Toward Quantity-of-Interest Preserving Lossy Compression for Scientific Data

Pu Jiao
University of Kentucky
pujiao@uky.edu

Sheng Di
Argonne National Laboratory
sdi1@anl.gov

Hanqi Guo
The Ohio State University
guo.2154@osu.edu

Kai Zhao
University of Alabama at Birmingham
kzhao@uab.edu

Jiannan Tian
Indiana University
jti1@iu.edu

Dingwen Tao
Indiana University
ditao@iu.edu

Xin Liang*
University of Kentucky
xliang@uky.edu

Franck Cappello
Argonne National Laboratory
cappello@mcs.anl.gov

ABSTRACT

Today's scientific simulations and instruments are producing a large amount of data, leading to difficulties in storing, transmitting, and analyzing these data. While error-controlled lossy compressors are effective in significantly reducing data volumes and efficiently developing databases for multiple scientific applications, they mainly support error controls on raw data, which leaves a significant gap between the data and user's downstream analysis. This may cause unqualified uncertainties in the outcomes of the analysis, a.k.a quantities of interest (QoIs), which are the major concerns of users in adopting lossy compression in practice. In this paper, we propose rigorous mathematical theories to preserve four families of QoIs that are widely used in scientific analysis during lossy compression along with practical implementations. Specifically, we first develop the error control theory for univariate QoIs which are essential for computing physical properties such as kinetic energy, followed by multivariate QoIs that are more commonly used in real-world applications. The proposed method is integrated into a state-of-the-art compression framework in a modular fashion, which could easily adapt to new QoIs and new compression algorithms. Experiments on real-world datasets demonstrate that the proposed method provides faithful error control on important QoIs including kinetic energy, regional average, and isosurface without trials and errors, while offering compression ratios that are up to 4× of the compression ratios provided by state-of-the-art compressors.

PVLDB Reference Format:
Pu Jiao, Sheng Di, Hanqi Guo, Kai Zhao, Jiannan Tian, Dingwen Tao, Xin Liang, and Franck Cappello. Toward Quantity-of-Interest Preserving Lossy Compression for Scientific Data. PVLDB, 16(4): 697 - 710, 2022.
doi:10.14778/3574245.3574255

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/szcompressor/SZ3/tree/qoi_error_control.

*Corresponding author: Xin Liang, Department of Computer Science, University of Kentucky, Lexington, KY 40506.
This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 16, No. 4 ISSN 2150-8097.
doi:10.14778/3574245.3574255

1 INTRODUCTION

Today's large-scale simulation and high-resolution instruments are producing data at an unprecedented velocity and amount. For instance, climate simulations can generate over 200 TB of data in 16 seconds [33], and fusion simulations produce over 200 PB of data in a single run [3]. As such data velocity and amount significantly outpace storage and I/O systems, there is an urgent need for data systems to efficiently store and query these scientific data.

Compression is the most promising way to address the big data problem and has been widely used in designing databases [18, 22, 25, 54, 58] and accelerating queries [14, 21] or analytics [60, 61]. However, traditional compression techniques face critical challenges when dealing with scientific data. On the one hand, lossless compressors [13, 23, 26] suffer from limited compression ratios (usually ≤ 2 according to [46, 52]) because of the random mantissas in the floating-point format. On the other hand, although lossy compressors [37, 50, 56] can obtain fairly high compression ratios, they have certain data loss during the compression, which may cause unqualified reconstructed data for post-processing.

Error-controlled lossy compression [10, 16, 36, 40, 45, 47, 53, 62] has been proposed and developed in recent years to significantly reduce scientific data volume and provide error controls at the same time. Using these compressors, scientists can specify an error tolerance (usually in the forms of L^∞ norms and/or L^2 norms) and gurantee the decompressed data is within the given error tolerance compared with the original data. As such, scientists can control the impact of the lossy compression on their post hoc analysis as needed. Error-controlled lossy compressors have been widely used in different applications including cosmology [57], quantum chemistry [27], molecular dynamics [64], and climate studies [15]. For the success of error-controlled lossy compressors in multiple disciplines, there is an ever-increasing trend of using them in the development of today's databases [30, 64].

Because of the gap between the error tolerance applied on the raw data and that on the user's post hoc analysis, how the reconstructed data can be guaranteed to produce correct outcomes in the downstream analysis, a.k.a., Quantities of Interest (QoIs), is still the most critical and challenging issue. Typical QoIs, including physical properties such as kinetic energy and topological information such as isosurface [48], contain important information that should not be altered too much during compression, otherwise the reconstructed data can lead to misinterpretation. However, most existing error-controlled lossy compressors, including SZ [62] and ZFP [45], have no error control for QoIs. As a result, determining a proper setting

Table 1: Error control provided by existing compressors

Compressor	L^∞	L^2	Pointwise Relative	Linear QoI	Quadratic QoI	Topological QoI
ISABELA [36]			✓			
FPZIP [47]			✓			
ZFP [45]	✓					
SZ [62]	✓	✓	✓			
MGARD [11]	✓	✓		✓		
cpSZ [41]			✓			✓
Our method	✓	✓	✓	✓	✓	✓

to ensure acceptable QoIs loss with these compressors requires trial and error, which is contrary to the benefits of adding error control.

While being understudied in the community, QoI-preserving compression has already been explored for a few specific data analyses. MGARD [11] is the first compressor that considers QoI preservation via rigorous mathematical derivation, but it can only guarantee errors in bounded-linear QoIs while many important QoIs such as kinetic energy is nonlinear. The feature-preserving compressor in [41] proposes an elegant way to keep the locations and types of critical points in 2D and 3D vector fields, but it is specialized for that particular feature. Table 1 summarizes the error control provided by state-of-the-art lossy compressors for scientific data. To the best of our knowledge, there is no compression software that preserves families of generic nonlinear QoIs such as the polynomial ones.

In this work, we develop both the theory and the implementation of a general QoI-preserving lossy compression framework. Inspired by [41], we leverage a pointwise error bound for each data point to convey the constraints from QoI to data. Specifically, we formulate the QoI error preservation problem as an error bound derivation problem for the compression of raw data. We theoretically establish a series of mappings from the error of target QoIs to the pointwise error bound on raw data. In particular, we show that four families of important QoIs can be preserved in this way and the set of preservable QoIs is closed under certain arithmetic operations. This method further generalizes to multivariate QoIs by taking advantage of the prediction-based compression pipeline where decompressed data is immediately accessible during compression. To this end, we develop a QoI-preserving compressor based on a state-of-the-art compression framework and validate it using real-world datasets on a cluster. Our contributions are summarized as follows.

- We develop rigorous error preservation theories for four families of important univariate QoIs. We also prove that QoIs composed through the operations such as addition, multiplication, and composition, can also be preserved.
- We extend our theories to preserve multivariate QoIs based on the coupled compression scheme in [41]. By leveraging the immediately available decompressed data during compression, we reduce the preservation of multivariate QoIs to that of univariate QoIs, which can be solved using the proposed error preservation theories.
- We implement a general QoI-preserving lossy compression framework based on our theories and a state-of-the-art lossy compressor. Particularly, we decouple the QoI derivation from the compressor so that 1) new QoI can be easily integrated and 2) the QoI preservation method can adapt easily to new compression algorithms.
- We evaluate our framework using four scientific datasets from real applications on a cluster. Experiments demonstrate that our method delivers compression ratios that are up to 4× of the compression ratios provided by the best existing compressors under the same QoI tolerance. The proposed method is also able to preserve the isosurface, leading to almost no difference in the underlying visualization. In addition, the localized error constraints make it easy and convenient for our algorithm to preserve multiple QoIs at the same time.

The rest of the paper is organized as follows. In Section 2, we discuss about the background and related works. In Section 3, we formulate the research problem and present an overview of the compression framework. In Section 4, we introduce error preservation theories for both univariate and multivariate QoIs, which serves as the foundation for the proposed work. In Section 5, we describe the implementation of the proposed framework. In Section 6, we present our evaluation results with real-world datasets on a cluster. In Section 7, we conclude with a vision for future work.

2 BACKGROUND AND RELATED WORKS

In this section, we present the background of Quantities-of-Interest (QoIs) and related works on scientific data compression.

2.1 Quantities-of-Interest in Scientific Data

Scientific data generated by simulations and instruments will be used in data analytics to identify patterns and extract features. As studied in [11], Quantities-of-Interest (QoIs) are used to refer to the outcomes of the underlying analytics, derived statistics from raw data. Typical QoIs include physical properties such as mass and momentum and topological features such as critical points [7] and isosurface [48], and they can be generalized to any information that is extracted from the data.

We classify QoIs into two categories and define them as follows. Univariate QoIs are defined as the downstream quantities computed using only one data point. These include momentum $p = mv$ and kinetic energy $E = \frac{1}{2}mv^2$ for particles, and logarithmic mapping $y = \log x$ for log-scale visualization, etc. Multivariate QoIs are defined as the downstream quantities computed using multiple data points. Statistical measurements such as regional average and topological information such as isosurface fall into this category.

In our paper, we focus on how to preserve four families of QoIs widely used in domain scientific analytics, with the corresponding visualization on a sample data field in Fig. 1. It is observed that these QoIs exhibit very different properties that are needed by different data analytics. The detailed mathematical formulation of these QoIs will be presented in Section 3.1.

Polynomials: Polynomials are univariate QoIs that involve only the operations of addition, subtraction, multiplication, and non-negative integer exponentiation of a data point . As a typical example, kinetic energy $E = \frac{1}{2}mv^2$ is the energy that an object possesses due to its motion, where m and v are the mass and velocity of the object. Because m is usually a constant (especially for particle simulations), kinetic energy is a quadratic function of v. As one of the key properties needed in many physics-related domains, the preservation of kinetic energy is important for many physics simulations including fusion energy science [19] and cosmology [29].

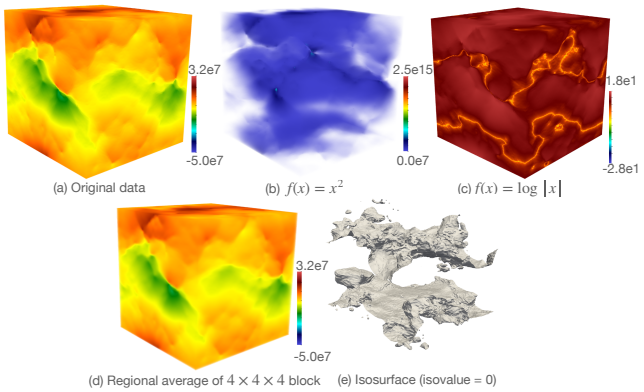

Figure 1: Visualization of original data and the four QoIs on a scientific data field (NYX velocity_x).

