Region-adaptive, Error-controlled Scientific Data Compression using Multilevel Decomposition

Qian Gong
Oak Ridge National Laboratory
Oak Ridge, TN, USA

Ben Whitney
Oak Ridge National Laboratory
Oak Ridge, TN, USA

Chengzhu Zhang
Lawrence Livermore National Laboratory
Livermore, CA, USA

Xin Liang
Missouri Science & Technology
Rolla, MO, USA

Anand Rangarajan
University of Florida
Gainesville, FL, USA

Jieyang Chen
Oak Ridge National Laboratory
Oak Ridge, TN, USA

Lipeng Wan
Oak Ridge National Laboratory
Oak Ridge, TN, USA

Paul Ullrich
UC Davis
Davis, CA, USA

Qing Liu
New Jersey Institute of Technology
Newark, NJ, USA

Robert Jacob
Argonne National Laboratory
Lemont, IL, USA

Sanjay Ranka
University of Florida
Gainesville, FL, USA

Scott Klasky
Oak Ridge National Laboratory
Oak Ridge, TN, USA

ABSTRACT

The increase of computer processing speed is significantly outpacing improvements in network and storage bandwidth, leading to the big data challenge in modern science, where scientific applications can quickly generate much more data than that can be transferred and stored. As a result, big scientific data must be reduced by a few orders of magnitude while the accuracy of the reduced data needs to be guaranteed for further scientific explorations. Moreover, scientists are often interested in some specific spatial/temporal regions in their data, where higher accuracy is required. The locations of the regions requiring high accuracy can sometimes be prescribed based on application knowledge, while other times they must be estimated based on general spatial/temporal variation. In this paper, we develop a novel multilevel approach which allows users to impose region-wise compression error bounds. Our method utilizes the byproduct of a multilevel compressor to detect regions where details are rich and we provide the theoretical underpinning for region-wise error control. With spatially varying precision preservation, our approach can achieve significantly higher compression ratios than single-error bounded compression approaches and control errors in the regions of interest.

We conduct the evaluations on two climate use cases – one targeting small-scale, node features and the other focusing on long, areal features. For both use cases, the locations of the features were unknown ahead of the compression. By selecting approximately 16% of the data based on multi-scale spatial variations and compressing those regions with smaller error tolerances than the rest, our approach improves the accuracy of post-analysis by approximately 2× compared to single-error-bounded compression at the same compression ratio. Using the same error bound for the region of interest, our approach can achieve an increase of more than 50% in overall compression ratio.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; • **Mathematics of computing → Mathematical analysis**.

KEYWORDS

Region-adaptive Lossy Compression, Error Control, Climate Data Compression

ACM Reference Format:
Qian Gong, Ben Whitney, Chengzhu Zhang, Xin Liang, Anand Rangarajan, Jieyang Chen, Lipeng Wan, Paul Ullrich, Qing Liu, Robert Jacob, Sanjay Ranka, and Scott Klasky. 2022. Region-adaptive, Error-controlled Scientific Data Compression using Multilevel Decomposition. In *34th International Conference on Scientific and Statistical Database Management (SSDBM 2022), July 6–8, 2022, Copenhagen, Denmark.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3538712.3538717

1 INTRODUCTION

Compression serves an important role in data transmission, archiving, analysis, and visualization. This is especially true for data generated by large-scale scientific simulations as the number of cores in supercomputers continues to increase more rapidly than storage and network bandwidths [34]. Lossless compression [9, 12, 14, 15, 23], though desirable, only achieves limited compression ratios due to random mantissas in the floating-point scientific representation [26]. Error-controlled lossy compressors such as MGARD [4], SZ [16], ZFP [29], and FPZIP [30] have recently received a great

SSDBM 2022, July 6–8, 2022, Copenhagen, Denmark

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3538712.3538717

deal of attention as they provide high compression ratios along with insights on the extent to which the data has been modified. The goal of compression is to control not only errors in the raw data but also errors in derived quantities of interest (QoI). Data points collected from experiments or simulation do not always impose equal contribution to the outcome of analysis pipeline. For example, it's critically important for the gyrokinetic particle simulation code XGC [24] to preserve the particle interaction around the edge of tokamak plasmas. These types of requirements motivate the design of a compression strategy with region-adaptive error bounds to better preserve data in the regions of interest (RoI) and/or further reduce the storage cost by compressing the less important regions more significantly. The goal of this paper is to introduce an adaptive lossy compressor that can detect critical regions and apply various error bounds on data at different regions. Unlike compressors that are customized to preserve certain specific features in specific data types [10, 22, 27, 33, 35, 40], our technique generally focuses on regions where the data are more turbulent than the average level. This choice has been made due to the fact that the features are variously defined for different tasks and applications. For example, climate events analysis code [38] may require compressors to better preserve topological features, whereas the analysis code for wind turbine simulation may focus on quantities near airfoil blades [17]. By focusing on regions where features are *likely* to be present rather than features themselves, we provide a generalized compression pipeline suitable for data which will be used for unspecified scientific tasks.

Our adaptive compression pipeline automatically determines RoIs utilizing the decomposed coefficients produced by a multilevel lossy compressor, MGARD, during its execution. Drawn from the multigrid linear solver and finite element methods, MGARD defines data in a nested grid structure. Data are decomposed into a collection of coefficients, which are then quantized to meet user-prescribed error bounds through rigorous mathematical analysis [28]. MGARD uses a multilinear interpolation operator with L^2 projection for data decomposition and implements the procedure iteratively in a set of nested grids. The coefficients output from the decomposition capture spatial variations in different scales defined by the grid resolutions. Next, inspired by the Adaptive Mesh Refinement (AMR) [7] used in numerical analysis, we partition the coefficients into grid cells, selecting and recursively refining the cells whose coefficients' magnitudes are considerably large. Then, for each coefficient in the refined meshes, we draw a blob around it with the radius determined by the level of the selected coefficient in the pyramid of grids. These blobs are then merged together, transforming them into large, continuous RoIs.

In general, the shapes of detected RoIs are irregular and cannot be easily segmented using bounding box approaches [6]. In addition, data segmentation adds additional storage cost as the coordinates of RoIs and non-RoIs must be recorded so they can be put back together after reconstruction. Our approach implements point-wise error-control and encodes error bounds in the quantized coefficients, so that RoIs and non-RoIs can be compressed and decompressed together. This strategy, however, imposes challenges in managing errors. In particular, due to the L^2 projection used in the multilevel recomposition, error at one data-point propagates to the entire coordinate space. Therefore, a simple strategy that varies the quantization error bounds per region/data-point will not guarantee that data in different regions are reconstructed to the prescribed accuracy. We present mathematical analysis for the error propagation during multilevel recomposition and provide a strategy to ensure that the accumulated errors in the selected regions do not exceed the prescribed bounds.

In summary, the main contributions of this paper are:

- A region-adaptive compression method which can compress data using regional varying error bounds. The compression does not rely on region segmentation and the decompression does not require RoI masks.
- Mathematical studies on the cross-region error propagation during the multilevel recomposition.
- A method to detect candidate critical regions using the byproduct of compression and mesh refinement.
- Evaluations using two climate use cases show improved data compression ratios and lower errors in post-analysis compared to single-error-bounded approaches.

2 PROBLEM AND RELATED WORKS

2.1 Region-of-Interest based compression

Conventional compressors reduce every data point using a single error bound. However, important information does not scatter the entire space uniformly for most cases of scientific data. The motivation behind RoI-based compression counts on the non-uniform distribution of information. The goal is to maintain certain key regions with high quality and compress other regions with lower quality, so that larger compression ratios can be achieved and meanwhile task-interested information are preserved.

RoI-based compressors can be divided into static and dynamic categories depending on the availability of prior knowledge. Static approaches set RoIs at constant locations and assume their locations remain unchanged in the same scene across different timesteps. For example, tokamak plasma studies are particularly interested in particles hitting the divertor and magnetic X-point regions [18]. Dynamic RoI-based approaches, in comparison, require region detection before compression. For example, Liang et al. [27] derive local error bounds and leverage them for error-controlled compression to retain critical points in 2D/3D vector fields based on the underlying feature extraction algorithm. Xu et al. [40] create a hierarchical perception model to track face features and a weight-based rate-quantization scheme to improve the visual quality of the RoI. Machine learning techniques have also been applied to RoI-based compression. Cai et al. [10] transform input images into multiscale representations and train encoder and decoder networks to predict RoIs in a supervised manner. Song et al. [35] build a spatial feature transform network to produce task-aware feature maps, and combine it with a variable-rate bit allocation algorithm to compress images with spatially-varying quality.

In general, RoI-based compressors need to store the coordinates of features and background so they can be put back together after reconstruction. Commonly used approaches for coordinate registration include pixel-wise maps [27, 40] and bounding boxes [1, 6, 32]. Generally speaking, pixel-wise maps lead to higher efficiency when the RoI shapes are irregular while bounding boxes could be better choices when RoI points are heavily clustered.

