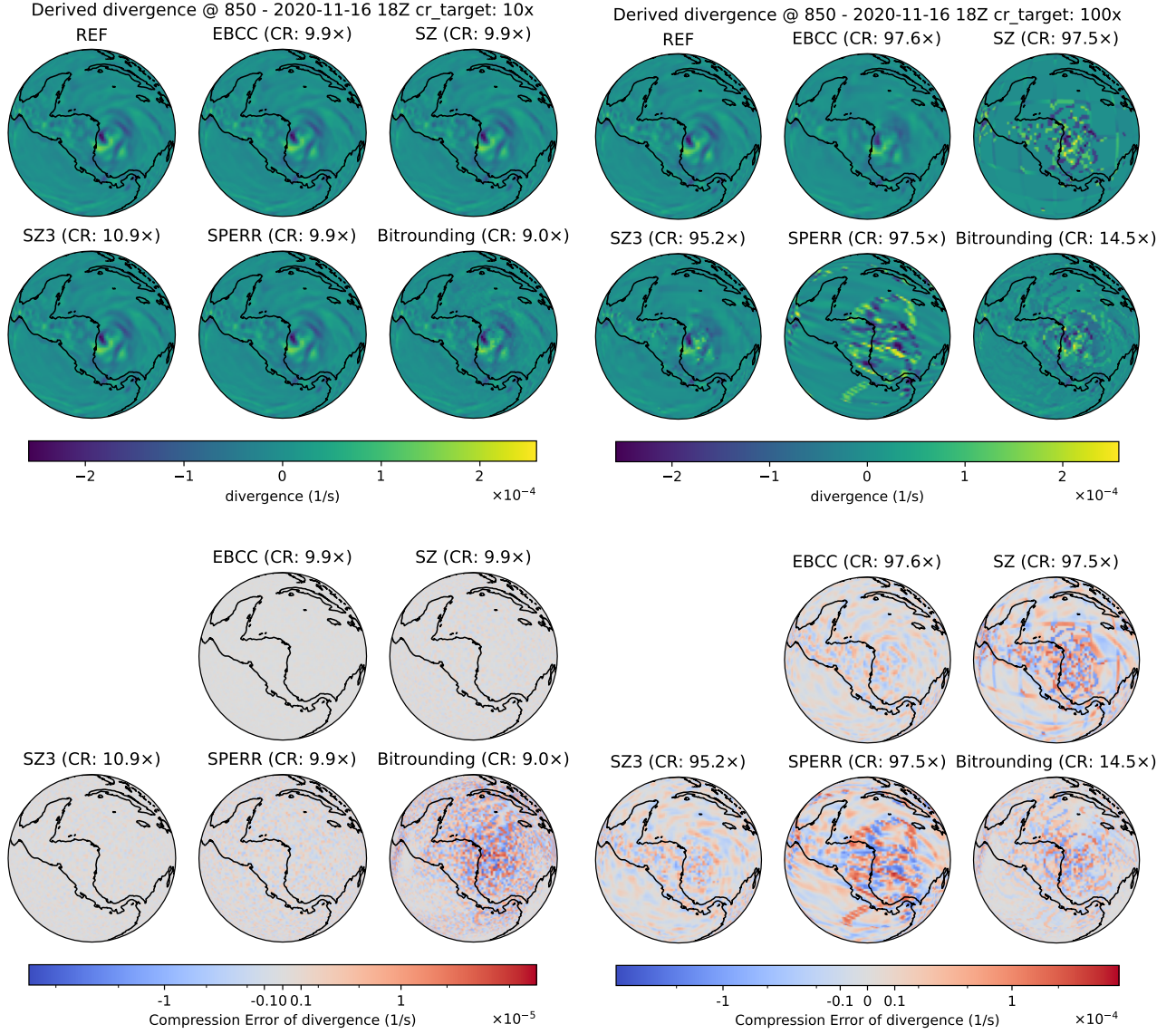


Figure 7. Top: Plots of derived 850-hPa horizontal wind divergence from compressed horizontal winds with compression ratios around 10 $\times$ (left) and 100 $\times$ (right). **Bottom:** Plots of compression errors with corresponding compression ratios as top plots.

As shown in Figure 8, R computed from compressed data is close to the reference except for SZ3 with 5% and 10% relative error targets. These two cases result in significant differences in R although achieving high compression ratios of 283 $\times$ and 524 $\times$. In contrast, R computed from EBCC compressed data is closer to the reference. Quantitatively, we calculate the RMSE of R for the compressed data and plot the rate-distortion plot for EBCC and SZ3 (Figure 9 left). It is clear that EBCC outperforms SZ3 in RMSE at compression ratios ranging from 20 $\times$ to 300 $\times$, while EBCC typically allows a higher

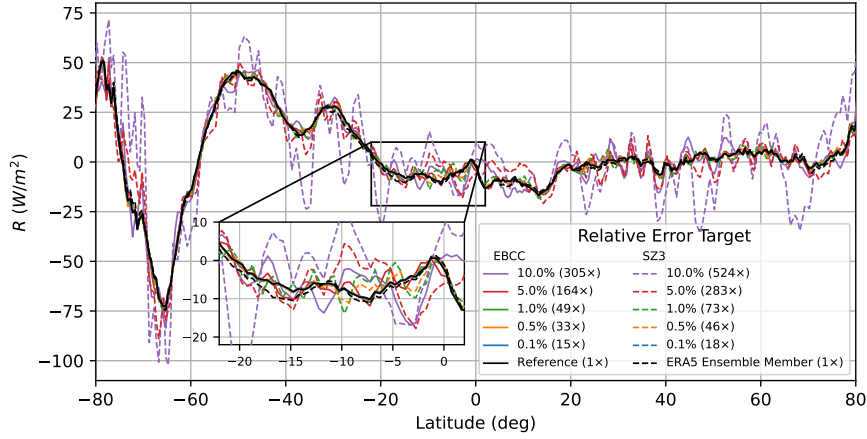


Figure 8. Monthly zonal mean energy budget residual R for compressed and original data for September 2016. The relative error target is set to 0.1%, 0.5%, 1%, 5%, and 10% respectively. The resulting compression ratios are marked in the legend.

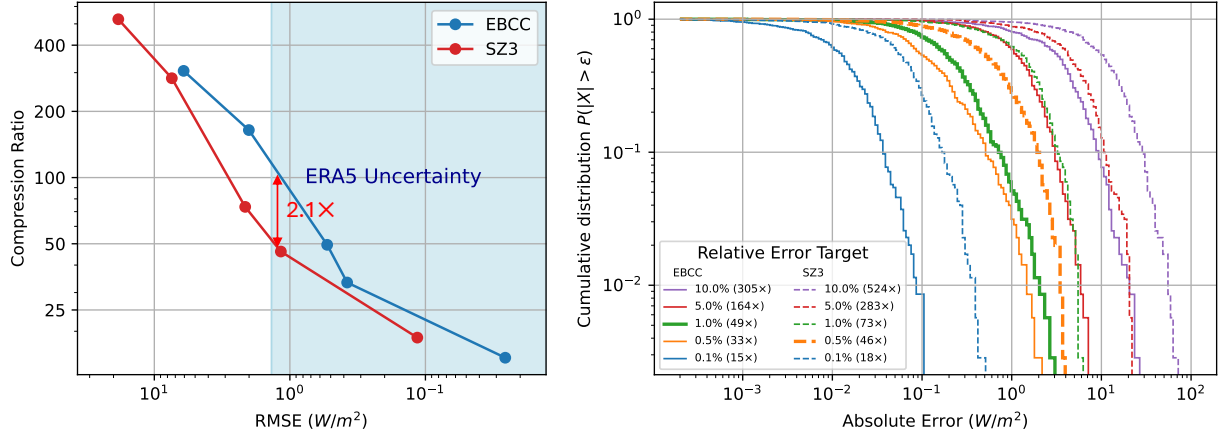


Figure 9. Left: Scatter plot of compression ratio and RMSE of compressed R (rate-distortion plot). **Right:** Cumulative distribution function (CDF) of absolute errors of R induced by compression. EBCC with 1% relative error target and SZ3 with 0.5% error target are highlighted. EBCC's CDF dominates SZ3's despite of having similar compression ratios (49 $\times$ and 46 $\times$).

error bound compared with SZ3 to achieve the same compression ratio. In the plot, we highlight the RMSE due to ERA5 uncertainty representing the natural variability. It is created by running the benchmark on each of the 10 members representing ERA5 uncertainty and taking the largest RMSE of the 10 members. It can help decide the error target or compression ratio for lossy compression. In this case, EBCC can reach a 100 $\times$ compression ratio with resulting errors within natural variability, while SZ3 can only reach 50 $\times$.