Logarithmic Mappings: Logarithmic mappings are univariate QoIs that map original data to the logarithmic domain with a given base. As demonstrated in previous work [39], this QoI is particularly useful for visualizing data with clustered values close to 0, which is usually the case for certain data fields in scientific datasets (such as dark matter density in cosmology simulations [2]).

Weighted Sum: Weighted sum is a family of multivariate QoIs that compute aggregated information in local regions, where the regional average is a particular example. The regional average is usually used to represent data in coarse resolution for either visualization or exploratory analysis, as doing that on the entire data is costly while sampling approaches based on decimation may not be accurate due to the negligence of unselected data points. For instance, it is a required preprocessing step for images produced by the x-ray diffraction in [1] due to the ultra-fast data generation rate.

Isoline/Isosurface: Isoline, also known as contour line, is the line connecting specified constant values (a.k.a isovalues) in 2D data and isosurface is its generalization to 3D cases. Similar to weighted sum, they are multivariate QoIs whose computation involves data from local regions. They are widely used in various applications to recognize and understand patterns and relationships in the data. For instance, isobar used in [51] is the isoline used to represent points of equal atmospheric pressure and isotherm studied in [32] is the one for equal temperature in climate research.

2.2 Compression for Scientific Data

Many approaches have been proposed in the literature to address the imbalanced growth between data and storage systems for scientific applications. Lossless compression techniques such as GZIP [26], ZSTD [23], and BLOSC [13] can recover the exact data during decompression, but they suffer from limited compression ratios on scientific data due to the random mantissas in the floating-point format. Recently studies show that lossless compression can only achieve 2× reduction in most cases [46, 52], while at least 10× is required in many use cases [17].

General lossy compressors such as JPEG/JPEG2000 [50, 56] and VAPOR [37] are able to trade-off accuracy for high compression ratios, but they are not trusted by many scientists because of the unbounded loss in decompressed data. To mitigate this issue, error-controlled lossy compression is proposed to reduce data while maintaining a certain level of accuracy.

There are two main models for error-controlled lossy compression, namely prediction-based [36, 40, 47, 53, 62] and transform-based [10, 45]. Prediction-based compression models such as SZ [53, 62] usually follow a general compression pipeline [43], which consists of four stages including prediction, quantization, entropy encoding, and lossless compression. In the first two stages, each data value is predicted using specific predictors, e.g., Lorenzo predictor in [53] and spline predictor in [62], to take advantage of the spatiotemporal correlation in the data, and then the difference between the predicted value and the original value is quantized to an integer value to reduce the entropy while enforcing the error bound. Please note that decompressed data is required for prediction during compression to ensure the enforcement of error bound during decompression. The quantized integer values are then encoded by entropy encoders such as Huffman encoder [31] and arithmetic encoder [59], followed by another lossless compression stage with GZIP [26] or ZSTD [23] that further reduces the size. ZFP is a transform-based compressor that compresses data in separate blocks. During compression, it first converts data in each block to fixed-point format under the same exponents, and then applies a near-orthogonal transform to generate coefficients in the transformed domain, which are further quantized and encoded using embedded encoding. Although these compressors provide error control on the decompressed data, uncertainties may arise in QoIs derived from raw data because their compression stages are completely QoI-agnostic.

Recently, the developers of MGARD proposed to preserve QoI during lossy compression and managed to control the error for the family of bounded-linear QoIs [11]. Drawn from the wavelet theories and finite element analysis, the original MGARD [10] relies on multilinear interpolation with L^2 projection to transform data into multilevel coefficients, which are then quantized and encoded using linear-scaling quantization [53] and lossless compression. In the latest studies, the MGARD team proposed an operator norm to integrate QoI information, which is used to adjust the quantization strategy based on prior knowledge. Through careful derivations on both the QoI and error propagation of multilevel coefficients, they showed that the errors in bounded-linear QoIs, such as mass and streamlines, can be preserved. Nevertheless, the QoI error control in MGARD is a little loose due to pessimistic estimations, and it cannot control the errors in nonlinear QoIs.

In [41], the authors proposed a novel way to preserve critical points in 2D/3D vector fields. Specifically, they transform the requirements of retaining critical points in local cells to sufficient error bounds on the data points based on how critical points are extracted, and use them to reduce data with two compression schemes. However, the derivation is specific to critical point extraction and thus cannot be generalized to other use cases. There are contradictory deficiencies between the proposed two schemes as well: the decoupled scheme can adapt to different compression methods but suffers from limited compression ratios, while the coupled scheme is tightly integrated with a specific compression method and thus is hard to adapt when new compression algorithms are developed.

In this work, we develop a QoI-preserving compression framework based on the coupled scheme in [41], but significantly expand

Table 2: Notations

Symbol	Description
τ	User-specified error tolerance on QoIs.
ε	Actual error of QoIs using decompressed data.
n	Number of data points.
d	Dimentionality of data.
x	Original data (single point).
x'	Decompressed data (single point).
ϵ	Derived error bound for a single data point.
ξ	The error of a data point ($\xi = x - x' \in [-\epsilon, \epsilon]$).
$\mathbf{x}, \mathbf{x}', \boldsymbol{\epsilon}$	Vectors of x, x', ϵ in multivariate cases.
x_i, x_i', ϵ_i	The i-th element in $\mathbf{x}, \mathbf{x}', \boldsymbol{\epsilon}$.
f	Abstraction for univariate QoI.
g	Abstraction for multivariate QoI.
$Q(f, \tau, x)$	A univariate error bound derivation problem.
$\lvert\cdot\rvert$	Operator of getting absolute value.
$\lvert\cdot\rvert_{L^\infty}$	Operator of getting L^∞ norm.

the functionality to guarantee the preservation of a wide range of QoIs and easy adaption to any prediction-based compression method. Specifically, we develop a general error control theory to map the tolerance of QoIs to that of data for four families of important QoIs, and prove its applicability for any new QoIs of their composition through certain arithmetic operations. We further design and implement a flexible and modular compression framework to decouple QoIs from the actual compression method for easy adaption. Specifically, our framework is integrated into the latest interpolation-based compressors [62] whose predictor switches according to the user-specified tolerance, which usually yields higher compression ratios.

3 OVERVIEW

In this section, we formulate our research problem and describe the quality measurement, followed by an overview of the proposed QoI-preserving lossy compression framework. Common notations used in this paper are summarized in Table 2. The focused QoIs that are preserved in this paper as summarized in Table 3.

Table 3: Target QoIs

Name	Category	Formula	Goal
Polynomials	univariate	$f(x) = \sum a_i x^i$	$\varepsilon \leq \tau$
Logarithmic Mapping	univariate	$f(x) = a\log_b x + c$	$\varepsilon \leq \tau$
Weighted Sum	multivariate	$g(\mathbf{x}) = \sum a_i f(x_i)$	$\varepsilon \leq \tau$
Isoline/isosurface	multivariate	$g(\mathbf{x}) = \{\boldsymbol{x} \lvert u(\boldsymbol{x}) = z\}$	$\varepsilon^* = 0$

* Using the localized metric defined in Section 3.

3.1 Problem Formulation

The objective of QoI-preserving lossy compression is to provide error control on the underlying QoIs while compressing raw data lossily, such that scientists can effectively reduce the volumes of their data based on their actual needs. We first define QoI in this paper as follows, and formulate the research objectives thereafter.

Definition 1. *A univariate QoI $f : \mathbb{R} \to \mathbb{R}$ is a function that maps a data value to a quantity, e.g., $f(v) = \frac{1}{2}mv^2$ maps velocity v to kinetic energy. A multivariate QoI $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}$ is a function that maps a vector of data values to a quantity, e.g., $\boldsymbol{g}(\mathbf{x}) = \sum_{x \in \mathbf{x}} x$ maps the data vector to its mean value. Note that this definition easily generalizes to QoIs that map original data to multiple quantities.*

Given d-dimensional scientific data $\mathbf{x}$ and QoI tolerance τ, the general goal is to achieve maximum compression ratio while ensuring $\varepsilon = |f(\mathbf{x}) - f(\mathbf{x}')|_{L^\infty} \leq \tau$ for univariate QoI f or $\varepsilon = |g(\mathbf{x}) - g(\mathbf{x}')| \leq \tau$ for multivariate QoI g in decompressed data $\mathbf{x}'$. Note that some statistical QoIs such as Mean Square Errors (MSE) are defined in the form of $g(\mathbf{x}, \mathbf{x}')$ instead of $g(\mathbf{x})$. In this case, we re-define $g'(\mathbf{y}) = g(\mathbf{x}, \mathbf{y})$ (using $\mathbf{y}$ as the independent variable), such that these QoIs can be formulated in the same way.

With this general formulation, we then formulate examples of the four QoIs that will be used in our evaluation. Specifically, we will preserve $f(x) = x^2$ which represents the order of kinetic energy, and $f(x) = \log x$ which is a general logarithmic function for polynomials and logarithmic mapping, respectively. We will then consider the preservation of the regional average of x^2 and the isosurface with the specified isovalue as two multivariate QoIs. As the general formulation directly applies to the univariate examples, we mainly focus on the formulation of the latter two.

Regional average of x^2: We use the regional average of x^2 instead of x as our example because 1) it cannot be preserved by existing compressors and 2) it can be compared with the univariate preservation of $f(x) = x^2$. Given data of dimension $n_1 \times n_2 \times \cdots \times n_d$ and block size B, we can segment the data into blocks of B^d. The regional average QoI $\boldsymbol{g}$ maps each block of data to a single quantity that equals to the average of their squares and produce a dataset with a coarse resolution $\lceil \frac{n_1}{B} \rceil \times \lceil \frac{n_2}{B} \rceil \times \cdots \times \lceil \frac{n_d}{B} \rceil$. The goal is then to control the maximal error in the coarse-resolution quantities.

Isoline/Isosurface: Formally, the isoline of an isovalue z can be described as the collection $I(z) = \{(x, y) | f(x, y) = z \text{ and } (x, y) \in \Omega\}$ where f is the functional representation of data and Ω is the corresponding domain, and isosurface is its direct generalization to 3D cases. As there is no general metric to evaluate the error of the isoline or isosurface, we leverage a localized definition similar to that of critical points in [41].

We first introduce the classical marching squares algorithm which extracts isolines (marching cubes for isosurfaces [48]), followed by our definitions for correct and erroneous cells. In particular, the algorithm takes one cell with 4 neighbor locations at a time (8 for 3D cases) and determines the shape of the polygon based on their relative values compared to the isovalue. An edge is identified as required for representing the isoline if one of the node values on the edge is larger than the isovalue and the other is smaller. Then, linear interpolations are performed between the two nodes on the required edges to find the exact position and generate the isoline. An example of isoline extraction is illustrated in Fig. 2(a).