Compressors also use hierarchical data formats to achieve adaptive compression quality in different regions. Octrees and AMR are the most-used formats for hierarchical data representation. For instance, Skylar et al. [39] leverage an octree-based structure to represent data in hierarchical super resolution using neural networks. Bhatia et al. [8] introduce an AMR-based data structure, namely Adaptive Multilinear Meshes (AMM), to allow for incremental updates in both spatial resolution and numerical precision using the basis functions of tensor products of linear B-spline wavelets. However, these existing hierarchical data representation-based compression techniques implement spatial refinement primarily on raw data. In comparison, our critical region detection method implements mesh refinement on the compressor-decomposed coefficients, which is a multiscale vector capturing local data variation.

2.2 Error-bounded lossy compression

Error-controlled compression has been proposed to reduce scientific data while providing quantifiable error bounds toward user's requirements. These compressors can be classified as prediction-based or transform-based in general, depending on how they decorrelate the data. Prediction-based compressors, such as ISABELA [25], SZ [36, 42], and FPZIP [30], rely on various predictors to exploit the inherent correlation in data. In contrast, transform-based compressors such as ZFP [29] and MGARD [2] decorrelate data via specific transforms, where data are transformed into formats which are amenable to compression. These transformed data are then quantized and encoded into reduced representations.

Error-controlled lossy compressors have proven to be useful for applications that are strict on how much error can be tolerated [11]. Nevertheless, similar to most lossless compressors, they a use single error bound, which may lead to sub-optimal compression ratios when the requested accuracy is defined on QoIs. In this work, we expand the multilevel compressor MGARD to perform region-adaptive, multi-error-bounded compression. We focus on MGARD because its multilevel decomposition captures multi-scale features, which can be used for identifying regions where details are rich, and can be applied to non-uniform and unstructured grids as well [5]. This is a significant departure from the current design of MGARD, which is RoI-agnostic.

3 BACKGROUND OF MULTILEVEL LOSSY COMPRESSION

Our work on RoI detection and region-wise error control is built upon the MGARD compression algorithm. In this section, we describe MGARD's compression, decompression, and error control mechanisms. This description focuses on the decomposition stage, as we will use its byproduct for RoI detection. The decompression is an invertible operation.

MGARD is a multilevel decomposition-based compressor. The goal of the decomposition is to transform input data into coefficients amenable to compression. MGARD interprets an input array u in d dimensions as the values taken by a continuous function u on a grid $\mathcal{N}_L$ having the same grid structure as u. This grid can be downsampled into a hierarchy of subgrids $\mathcal{N}_{L-1}, \ldots, \mathcal{N}_0$. The ratio between the number of nodes on $\mathcal{N}_{l-1}$ and $\mathcal{N}_l$ is approximately

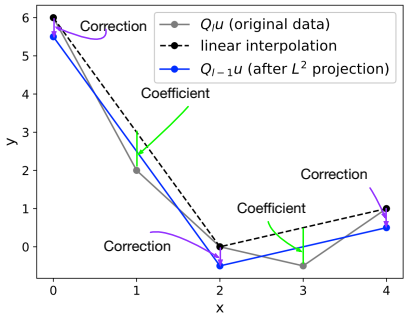


Figure 1: MGARD implements data reduction based on a multilevel decomposition. The transform coefficients in MGARD are residues from multilinear interpolation of L^2 projections.

$1/2^d$. We denote the node set $\mathcal{N}_l \setminus \mathcal{N}_{l-1}$ by $\mathcal{N}_l^*$. MGARD's decomposition starts from level L, the finest grid, and stops at level 0, the coarsest grid. The decomposition is achieved using two operations: L^2 projection, denoted Q_l for level l, and multilinear interpolation, denoted Π_l for level l. An example is illustrated in Fig. 1, which was originally presented in [13]. As shown in the figure, the multilevel coefficients on $\mathcal{N}_l^*$ are obtained by subtracting from the projection $Q_l u$ to the current grid its interpolation $\Pi_{l-1} Q_l u$ on the next coarser grid $\mathcal{N}_{l-1}$. These coefficient are then projected to $\mathcal{N}_{l-1}$ to obtain a *correction* used to transform $\Pi_{l-1} Q_l u$ to $Q_{l-1} u$. $Q_{l-1} u$ is then used to compute the coefficients at the next coarser level $\mathcal{N}_{l-1}^*$, and the procedure repeats.

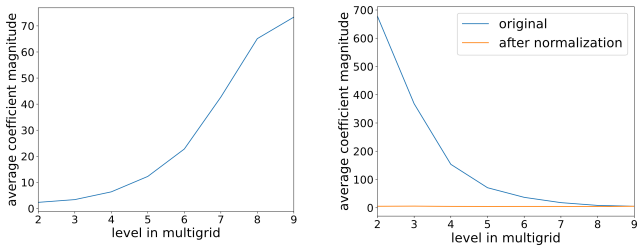
We summarize the decomposition procedure as follows:

(1) The decomposition starts with $Q_L u = u$.
(2) Compute the piecewise linear interpolant $\Pi_{l-1} Q_l u$ and subtract it from $Q_l u$ to get the multilevel coefficients u_mc at level l. These coefficients encode $(I - \Pi_{l-1}) Q_l u$.
(3) Project the multilevel coefficients to the next coarser level to obtain the *correction* and add it to the interpolant $\Pi_{l-1} Q_l u$ to obtain $Q_{l-1} u$, the L^2 projection to the next coarser level.
(4) Repeat the above process until $l = 0$.

The original input u is transformed into a set of multilevel coefficients u_mc after decomposition. Next, given a user-prescribed L^2 error bound τ, MGARD quantizes each coefficient u_mc[x] into ũ_mc[x] such that

$$\|u - \tilde{u}\|_{L^2} \leq \left(\sum_{l=0}^{L} \sum_{x \in \mathcal{N}_l^*} \mathrm{vol}(x) \, |\mathtt{u\_mc}[x] - \tilde{\mathtt{u}}\_\mathtt{mc}[x]|^2 \right)^{1/2} \leq \tau, \quad (1)$$

where $\tilde{u}$ is a reduced representation of u after quantization and $\mathrm{vol}(x)$ is the volume of an element centered at x measured in the corresponding grid. For the mathematical details on how to obtain the relation between $\|u - \tilde{u}\|_{L^2}$ and $|\mathtt{u\_mc}[x] - \tilde{\mathtt{u}}\_\mathtt{mc}[x]|$, please refer to [2, 3]. MGARD also provides error control on L^∞ [4] and quantities derived from u through any bounded linear operator [3]. We omit the review on these two as they are not directly related to the work in this paper.



(a) Magnitude of coefficients of random data.

(b) Magnitude of coefficients of real data.

Figure 2: The multilevel coefficients must be normalized before mesh refinement to take out the level-wise magnitude variation caused by spatial correlation (2b) and lack of thereof (2a). Level 9 is the finest grid and level 2 is the second coarsest grid.

4 THEORETICAL FOUNDATION

In this section, we describe how to perform region-wise error control and critical region detection during compression. We explore how a point-wise error propagates during the multilevel recomposition and we propose error control methods based on buffer zones and linear quantization.

4.1 Critical Region Detection

As discussed in Sec. 3, MGARD uses a hierarchy of nested grids to decompose data u into a collection of multilevel coefficients u_mc. Each multilevel coefficient u_mc[x] captures data variations at a particular location and scale. In this subsection, we explore how to make use of the multilevel coefficients for critical region detection.

4.1.1 Multilevel coefficient preparation. Due to differing scales across the grid hierarchy, the magnitudes of coefficients at coarse and fine levels cannot be directly compared. To understand the relationship between the level in the hierarchy and the magnitude of the coefficients, we compute the multilevel coefficients for a pointwise randomly generated 2D dataset and plot the average magnitude of coefficients by level in Fig. 2a. As seen in the figure, the average magnitude of the coefficients increases from coarse to fine levels. This is caused by high-frequency oscillations found in this random dataset.

Next, we study the multilevel coefficients of real data. The blue line in Fig. 2b shows the multilevel coefficients of a 2D sea level pressure (PSL) variable taken from an E3SM [19] climate simulation dataset. Opposite to the randomly generated data, the magnitude of level-wised coefficients of real data decreases as the level becomes finer. This is because the real data is spatially correlated.This measure of smoothness results in increasingly small multilevel coefficients; see [3] for details.

To avoid the bias caused by coefficients at fine levels in weakly correlated data and coefficients at coarse levels in strongly correlated data, we perform data normalization. Specifically, each multilevel coefficient at level l will be normalized by $\texttt{u\_mc}'[x] = \texttt{u\_mc}[x]/\alpha^{L-l}$ before mesh refinement, where α is a constant calculated by curve fitting the multilevel coefficients decomposed from a sample dataset. For the example of E3SM climate simulation data, we choose $\alpha = 2$. As shown by the orange line in Fig. 2b, the magnitudes of coefficients at coarse and fine levels are normalized to the same scale (though a large coefficient at coarser level still mean data variations in wider expansion).