In the right plot of Figure 9, we show the cumulative distribution function (CDF) of absolute errors of R , representing the fraction of errors whose absolute value is larger than a certain value. The CDFs for EBCC are smaller than SZ3's at the same relative error target, indicating that EBCC has a smaller portion of large errors. This even holds true when the two have similar compression ratios (e.g., EBCC 1% and SZ3 0.5% relative error target). In sum, EBCC is the preferable choice to conserve the zonal mean energy budget terms at a monthly scale.

3.4 Lagrangian Trajectory Simulation

In the end, we present the Lagrangian air parcel trajectory simulation as an extreme benchmark. Trajectories are highly sensitive to errors in the wind data, and small errors in the beginning will amplify in following time as the simulator integrates wind speeds at the positions of the air parcels. We design an experiment simulating the trajectories of air parcels around the Fukushima nuclear incident for one week using the Lagrangian analysis tool LAGRANTO (Sprenger and Wernli, 2015), and examine the deviation of trajectories caused by using compressed wind data. Similar to Section 3.3, we run trajectory simulations on ERA5 uncertainty members and take the smallest error metric of all the members to estimate the natural variability of the data.

The deviation of trajectories is measured through the RMSE of particle densities at every time step, where we treat each tracked air parcel as an abstract particle. This effectively measures the difference of the particle distribution. Figure 10 shows the rate-distortion plot for EBCC and SZ3. EBCC has approximately $1.9\times$ compression ratio of SZ3 at the same RMSE level. Similarly, it can achieve around 50% RMSE at the same compression ratio. The baseline RMSE derived from trajectories of ERA5 uncertainty members is relatively high. This allows a 10% error target for EBCC and 1% - 5% error target for SZ3, which results in compression ratios of $282\times$ and around $150\times$ respectively.

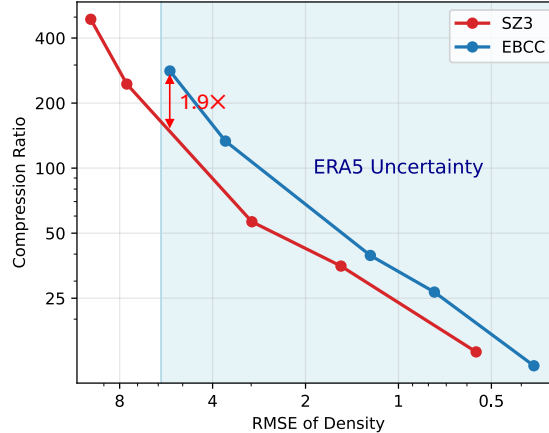


Figure 10. Scatter plot of compression ratio and RMSE of particle densities at every timestep (rate-distortion plot).

Figure 11 shows the simulated trajectory using original, ERA5 uncertainty ensemble member, and compressed data. We select EBCC with relative error targets of 10% and 1%, and SZ3 with targets of 5% and 0.5%, since the two pairs have similar

compression ratios. As indicated in Figure 10, the difference between trajectories of ERA5 ensemble members is evident, as more lower level particles circle around the north-western Pacific in member 1. EBCC with 1% relative error results in a similar trajectory plume as the reference with only a slight change in the thickness of low level ($> 950\text{hPa}$) trajectories around 50°N , 160°E . When relaxing the error target to 10%, the changes in near-surface trajectories (green and yellow) become more apparent while there are fewer trajectories going north at higher levels (blue). For SZ3, the difference in trajectories at 0.5% relative error is already comparable to EBCC at 1% relative error. The trajectories for SZ3 with 5% relative error are visibly different from the reference trajectories with ERA5 member 0 data and such difference is larger than the difference between ensemble members. In Figure 12, we examine the density of tracked particles one week after the Fukushima incident. Both EBCC at 1% and SZ3 at 0.5% relative error targets produce perceptually similar distributions as the reference. At 10% error target, our method keeps the general distribution, while SZ3 at 5% error target results in fewer air parcels near the south-east region of the Kamchatka peninsula and the US west coast.

4 Conclusions

This study introduces a new error-bounded compression method EBCC targeted for weather and climate data. It contains a base compression layer using JPEG2000 to capture the bulk part of the data and a residual compression layer to record the sparse signal that exceeds the given error bound. The residual compression layer encodes the residue of the previous layer using the SPIHT algorithm. We also introduce feedback-based rate control for both layers that adjusts the compression ratios to achieve the given error bound. We implement EBCC as a standalone C library with CDO and Zarr plugins, which transparently integrates into existing toolchains.

We test EBCC with established compression methods on a suite of benchmarks including error statistics, a case study of a tropical cyclone, the closure of the Earth’s energy budget, and a Lagrangian air parcel trajectory simulation. EBCC is superior in all benchmark cases at given range relative error targets from 0.1% to 10% and compression ratios from $16\times$ to more than $300\times$, respectively. Our method can reconstruct the derivatives of the compressed data with high fidelity. As the compression ratio or error bound increases, the EBCC compressed data degrades to a smoothed dataset instead of adding high-frequency artifacts found in other compression methods. In energy balance analysis and Lagrangian trajectory simulations, EBCC results in less distortion compared to the best existing method SZ3 at comparable compression ratios. It can achieve more than $100\times$ compression ratios while keeping downstream error metrics within natural variability derived from ERA5 uncertainty members. EBCC is an important step forward to cope with the growing difficulties in storing weather and climate data.

5 Code Availability

The source code of EBCC is available in github.com/spcl/EBCC.