Since the shape of the isoline is solely determined by the relative values of the nodes compared to the isovalue, we define the metric for isoline preservation as shown in Fig. 2(b). Specifically, we define a cell as "matched" if the relative values of all nodes (compared with the isovalue) keep the same in original and decompressed data, or formally $(x - z)(x' - z) > 0$ for an isovalue z and any node x belonging to the cell. Please note that we overlook the perturbations on the required edges as they will not affect the shape or trend of the isoline. Mismatches happen in the following 3 cases: (1) False Negative (FN) if an isoline is present in the original data but absent

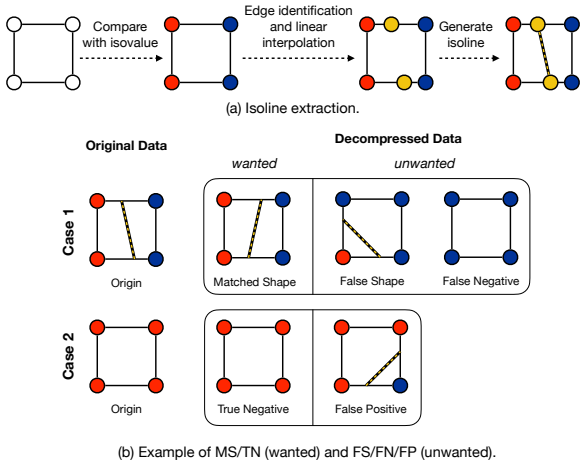

Figure 2: Isoline extraction and examples of False Negative (FP), False Positive (FN), and False Shape (FS) cells. Red nodes have larger values than the isovalue while blue nodes have smaller values. Dotted lines indicate the extracted isoline.

in the decompressed data; (2) False Positive (FP) if an isoline is absent in the original data but appears in the decompressed data; (3) False Shape (FS) for any other mismatches that lead to a wrong connection of an isoline to adjacent cells. The number of FN/FP/FS will indicate how the isoline is preserved during lossy compression.

3.2 Quality Measurement

We use rate-distortion as a quality measurement in this paper as it is an important and widely used metric in the community. Rate here stands for bit-rate, which is computed by the size of the original data type (e.g., 32 for single-precision floating point and 64 for double) over compression ratio thus indicating the mean number of bits used in the compressed format. When multiple fields are present in a dataset, the aggregated bit-rate is the average bit-rate of all the fields. We use the error of the underlying QoI as our distortion metric. Thus, higher compression quality is indicated by a lower QoI error under the same bit-rate or lower bit-rate (i.e., higher compression ratios) under the same QoI error. We also present the compression performance/throughput, which is evaluated by the size of data divided by the corresponding compression/decompression time.

3.3 System Overview

We present the overview of our QoI-preserving compression framework in Fig. 3 with the bottom line depicting stages in existing prediction-based lossy compressors (e.g., SZ [53] and FPZIP [47]). Unlike existing compressors which utilize a uniform global error bound, our framework leverages a QoI module to estimate the error bound for each data point and use it during quantization. Specifically, the pointwise error bound is estimated by a univariate QoI solver based on original and already decompressed data following the theories to be presented in Section 4, and quantized by a dedicated error bound quantizer to reduce storage cost. Then, the decompressed value of the error bound (instead of the uniform global error bound) is fed to the data quantizer in the prediction-based compression pipeline. When all the data points have been processed, the quantized values of both error bounds and data are compressed using Huffman encoding and lossless compression. Note that the workflow of the proposed framework is similar to the online compression scheme proposed in [41], but differs in the decoupled design for generalization and easy extension to both new QoIs and new predictors/compressors. The detailed implementation and integration with the target QoIs will be introduced in Section 5.

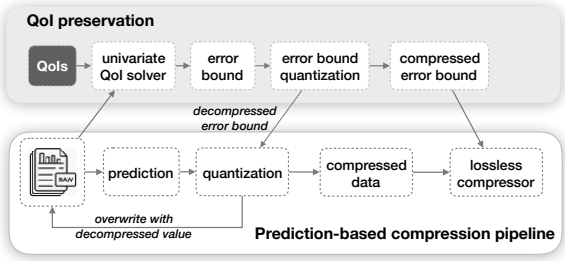

Figure 3: Design of QoI-preserving lossy compression framework. The QoI module is fully decoupled from the pipeline for easy adaption to new QoIs and diverse predictors.

4 THEORETICAL FOUNDATION

In this section, we present the theoretical foundation of our QoI-preserving lossy compression framework. Specifically, we aim to derive the analytical solutions to preserve certain families of univariate and multivariate QoIs using the pointwise error bounds proposed in [41]. Note that the theories in this section only deliver a sufficient solution to demonstrate the feasibility of this approach for the target families of QoIs. Integration of the example QoIs and the corresponding optimizations will be introduced in Section 5.

4.1 Univariate QoI Preservation

Our key idea for enabling QoI preservation is to derive the proper error bound on each point of the raw data according to the QoIs. As such, we formulate the error bound derivation problem for univariate QoI preservation as follows.

Definition 2. *Given a univariate QoI $f : \mathbb{R} \to \mathbb{R}$ and an acceptable QoI tolerance τ, the error bound derivation problem $Q(f, \tau, x)$ for a data point x solves an error bound ϵ in the form of user-specified tolerance τ, such that $|f(x) - f(x')| \leq \tau$ for any x' satisfying $|x - x'| \leq \epsilon$.*

Per the definition, solving an error bound derivation problem $\epsilon = Q(f, \tau, x)$ returns the required error bound ϵ on the raw data point x, which leads to a maximal error of τ in the QoI f. This definition can be easily generalized to cover QoIs that map original data x to multiple quantities by treating each mapping as a separate QoI. For instance, it is straightforward to have $Q(f, \tau, x) = \min(Q(f_1, \tau, x), Q(f_2, \tau, x))$ when $f(x) = (f_1(x), f_2(x))$. Once ϵ is identified, compressing x with the error bound ϵ guarantees that the error in the underlying QoI is less than τ. Note that ϵ must exist, because $\epsilon = 0$ is a feasible solution. Based on this property, we further define the preservability of QoIs as follows.

Definition 3. *A univariate QoI f is **preservable** if there is an analytical solution for the error bound $\epsilon = Q(f, \tau, x)$.*

This definition indicates a sufficient condition for univariate QoIs that can be preserved in this paper. Also, when multiple preservable QoIs are present at the same time, selecting the minimal derived error bound will guarantee the required errors on them all, as summarized in the following corollary.

Corollary 1. *$\min(Q(f_1, \tau_1, x), Q(f_2, \tau_2, x))$ provides a sufficient error bound that satisfies the error requirements of two QoIs (τ_1 for f_1 and τ_2 for f_2) at the same time.*

We seek for sufficient bounds to solve the error bound derivation problems in practice, because the optimal solution that yields the largest ϵ is usually hard to find especially for complex nonlinear QoIs. In the following, we present and prove some general properties for preservable QoIs at first, and then the sufficient solutions for certain families of QoIs.

First, we show that the preservable QoIs are closed under certain arithmetic operations including addition, multiplication, and composition, as shown in the following lemmas. These properties can be used to compose complex preservable QoIs based on existing ones thus increasing the coverage of the proposed method.

Lemma 1. *If ϵ_1 and ϵ_2 are sufficient solutions for $Q(f_1, \tau_1, x)$ and $Q(f_2, \tau_2, x)$, respectively, then $\min(\epsilon_1, \epsilon_2)$ is a sufficient solution for $Q(f_1 + f_2, \tau_1 + \tau_2, x)$.*

Proof. Let $f = f_1 + f_2$. We have $|f(x) - f(x')| \leq |f_1(x) - f_1(x')| + |f_2(x) - f_2(x')| \leq \tau_1 + \tau_2$ using the triangular inequality. □

Lemma 2. *If ϵ_1 and ϵ_2 are sufficient solutions for $Q(f_1, \tau_1, x)$ and $Q(f_2, \tau_2, x)$, then $\min(\epsilon_1, \epsilon_2)$ is a sufficient solution for $Q(f_1 f_2, \tau_1\tau_2 + |f_1(x)|\tau_1 + |f_2(x)|\tau_2, x)$.*

Proof. Let $f = f_1 f_2$. We have the following relationship:

$$\begin{aligned}
|f(x) - f(x')| &= |f(x') - f(x)| = |f_1(x')f_2(x') - f_1(x)f_2(x)| \\
&= |(f_1(x') - f_1(x))(f_2(x') - f_2(x)) + (f_1(x') - f_1(x)) \\
&\quad \cdot f_2(x) + f_1(x)(f_2(x') - f_2(x))| \\
&\leq |f_1(x') - f_1(x)||f_2(x') - f_2(x)| + |(f_1(x') - f_1(x))| \\
&\quad \cdot |f_2(x)| + |f_1(x)||(f_2(x') - f_2(x)| \\
&\leq \tau_1\tau_2 + |f_1(x)|\tau_1 + |f_2(x)|\tau_2
\end{aligned}$$

□

By trivially setting $\tau_1 = \tau_2$, these lemmas can be used to solve the error bound derivation problem for QoIs composed by additive and/or multiplicative operations, as shown in the below corollaries.

Corollary 2. *$\min(Q(f_1, \tau/2, x), Q(f_2, \tau/2, x))$ is one sufficient solution for $Q(f_1 + f_2, \tau, x)$.*

Corollary 3. *Let $f^+(x) = |f_1(x)| + |f_2(x)|$, $\tau' = \frac{-f^+(x) + \sqrt{4\tau + f^+(x)^2}}{2}$. $\min(Q(f_1, \tau', x), Q(f_2, \tau', x))$ is one sufficient solution for $Q(f_1 f_2, \tau, x)$.*

Lemma 3. *$Q(f_2, Q(f_1, \tau, f_2(x)), x)$ is a sufficient solution for $Q(f_1 \circ f_2, \tau, x)$ where $\circ$ is composition operation.*

Proof. $Q(f_1, \tau, f_2(x))$ gives a sufficient error bound for $f_2(x)$ so a sufficient error bound for x would be $Q(f_2, Q(f_1, \tau, f_2(x)), x)$. □

Given these properties, we then identify the several preservable QoIs including the families of polynomials, logarithmic functions, and radical functions.