4.1.2 Mesh refinement on multilevel coefficients. After normalization, we propose a mesh refinement approach to identify regions where large-value coefficients are clustered. These regions are places where features of interest are likely to be found, and they will be compressed with small errors. Our mesh refinement-based region detection algorithm is summarized with the following steps:

(1) Begin with the entire full resolution space S.
(2) Partition the original space S into a grid D with k the starting grid spacing.
(3) Aggregate the coefficients in each grid cell and select the grid cells where the aggregated magnitude is above the p^{th} percentile.
(4) Check to see if the stop criteria is met (details in following paragraphs).
    - If not, partition each selected grid cell with k' the new grid spacing and recurse step (3).
    - If yes, return the selected grid cells from the final level of refinement.

Our method is similar to adaptive mesh refinement in that we keep partitioning the mesh grid if the data inside is not sufficiently *smooth*. Fig. 3 shows the mesh grids after 3 levels of refinement. We have two stopping criteria for any grid cell to not be subdivided further. First, if the mesh interval k is less or equal to a user-specified hyperparameter k_{min} or the original data resolution. Second, if the aggregated magnitude of coefficients in the cell is smaller than a threshold A_c, which is derived from the global distribution of coefficient magnitudes.

In our implementation, the starting grid space k, threshold k_{min}, A_c used for stopping refinement, the downscaling factor (i.e., k/k') used at each refinement step, and the threshold p^{th} used for selecting coefficient after are hyperparameters determined by the features in post-processing. In general, k_{min} sets the scale of features to be captured; A_c indicates the degree of variations in selected regions comparing to global histogram; k, k/k' and p^{th} together determine the shape of features, i.e., whether the features are clustered in small-scale, high turbulent regions or expanded in wide, smoothly varying space.

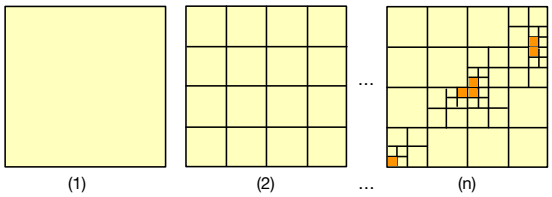


Figure 3: An example showing how to selected the most significant coefficients (orange cells) by recursively refining mesh grids based on the encapsulated coefficients.

The multilevel coefficient at node x_l represents the changes between the value $Q_l u$ takes at x_l and the nodes at a distance h_l to the left and right of x_l (in 1D), where $h_l = 2^{L-l}$. Thus, for each coefficient selected by the mesh refinement, we add the data-points which fall under a radius of h_l in the multidimensional space from the selected coefficient. The final output RoIs are larger, continuous regions formed by merging these blobs.

4.2 Error Propagation in Multilevel Recomposition

The quantization error between u_mc and ũ_mc induces a compression error between u and $\tilde{u}$. Because the recomposition procedure is linear, the compression error is obtained by recomposing the quantization error:

$$u - \tilde{u} = \sum_{l=0}^{L} \sum_{x \in \mathcal{N}_l^*} (\texttt{u\_mc[}x\texttt{]} - \tilde{\texttt{u}}\texttt{\_mc[}x\texttt{]})(I - Q_{l-1})\phi_l(\,\cdot\,; x).$$

Here $\{\phi_l(\,\cdot\,; x) : x \in \mathcal{N}_l\}$ is the Lagrange basis for the space of piecewise multilinear functions with knots $\mathcal{N}_l$. $\phi_l(\,\cdot\,; x)$ is the basis function that is 1 at x and 0 at every other node in $\mathcal{N}_l$. We can bound the magnitude of the compression error at some point y in the RoI as follows:

$$|(u - \tilde{u})(y)| \le \sum_{l=0}^{L} \sum_{x \in \mathcal{N}_l^*} \begin{array}{l} \left|\texttt{u\_mc[}x\texttt{]} - \tilde{\texttt{u}}\texttt{\_mc[}x\texttt{]}\right| \\ \quad \times \left|(I - Q_{l-1})\phi_l(\,\cdot\,; x)(y)\right|. \end{array} \tag{2}$$

$(I - Q_{l-1})\phi_l(\,\cdot\,; x)$ is the error incurred by the quantization of u_mc[x]. See Fig. 4 for an illustration. It is an oscillatory function which decays with the distance from x. To bound the compression error at y, we require a bound on the magnitude of each error $(I - Q_{l-1})\phi_l(\,\cdot\,; x)$ at y. The page limit precludes the derivation of this bound in this article. The proof will appear in a forthcoming article [21], and we will give a brief summary of the main result here.

Consider first the 1D case. Define a distance function d_{l-1} by $d_{l-1}(x, y) = |x - y|/h_{l-1}$. Take $x \in \mathcal{N}_l^*$ and $y \in \mathcal{N}_{l-1}$. Let a be the grid endpoint closest to x, and let b be the grid endpoint closest to y. It is shown in [4, p. A1300] that, with $\Lambda_1 = 2 + \sqrt{3}$ and C some constant,

$$\left|Q_{l-1}\phi_l(\,\cdot\,; x)(y)\right| = C \frac{\begin{array}{c}\left[\Lambda_1^{d_{l-1}(a,x)} - \Lambda_1^{-d_{l-1}(a,x)}\right] \\ \times \left[\Lambda_1^{d_{l-1}(y,b)} + \Lambda_1^{-d_{l-1}(y,b)}\right]\end{array}}{\left[\Lambda_1^{d_{l-1}(a,b)} - \Lambda_1^{-d_{l-1}(a,b)}\right]}.$$

Observe that $d_{l-1}(a, x) + d_{l-1}(x, y) + d_{l-1}(y, b) = d_{l-1}(a, b)$. In the typical case, where $d_{l-1}(a, b)$ is large and neither $d_{l-1}(a, x)$ nor $d_{l-1}(y, b)$ is too small, we have

$$\left|Q_{l-1}\phi_l(\,\cdot\,; x)(y)\right| \approx C \frac{\Lambda_1^{d_{l-1}(a,x)} \Lambda_1^{d_{l-1}(y,b)}}{\Lambda_1^{d_{l-1}(a,b)}} = C\Lambda_1^{-d_{l-1}(x,y)}$$

This approximation can be also be shown to hold when not in the 'typical' case (when any of $d_{l-1}(a, b)$, $d_{l-1}(a, x)$, or $d_{l-1}(y, b)$ is small), and in 2D (with a different constant, and with d_{l-1} defined as in Claim 1). Furthermore, it can be shown that this decay rate also holds for the values taken by $(I - Q_{l-1})\phi_l(\,\cdot\,; x)$, including at points in between the nodes of $\mathcal{N}_{l-1}$. See [21] for details. The following bound is the result.

Claim 1. *Let $x \in \mathcal{N}_l^*$. For any y in the domain,*

$$\left|(I - Q_{l-1})\phi_l(\,\cdot\,; x)(y)\right| \le C(2 + \sqrt{3})^{-d_{l-1}(x,y)}$$

where $d_{l-1}(x, y) = |x - y|/h_{l-1}$ in 1D and $d_{l-1}(x, y) = |x_1 - y_1|/h_{l-1} + |x_2 - y_2|/h_{l-1}$ in 2D.

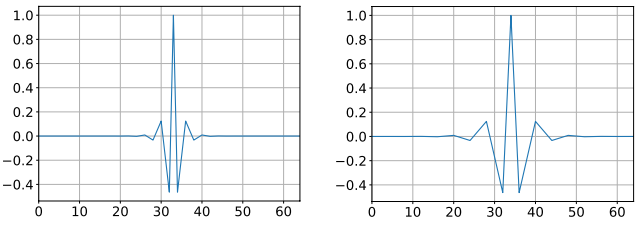
Combining Claim 1 with Equation (2), we find that

$$|(u - \tilde{u})(y)| \le C \sum_{l=0}^{L} \sum_{x \in \mathcal{N}_l^*} \left|\texttt{u\_mc[}x\texttt{]} - \tilde{\texttt{u}}\texttt{\_mc[}x\texttt{]}\right| (2 + \sqrt{3})^{-d_l(x,y)}$$

in 1- and 2D. This result has three main implications for RoI error control:

(1) The compression error induced by quantizing a multilevel coefficient at a node x decays exponentially in the distance from x. As a result, the compression error in a region of interest is chiefly attributable to the quantization of the coefficients in the region itself and in a thin surrounding buffer zone.
(2) The rate of error decay is scaled by the distance between the nodes of the level. As a result, the physical width of the buffer zone scales by the same distance, and the number of nodes included in it is essentially constant from level to level.
(3) The particular notion of distance that is a scaled Manhattan metric. As a result, the buffer zone is diamond-shaped.

The pointwise decay rate give in Claim 1 does not immediately yield an L^2 error estimate, and it also does not account for the possibility of cancellation between the components of the compression error, so in the next subsection we augment this theoretical result with an empirical investigation of the bounds used for region-wise L^2 error control.



(a) Induced error centered on a node at level L.

(b) Induced error centered on a node at a level coarser than L.

Figure 4: The compression error induced by a quantization error at a node on a coarser level is propagated to a wider region by the multilevel recomposition procedure. The error decays exponentially at a rate proportional to the level's internode spacing and flips sign from node to node.