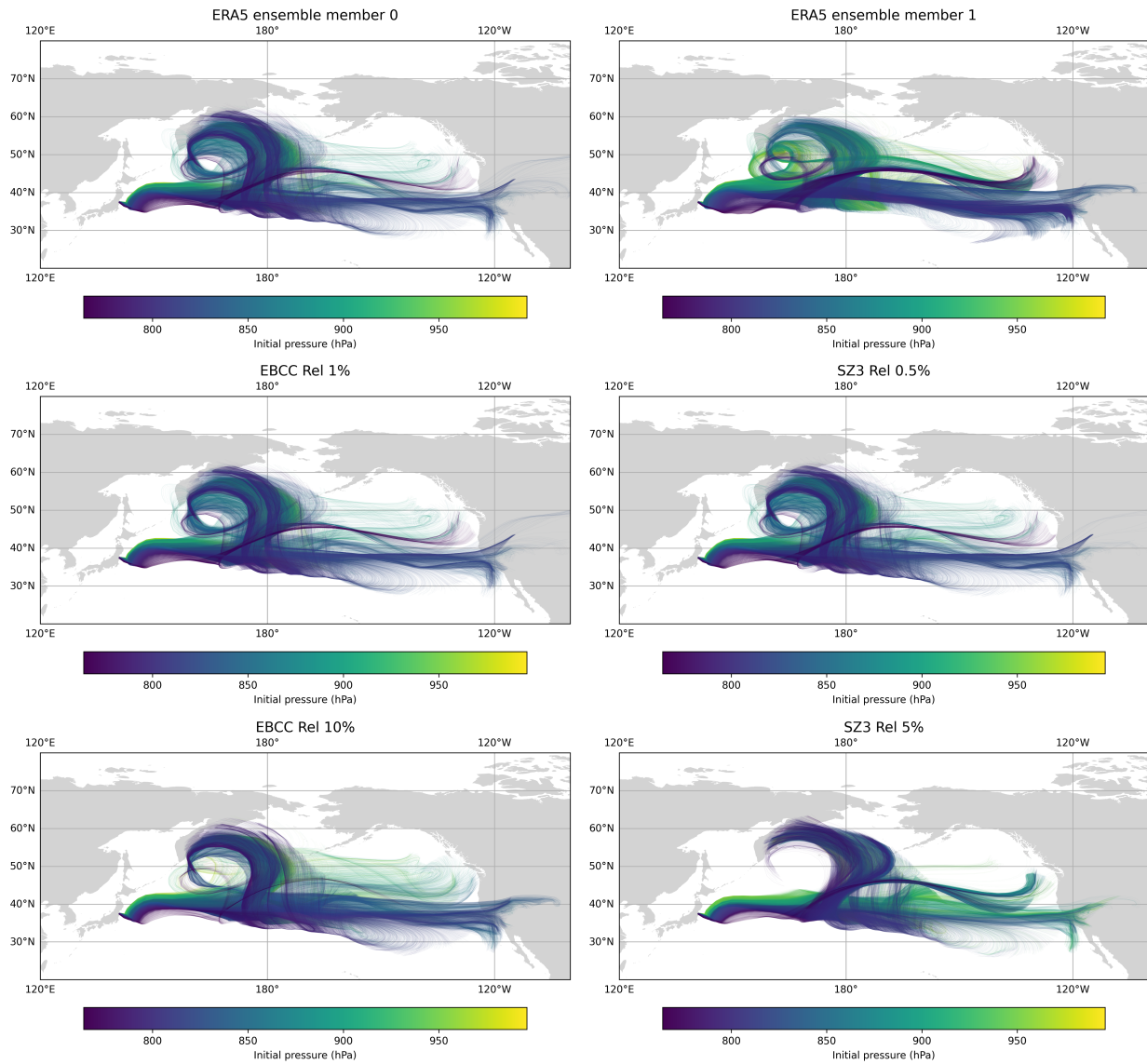


Figure 11. Simulated trajectories of particles around the Fukushima incident in 1 week period colored by the initial pressure level of particles. The simulations are performed with original wind speed data from ERA5 (ensemble member 0), ERA5 uncertainty ensemble member 1, compressed data with EBCC relative error targets at 1% and 10% (compression ratio: 39 $\times$, 282 $\times$), and SZ3 relative error targets at 0.5% and 5% (compression ratio: 35 $\times$, 245 $\times$).

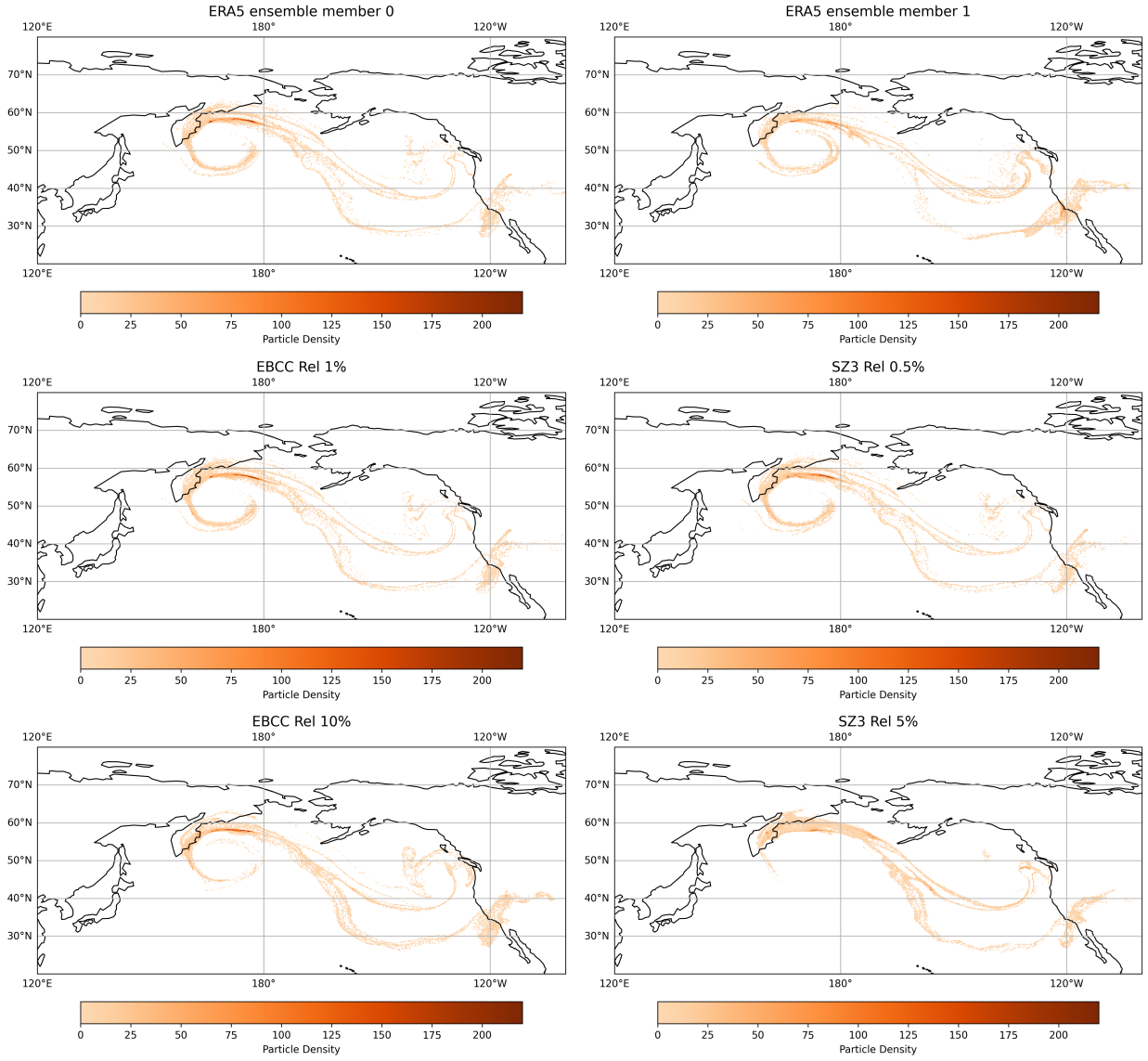


Figure 12. Density of simulated particles 1 week after the Fukushima incident. The simulations are performed with original wind speed data from ERA5 (ensemble member 0), ERA5 uncertainty ensemble member 1, compressed data with EBCC relative error targets at 1% and 10% (compression ratio: $39\times$, $282\times$), and SZ3 relative error targets at 0.5% and 5% (compression ratio: $35\times$, $245\times$).