Lemma 4. *For non-degenerative linear QoI $f(x) = ax + b$, $\epsilon = \tau/|a|$ is a sufficient solution for $Q(f, \tau, x)$.*

Proof. By directly applying the triangular inequality, we have $|f(x) - f(x')| \leq |a||x - x'| \leq |a|\epsilon = \tau$. □

Theorem 1. *All polynomial QoIs are preservable.*

Proof. Because an order n polynomial can be written as the multiplication of a linear polynomial and an order $n-1$ polynomial, this can be proved by mathematical induction with Lemma 1/2/4. □

Theorem 2. *For logarithmic QoIs $f(x) = a\log_b x + c$ with $b > 1$, $\epsilon = |x| \min(1 - b^{-\tau/|a|}, b^{\tau/|a|} - 1)$ is a sufficient solution for $Q(f, \tau, x)$.*

Proof. Let $\xi = x - x' \in [-\epsilon, \epsilon]$. We have

$$\begin{aligned}
|f(x) - f(x')| &= |f(x') - f(x)| = |a\log_b(1 + \xi/x)| \\
&\leq |a||\log_b(1 + \xi/x)| \leq |a||\tau|/|a| = \tau
\end{aligned}$$

□

Theorem 3. *For radical QoIs $f(x) = \sqrt{x}$, $\epsilon = \tau^2 - 2\tau\sqrt{x}$ is a sufficient solution for $Q(f, \tau, x)$.*

Proof. Let $\xi = x - x' \in [-\epsilon, \epsilon]$. We have

$$|f(x) - f(x')| = \left| \frac{\xi}{\sqrt{x+\xi} + \sqrt{x}} \right| < \tau.$$

This reduces to $(\xi - \tau\sqrt{x})^2 < \tau^2(x + \xi)$ and $(\xi + \tau\sqrt{x})^2 > \tau^2(x + \xi)$, or equivalently $\xi^2 - (2\tau\sqrt{x} + \tau^2)\xi < 0$ and $\xi^2 + (2\tau\sqrt{x} - \tau^2)\xi < 0$. Similarly, it has closed-form solution $\epsilon = \min(\tau^2 + 2\tau\sqrt{x}, \tau^2 - 2\tau\sqrt{x}) = \tau^2 - 2\tau\sqrt{x}$. □

4.2 Multivariate QoI Preservation

We define the error bound derivation problem and preservability of multivariate QoIs in a way similar to those of the univariate ones, which is detailed as follows.

Definition 4. *Given a multivariate QoI $g : \mathbb{R}^n \to \mathbb{R}$ and an acceptable QoI tolerance τ, the error bound derivation problem $Q(g, \tau, \mathbf{x})$ for input d-dimensional data $\mathbf{x}$ solves an error bound $\boldsymbol{\epsilon}$, such that $|g(\mathbf{x}) - g(\mathbf{x}')| \leq \tau$ for any $\mathbf{x}'$ satisfying $|\mathbf{x} - \mathbf{x}'| \leqq \boldsymbol{\epsilon}$, where $|\mathbf{x} - \mathbf{x}'| \leqq \boldsymbol{\epsilon}$ means $|x_i - x_i'| \leq \epsilon_i$ for any i.*

The definition is a direct extension of Definition 2 which works for univariate QoIs. Instead of solving a single error bound ϵ for a single data point x in $Q(f, \tau, x)$, $Q(g, \tau, \mathbf{x})$ needs to solve the sufficient error bounds for all data points that are involved in the computation of the multivariate QoI g.

Definition 5. *A multivariate QoI g is **preservable** if there is an analytical solution for the error bound vector $\boldsymbol{\epsilon} = Q(g, \tau, \mathbf{x})$.*

Similar to Definition 3, this definition indicates a sufficient condition for multivariate QoIs that can be preserved in this paper, and can be generalized to cover QoIs that map the original data to multiple quantities. Based on this definition, we have the following theorem for multivariate QoI preservation.

Theorem 4. *Preservation of a multivariate QoI can be reduced to that of a family of univariate QoIs when the prediction-based compression pipeline is used.*

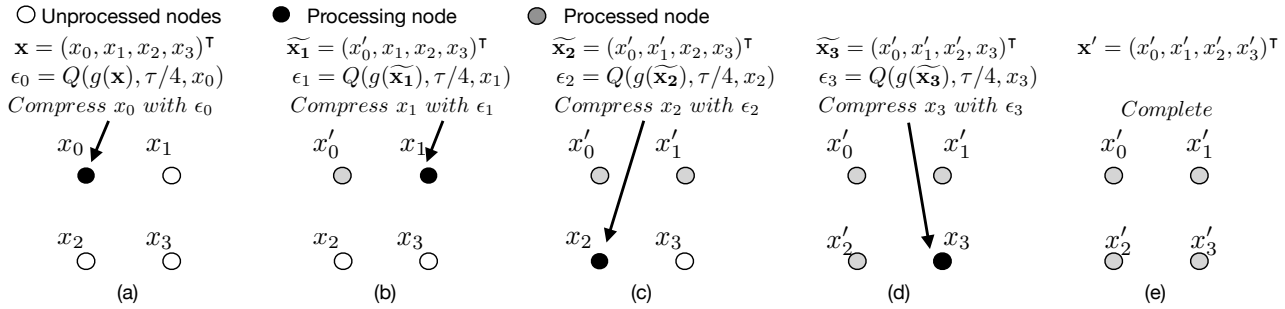

Figure 4: Online error bound derivation and compression with prediction-based compression pipeline for multivariate QoIs.

Proof. We reduce the multivariate error bound derivation problem $Q(g, \tau, \mathbf{x})$ to a family of univariate problems as follows. As mentioned in Section 2, prediction-based compressors such as SZ [53] and FPZIP [47] process input data point one by one, and the decompressed value for any data point is known immediately after it is processed. Consider the case when the i-th point is being processed. As x_i is the only value that will be changed in this iteration, the multivariate QoI $g(\mathbf{x})$ can be regarded as a univariate one when $x_{j(j<i)}$ are treated as constants with decompressed values and $x_{j(j>i)}$ are treated as constants with the original value, respectively. Specifically, let $\widetilde{\mathbf{x_i}} = (x'_0, \ldots, x'_{i-1}, x_i, x_{i+1}, \ldots, x_{n-1})^\intercal$ and $g_i(x_i) = g(\widetilde{\mathbf{x_i}})$ be a univariate QoI of x_i, and we have $\widetilde{\mathbf{x_0}} = \mathbf{x}$, $\mathbf{x}' = \widetilde{\mathbf{x_n}}$, and $g_i(x'_i) = g(\widetilde{\mathbf{x_{i+1}}})$. Then, the error bound derivation problem at the current iteration with τ_i as the allowed tolerance is reduced to $Q(g_i, \tau_i, x_i)$ based on the definitions in the previous section. If $Q(g_i, \tau_i, x_i)$ is preservable for any i, the multivariate problem is divided into n univariate problems and solved thereafter by letting $\tau_i = \tau/n$, based on the fact that $|g(\mathbf{x}) - g(\mathbf{x}')| \leq |g(\mathbf{x}) - g(\widetilde{\mathbf{x_1}})| + |g(\widetilde{\mathbf{x_1}}) - g(\widetilde{\mathbf{x_2}})| + \cdots + |g(\widetilde{\mathbf{x_{n-1}}}) - g(\mathbf{x}')| = \sum_i |g_i(x_i) - g_i(x'_i)| \leq n \cdot (\tau/n) = \tau$. To illustrate the processing steps, we present an example with 2×2 input data $\mathbf{x} = (x_0, x_1, x_2, x_3)^\intercal$ in Fig. 4. □

One can prove that a wide range of QoIs are preservable given Lemma 1, 2, and 3, and the four theorems. For instance, the norm of velocity $v = \sqrt{v_x^2 + v_y^2}$ is preservable because it is composed by a radical function and a polynomial. As a more complex example, if the original data is a probability function, the Kullback–Leibler divergence [35] between the original and decompressed distributions is preservable as it is a weighted average of logarithmic functions.

5 IMPLEMENTATION AND INTEGRATION

In this section, we introduce the implementation of our lossy compression framework and the integration of the example QoIs. Compared with existing error-controlled compressors, the proposed framework features guaranteed QoI error control, non-iterative compression, and high extensibility to new QoIs and predictors.

5.1 Algorithm and Implementation

We present our QoI-preserving compression algorithm in Algorithm 1 based on a modular prediction-based compression framework [43], with the highlighted changes in blue (compared with the general compression algorithms). Specifically, a QoI object is initiated in the beginning (line 1) with user-specified error tolerance τ. In each iteration, the QoI information is used to estimate the pointwise error bound. This estimated pointwise error bound is the minimum of the global error bound and the one derived from QoI (line 3-4). Then, this estimated error bound is quantized using a log-scale quantizer [41] and immediately recovered as the actual pointwise error bound to guide the quantization of data points (line 5-7) using the existing algorithms [53]. Next, an optional sanity check (line 8-12) is performed to eliminate corner cases where the error bound is not met due to round-off errors. By the end of each iteration, tolerance in the QoI is updated using decompressed data (line 13). In the end, the quantizer and encoder for the derived error bounds are both stored to ensure complete information during decompression (line 17 and line 19). Such a design decouples QoI preservation (which includes error bound derivation and error bound compression) with data compression (which compresses data based on the derived error bounds), thus facilitating the integration of the former to diverse prediction-based compression algorithms.

Algorithm 1 QoI-preserving Lossy Compressor

Input: input data d of size n, QoI error τ, global error bound ϵ_g
Output: compressed data cc

1: qoi.init(τ) /*Initiate QoI information*/
2: **for** $i = 1 \rightarrow n$ **do**
3: eb $\leftarrow$ qoi.estimate_eb(d[i]) /*Estimate error bound for data*/
4: eb $\leftarrow \min(\epsilon_g, eb)$ /*Ensure global error bound*/
5: eb, q_e[i] $\leftarrow$ quantizer_eb.quantize(eb) /*Quantize computed error bound*/
6: $p \leftarrow predictor$.predict($d[i]$) /*perform prediction*/
7: $q[i], d'[i] \leftarrow quantizer$.quantize($d[i], p, eb$) /*perform quantization based on the estimated error bound*/
8: /*Optional condition check for QoIs*/
9: **if** qoi.check_compliance(d[i], d'[i]) **then**
10: $eb \leftarrow 0, q\_e[i] \leftarrow 0$ /*Set error bound to 0*/
11: $q[i] \leftarrow 0, d'[i] \leftarrow d[i]$
12: **end if**
13: qoi.update_tolerance(d[i], d'[i]) /*Update the error tolerance for aggregated QoIs*/
14: **end for**
15: $c \leftarrow$ allocate_memory()
16: $predictor$.save(c) /*save predictor*/
17: $quantizer\_eb$.save(c) /*save error bound quantizer*/
18: $quantizer$.save(c) /*save data quantizer*/
19: $encoder\_eb$.encode(q, c) /*perform encoding for error bounds*/
20: $encoder$.encode(q_e, c) /*perform encoding for data*/
21: $encoder$.save(c) /*save encoder*/
22: $cc \leftarrow lossless\_compressor$.compress($c$) /*perform lossless compression*/
23: **return** cc

We integrate the proposed method to two families of predictors, namely multi-layer Lorenzo predictors [53] and interpolation

predictors [62], because recent studies show that they lead to the best rate-distortion under relatively low and high error bounds, respectively. In practice, we notice that the final compression ratio will first increase and then decrease when the global error bound decreases. This is because a very large global error bound will lead to large errors and low prediction accuracy (thus low compression ratios) for certain datasets, while a very small global error bound will over-preserve the data in most cases. As such, we set the most appropriate global error bound based on a sampling approach if it is not specified by the users. Specifically, we start with the largest allowed error bound (or a rough estimation) and use the selected predictors to perform compression on sampled data. The error bound is decreased by half every time and this process is repeated until the current error bound leads to a lower compression ratio compared with the previous iteration.