4.3 Region-wise Error Control

In the previous section, we show the error is propagated and decayed in a modified distance $d_l(x, y)$ during the multilevel recomposition. In this section, we describe the buffer zone and the error bounds used for region-wise error control.

4.3.1 Buffer Zone. The compression error in the RoI after recomposition is attributable to errors incurred quantizing coefficients both inside and outside the RoI (with outside errors propagating inward). We focus on L^2 error in this paper, and so errors accumulated from coefficients inside the RoI can be preserved using MGARD's error control Equation (1). Next, motivated by the exponential rate of error decay in Claim 1, we add a buffer zone to prevent an excessive amount of error propagating from coefficients outside. By preserving the data in a thin region surrounding the RoI with high accuracy, we guarantee the quality of the data inside the RoI and in the meantime can use a larger error bound to compress data outside the RoI.

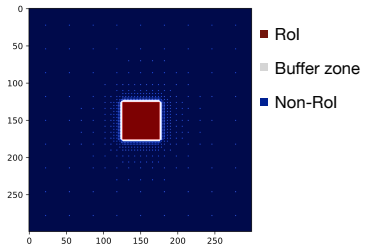


Figure 5: Our region-wise error control is implemented using a buffer zone. A buffer zone (white) consists of nodes which fall within a certain Manhattan distance of the RoI (red). The distance varies for nodes at different grid levels.

According to Claim 1, the point-wise compression error decays exponentially in $d_l(x, y)$, the distance between x and y scaled by the grid spacing at the associated level. Accordingly, the buffer zone is a discontinuous region filled by nodes at different grid levels. We define the radius of the buffer zone, R_{bz}, as the number of grid intervals at level $l-1$ when the quantization error is incurred at a node at level l. Fig. 5 shows an example of the buffer zone nodes with $R_{bz} = 2$ for a square RoI in 2D space. In real cases, our RoI is composed by numerous smaller, square RoI-blobs, which are expanded from coefficients selected by mesh refinement. For each coefficient x at an RoI-blob's edge, we check whether it overlaps with another RoI-blob. If not, we use Algorithm 1 to compute a buffer zone around it. Because the number of coefficients at coarser levels is few while their error propagation steps are wide (as illustrated in Fig. 4), we accelerate the search by first adding all coefficients below level k into the buffer zone. The search can then be limited into a smaller region with a maximum distance $R_{rz}h_{k-1}$.

We apply the same error bound, τ_0, used for compressing the RoI to compress the buffer zone, and apply a larger error bound, τ_1, to compress non-RoI regions. Supposing a quantization error q is incurred at node x outside the RoI, a buffer zone of radius R_{rz} ensures that the compression error propagated from x to a node y at the edge of the RoI is at most $Cq(2+\sqrt{3})^{-R_{rz}}$, where C is some constant. Increasing R_{rz} allows to use a larger error bound for compressing the non-RoI regions, but in the meantime more data will be compressed with the low error bound τ_0 because of the increased size of the buffer zone.

Algorithm 1 Create level-wise buffer zone.

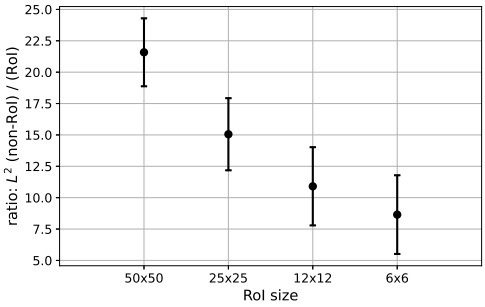
Input: coefficient x on the boundary of RoI; lookup table u_l to find the level of a coefficient in grid; maximum level L; maximum searched level k; radius of buffer zone R_{bz}.

for $l = L \rightarrow k$ **do**
  $h_l = 2^{L-l}$, $R_l = R_{bz} \times h_{l-1}$ ▷ radius of searched region at level l
  for y in $[-R_l, R_l]$ **do** ▷ centered at x in d-dimensional space
    if u_l$[y] == l$ **and** Manhattan_distance$(x, y) \leq R_l$ **then**
      bz_map$[y] = 1$ ▷ buffer zone points
    else
      bz_map$[y] = 0$ ▷ non-buffer zone, non-RoI points
    end if
  end for
end for
return bz_map

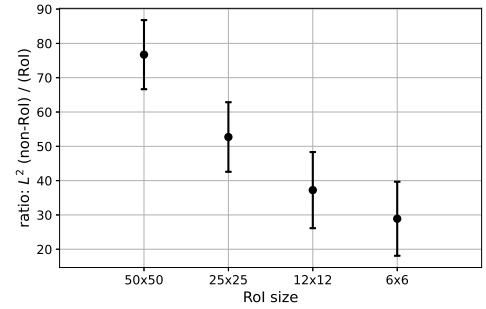
Theories provided in Sec. 4.2 indicate that bounds used for region-wise error control can be translated into a ratio between the error bounds used for data inside and outside the RoI. Claim 1 provides the error bounds when quantization error is incurred at a single node. To estimate the compression error of a node inside the RoI, we must accumulate the quantization errors propagated from every node outside the RoI. Checking the distance between every node inside and outside the RoIs is computationally prohibitive, considering that the RoI shapes are irregular and the data size is usually large. In addition, The error bound derived using the summation formula with the bound provided in Claim 1 will be extremely pessimistic, as the quantization error incurred at each node are different and errors at neighboring nodes may cancel each other out as the error propagation is an oscillatory function across nodes (shown in Fig. 4). We therefore estimate the ratio, $R_\tau = \tau_1 : \tau_0$, through empirical studies.

We conduct empirical studies as follows. Given an RoI in a 2D space, we use Algorithm 1 to build a buffer zone. We set the coefficients inside the RoI/buffer zone as zeros and the rest as random numbers between $[-1, 1]$. Next, we recompose the coefficients and check the L^2 errors for RoIs with different sizes. We demonstrate the trade-off between the radius of buffer zone and the maximally allowed non-RoI error bounds in Figs. 6a and 6b using $R_{bz} = 1$ and $R_{bz} = 2$. For each case, we measure the L^2 error inside the RoI. Because the "quantization errors" inside the RoI are zeros, errors measured after the recomposition are purely from the quantization errors propagated from coefficients outside. For each RoI size, we collected measurements from 10,000 trials and plot the mean and standard deviation in Fig. 6. By using a radius of $2h_{l-1}$ instead of $1h_{l-1}$, we could use an error bound approximately 3.5× larger to compress data in non-RoI. The trade-off indicates that a wider buffer zone (e.g., $2h_{l-1}$) is preferred when the requested RoI size is small, and vice versa. Error bounds used in the experiments of this paper are derived from the numbers in Fig. 6b.

4.3.2 Error quantization & RoI map encoding. The error-bounded quantization is performed using a linear-scaling encoder described in [36] and MGARD error control Equation (1). Decompressors need to know the quantization error bounds used at different data points so they can be appropriately reconstructed. In contrast to conventional RoI-based compression, we don't physically segment the data. Instead, we design a metadata-free quantization/dequantization algorithm to manage the varied error bounds. Using a linear-scaling encoder, an input datum x can be encoded into an integer by $\lceil \frac{x}{eb} \rceil$,



(a) Radius of a buffer zone: $1h_{l-1}$.



(b) Radius of a buffer zone: $2h_{l-1}$.

Figure 6: Empirical studies on the maximal error bound (τ_1) allowed for the non-RoI given an error bound τ_0 used for the RoI. The ratio $\tau_1 : \tau_0$ is determined by the size of the RoI and buffer zone. (a) uses a radius of $1h_{l-1}$ and (b) uses a radius of $2h_{l-1}$ for the buffer zone. We show the mean (circle) and standard deviation over 10,000 trials. A larger buffer zone radius means smaller sizes for the non -RoI, but data inside the non-RoI can be compressed more aggressively.

where eb is a scaled number originating from the input error bound τ taking into account the error control. With $\tau_1 = R_\tau \times \tau_0$, the eb_0 and eb_1 used for data inside and outside the RoI/buffer zone are subject to the same linear relation. Our metadata-free RoI quantization/dequantization algorithm is summarized as the following:

(1) In the quantization stage, if a node x is inside the RoI, we quantize it into $n_x = \lceil \frac{x}{eb_0} \rceil$; otherwise, we quantize it into $n_x = \lceil \frac{x}{eb_1} \rceil \times eb_1$.
(2) In dequantization stage, we convert n_x back to x' with $x' = n_x \times eb_0$, regardless of which eb has been used for x in quantization stage.