References

- Acosta, M. C., Palomas, S., Paronuzzi Ticco, S. V., Utrera, G., Biercamp, J., Bretonniere, P.-A., Budich, R., Castrillo, M., Caubel, A., Doblas-Reyes, F., et al.: The computational and energy cost of simulation and storage for climate science: lessons from CMIP6, *Geoscientific Model Development*, 17, 3081–3098, 2024.
- Baker, A. H., Hammerling, D. M., and Turton, T. L.: Evaluating image quality measures to assess the impact of lossy data compression applied to climate simulation data, in: *Computer Graphics Forum*, vol. 38, pp. 517–528, Wiley Online Library, 2019.
- Ballester-Ripoll, R., Lindstrom, P., and Pajarola, R.: TTHRESH: Tensor compression for multidimensional visual data, *IEEE transactions on visualization and computer graphics*, 26, 2891–2903, 2019.
- Bauer, P., Dueben, P. D., Hoefler, T., Quintino, T., Schulthess, T. C., and Wedi, N. P.: The digital revolution of Earth-system science, *Nature Computational Science*, 1, 104–113, 2021.
- Boer, G. J. and Sargent, N. E.: Vertically Integrated Budgets of Mass and Energy for the Globe, *Journal of the Atmospheric Sciences*, 42, 1592–1613, [https://doi.org/10.1175/1520-0469\(1985\)042<1592:VIBOMA>2.0.CO;2](https://doi.org/10.1175/1520-0469(1985)042<1592:VIBOMA>2.0.CO;2), 1985.
- Chao, C.-c. and Gray, R. M.: Image compression with a vector speck algorithm, in: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, pp. II–II, IEEE, 2006.
- Cox, T., Donohoe, A., Armour, K. C., Roe, G. H., and Frierson, D. M.: A new method for calculating instantaneous atmospheric heat transport, *Journal of Climate*, 37, 4337–4346, 2024.
- Di, S. and Cappello, F.: Fast error-bounded lossy HPC data compression with SZ, in: *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 730–739, IEEE, 2016.
- ECMWF: Annual Report 2023, <https://annualreport.ecmwf.int/2023>, accessed: 2024-04-22, 2023.
- Folk, M., Cheng, A., and Yates, K.: HDF5: A file format and I/O library for high performance computing applications, in: *Proceedings of supercomputing*, vol. 99, pp. 5–33, 1999.
- Govett, M., Bah, B., Bauer, P., Berod, D., Bouchet, V., Corti, S., Davis, C., Duan, Y., Graham, T., Honda, Y., et al.: Exascale computing and data handling: Challenges and opportunities for weather and climate prediction, *Bulletin of the American Meteorological Society*, 105, E2385–E2404, 2024.
- Graversen, R. G. and Burtu, M.: Arctic amplification enhanced by latent energy transport of atmospheric planetary waves, *Quarterly Journal of the Royal Meteorological Society*, 142, 2046–2054, 2016.
- Han, T., Chen, Z., Guo, S., Xu, W., and Bai, L.: Cra5: Extreme compression of era5 for portable global climate and weather research via an efficient variational transformer, *arXiv preprint arXiv:2405.03376*, 2024.
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al.: The ERA5 global reanalysis, *Quarterly journal of the royal meteorological society*, 146, 1999–2049, 2020.
- Hoefler, T., Stevens, B., Prein, A. F., Baehr, J., Schulthess, T., Stocker, T. F., Taylor, J., Klocke, D., Manninen, P., Forster, P. M., et al.: Earth Virtualization Engines: a technical perspective, *Computing in Science & Engineering*, 25, 50–59, 2023.
- Hoyer, S. and Hamman, J.: xarray: N-D labeled arrays and datasets in Python, *Journal of Open Research Software*, 5, <https://doi.org/10.5334/jors.148>, 2017.
- Huang, L. and Hoefler, T.: Compressing multidimensional weather and climate data into neural networks, *arXiv preprint arXiv:2210.12538*, 2022.

- Huang, X., Ni, Y., Chen, D., Liu, S., Fu, H., and Yang, G.: Czip: A Fast Lossless Compression Algorithm for Climate Data, *International Journal of Parallel Programming*, 44, 1248–1267, 2016.
- ISO Central Secretary: Information technology — JPEG 2000 image coding system Part 1: Core coding system, Standard ISO/IEC 15444-1:2024, International Organization for Standardization, Geneva, CH, <https://www.iso.org/standard/87632.html>, 2024.
- Jian, Z., Di, S., Liu, J., Zhao, K., Liang, X., Xu, H., Underwood, R., Wu, S., Huang, J., Chen, Z., et al.: Cliz: Optimizing lossy compression for climate datasets with adaptive fine-tuned data prediction, in: 2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 417–429, IEEE, 2024.
- Klöwer, M., Razingier, M., Dominguez, J. J., Düben, P. D., and Palmer, T. N.: Compressing atmospheric data into its real information content, *Nature Computational Science*, 1, 713–724, 2021.
- Li, S., Lindstrom, P., and Clyne, J.: Lossy Scientific Data Compression With SPERR, in: 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1007–1017, <https://doi.org/10.1109/IPDPS54959.2023.00104>, 2023.
- Liang, X., Zhao, K., Di, S., Li, S., Underwood, R., Gok, A. M., Tian, J., Deng, J., Calhoun, J. C., Tao, D., et al.: Sz3: A modular framework for composing prediction-based error-bounded lossy compressors, *IEEE Transactions on Big Data*, 9, 485–498, 2022.
- Lindstrom, P. G.: Fpzip, Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States), 2017.
- Mayer, M., Haimberger, L., Edwards, J. M., and Hyder, P.: Toward consistent diagnostics of the coupled atmosphere and ocean energy budgets, *Journal of Climate*, 30, 9225–9246, 2017.
- Miles, A., Kirkham, J., Hamman, J. J., et al.: zarr-python: v2.18.5, <https://doi.org/10.5281/zenodo.15101732>, 2025.
- Mirowski, P., Warde-Farley, D., Rosca, M., Grimes, M. K., Hasson, Y., Kim, H., Rey, M., Osindero, S., Ravuri, S., and Mohamed, S.: Neural Compression of Atmospheric States, arXiv preprint arXiv:2407.11666, 2024.
- Mummadisetty, B. C., Puri, A., Sharifahmadian, E., and Latifi, S.: Lossless Compression of Climate Data, in: *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*, pp. 391–400, Springer, 2015.
- Neelin, J. D. and Held, I. M.: Modeling Tropical Convergence Based on the Moist Static Energy Budget, *Monthly Weather Review*, 115, 3–12, [https://doi.org/10.1175/1520-0493\(1987\)115<0003:MTCBOT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1987)115<0003:MTCBOT>2.0.CO;2), 1987.
- Pinard, A., Hammerling, D. M., and Baker, A. H.: Assessing differences in large spatio-temporal climate datasets with a new python package, in: 2020 IEEE International Conference on Big Data (Big Data), pp. 2699–2707, IEEE, 2020.
- Puchinger, R.: ImShrinker - SPIHT algorithm implementation, <https://github.com/gilson27/imshrinker>, <https://github.com/gilson27/imshrinker>, last modified in 2017. Originally created in 2008, revised in 2013. A minimal console implementation of the SPIHT algorithm from the paper by A. Said and W. A. Pearlman (1996)., 2017.
- Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N.: WeatherBench: a benchmark data set for data-driven weather forecasting, *Journal of Advances in Modeling Earth Systems*, 12, e2020MS002 203, 2020.
- Rice, R. F.: Lossless coding standards for space data systems, in: *Conference Record of the Thirtieth Asilomar conference on Signals, Systems and Computers*, vol. 1, pp. 577–585, IEEE, 1996.
- Robertson, F. R., Roberts, J. B., Bosilovich, M. G., Bentamy, A., Clayson, C. A., Fennig, K., Schröder, M., Tomita, H., Compo, G. P., Gutenstein, M., et al.: Uncertainties in ocean latent heat flux variations over recent decades in satellite-based estimates and reduced observation reanalyses, *Journal of Climate*, 33, 8415–8437, 2020.
- Said, A. and Pearlman, W. A.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Transactions on circuits and systems for video technology*, 6, 243–250, 1996.
- Schulzweida, U.: CDO User Guide, <https://doi.org/10.5281/zenodo.10020800>, 2023.