5.2 QoI Integration

We demonstrate how the examples of the target QoIs can be integrated into our framework. To this end, we further show that our framework is adept at preserving multiple QoIs at the same time.

Quadratic function $f(x) = x^2$: Since this QoI is quadratic, it falls into the polynomial family and the corresponding error bound derivation problem can be solved by the theory in Section 4. According to Corollary 3, a sufficient error bound for this problem would be $Q(f, \tau, x) = -|x| + \sqrt{x^2 + \tau}$ because $f^+(x) = 2|x|$. There is no need to update tolerance in this case as it is a univariate QoI.

Logarithmic mapping $f(x) = \log_2 x$: The sufficient error bound for this QoI can be directly derived using Theorem 2 with $a = 1$, $b = 2$, and $c = 0$. Again, there is no need to update tolerance in this case as it is a univariate QoI.

Regional average of x^2: Based on our theory, this QoI can be treated as $\Pi_{j=1}^{d} n_j$ decomposed QoIs where each of them operates on a data block independently. Specifically, denoting the set of data in the i-th block as Ω_i, the corresponding QoI in this block is translated to $g_i(\mathbf{x}) = \frac{\sum_{x \in \Omega_i} x}{\mathbf{card}(\Omega_i)}$ in this case, where **card** is the cardinality, a.k.a number of data points in the block. As each g_i is a multivariate linear QoI, they can be preserved using Theorem 4. Specifically, the sufficient error bound for the j-th data point in the block would be $Q(g_i, \tau, x_j) = \frac{\tau * \mathbf{card}(\Omega_i)}{\mathbf{card}(\Omega_i)} = \tau$. This reduces to the uniform error bound of τ on all the data points, which certainly preserves the error in the regional average.

We further optimize our method in this case, since the direct derivation above does not take the cancellation into consideration when computing the average, leading to over-conservative error bounds. Specifically, we accumulate an error value $e_i = \sum_{x \in \Omega_i}$ and x is processed $x - x'$ in each block during compression, which sums up the error in all decompressed data at the current stage. Then, a possible better solution could be $Q(g_i, \tau, x_j) = \frac{\tau * \mathbf{card}(\Omega_i) - e_i}{\mathbf{card}(\Omega_i) - j}$ which accounts for both the cancellation and the number of points left. After this data point is processed, the accumulated tolerance will be updated to $e_i = e_i + x_j - x'_j$. As this method takes the cancellation during the summation into consideration, we find in practice that it always leads to better compression ratios than compressing with a uniform QoI error bound τ.

Isoline/Isosurface: Based on the marching cube algorithm and our metrics described in Section 3.1, a possible error bound estimation function for a data point x with respect to an isovalue z could be $|x - z|$. In a general case where multiple isovalues $\{z_i\}$ need to be preserved, the estimated error bound be $\min_i |x - z_i|$. In our implementation, we optimize the estimation process by sorting the isovalues at first and then identifying the most adjacent ones with the binary search for comparison.

Preservation with multiple QoIs: Our framework easily generalizes to preserve multiple desired QoIs. Based on Corollary 1, this can be done by solving the error bounds for all QoIs and choosing the minimal one as the final error bound.

6 EXPERIMENTAL EVALUATIONS

We evaluate our method with the four QoIs mentioned above with four real-world datasets from Scientific Data Reduction Benchmarks [63]. Specifically, we mainly compare the compression quality and QoI error control capability of our method with three state-of-the-art error-controlled lossy compressors, namely SZ-interp [62], ZFP [45], and MGARD [10, 11]. For all compressors we have benchmarked, the latest releases from their master branches were used as of Feb. 1st, 2022. Throughout the evaluation, we use "CR" to denote compression ratio, "S_C" for compression speed in megabytes per second (MB/s), "S_D" for decompression speed in MB/s, "NMAE" for normalized maximal absolute error, and "#FN/#FP/#FS" for the number of false negative/false positive/false shape cells in the isoline/isosurface preservation, respectively.

6.1 Experiment Setup

We evaluate four scientific datasets from diverse applications domains (shown in Table 4), including Hurricane Isabel climate simulation [24], NYX cosmology simulation [12], SCALE climate simulation [44], and QMCPACK quantum Monte Carlo simulation [34].

Table 4: Benchmark datasets

Dataset	#Fields	Dimensions	Size
Hurricane	13	$100 \times 500 \times 500$	1.21 GB
NYX	6	$512 \times 512 \times 512$	3.00 GB
SCALE	12	$98 \times 1200 \times 1200$	6.31 GB
QMCPACK	1	$288 \times 115 \times 69 \times 69$	0.59 GB

All the experiments are conducted on a high-performance cluster [5], where each compute node is equipped with 2 AMD EPYC 7502 processors containing 64 physical cores in total and 128 GB of DDR4 memory. GCC 9.2 is used as the compiler for all the compressors. Any experiment related to compression/decompression speed is performed 5 times and the average values are reported. We present the aggregated results over the datasets in most of the experiments and use two representative data fields, namely Uf48 and Pf48 from Hurricane ISABEL, to demonstrate the effectiveness of our method on error control and isosurface preservation.

6.2 Preservation of x^2

We first compare the compression quality and performance of our method with other state-of-the-art lossy compressors when preserving $f(x) = x^2$. Since none of the existing compressors preserve this

non-linear QoI, we iteratively tune them to make the error of this QoI in their decompressed data as close to the target as possible. This usually requires several compress-decompress-verification processes that are extremely time-consuming and inefficient. We test two modes of MGARD: the L^∞ mode [10] (labeled "MGARD(inf)" in the figures) which better preserves pointwise error and the L^2 mode [9] (labeled "MGARD(0)") with smooth parameter $s = 0$ which better preserves sum of squared errors.

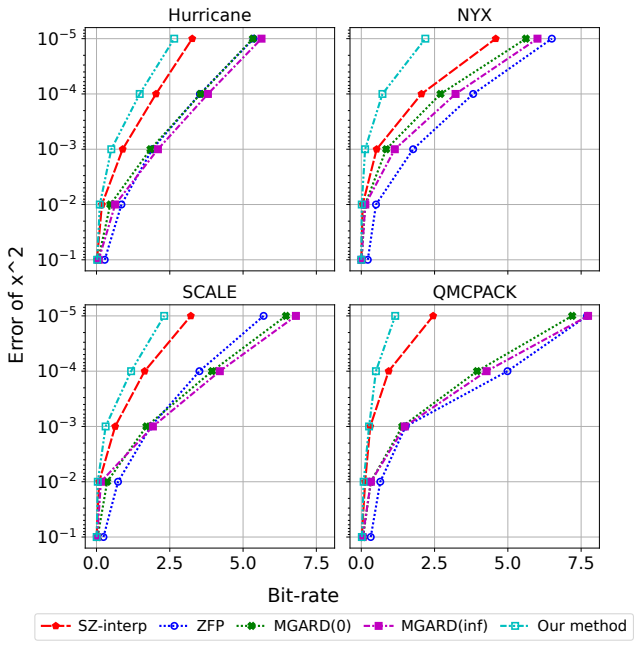

Figure 5: Rate-distortion for preserving $f(x) = x^2$.

Fig. 5 shows the overall rate-distortion of all the evaluated compressors on the four datasets. It is observed that our method always leads to the best compression ratios under the same QoI error bound. In absolute terms, the compression ratios of our method are up to 1.77× (1E-3 on Hurricane), 4.03× (1E-3 on NYX), 2.06× (1E-2 on SCALE), and 2.12× (1E-5 on QMCPACK) of those of the best existing compressors, respectively. Such benefits demonstrate the necessity of leveraging pointwise error bounds against a uniform error bound, because the latter usually over-preserves data in many regions.

We also present the compression/decompression speed in Fig. 6 and Fig. 7, respectively, using log-scale because of the large gap between the fastest and slowest compressors. Similar to previous works [38, 42, 62], ZFP leads in both the compression speed and decompression speed in most cases because its optimized implementation for orthogonal transforms and embedded encoding. SZ-interp is generally slower compared with ZFP, but it outperforms ZFP in some small error bounds because the embedded encoding in ZFP becomes costly in those cases. The two modes of MGARD are usually the slowest, because they use a general implementation that is applicable to broader use cases, e.g., compressing data in non-uniform and unstructured grids. Our method has a slower speed compared with SZ-interp, because it involves extra computation including error bound estimation, quantization, and encoding.

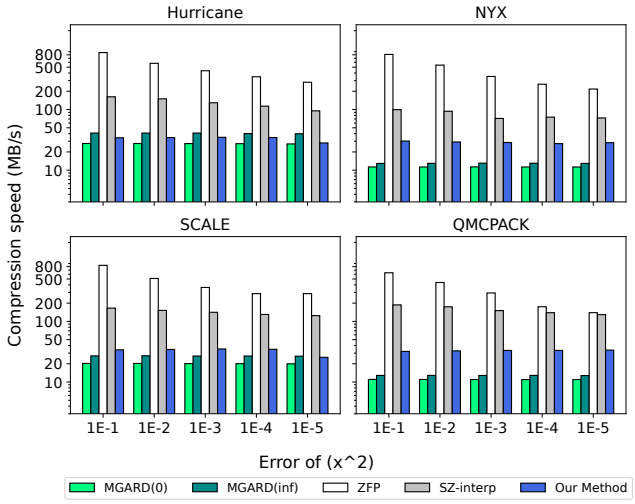

Figure 6: Compression speed of different compressors.