We use an example to explain the procedure. For an RoI point $x_0 = 100.52$ and a non-RoI point $x_1 = 100.83$, we set $\tau_0 = 0.1$ and $\tau_1 = 10 \times \tau_0$. After quantization, we get $n_0 = \lceil \frac{100.52}{0.1} \rceil = 1005$ and $n_1 = \lceil \frac{100.83}{0.1 \times 10} \rceil \times 10 = 1000$. Next, for dequantization, n_0 and n_1 are converted back to $x_0' = 100.5$ and $x_1' = 100$. The quantization error at x_0 is $|x_0 - x_0'| = 0.02$ and the quantization error at x_1 is $|x_1 - x_1'| = 0.83$, each satisfying the prescribed error bound of τ_0 and τ_1. With the above algorithm, our compressor only needs to record the compressed data, not the coordinates of regions. Data reduced by our region-adaptive compressor can be reconstructed back using regular, non-adaptive decompressors.

5 REGION-ADAPTIVE COMPRESSION PIPELINE

In this section, we summarize our region-adaptive compression procedure. Fig. 7 depicts the pipeline. The original MGARD compression pipeline consists of two steps: decomposition and error-bounded quantization. Our pipeline detects candidate critical regions, searches the buffer zone, and imposes multi-error-bounded linear quantization for region-wise error control. The following summarizes the compression steps:

(1) Begin with the coefficients produced by MGARD multilevel decomposition.
(2) Normalize the magnitude of the coefficients using the algorithm described in Sec. 4.1.1.
(3) Apply the mesh refinement algorithm described in Sec. 4.1.2 to select sub-regions where large coefficients are clustered.
(4) For each coefficient in the selected sub-region, include coefficients falling within a distance of h_l to form an RoI-blob, where $h_l = 2^{L-l}$.
(5) For each RoI-blob, check the surrounding coefficients by level to see if they fall within a buffer zone of a designated radius R_{bz} using the steps described in Algorithm 1.
(6) Quantize the coefficients in the RoI/buffer zone using the user-prescribed error bound τ_0 and the coefficients in the non-RoI region using τ_1, where $\tau_1 = R_\tau \times \tau_0$ and R_τ is derived based on the studies described in Sec. 4.3.
(7) Apply encoding and lossless compression to the quantized coefficients to obtain the final reduced representation.

Users can control how large a RoI to be retained by tuning the hyper-parameters in mesh refinement Algorithm described in 4.1.2. In the case of using different τ_0^i for multiple RoIs, $\tau_1 = R_\tau \times \min(\tau_0^i)$.

6 EXPERIMENTS

We compare our method to three baseline compression approaches using the atmospheric field output from the Energy Exascale Earth System Model (E3SM) version 1. We evaluate the L^2 error control and improvement of compression ratios on the raw data by comparing our approach to MGARD using single error bound and MGARD using multiple error bounds on region-segmented data. As indicated by the work in [20], the compression ratios of the state-of-the-art lossy compressors are comparable respect to L^2 error metrics. Next, we evaluate the improvement of region-adaptive compression on post-processing with two climate use cases – Tropical Cyclone (TC) detection and Atmospheric River (AR) tracking. Finally, we profile the overhead in computation resulted from the region-adaptive approach. The experiments in this paper were conducted on OLCF Andes cluster [31], where each node on the system has two 16-core AMD EPYC 7302 processors and 256 GB of memory.

6.1 Datasets

E3SM is a fully coupled Earth system and climate model used in mission-defined efforts in the U.S. Department of Energy, as

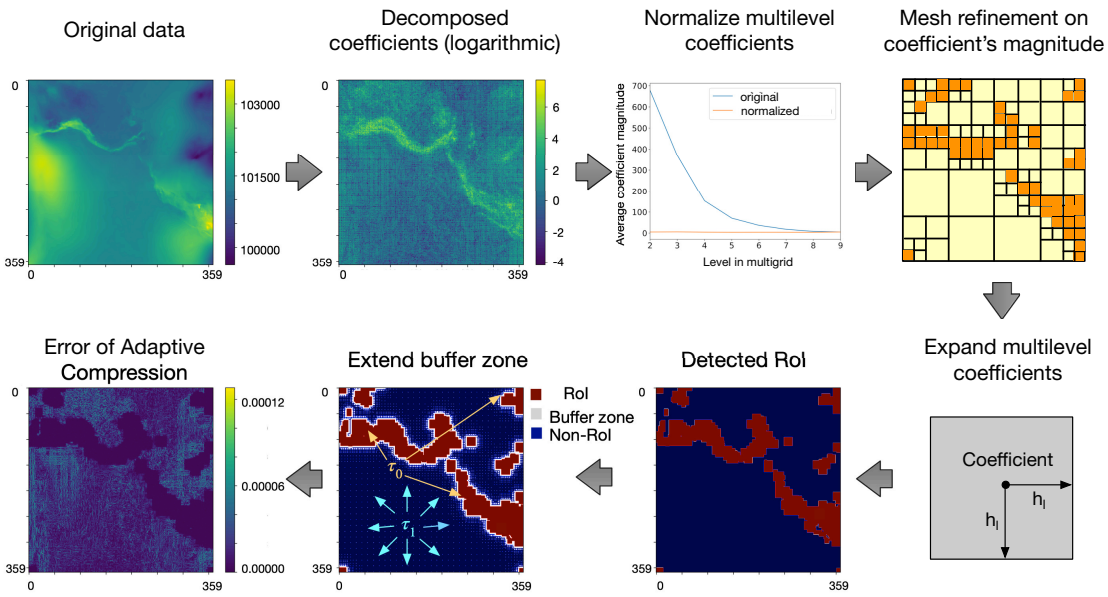


Figure 7: Region-adaptive compression pipeline. The pipeline is built upon a multilevel compressor, MGARD. We detect critical regions and compress regions with less details more aggressively using a larger error bound. RoIs are detected by applying mesh refinement on MGARD decomposed coefficients. To prevent non-RoIs (blue region) errors propagating into RoIs (red regions) after recomposition, we add buffer zones (white regions) in between. The bound used for error bound is derived from both theoretical and empirical studies.

well as several international model inter-comparison efforts [19]. Our experiments use the atmospheric data coming from its High-Resolution (HR) configuration coupled simulation. The atmosphere grids are based on a cubed-sphere topology. The HR grid configuration is characterized by 120 quadrilateral spectral elements in both x and y directions of each face of the cube sphere corresponding to an approximate grid spacing of 25 km and a total of ~800,000 columns per variable per snapshot. Our experiments use a dataset spanning 5 years at a temporal resolution of 6 hours and stored in `float32` precision. To achieve better compression ratios, for each snapshot, we transform variables from their original 1D grids to three 2D snapshots correlated in longitude and latitude on the cubed sphere. The compression is performed in 2D space for both region-adaptive and single-error-bounded cases.

6.2 Event analysis codes

To determine whether the data coming out of lossy compression is acceptable, we evaluate the impact of compression on both the raw data (i.e., relative L^2 error) and post-analysis statistics. We evaluate the impact on post-processing with two use cases – TC tracking and AR detection, and we use TempestExtreme Version 2.1 [38] in both cases. TempestExtreme is a software package which performs a variety of feature tracking and scientific analysis for global Earth-system data.

A TC is an intense circular storm that originates over warm tropical oceans. TempestExtreme identifies candidate TC locations first based on a minimum sea-level pressure, then with the associated upper-level warm cores. All candidate nodes are stitched together to form tracks with multiple conditions imposed, such as maximum distances between any two candidate nodes in one track, minimal nodes in a track, and minimal wind speed. For the above algorithm, five 2D variables are required as input: pressure at sea-level (PSL), temperature at 500 hPa and 200 hPa (T500, T200) to determine the upper-level warm core, zonal and meridional wind speeds (UBOT, VBOT) to derive wind speed at the lowest/bottom model level.

Atmospheric rivers (ARs) are thin and long filamentary structures characterized by high integrated vapor transport (IVT) and often resulting in intensive rainfall over impacted areas. The TempestExtreme detection algorithm detects ARs as ridges using the Laplacian of the IVT field. Only points whose Laplacian is less than a threshold are retained. And typically, features too near the Equator and those that are deemed too small filter out TCs. The final step is to stitch the binary map and labels individual ARs with different tags. For this algorithm, a 2D variable, magnitude of IVT, is used as input.

In both use cases, the weather events are characterized by rich regional features, making them ideal candidates for our region-adaptive compression algorithm. From the perspective of detection algorithms, TC tracking focuses on point-wise features (nodes), whereas AR detection focuses on areal feature (blobs) characterized by points in a continuous region. We evaluate our region-adaptive compression on these two distinct features.

6.3 Error control and data reduction evaluation

For TC analysis, we compress the five 2D input variables with a requested error bound $\tau_0 = 5 \times 10^{-5}$ for PSL and $\tau_0 = 1 \times 10^{-3}$ for

T200, T500, UBOT, and VBOT. We choose a smaller error bound for PSL as the detection thresholds defined on vortex intensity (i.e., depth of sea level pressure minimum) drive the major TC output sensitivity [41]. Since TC analysis focuses on small-scale structures (a radius of 4°–6.5° is used by most detection algorithms), we choose the starting k as 8 and p^{th} in the range of 10–15% for mesh refinement. These numbers are chosen by evaluating the TC analysis results using a sample dataset spanning 1 month. The resultant thresholds are then applied to the whole dataset spanning 5 years. The selected region accounts for approximately 16% of the total dataset after the coefficient to RoI-blob expansion. Since the location of features should not change among 5 TC tracking variables, we implement region detection only on PSL and use the obtained RoI map for the other 4 variables at the same snapshot.