- Selz, T. and Craig, G. C.: Can artificial intelligence-based weather prediction models simulate the butterfly effect?, *Geophysical Research Letters*, 50, e2023GL105 747, 2023.
- Shannon, C. E.: A mathematical theory of communication, *The Bell system technical journal*, 27, 379–423, 1948.
- Shaw, T. A., Barpanda, P., and Donohoe, A.: A moist static energy framework for zonal-mean storm-track intensity, *Journal of the Atmospheric Sciences*, 75, 1979–1994, 2018.
- Silver, J. D. and Zender, C. S.: The compression–error trade-off for large gridded data sets, *Geoscientific Model Development*, 10, 413–423, 2017.
- Sprenger, M. and Wernli, H.: The LAGRANTO Lagrangian analysis tool–version 2.0, *Geoscientific Model Development*, 8, 2569–2586, 2015.
- Tintó Prims, O., Redl, R., Rautenhaus, M., Selz, T., Matsunobu, T., Modali, K. R., and Craig, G.: The effect of lossy compression of numerical weather prediction data on data analysis: a case study using enstools-compression 2023.11, *Geoscientific Model Development*, 17, 8909–8925, 2024.
- Trenberth, K. E., Fasullo, J. T., and Mackaro, J.: Atmospheric moisture transports from ocean to land and global energy flows in reanalyses, *Journal of climate*, 24, 4907–4924, 2011.
- Unidata: NetCDF-C Library, <https://doi.org/10.5065/D6H70CW6>, 2025.
- Vincent, T., Solé, V. A., Kieffer, J., Prims, O. T., Kittisopikul, M., Florian-g, Plaswig, F., i Balaguer, I. V., Steinbach, P., Valls, V., Klein, J., Gerstel, M., junyuewang, and payno: silx-kit/hdf5plugin: 5.1.0: 31/03/2025, <https://doi.org/10.5281/zenodo.15111354>, 2025.
- Wallace, G. K.: The JPEG Still Picture Compression Standard, *Communications of the ACM*, 34, 30–44, <https://doi.org/10.1145/103085.103089>, 1991.
- WMO: Manual on Codes – Volume I.2: FM 92 GRIB, Edition 2, World Meteorological Organization, Geneva, Switzerland, <https://community.wmo.int/activity-areas/wis/publications/306-vI.2/latest-version>, wMO-No. 306; valid 19 May 2025, 2025.
- Zender, C. S.: Analysis of self-describing gridded geoscience data with netCDF Operators (NCO), *Environmental Modelling & Software*, 23, 1338–1342, 2008.
- Zender, C. S.: Bit Grooming: statistically accurate precision-preserving quantization with compression, evaluated in the netCDF Operators (NCO, v4. 4.8+), *Geoscientific Model Development*, 9, 3199–3211, 2016.

Appendix A: Definitions of Error Metrics

Denote A as the original data chunk, A' as the reconstructed chunk after lossy compression and decompression, and n as the number of elements in A . The definitions of error metrics are as follows.

Table A1. Error metrics and their mathematical definitions.

Metric	Formula
Maximum absolute error	$\max_{\forall i \in [1, n]} A'_i - A_i $
Range-relative maximum absolute error	$\frac{\max_{\forall i \in [1, n]} A'_i - A_i }{\max_{\forall i \in [1, n]} A_i - \min_{\forall i \in [1, n]} A_i}$
Root mean square error (RMSE)	$\sqrt{\frac{\sum_{i=1}^n (A'_i - A_i)^2}{n}}$

Appendix B: Histograms

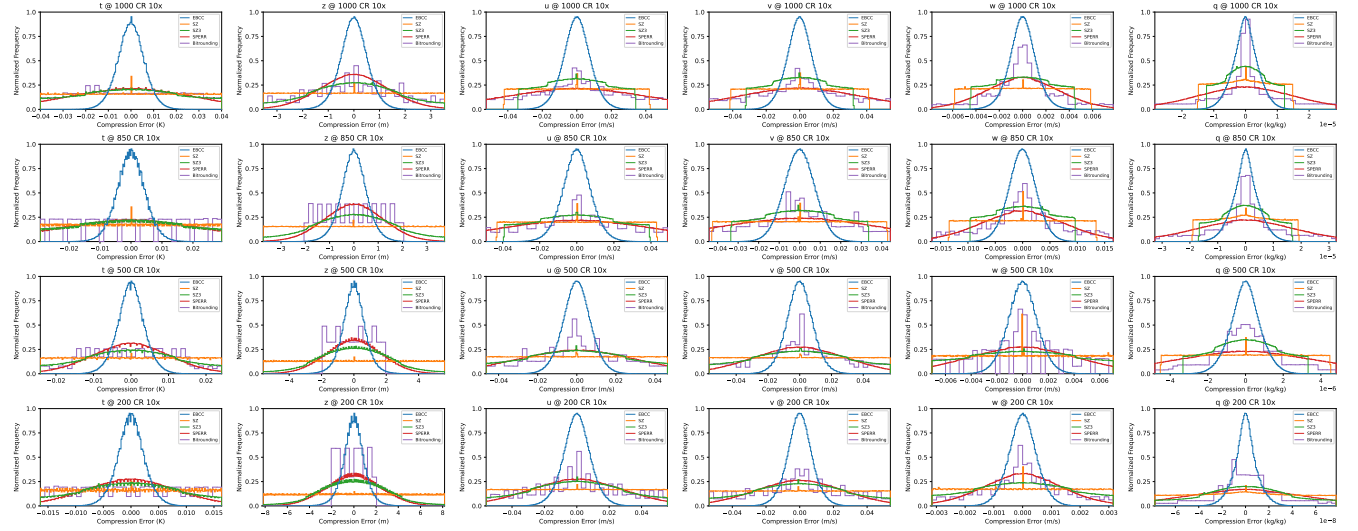


Figure B1. Histograms of compression errors for pressure-level variables at a compression ratio of $10\times$.

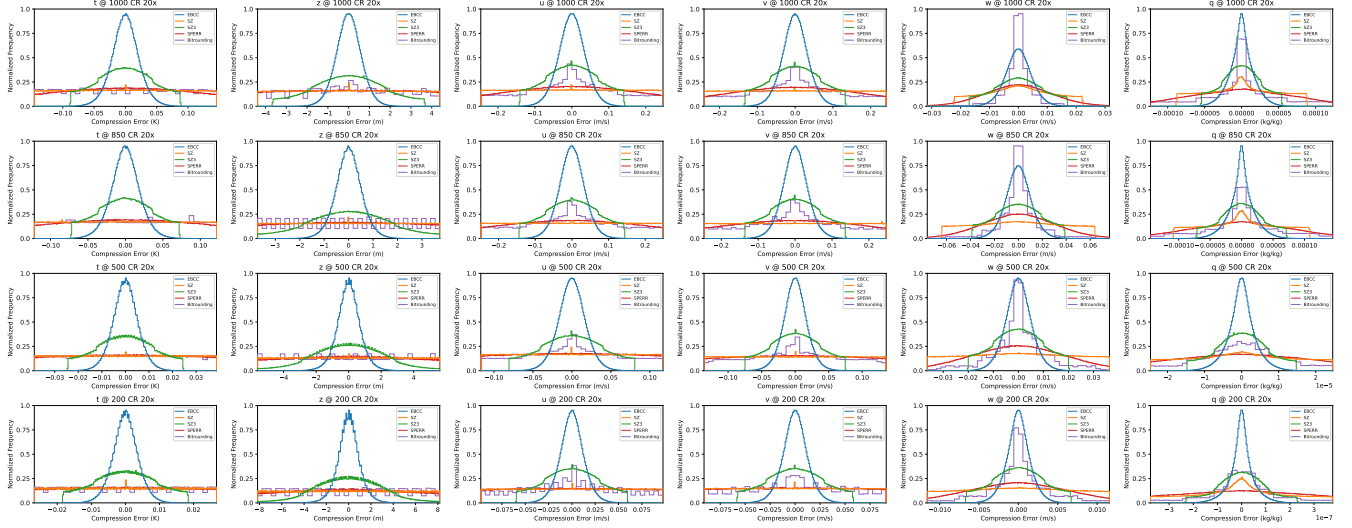


Figure B2. Histograms of compression errors for pressure-level variables at a compression ratio of 20 $\times$.