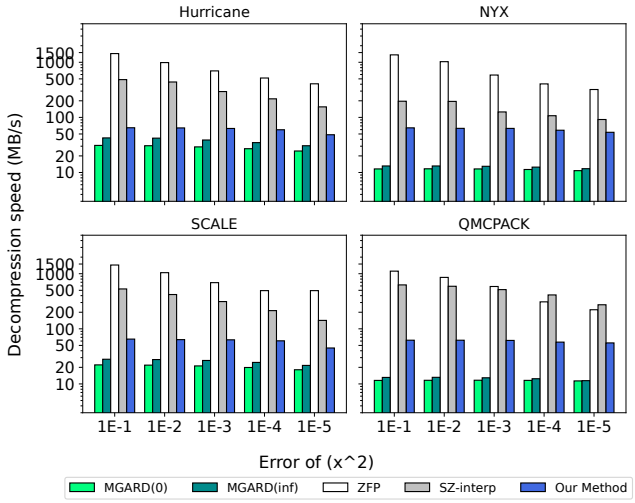

Figure 7: Decompression speed of different compressors.

Since the results above overlook the tuning process, we further perform a more fair comparison between our method and FRaZ [55], a state-of-the-art tuning-based compressor that is able to preserve QoIs via trials and errors. Specifically, FRaZ starts with a preset error bound and compressor, and then iteratively refines the error bound by decompressing the data and checking the tolerance of the target QoI. We present the results with FRaZ in Table 5. Note that we only test FRaZ with SZ and ZFP because our method already delivers comparable speed with higher compression ratios compared to MGARD. According to the table, FRaZ cannot achieve high compression ratios unless lower-bound is set close to the target. The only case FRaZ outperforms our method in compression ratio is when $\tau = 1E - 1$ on Pf48, because the compressed error bounds in our method manifest high overhead when compression ratios become extremely high. Also, FRaZ yields similar (and lower when the number of iterations becomes large) compression speed

Table 5: Preservation of $f(x) = x^2$ compared with FRaZ. MT indicates "manually tuned" results. LB is the lower-bound of acceptable errors in FRaZ and "#Iter" indicates the number of iterations used to find the appropriate setting.

			Uf48					Pf48				
τ	Compressor	LB	#Iter	CR	S_C	S_D	NMAE	#Iter	CR	S_C	S_D	NMAE
1E-1	SZ3-FRaZ	1E-2	3	1034.90	28.72	394.08	7.75E-2	3	2548.36	30.63	397.36	8.35E-2
		5E-2	3	1034.90	28.73	394.08	7.75E-2	3	2526.46	30.66	397.36	8.35E-2
		9E-2	25	1430.62	4.45	399.03	9.73E-2	36	2913.84	3.13	399.03	9.63E-2
	SZ3-MT	-	-	1517.68	154.13	507.85	9.91E-2	-	**3101.16**	173.20	532.42	9.96E-2
	ZFP-FRaZ	1E-2	3	59.46	52.78	472.12	8.19E-2	5	57.24	36.71	489.06	4.90E-2
		5E-2	3	59.46	52.81	472.12	8.19E-2	N/A	-	-	-	-
		9E-2	N/A	-	-	-	-	N/A	-	-	-	-
	ZFP-MT	-	-	59.46	452.44	955.20	8.19E-2	-	44.74	408.32	942.41	2.33E-2
	Our method	-	-	**1619.30**	33.24	65.46	9.79E-2	-	2112.65	33.12	64.88	9.89E-2
1E-2	SZ3-FRaZ	1E-3	3	55.85	24.97	338.18	7.98E-3	3	237.63	27.50	364.00	8.78E-3
		5E-3	3	55.85	28.73	394.08	7.98E-3	3	237.63	27.52	364.00	8.78E-3
		9E-3	25	68.75	3.72	350.62	9.89E-3	66	261.96	1.65	368.21	9.90E-3
	SZ3-MT	-	-	66.71	135.08	422.66	9.61E-3	-	260.82	142.48	443.08	9.87E-3
	ZFP-FRaZ	1E-3	5	16.56	26.06	346.79	7.02E-3	5	25.76	30.21	416.51	8.18E-3
		5E-3	5	16.56	26.02	348.06	7.02E-3	5	25.76	30.20	416.52	8.18E-3
		9E-3	N/A	-	-	-	-	N/A	-	-	-	-
	ZFP-MT	-	-	16.56	289.91	553.51	7.02E-3	-	25.76	336.23	751.05	8.17E-3
	Our method	-	-	**155.79**	33.33	64.22	9.98E-3	-	**329.71**	34.11	64.83	9.98E-3
1E-3	SZ3-FRaZ	1E-4	3	13.47	17.70	172.77	8.09E-4	3	34.77	23.43	294.43	9.07E-4
		5E-4	3	13.45	17.70	172.77	8.09E-4	3	34.77	23.43	293.44	9.07E-4
		9E-4	25	14.06	2.73	138.41	9.88E-4	3	34.77	23.44	293.44	9.07E-4
	SZ3-MT	-	-	13.97	88.12	143.24	9.61E-4	-	37.51	126.81	331.89	9.81E-4
	ZFP-FRaZ	1E-4	3	7.78	29.88	270.16	8.77E-4	3	11.38	35.65	317.89	8.44E-4
		5E-4	3	7.78	29.89	271.70	8.77E-4	3	11.38	35.66	318.95	8.44E-4
		9E-4	N/A	-	-	-	-	N/A	-	-	-	-
	ZFP-MT	-	-	7.78	224.94	378.07	8.77E-4	-	11.38	259.28	510.04	8.43E-4
	Our method	-	-	**27.17**	33.10	60.55	9.99E-4	-	**62.68**	34.07	62.50	9.99E-4

compared to our method because of the costly iterative tuning. This table also demonstrates the tight error bound in our method, as our NMAEs are very close to the target error bounds.

6.3 Preservation of Regional Average of x^2

We then present how the optimization proposed in Section 5.2 improves the compression ratios when the QoI is the regional average of x^2. Again, we first show in Table 6 that our method successfully preserves the regional average of x^2 for various block sizes. It is noticed that the error control becomes looser when the block size increases. This is because larger block sizes lead to less number of blocks and it becomes less likely for the last point in the block to have an error that is close to the error bound.

Table 6: NMAE of regional average of x^2 (τ=1E-3)

Block size	1	2	3	4
Uf48	9.99426E-4	9.99285E-4	9.17413E-4	8.66270E-4
Pf48	9.99293E-4	9.87116E-4	9.05715E-4	7.06676E-4

We also show how this optimization improves compression quality by presenting the rate-distortion in Fig. 8, where the case of block size equals 1 reduces to the preservation of $f(x) = x^2$. According to the figures, the QoI integration with cancellation generally leads to 10% $\sim$ 30% improvement on the compression ratios when the block size equals 4, and this improvement comes with the fact that the regional average applies a uniform weight on all the data in the block, which is not the most advantageous case for this approach. A larger compression ratio gain is expected when a weighted average QoI with non-uniform weights needs to be preserved.

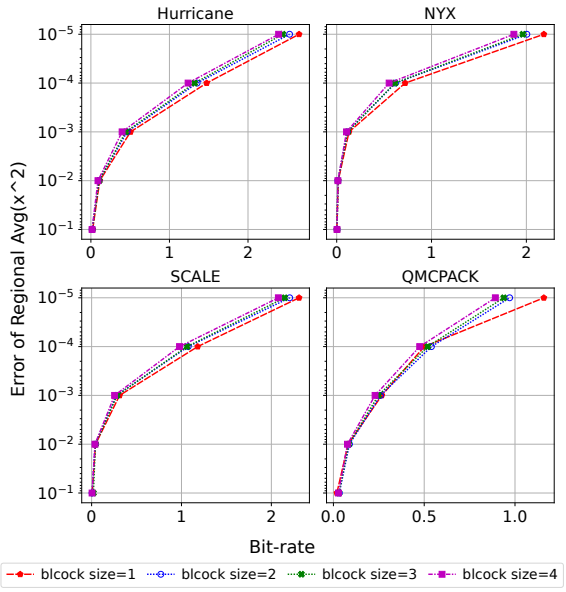

Figure 8: Rate-distortion in terms of the regional average of x^2 using different region block sizes.

6.4 Preservation of Isoline/Isosurface

Next, we present both the quantitative and qualitative analysis of isosurface preservation using the two example data fields. Specifically, we use the number of #FN/#FP/#FS cells for the quantitative

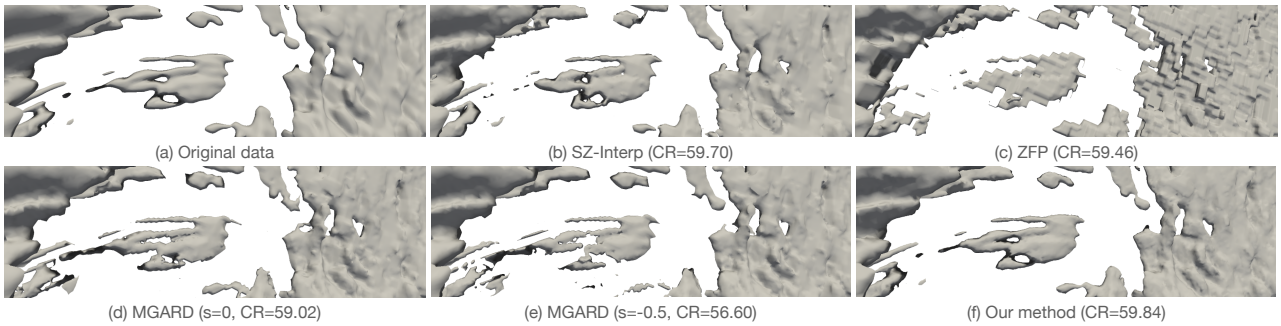

Figure 9: Isosurface of a zoomed-in region on Uf48 (isovalue equals to the mean of data).

study and the visualization result from ParaView [8] for the qualitative study. The compression ratios of all the compressors are tuned to the same level for a fair comparison. For MGARD, we evaluate the modes $s = 0$ and $s = -0.5$, and leave out the L^∞ mode for QoI preservation as recommended by the authors [11].