For AR analysis, we compress the IVT variable with a requested error bound $\tau_0 = 2\times10^{-3}$. ARs are long continuous regions filled by blobs detected using a Laplacian operator, and the fluctuations in AR regions are less intense than those in TC regions. To cover AR regions, our region-adaptive method must keep more coefficients after mesh refinement. An example can be found in Fig. 8, which shows a snapshot of the IVT field, AR masks detected by TempestExtreme, and the RoI mask coming out of our region-adaptive compressor. In this example, we use a starting k as 16 and p^{th} as 17.5 − 50%, resulted in a selection of 33–38% across all 2D IVT snapshots.

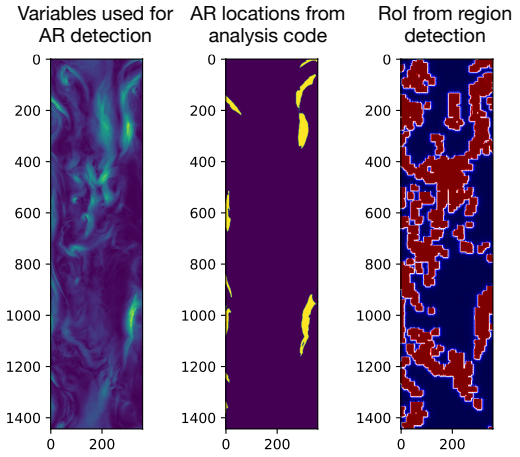
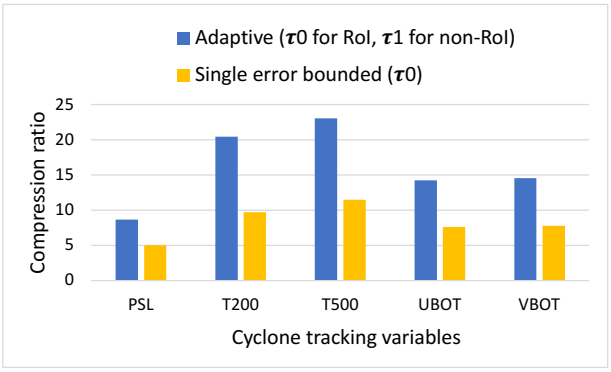


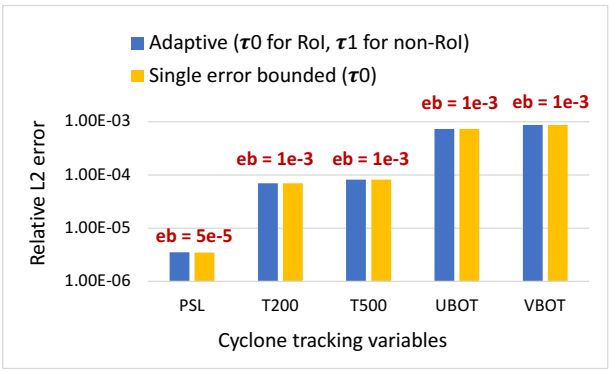
Figure 8: ARs show a long, filamentary structure. Since our region detection is not customized for AR tracking, a large portion of the total region (≥ 35% in our experiments) must be preserved to encapsulate the AR locations. Large RoIs limit the compression ratio achieved by our adaptive method.

Throughout the experiments, we use a buffer zone of radius $R_{rz} = 2$. We choose the ratio between the error bounds used for the RoI and non-RoI as $\tau_1 : \tau_0 = 23$ by interpolating the bottom of error bar in Fig. 6b with an RoI size of 8×8. We set the ratio of $\tau_1 : \tau_0$ based on the narrowest island in the detected RoI. We estimate the values based on the resolution of the finest grid used in mesh refinement and the average size of RoI-blobs drawn in the subsequent expansion. Please note the derived error bound is pessimistic because RoI-blobs may connect to form regions of much larger size, as shown in Fig. 8, so as can tolerate larger error bounds with surrounded data points.

We first check whether the proposed error control approach can guarantee that the compression errors in different regions respect a user-prescribed error tolerance in the L^2 norm. We compare our approach to a single-error-bounded compressor, MGARD. The latter reduces the 5 TC analysis variables using the same error bound we used for RoI compression. Fig. 9 shows the relative L^2 error measured in the recomposed RoIs for 5 TC analysis variables. The errors of our region-adaptive compressed data are indistinguishable from those measured in data compressed using a single error bound, and strictly less than the requested error bounds. Meanwhile, the compression ratio obtained using our region-adaptive approach is approximately 1.74× better for PSL, 2.05× better for two temperature variables, and 1.87× better for the velocity vector than the single-error-bounded MGARD.



(a) Compression ratios: adaptive vs. single-error-bounded.



(b) L^2 errors in RoI regions: adaptive vs. single-error-bounded.

Figure 9: Compressed variables used for tropical cyclone (TC) detection with an input error bound (eb) of τ_0. The adaptive method applies τ_0 to RoIs output from its region detection algorithm, which counts for ~16% of the total region, whereas uniform compression applies τ_0 for the whole region.

Next, we compare the compression ratios of our method to the standard RoI-based compression procedure. Consider a scenario where the location of an RoI has been obtained using some feature

tracking algorithms. The subsequent approach will then extract the RoI data, separate RoI and non-RoI data into a different sets, and compress each using different error bounds. This segmentation-based approach perfectly avoids cross-region error propagation, but it comes at the cost of extra metadata spent on saving RoI masks. Bounding box approaches cannot be applied effortlessly for our case. As an example shown in Fig. 8, RoIs in E3SM atmospheric field coming out of our detection method have irregular boundaries because the atmospheric data are captured over a wide span (i.e., global earth system) and our detection algorithm is not customized for capturing specific features. Alternatively, for the experiment used for comparison, we label points in different regions on a binary mask and save the mask in bit-format (i.e., 1 byte for 8 grid points). RoI and non-RoI data are linearized and compressed separately by single-error-bounded MGARD using error bounds τ_0 and τ_1. The compressed data comprises both the quantized bytes and associated masks. The ratios of 5 TC variables compressed using region-segmented and our region-adaptive approaches are shown in Fig. 10. The compression ratios of our approach are approximately 1.36× better for PSL, 1.8–2.0× better for the two temperature variables, and 1.43× better for the velocity vector than the region-segmented approach using the same set of error bounds. Our advantage comes from two aspects: (1) our approach don't save region labels, and (2) data correlations are better exploited by compressing RoI and non-RoI data together. For example, eliminating the cost of RoI masks, the compress ratio of our method is still 1.27× better than region-segmented approach for temperature variable, even though the data compressed using τ_0 is ~50% more in our case than those in region-segmented approach due to the data in buffer zone.

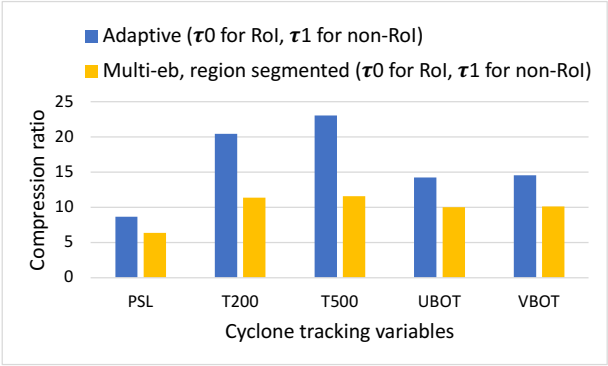


Figure 10: Compression ratios obtained on the input variables used for TC detection with adaptive and segmented approaches. The segmented approach splits RoI and non-RoI data using the same RoI mask output from the region-adaptive method and compresses RoI and non-RoI datasets separately using two error bounds.