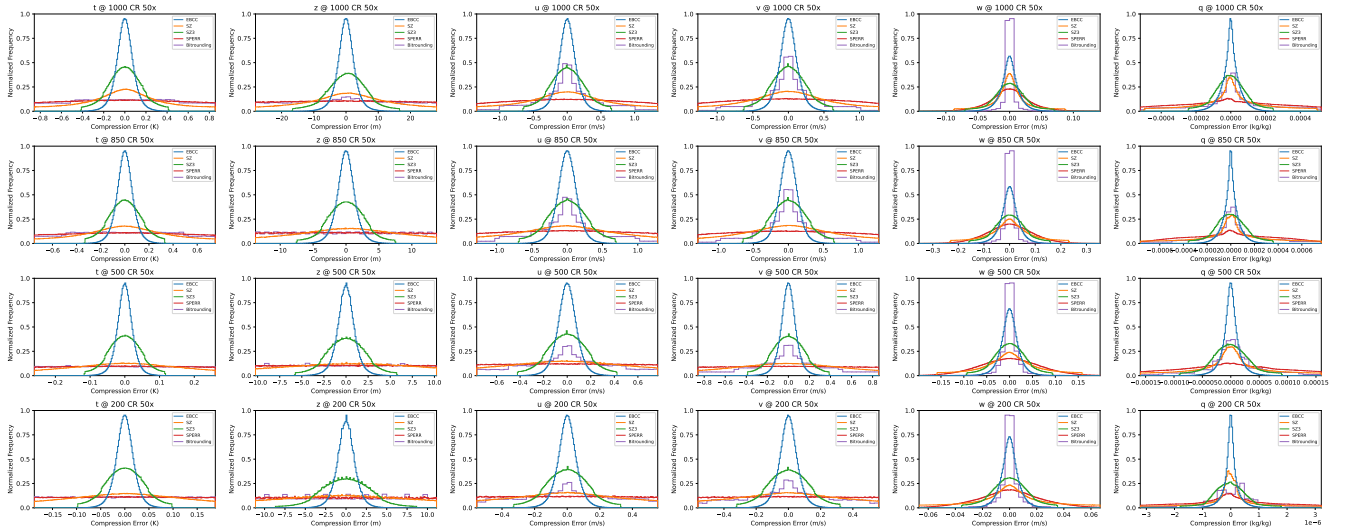


Figure B3. Histograms of compression errors for pressure-level variables at a compression ratio of 50 $\times$.

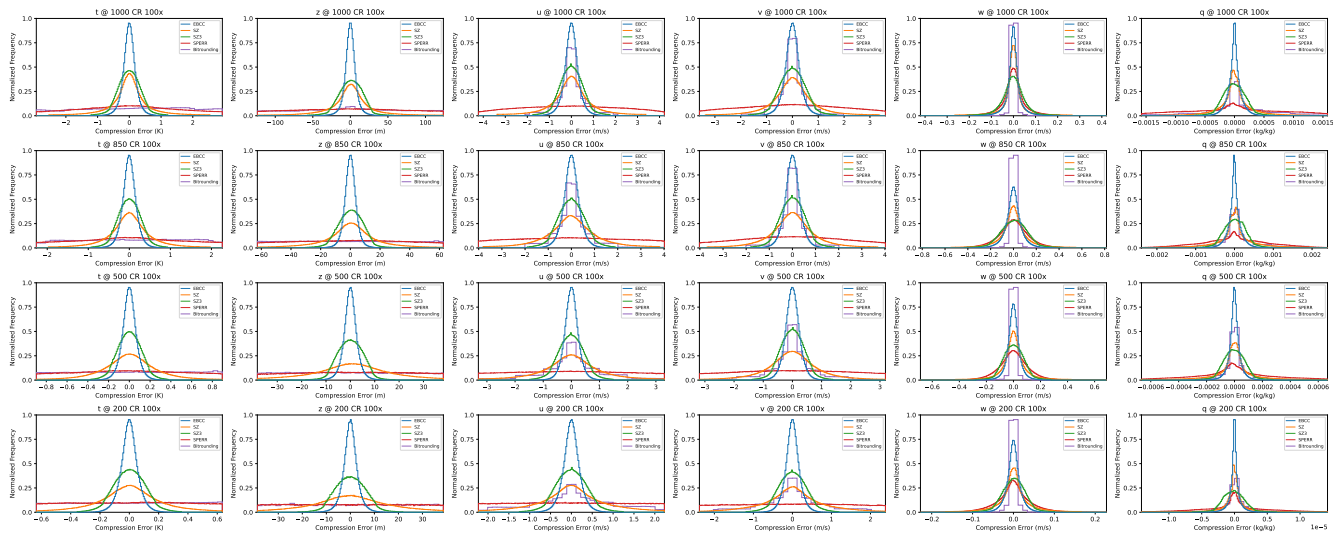


Figure B4. Histograms of compression errors for pressure-level variables at a compression ratio of 100 $\times$.