Table 7: Quantitative error for isosurface (Hurricane Uf48)

Compressor	CR	#FN	#FP	#FS	S_C	S_D
SZ-interp	59.71	33168	43321	369902	147.17	510.35
ZFP	59.46	93798	83806	553660	442.23	931.24
MGARD(s=0)	59.02	29854	40198	354863	26.74	29.85
MGARD(s=-0.5)	56.60	29591	37855	345242	26.82	29.61
Our method	59.84	0	0	0	31.94	61.04

The quantitative results in terms of erroneous cells for the isosurface on Uf48 are presented in Table 7, where the isovalue is the mean of the data. It is observed that, under this compression ratio, all the other compressors have a large number of #FN/#FP/#FS cells, while our method preserves all the cells. However,our method is slightly slower than SZ-interp and ZFP.

Fig. 9 shows the qualitative results over a zoomed-in region of the extracted isosurface with the same setting. Although SZ-interp maintains the rough shape of the isosurface, there are obvious disjoints (e.g., islands on the left), extra joints (islands in the middle top), distorted shapes (e.g., holes on the right), etc. ZFP leads to larger distortions with blockwise artifacts owing to its block-transform-based design. Compared with SZ-interp, the two modes of MGARD have relatively better preservation of the holes on the right, but exhibit larger distortion in the middle. Also, $s = 0$ has better quality in this region because it cares more about high frequencies related to the local details compared with $s = -0.5$, although it has less #FN/#FP/#FS cells in the global view. In contrast, our method keeps almost all isosurface details and exhibits negligible visual distortions.

Table 8: Quantitative error for isosurface (Hurricane Pf48)

Compressor	CR	#FN	#FP	#FS	S_C	S_D
SZ-interp	130.05	17252	51760	98320	152.78	539.46
ZFP	129.71	3798289	33408	358907	730.06	1224.94
MGARD(s=0)	126.17	8665	56591	95248	23.62	24.76
MGARD(s=-0.5)	124.97	8222	55365	92958	23.62	24.64
Our method	133.90	0	0	0	31.75	63.05

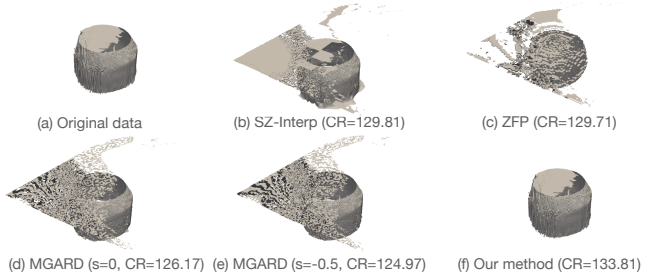

Figure 10: Isosurface on Pf48 (isovalue equals to 0).

We also present the results on Pf48, using a special isovalue 0 because 1) it represents some important features such as critical points in vector fields, and 2) it is easily distorted by existing compressors. The corresponding quantitative and qualitative results are shown in Table 8 and Fig. 10, respectively. According to the figures, severe artifacts are present in almost all the other compressors because the existing compression algorithm easily flushes values to 0, which is also evidenced by the large number of FP cells in Table 8.

6.5 Preservation of Multiple QoIs

Finally, we present the results of the preservation of multiple QoIs. The selected QoIs and error requirements are: (1) $f_1 = x^2$ with normalized error bound 1E-3; (2) $f_2 = \log_2 x$ with error bound 1E-2; (3) f_3 is the isosurface extraction with mean as the only isovalue. We omit the regional average of x^2 because it is overlapped with f_1. Also, our error bound for f_2 is 10× larger than f_1 as it usually poses much more stricter constraints than f_1 under the same error bound. We evaluate all 7 combinations of the three QoIs on the two example data fields, and present the results in Table 9.

As shown in the table, our method faithfully preserves the QoIs as requested. For instance, when f_1 is enabled, the NMAE of x^2 is always smaller than the error bound 1E-3. Nevertheless, such bound is exceeded when f_1 is not enabled, even though both f_2 and f_3 are enabled (row 6). Furthermore, our method successfully preserves multiple QoIs at the same time. It is observed that all the corresponding error bounds are respected when all three QoIs are enabled and some of them might be over-preserved because of the combined constraints.

Table 9: Preservation of multiple QoIs: $f_1 = x^2$ ($\tau = 1E-3$), $f_2 = \log_2 x$ ($\tau = 1E-2$), and isosurface f_3 (mean as the isovalue)

Field	f_1	f_2	f_3	NMAE (x^2)	NMAE ($\log x$)	$\#FN$	$\#FP$	$\#FS$	S_C	S_D	CR
Uf48	✓			9.99E-4	INF	27389	35903	329811	33.16	60.49	27.17
		✓		1.07E-2	1.00E-2	5707	6939	88540	20.94	55.59	6.75
			✓	1.84E-2	INF	0	0	0	38.18	62.80	59.95
	✓	✓		8.57E-4	9.99E-3	4712	5508	71574	19.38	52.69	7.26
	✓		✓	9.99E-4	INF	0	0	0	27.86	59.86	24.21
		✓	✓	1.15E-3	9.99E-3	0	0	0	20.03	52.21	7.34
	✓	✓	✓	8.57E-4	9.99E-3	0	0	0	18.40	52.58	7.20
Pf48	✓			9.99E-4	INF	5949	5236	46651	34.13	62.38	62.68
		✓		2.36E-2	1.00E-2	423	418	6095	23.25	67.76	9.46
			✓	1.85E-2	INF	0	0	0	39.13	64.32	244.96
	✓	✓		9.99E-4	9.99E-3	347	399	5080	20.41	57.73	15.38
	✓		✓	9.99E-4	INF	0	0	0	28.39	62.22	58.54
		✓	✓	2.29E-3	9.99E-3	0	0	0	21.12	57.32	16.01
	✓	✓	✓	9.99E-4	9.99E-3	0	0	0	19.41	57.73	15.35

An interesting trend in the table is that the compression ratio and speed will be dominated by the strictest constraint, which is f_2 in this case. One can see degraded compression ratios and speed every time when f_2 is included (e.g., row 1 and row 4). Nevertheless, an increase in compression ratio is noticed when other QoIs are included in the target QoI sets where f_2 is present (e.g., row 2 and row 4). This is because f_2 imposes a very large error bound on data points with a large value, which may negatively impact the prediction accuracy (because prediction is performed using the decompressed data [53]). Under those circumstances, adding constraints from additional QoIs could decrease the error bounds that are actually assigned to those large values, thus improving the prediction accuracy, and in turn leading to better compression ratios.

6.6 Discussion

Coverage and limitations: The proposed method in this paper provides guaranteed error control for four families of QoIs and any QoI of their composition through certain arithmetic operations. These QoIs cover a wide range of frequently used analyses in scientific simulations, including quantity derivation (e.g., kinetic energy, the magnitude of velocity, and density, etc.) in fusion and cosmology simulation [28], statistical analysis (e.g., mean, standard derivation, and contrast variations, etc.) in climate research [49], and isosurface extraction in almost all the domains [20, 32, 51]. The same idea can be generalized to other analyses, as long as the mapping from the target QoI error to those of raw data is derivable. One limitation is that this method does not apply to QoIs without such an error mapping, which is typical when the QoIs are irreversible. For instance, it does not work for the halo finding analysis in HACC cosmology simulation [29], which requires a random down-sampling step that prevents the derivation of the error bounds. In such cases, an iterative approach would be needed to control the errors.

Trade-offs between compression ratio and speed: There are two urgent needs for data compression in the scientific domains: mitigation of storage overhead and improvement of I/O performance. For the former, higher compression ratios are always preferred. For the latter, the trade-offs between compression ratio and speed need to be considered. As evidenced by previous works, compression ratios have a higher impact when I/O dominates the overall time [38] (which is typical for medium-scale clusters such as [4]), and high compression speed would be more important after compression time occupies a certain portion of the running time (which is observed on high-end computing systems such as [6]). In this work, we focus on developing the theory and implementation of QoI-preserving lossy compression that yields high compression ratios. We will investigate how to improve our compression speed in the future.

7 CONCLUSION AND FUTURE WORK

In this paper, we design and develop a novel lossy compression framework for scientific data that is capable of preserving the errors in the downstream QoIs in addition to those of the raw data. Specifically, we develop rigorous theories to preserve four families of important QoIs, which cover a wide range of data analytics performed by computational scientists in real applications. The theories and methods are further integrated into a state-of-the-art lossy compression framework in a modular fashion, such that new QoIs and novel compression algorithms can be supported easily. Extensive evaluations on four real-world scientific datasets demonstrate that our framework provides strict error guarantees on the underlying QoIs while offering up to 4× compression ratios compared to state-of-the-art error-controlled compressors. In the future, we will work on improving the compression/decompression speed using advanced optimization techniques and/or accelerators, as well as designing efficient database systems with compression for scientific data while providing quality guarantees.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grants OAC-2003709, OAC-2042084/2303064, OAC-2104023, and OAC-2153451. The material was supported by the U.S. Department of Energy, Office of Science and Office of Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357. This research was also supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations – the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nation's exascale computing imperative. This work used the Foundry cluster that was supported by the National Science Foundation under Grant OAC-1919789.

REFERENCES

[1] 2019. ExaFEL: Data Analytics at the Exascale for Free Electron Lasers. https://www.exascaleproject.org/wp-content/uploads/2019/10/ExaFEL.pdf.

[2] 2021. Nyx: An adaptive mesh, cosmological hydrodynamics simulation code. https://amrex-astro.github.io/Nyx/. Online.

[3] 2021. Team at Princeton Plasma Physics Laboratory employs DOE supercomputers to understand heat-load width requirements of future ITER device. https://www.olcf.ornl.gov/2021/02/18/scientists-use-supercomputers-to-study-reliable-fusion-reactor-design-operation. Online.

[4] 2022. Bebop_supercomputer. Available at https://www.lcrc.anl.gov/systems/resources/bebop. Online.

[5] 2022. Foundry Cluster. https://itrss.mst.edu/cluster/foundry.

[6] 2022. Theta Computing System. https://www.alcf.anl.gov/support-center/theta-and-thetagpu. Online.

[7] Robert A Adams and Christopher Essex. 1999. *Calculus: a complete course*. Vol. 4. Addison-Wesley Boston.

[8] James Ahrens, Berk Geveci, and Charles Law. 2005. Paraview: An end-user tool for large data visualization. *The visualization handbook* 717, 8 (2005).

[9] 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, 5-6 (2018), 65–76.

[10] 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.

[11] 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.

[12] Ann S Almgren, John B Bell, Mike J Lijewski, Zarija Lukić, and Ethan Van Andel. 2013. Nyx: A massively parallel AMR code for computational cosmology. *The Astrophysical Journal* 765, 1 (2013), 39.

[13] F Alted. [n.d.]. BlosC compressor. http://blosc.org/. Online.