We then evaluate the impact of compression on post-analysis. For TC analysis, we run TempestExtreme with the 5 input variables to identify candidate TC nodes in each snapshot and stitch them across time-series snapshots to form TC tracks. We fix the compression ratio (per-variable compression ratio displayed in Fig. 9a), and compare changes in TC analysis results caused by our approach to the ones caused by the single-error-bounded MGARD compression. Under the same compression ratio, the single-error-bounded compressor uses $\tau_0 = 3 \times 10^{-4}$ for PSL, $\tau_0 = 3.7 \times 10^{-3}$ for T200, $\tau_0 = 3.1 \times 10^{-3}$ for T500, $\tau_0 = 4.1 \times 10^{-3}$ for UBOT, and $\tau_0 = 3.9 \times 10^{-3}$ for VBOT. We measure the changes in location at each step of a TC track after lossy compression. We call TC tracks found in uncompressed 5-year ensemble $\{tc\}$, and tracks found in lossy compressed and reconstructed ensemble $\{\widetilde{tc_i}\}$. For every track tc_i in the set of $\{tc\}$, if there exists an equivalent $\widetilde{tc_i}$, we pair the two and compute the great-circle distance (gcd), r, between their nodes using the equation from [37]:

$$r(\lambda, \varphi; \tilde{\lambda}, \tilde{\varphi}) = \arccos\left(\sin\varphi \sin\tilde{\varphi} + \cos\varphi \cos\tilde{\varphi} \cos\left(\lambda - \tilde{\lambda}\right)\right)$$

where $\{\lambda, \varphi\}$ and $\{\tilde{\lambda}, \tilde{\varphi}\}$ are latitude-longitude coordinates of a node in tc_i and the corresponding node in $\{\widetilde{tc_i}\}$. Due to the variability of the detection algorithm, a TC track/step identified in the original ensemble may not show in the compressed ensemble. In that case, we use the shortest gcd between $\{\lambda, \varphi\}$ and any $\{\tilde{\lambda}_i, \tilde{\varphi}_i\}$ found at the same snapshot. We define the error as the number of grid-points changed in average for all steps in a TC track with the equation $d_{\mathrm{TC}} = \frac{1}{0.25N} \sum_i^N r(\lambda_i, \varphi_i; \tilde{\lambda}_i, \tilde{\varphi}_i)$, where N is the number of nodes in a TC track and 0.25° is the grid spacing in gcd. The average number of nodes per TC track is 16. We compute d_{TC} for each individual TC track and plot the statistical distribution of errors in all TC tracks found in 5 years. As shown in Fig. 11, d_{TC} computed from the data compressed using our region-adaptive method is more than 50% smaller than the error using the single-error-bounded approach across the whole distribution range. Moreover, 70% of the TC tracks computed from data compressed with our method match exactly with the ones found in uncompressed data, versus 50% with the single-error-bounded approach.

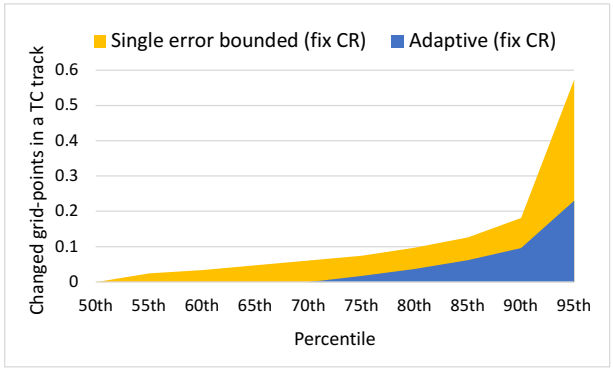


Figure 11: Fixing compression ratios (CR), evaluate the shifts in TC tracks when the detection algorithm uses lossy compressed data. The error is computed for each individual TC track and we plot the statistical distribution of errors among all TC tracks found in a dataset simulating 5-year climate.

For AR tracking, we run TempestExtreme with the IVT variable to detect AR masks in each snapshot and stitch the binary masks across time-series snapshots to label individual AR events. Similar to TC analysis, we fix the compression ratio as 16.7× and study

the impact of lossy compression on the detected AR events. Under the same compression ratio, our region-adaptive compressor uses $\tau_0 = 2 \times 10^{-3}$ for RoI data, and the single-error-bounded MGARD compressor uses $\tau_0 = 3 \times 10^{-3}$. We evaluate the changes in the size of a stitched AR event after lossy compression by the intersection over union (IoU). We use $\{M\}$ and $\{\tilde{M}\}$ to designate AR masks found in the original and compression data. For each AR event, we compute the intersection $M_i \cap \tilde{M}_i$ and the union $M_i \cup \tilde{M}_i$. The error is defined as $e_{\text{AR}} = (1 - \frac{M_i \cap \tilde{M}_i}{M_i \cup \tilde{M}_i}) * 100\%$. We plot the statistical distribution of e_{AR} in Fig. 12. Errors computed using data compressed by our region-adaptive approach are approximately 14–16% smaller than those computed by the single-error-bounded approach. This advantage is weaker compared to the one shown in TC analysis. As mentioned above, detecting features of blob structure (i.e., ARs) requires including more data in RoIs than detecing features of node structures (i.e., TCs), which diminishes the benefit of using a region-adaptive approach.

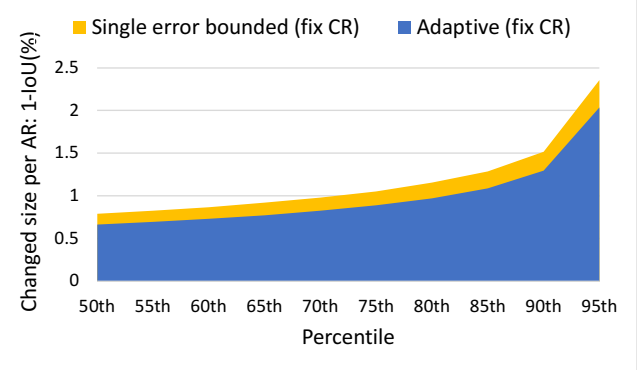


Figure 12: Fixing compression ratios (CR), evaluate how much the size of an AR changes when the detection algorithm uses lossy compressed data. The error is computed for each individual AR event using intersection of union (IoU) and we plot the statistical distribution of errors among all ARs found in a dataset simulating 5-year climate.

6.4 Throughput overhead

We further evaluate the performance overhead of region-adaptive compression. Our approach utilizes the decomposed coefficients from the MGARD compressor. Compared to single-error-bounded MGARD compression, the overhead mainly comes from mesh refinement and buffer zone searching. We use the 2D AR detection variable as the test data and vary the requested RoI size by tuning the threshold settings in mesh refinement. We plot the ratio of the region-adaptive overhead to the cost of the rest of the computation in Fig. 13. The overhead is observed to be small when the requested RoI size is small. For example, the overhead of doing region-adaptive compression for the variables used by TC analysis is detection is less than 10%. The buffer zone searching will eventually take almost the same amount of time as the RoI detection as RoI size grows, due to longer and more irregular boundaries.

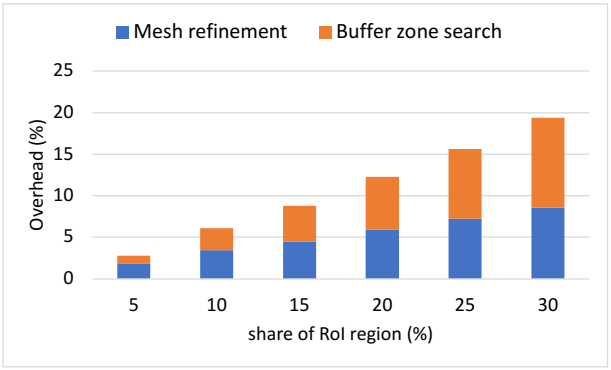


Figure 13: Overhead of region-adaptive compression. The overhead counts both region detection and buffer zone searching. We plot the ratio of execution time taken by overhead and the rest of the implementation; the latter is analogous to the standard, single-error-bounded compression.

7 CONCLUSION AND FUTURE WORK

In this paper, we present a region-adaptive lossy compression framework to tackle the needs of large compression ratios and region-wise error control by scientific applications. Our framework allows users to impose region-wise error bounds without region segmentation. Information of varied error bounds is embedded in the compressed data and the decompression can be performed using a regular, single-error-bounded compressor. Moreover, we also provide a method which detects candidate critical regions using the coefficients of a multilevel compressor in case the locations of RoIs are unavailable prior to compression. Experimental results demonstrate that for the 5 variables used for TC analysis, by selecting approximately 16% of the total region, our region-adaptive method can accurately capture the regions containing TC feature, and obtains approximately 2× and 1.6× compression ratios comparing to single-error-bounded approach and multiple error bounds, region-segmented approaches. Under the same compression ratio, the QoI in TC analysis is 2× accurate and the QoI in AR analysis is 1.15× accurate compared to the single-error-bounded approach.

One limitation of our approach is that in order to capture features with relatively large, continuous structures (e.g., AR), our method needs to keep a large region with high accuracy if the mask of RoI is not available as the input of compression. In future work, we plan to advance the theories in cross-region error control. In particular, we will derive bounds which can strictly control the accumulated errors from data outside the RoIs and eventually eliminate the bounds derived from empirical studies.

ACKNOWLEDGMENTS

This research was supported by the ECP CODAR, Sirius-2, and RAPIDS-2 projects through the Advanced Scientific Computing Research (ASCR) program of Department of Energy, and the LDRD project through DRD program of Oak Ridge National Laboratory. The climate simulation data were obtained from the Energy Exascale Earth System Model project, sponsored by the U.S. Department

of Energy, Office of Science, Office of Biological and Environmental Research.

REFERENCES

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. 94–105.

[2] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2018. Multilevel techniques for compression and reduction of scientific data—the univariate case. *Computing and Visualization in Science* 19, 5 (2018), 65–76.

[3] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2019. Multilevel techniques for compression and reduction of scientific data—quantitative control of accuracy in derived quantities. *SIAM Journal on Scientific Computing* 41, 4 (2019), A2146–A2171.

[4] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2019. Multilevel techniques for compression and reduction of scientific data—The multivariate case. *SIAM Journal on Scientific Computing* 41, 2 (2019), A1278–A1303.