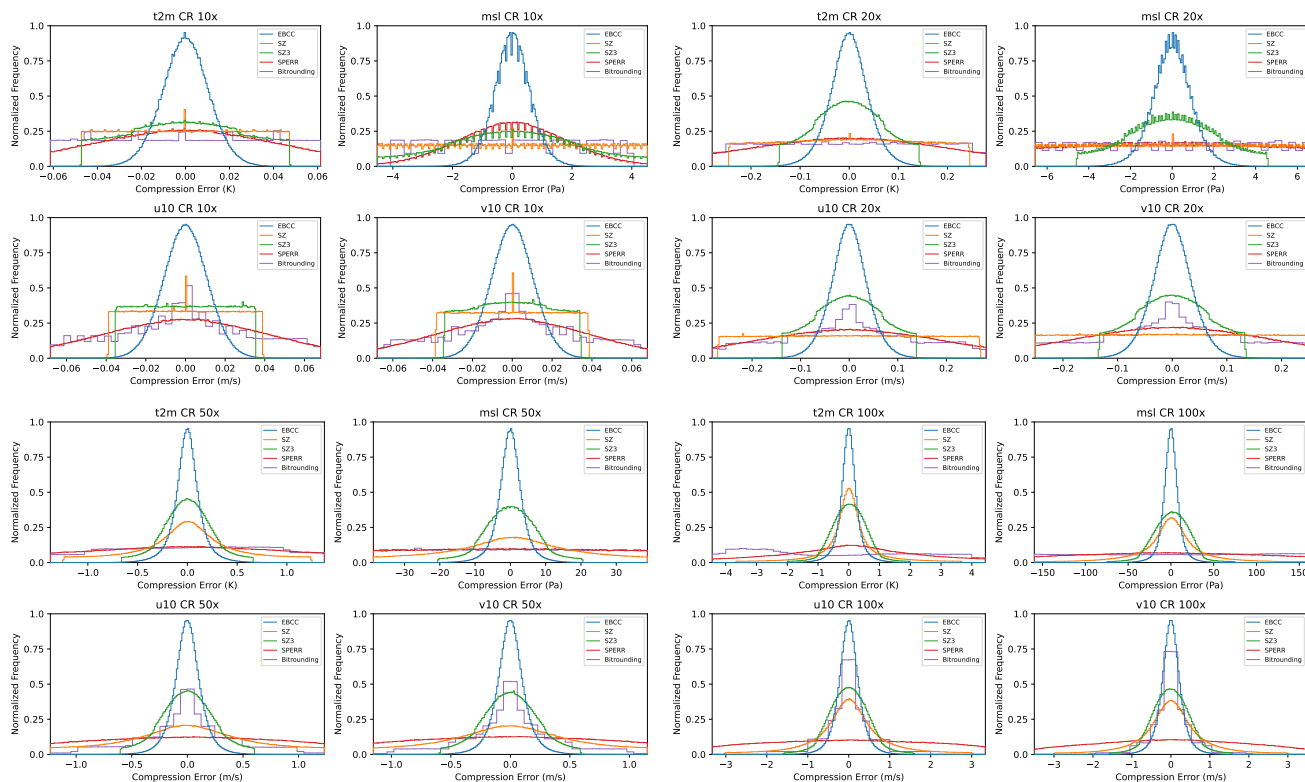


Figure B5. Histograms of compression errors for surface variables at compression ratios of 10 $\times$, 20 $\times$, 50 $\times$, and 100 $\times$.

Appendix C: Power Spectra

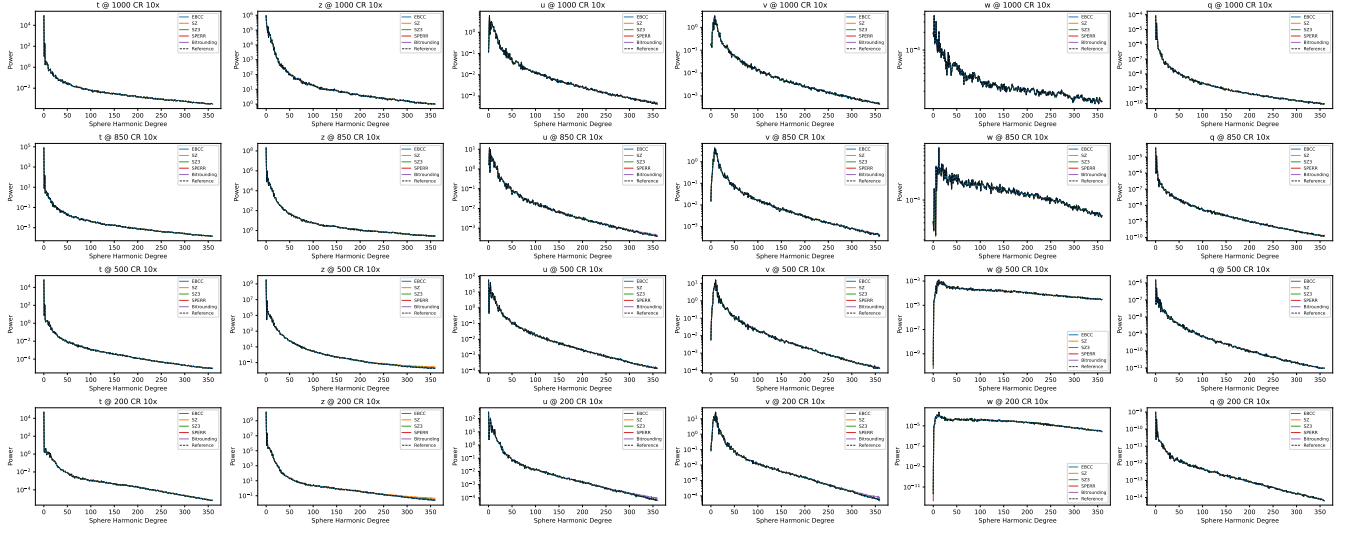


Figure C1. Spherical harmonics power spectra of compression errors for pressure-level variables at a compression ratio of 10 $\times$.

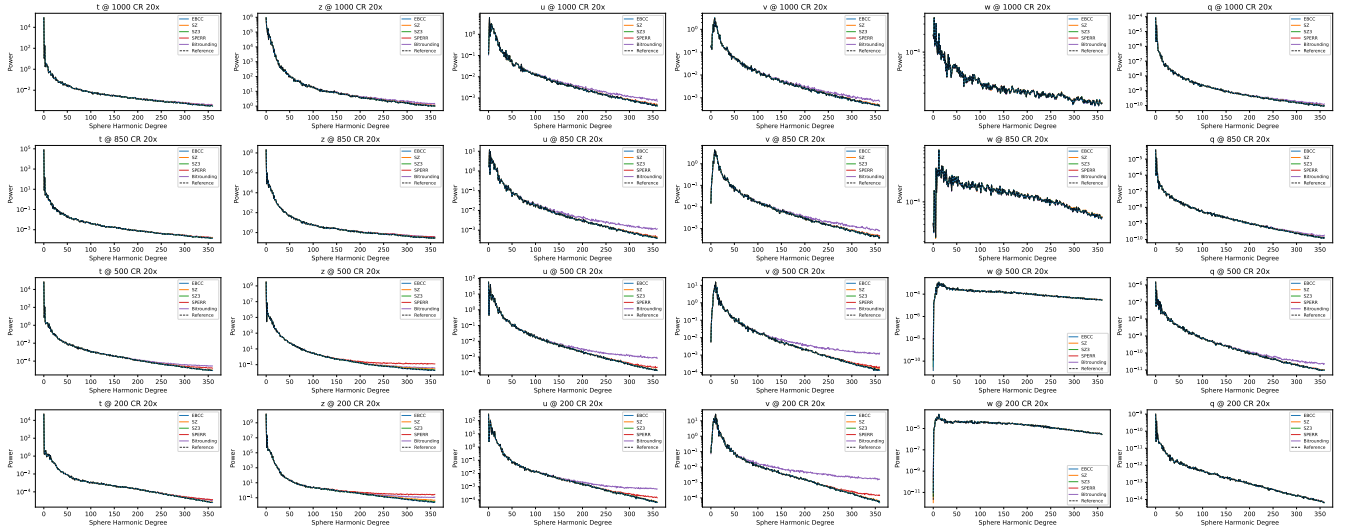


Figure C2. Spherical harmonics power spectra of compression errors for pressure-level variables at a compression ratio of 20 $\times$.