[14] Andrei Arion, Angela Bonifati, Ioana Manolescu, and Andrea Pugliese. 2007. XQueC: A query-conscious compressed XML database. *ACM Transactions on Internet Technology (TOIT)* 7, 2 (2007), 10–es.

[15] Allison H Baker, Haiying Xu, Dorit M Hammerling, Shaomeng Li, and John P Clyne. 2017. Toward a multi-method approach: Lossy data compression for climate simulation data. In *International Conference on High Performance Computing*. Springer, 30–42.

[16] Rafael Ballester-Ripoll, Peter Lindstrom, and Renato Pajarola. 2019. TTHRESH: Tensor compression for multidimensional visual data. *IEEE transactions on visualization and computer graphics* 26, 9 (2019), 2891–2903.

[17] 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.

[18] Chin-Chen Chang, Jun-Chou Chuang, and Yih-Shin Hu. 2004. Retrieving digital images from a JPEG compressed image database. *Image and Vision Computing* 22, 6 (2004), 471–484.

[19] Choong-Seock Chang. 2017. The fusion code XGC: Enabling kinetic study of multiscale edge turbulent transport in ITER. In *Exascale Scientific Applications*. Chapman and Hall/CRC, 529–552.

[20] Swetaprovo Chaudhuri, Hemanth Kolla, Himanshu L Dave, Evatt R Hawkes, Jacqueline H Chen, and Chung K Law. 2017. Flame thickness and conditional scalar dissipation rate in a premixed temporal turbulent reacting jet. *Combustion and Flame* 184 (2017), 273–285.

[21] Zhiyuan Chen, Johannes Gehrke, and Flip Korn. 2001. Query optimization in compressed database systems. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. 271–282.

[22] W Paul Cockshott, Douglas McGregor, and John Wilson. 1998. High-performance operations using a compressed database architecture. *Comput. J.* 41, 5 (1998), 283–296.

[23] Yann Collet. [n.d.]. Zstandard - Real-time data compression algorithm. http://facebook.github.io/zstd/. Online.

[24] Hurricane ISABEL dataset. [n.d.]. http://sciviscontest-staging.ieeevis.org/2004/data.html. Online.

[25] Luca Deri, Simone Mainardi, and Francesco Fusco. 2012. tsdb: A compressed database for time series. In *International Workshop on Traffic Monitoring and Analysis*. Springer, 143–156.

[26] Peter Deutsch. 1996. GZIP file format specification version 4.3. (1996).

[27] Ali Murat Gok, Sheng Di, Yuri Alexeev, Dingwen Tao, Vladimir Mironov, Xin Liang, and Franck Cappello. 2018. Pastri: Error-bounded lossy compression for two-electron integrals in quantum chemistry. In *2018 IEEE International Conference on Cluster Computing*. IEEE, 1–11.

[28] Qian Gong, Xin Liang, Ben Whitney, Jong Youl Choi, Jieyang Chen, Lipeng Wan, Stéphane Ethier, Seung-Hoe Ku, R. Michael Churchill, C.-S. Chang, Mark Ainsworth, Ozan Tugluk, Todd Munson, David Pugmire, Rick Archibald, and Scott Klasky. 2021. Maintaining trust in reduction: preserving the accuracy of quantities of interest for lossy compression. In *Smoky Mountains Computational Sciences and Engineering Conference*. Springer.

[29] Salman Habib, Vitali Morozov, Nicholas Frontiere, Hal Finkel, Adrian Pope, and Katrin Heitmann. 2013. Hacc: Extreme scaling and performance across diverse architectures. In *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–10.

[30] Katsumi Hagita, Takahiro Murashima, Masao Ogino, Manabu Omiya, Kenji Ono, Tetsuo Deguchi, Hiroshi Jinnai, and Toshihiro Kawakatsu. 2022. Efficient compressed database of equilibrated configurations of ring-linear polymer blends for MD simulations. *Scientific data* 9, 1 (2022), 1–9.

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

[32] Daniel J Isaak and Bruce E Rieman. 2013. Stream isotherm shifts from climate change and implications for distributions of ectothermic organisms. *Global Change Biology* 19, 3 (2013), 742–751.

[33] Jennifer E Kay, Clara Deser, A Phillips, A Mai, Cecile Hannay, Gary Strand, Julie Michelle Arblaster, SC Bates, Gokhan Danabasoglu, James Edwards, Marika Holland, Paul Kushner, Jean-Francois Lamarque, David Lawrence, Keith Lindsay, Adrianne Middleton, Ernesto Munoz, Richard Neale, Keith Oleson, Lorenzo Polvani, and Mariana Vertenstein. 2015. The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability. *Bulletin of the American Meteorological Society* 96, 8 (2015), 1333–1349.

[34] Jeongnim Kim, Andrew D Baczewski, Todd D Beaudet, Anouar Benali, M Chandler Bennett, Mark A Berrill, Nick S Blunt, Edgar Josué Landinez Borda, Michele Casula, David M Ceperley, Simone Chiesa, Bryan K Clark, Raymond C Clay, Kris T Delaney, Mark Dewing, Kenneth P Esler, Hongxia Hao, Olle Heinonen, Paul R C Kent, Jaron T Krogel, Ilkka Kylänpää, Ying Wai Li, M Graham Lopez, Ye Luo, Fionn D Malone, Richard M Martin, Amrita Mathuriya, Jeremy McMinis, Cody A Melton, Lubos Mitas, Miguel A Morales, Eric Neuscamman, William D Parker, Sergio D Pineda Flores, Nichols A Romero, Brenda M Rubenstein, Jacqueline A R Shea, Hyeondeok Shin, Luke Shulenburger, Andreas F Tillack, Joshua P Townsend, Norm M Tubman, Brett Van Der Goetz, Jordan E Vincent, D ChangMo Yang, Yubo Yang, Shuai Zhang, and Luning Zhao. 2018. QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids. *Journal of Physics: Condensed Matter* 30, 19 (2018), 195901.

[35] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[36] 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.

[37] Shaomeng Li, Stanislaw Jaroszynski, Scott Pearse, Leigh Orf, and John Clyne. 2019. VAPOR: A visualization package tailored to analyze simulation data in earth system science. *Atmosphere* 10, 9 (2019), 488.

[38] Xin Liang, Sheng Di, Sihuan Li, Dingwen Tao, Bogdan Nicolae, Zizhong Chen, and Franck Cappello. 2019. Significantly improving lossy compression quality based on an optimized hybrid prediction model. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–26.

[39] Xin Liang, Sheng Di, Dingwen Tao, Zizhong Chen, and Franck Cappello. 2018. An efficient transformation scheme for lossy data compression with point-wise relative error bound. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 179–189.

[40] Xin Liang, Sheng Di, Dingwen Tao, Sihuan Li, Shaomeng Li, Hanqi Guo, Zizhong Chen, and Franck Cappello. 2018. Error-controlled lossy compression optimized for high compression ratios of scientific datasets. In *2018 IEEE International Conference on Big Data*. IEEE, 438–447.

[41] 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.

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

[43] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M Gok, Jiannan Tian, Junjing Deng, Jon C Calhoun, Dingwen Tao, et al. 2022. SZ3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data* (2022).

[44] Guo-Yuan Lien, Takemasa Miyoshi, Seiya Nishizawa, Ryuji Yoshida, Hisashi Yashiro, Sachiho A. Adachi, Tsuyoshi Yamaura, and Hirofumi Tomita. 2017. The Near-Real-Time SCALE-LETKF System: A Case of the September 2015 Kanto-Tohoku Heavy Rainfall. *SOLA* 13 (2017), 1–6. https://doi.org/10.2151/sola.2017-001

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

[46] Peter Lindstrom. 2017. *Error distributions of lossy floating-point compressors.* Technical Report. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

[47] 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.

[48] William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 4 (1987), 163–169.

[49] Andrew Poppick, Joseph Nardi, Noah Feldman, Allison H Baker, Alexander Pinard, and Dorit M Hammerling. 2020. A statistical analysis of lossily compressed climate model data. *Computers & Geosciences* 145 (2020), 104599.

[50] Majid Rabbani. 2002. JPEG2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging* 11, 2 (2002), 286.

[51] Michelle Simões Reboita, Tércio Ambrizzi, Bruna Andrelina Silva, Raniele Fátima Pinheiro, and Rosmeri PorfiRio Da Rocha. 2019. The South Atlantic subtropical anticyclone: present and future climate. *Frontiers in Earth Science* (2019), 8.

[52] 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.

[53] 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*. IEEE, 1129–1139.

[54] Cuneyt Taskiran, Jau-Yuen Chen, Alberto Albiol, Luis Torres, Charles A Bouman, and Edward J Delp. 2004. ViBE: A compressed video database structured for active browsing and search. *IEEE Transactions on Multimedia* 6, 1 (2004), 103–118.

[55] Robert Underwood, Sheng Di, Jon C Calhoun, and Franck Cappello. 2020. Fraz: a generic high-fidelity fixed-ratio lossy compression framework for scientific floating-point data. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 567–577.

[56] Gregory K Wallace. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics* 38, 1 (1992), xviii–xxxiv.

[57] Daoce Wang, Jesus Pulido, Pascal Grosset, Sian Jin, Jiannan Tian, James Ahrens, and Dingwen Tao. 2022. Optimizing Error-Bounded Lossy Compression for Three-Dimensional Adaptive Mesh Refinement Simulations. *arXiv preprint arXiv:2204.00711* (2022).

[58] Till Westmann, Donald Kossmann, Sven Helmer, and Guido Moerkotte. 2000. The implementation and performance of compressed databases. *ACM Sigmod Record* 29, 3 (2000), 55–67.

[59] Ian H Witten, Radford M Neal, and John G Cleary. 1987. Arithmetic coding for data compression. *Commun. ACM* 30, 6 (1987), 520–540.

[60] Feng Zhang, Zaifeng Pan, Yanliang Zhou, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Xiaoyong Du. 2021. G-TADOC: Enabling efficient GPU-based text analytics without decompression. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1679–1690.

[61] Feng Zhang, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Wenguang Chen. 2018. Efficient document analytics on compressed data: Method, challenges, algorithms, insights. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1522–1535.

[62] 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.

[63] Kai Zhao, Sheng Di, Xin Lian, Sihuan Li, Dingwen Tao, Julie Bessac, Zizhong Chen, and Franck Cappello. 2020. SDRBench: Scientific data reduction benchmark for lossy compressors. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2716–2724.

[64] Kai Zhao, Sheng Di, Danny Perez, Zizhong Chen, and Franck Cappello. 2022. MDZ: An Efficient Error-bounded Lossy Compressor for Molecular Dynamics Simulations. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1–12.