[5] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2020. Multilevel Techniques for Compression and Reduction of Scientific Data—The Unstructured Case. *SIAM Journal on Scientific Computing* 42, 2 (apr 2020), A1402–A1427. https://doi.org/10.1137/19M1267878

[6] Marsha Berger and Isidore Rigoutsos. 1991. An algorithm for point clustering and grid generation. *IEEE Transactions on Systems, Man, and Cybernetics* 21, 5 (1991), 1278–1286.

[7] Marsha J Berger and Joseph Oliger. 1984. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics* 53, 3 (1984), 484–512.

[8] Harsh Bhatia, Duong Hoang, Garrett Morrison, Will Usher, Valerio Pascucci, Peer-Timo Bremer, and Peter Lindstrom. 2020. AMM: Adaptive Multilinear Meshes. *arXiv preprint arXiv:2007.15219* (2020).

[9] Martin Burtscher and Paruj Ratanaworabhan. 2008. FPC: A high-speed compressor for double-precision floating-point data. *IEEE Trans. Comput.* 58, 1 (2008), 18–31.

[10] Chunlei Cai, Li Chen, Xiaoyun Zhang, and Zhiyong Gao. 2019. End-to-end optimized ROI image compression. *IEEE Transactions on Image Processing* 29 (2019), 3442–3457.

[11] Franck Cappello, Sheng Di, Sihuan Li, Xin Liang, Ali Murat Gok, Dingwen Tao, Chun Hong Yoon, Xin-Chuan Wu, Yuri Alexeev, and Frederic T Chong. 2019. Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications* 33, 6 (2019), 1201–1220.

[12] Jianyu Chen, Maurice Daverveldt, and Zaid Al-Ars. 2021. FPGA Acceleration of Zstd Compression Algorithm. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 188–191.

[13] Jieyang Chen, Lipeng Wan, Xin Liang, Ben Whitney, Qing Liu, David Pugmire, Nicholas Thompson, Jong Youl Choi, Matthew Wolf, Todd Munson, et al. 2021. Accelerating multigrid-based hierarchical scientific data refactoring on gpus. In *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 859–868.

[14] Steven Claggett, Sahar Azimi, and Martin Burtscher. 2018. SPDP: An automatically synthesized lossless compression algorithm for floating-point data. In *2018 Data Compression Conference*. IEEE, 335–344.

[15] Peter Deutsch et al. 1996. GZIP file format specification version 4.3. (1996).

[16] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 ieee international parallel and distributed processing symposium (ipdps)*. IEEE, 730–739.

[17] Miloslav Feistauer, Jaromír Horáček, and Petr Sváček. 2015. Numerical simulation of airfoil vibrations induced by turbulent flow. *Communications in Computational Physics* 17, 1 (2015), 146–188.

[18] D Galassi, C Theiler, T Body, F Manke, P Micheletti, J Omotani, M Wiesenberger, M Baquero-Ruiz, I Furno, M Giacomin, et al. 2022. Validation of edge turbulence codes in a magnetic X-point scenario in TORPEX. *Physics of Plasmas* 29, 1 (2022), 012501.

[19] Jean-Christophe Golaz, Peter M Caldwell, Luke P Van Roekel, Mark R Petersen, Qi Tang, Jonathan D Wolfe, Guta Abeshu, Valentine Anantharaj, Xylar S Asay-Davis, David C Bader, et al. 2019. The DOE E3SM coupled model version 1: Overview and evaluation at standard resolution. *Journal of Advances in Modeling Earth Systems* 11, 7 (2019), 2089–2129.

[20] Qian Gong, Xin Liang, Ben Whitney, Jong Youl Choi, Jieyang Chen, Lipeng Wan, Stéphane Ethier, Seung-Hoe Ku, R Michael Churchill, C-S Chang, et al. 2021. Maintaining Trust in Reduction: Preserving the Accuracy of Quantities of Interest for Lossy Compression. In *Smoky Mountains Computational Sciences and Engineering Conference*. Springer, 22–39.

[21] Qian Gong, Xin Liang, Ben Whitney, and Scott Klasky. 2022. Improved L^∞ Error Control with MGARD. *in preparation* (2022).

[22] Hanqi Guo, David Lenz, Jiayi Xu, Xin Liang, Wenbin He, Iulian R Grindeanu, Han-Wei Shen, Tom Peterka, Todd Munson, and Ian Foster. 2021. FTK: A simplicial spacetime meshing framework for robust and scalable feature tracking. *IEEE Transactions on Visualization and Computer Graphics* 27, 8 (2021), 3463–3480.

[23] David A Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* 40, 9 (1952), 1098–1101.

[24] S Ku, Robert Hager, Choong-Seock Chang, JM Kwon, and Scott E Parker. 2016. A new hybrid-Lagrangian numerical scheme for gyrokinetic simulation of tokamak edge plasma. *J. Comput. Phys.* 315 (2016), 467–475.

[25] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Seung-Hoe Ku, Choong-Seock Chang, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F Samatova. 2013. ISABELA for effective in situ compression of scientific data. *Concurrency and Computation: Practice and Experience* 25, 4 (2013), 524–540.

[26] Xin Liang, Qian Gong, Jieyang Chen, Ben Whitney, Lipeng Wan, Qing Liu, David Pugmire, Rick Archibald, Norbert Podhorszki, and Scott Klasky. 2021. Error-controlled, progressive, and adaptable retrieval of scientific data with multilevel decomposition. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–13.

[27] Xin Liang, Hanqi Guo, Sheng Di, Franck Cappello, Mukund Raj, Chunhui Liu, Kenji Ono, Zizhong Chen, and Tom Peterka. 2020. Toward Feature-Preserving 2D and 3D Vector Field Compression.. In *PacificVis*. 81–90.

[28] Xin Liang, Ben Whitney, Jieyang Chen, Lipeng Wan, Qing Liu, Dingwen Tao, James Kress, David R Pugmire, Matthew Wolf, Norbert Podhorszki, et al. 2021. MGARD+: Optimizing multilevel methods for error-bounded scientific data reduction. *IEEE Trans. Comput.* (2021).

[29] Peter Lindstrom. 2014. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2674–2683.

[30] Peter Lindstrom and Martin Isenburg. 2006. Fast and efficient compression of floating-point data. *IEEE transactions on visualization and computer graphics* 12, 5 (2006), 1245–1250.

[31] OLCF. Accessed: March 14, 2022. Andes User Guide. *https://www.olcf.ornl.gov/olcf-resources/compute-systems/andes* (Accessed: March 14, 2022).

[32] Erik Schnetter. 2013. Performance and optimization abstractions for large scale heterogeneous systems in the cactus/chemora framework. In *2013 Extreme Scaling Workshop (xsw 2013)*. IEEE, 33–42.

[33] Maxime Soler, Mélanie Plainchault, Bruno Conche, and Julien Tierny. 2018. Topologically controlled lossy compression. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 46–55.

[34] Seung Woo Son, Zhengzhang Chen, William Hendrix, Ankit Agrawal, Wei-keng Liao, and Alok Choudhary. 2014. Data compression for the exascale computing era-survey. *Supercomputing frontiers and innovations* 1, 2 (2014), 76–88.

[35] Myungseo Song, Jinyoung Choi, and Bohyung Han. 2021. Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2380–2389.

[36] Dingwen Tao, Sheng Di, Zizhong Chen, and Franck Cappello. 2017. Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 1129–1139.

[37] Paul A Ullrich and Colin M Zarzycki. 2017. TempestExtremes: A framework for scale-insensitive pointwise feature tracking on unstructured grids. *Geoscientific Model Development* 10, 3 (2017), 1069–1090.

[38] Paul A Ullrich, Colin M Zarzycki, Elizabeth E McClenny, Marielle C Pinheiro, Alyssa M Stansfield, and Kevin A Reed. 2021. TempestExtremes v2. 1: a community framework for feature detection, tracking, and analysis in large datasets. *Geoscientific Model Development* 14, 8 (2021), 5023–5048.

[39] Skylar W Wurster, Han-Wei Shen, Hanqi Guo, Thomas Peterka, Mukund Raj, and Jiayi Xu. 2021. Deep Hierarchical Super-Resolution for Scientific Data Reduction and Visualization. *arXiv preprint arXiv:2107.00462* (2021).

[40] Mai Xu, Xin Deng, Shengxi Li, and Zulin Wang. 2014. Region-of-interest based conversational HEVC coding with hierarchical perception model of face. *IEEE Journal of Selected Topics in Signal Processing* 8, 3 (2014), 475–489.

[41] Colin M Zarzycki and Paul A Ullrich. 2017. Assessing sensitivities in algorithmic detection of tropical cyclones in climate data. *Geophysical Research Letters* 44, 2 (2017), 1141–1149.

[42] Kai Zhao, Sheng Di, Maxim Dmitriev, Thierry-Laurent D. Tonellot, Zizhong Chen, and Franck Cappello. 2021. Optimizing Error-Bounded Lossy Compression for Scientific Data by Dynamic Spline Interpolation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1643–1654.