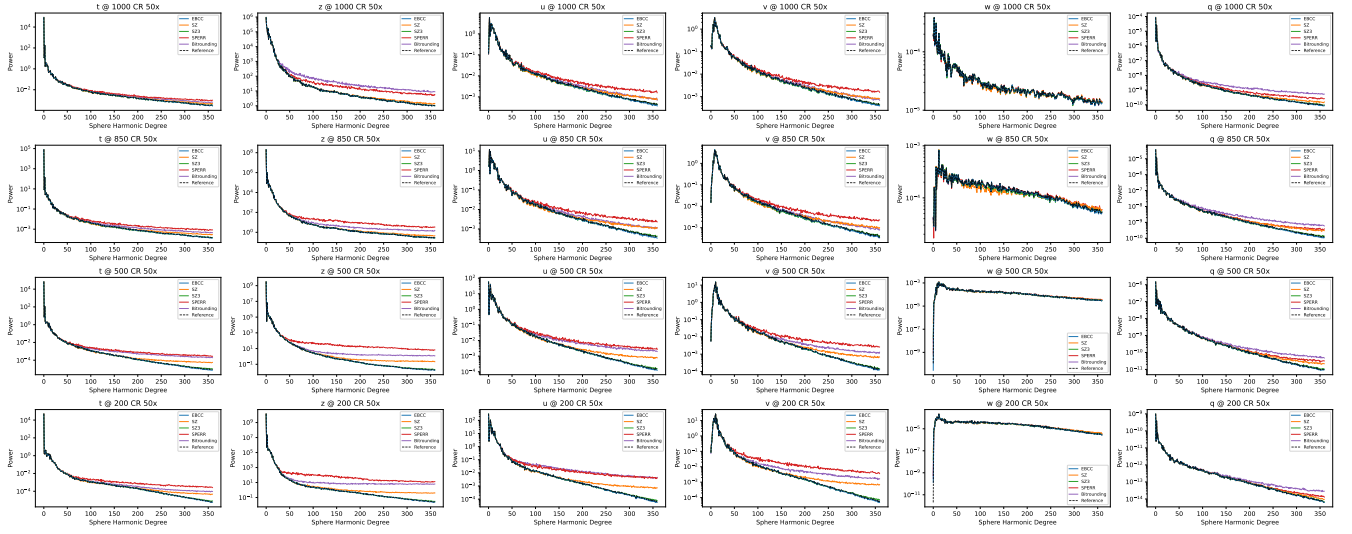


Figure C3. Spherical harmonics power spectra of compression errors for pressure-level variables at a compression ratio of 50 $\times$.

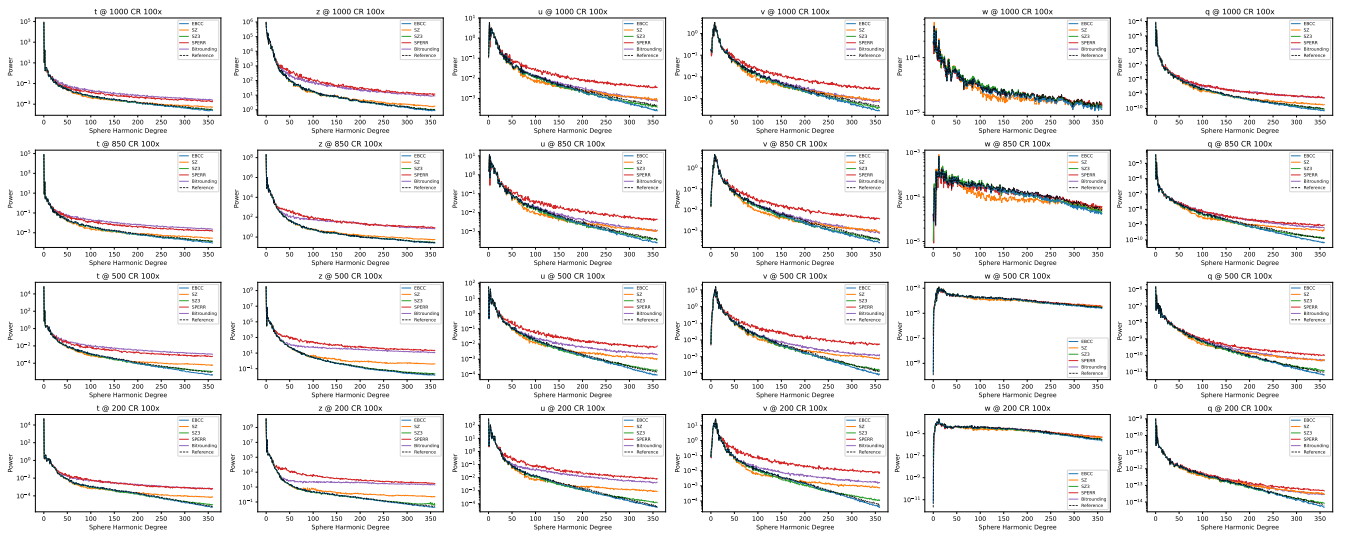


Figure C4. Spherical harmonics power spectra of compression errors for pressure-level variables at a compression ratio of 100 $\times$.

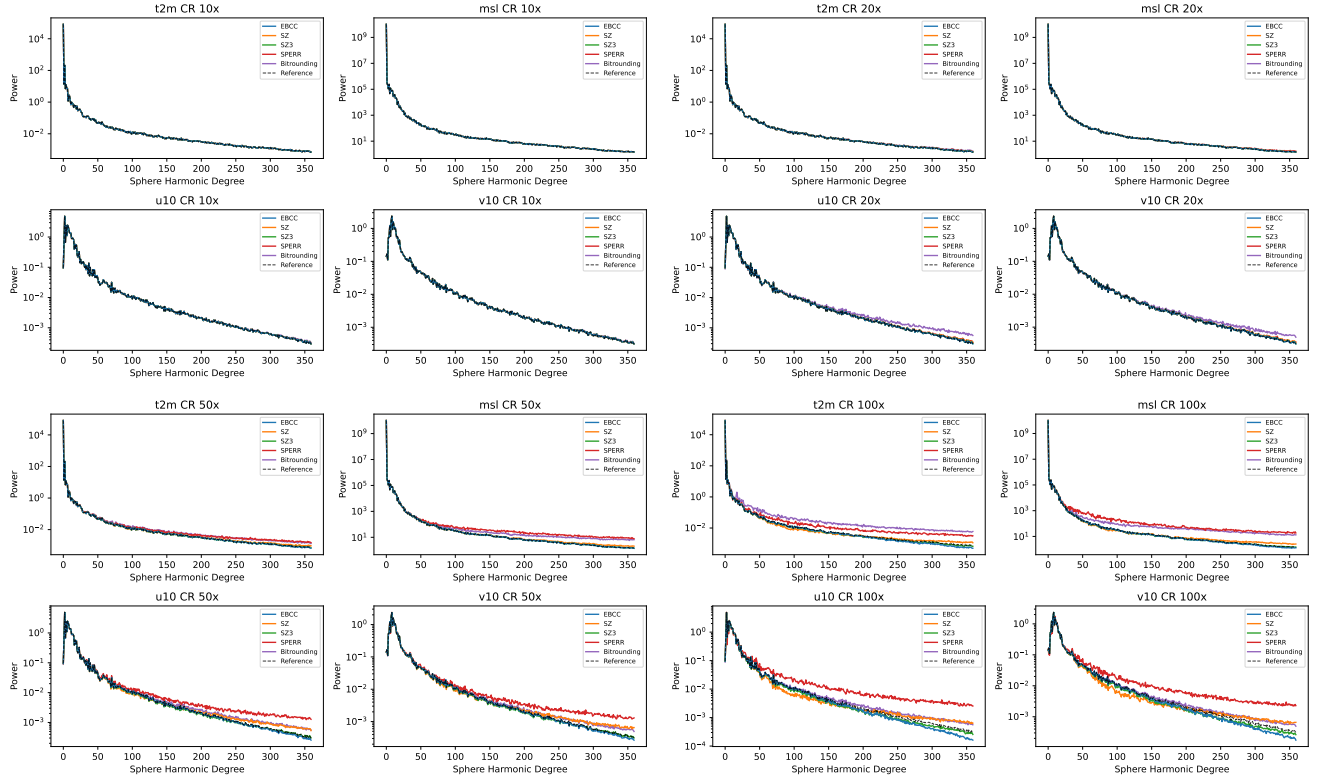


Figure C5. Spherical harmonics power spectra of compression errors for surface variables at compression ratios of 10 $\times$, 20 $\times$, 50 $\times$, and 100 $\times$.