FZ-VIS: A Visual Analytics Framework for Quantities-of-Interest-Aware Scientific Lossy Compression

Guoxi Liu, Yuxiao Li, Congrong Ren, Robert Underwood, Xin Liang, Bei Wang, Sheng Di, Franck Cappello, and Hanqi Guo

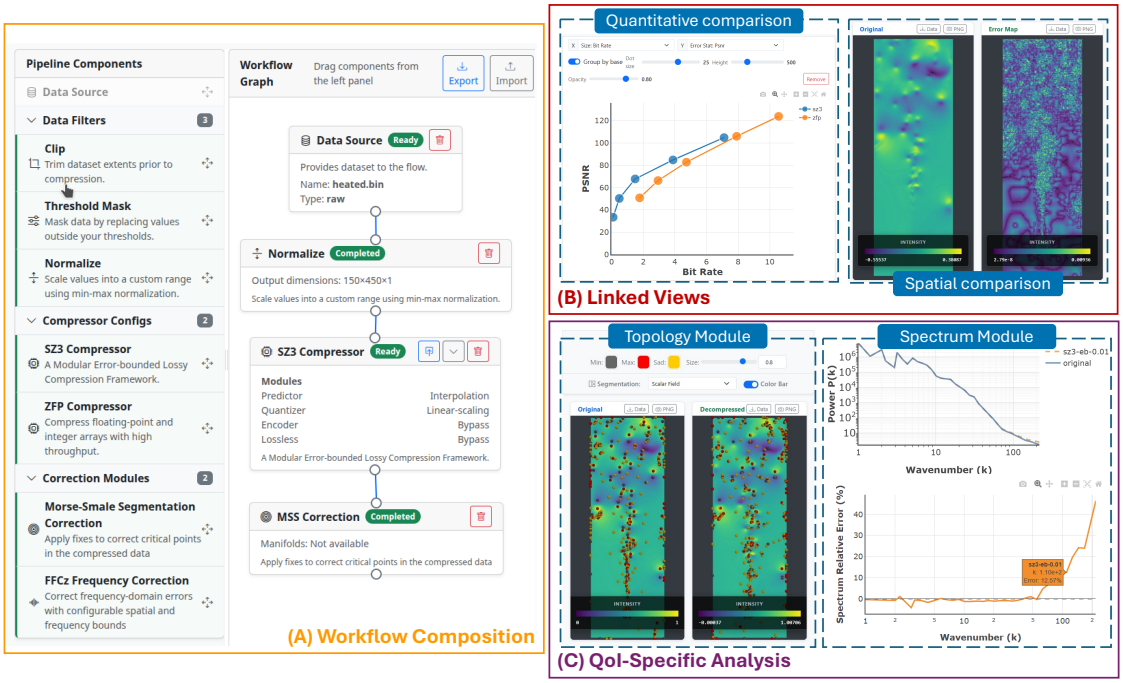


Figure 1: FZ-VIS supports QoI-aware lossy compression through (A) workflow composition, (B) linked quantitative and spatial views, and (C) QoI-specific analysis for topology- and spectrum-related fidelity validation.

Abstract—Modern scientific simulations generate massive volumes of data, making lossy compression essential for efficient storage and transmission. However, preserving critical quantities of interest (QoIs) under lossy compression is inherently data- and task-dependent, requiring domain scientists to navigate complex trade-offs between compression ratio and data fidelity. Exploring these trade-offs often involves large design and evaluation spaces, motivating human-in-the-loop approaches that combine interactive exploration with quantitative analysis. To address this challenge, we present FZ-VIS, an interactive framework for human-in-the-loop ***feature***-oriented lossy ***compression*** design and ***visual*** analytics. FZ-VIS provides a web-based interface for rapidly generating and comparing compression configurations, along with integrated visualization tools for assessing reconstruction fidelity and QoI preservation through both visual inspection and quantitative metrics. We demonstrate the utility of FZ-VIS through case studies involving three representative user groups: novice users selecting compression methods, compressor developers examining internal pipeline behavior, and domain scientists investigating feature preservation. The case studies show how FZ-VIS helps users efficiently navigate complex design spaces and make informed decisions that balance compression performance with application-specific QoI requirements.

Index Terms—Scientific data, lossy compression, feature-preserving compression, error control, quantities of interest, visual analytics

1 Introduction

Modern scientific workflows can generate vast amounts of data when running large-scale parallel applications or performing simulation tasks. In many situations, the volume of data produced is too large to efficiently transmit over networks, store in storage systems, or analyze using standard user tools. For example, the state-of-the-art Energy Exascale Earth System Model (E3SM) [11], funded by the U.S. Department of Energy, provides a kilometer-scale land model to accurately model geographical characteristics and extreme weather occurrences, which can generate terabytes of data from just a single-year simulation. Such large data sizes pose challenges in the visualization, comprehension, and prediction of climate patterns.

To address the challenge of data storage, transmission, and analysis, lossy compressors are considered a practical solution for drastically reducing the data size for scientific research, while still carefully controlling the loss of data accuracy based on user demand. However, scientific datasets often consist of floating-point data arrays, which

- *Guoxi Liu, Yuxiao Li, Congrong Ren, and Hanqi Guo are with The Ohio State University. E-mail: {liu.12722 | li.14025 | ren.452 | guo.2154}@osu.edu.*
- *Robert Underwood, Sheng Di, and Franck Cappello are with Argonne National Laboratory. E-mail: {runderwood | sdi1 | cappello}@anl.gov.*
- *Xin Liang is with Oregon State University. E-mail: lianxin@oregonstate.edu.*
- *Bei Wang is with University of Utah. E-mail: beiwang@sci.utah.edu.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

lossless compressors such as GZIP [7] cannot compress effectively [2]. Therefore, many lossy compressors targeting floating-point data arrays have been developed, some of which are considered ideal solutions for climate simulations by guaranteeing the validity of data after decompression [2, 40]. Furthermore, many error-bounded lossy compression techniques have been proposed to reduce data size while controlling the distortion [23, 28, 30, 31, 34, 54–57]. However, many other scientific research applications also require lossy compression, and the performance of existing lossy compressors on their data remains unclear. Moreover, users need the ability to select the compressors best suited to their specific requirements.

In practice, domain scientists often seek to preserve specific quantities of interest (QoIs) when applying lossy compression. For example, in the analysis of high-energy diffraction microscopy (HEDM) data, peaks associated with grain orientations and strain states in crystalline materials are essential for grain identification and the interpretation of internal strain distributions [26, 51]. Contour trees are critical for supporting a variety of post hoc visualization tasks in applications ranging from materials science to climate simulation [15, 21,58]. Similarly, the power spectrum of cosmological data is a key QoI for analyzing the distribution of matter and energy across spatial scales. These users therefore need a consistent, flexible, and high-performance interface to use and understand the effects of compression on these quantities. LibPressio [47] was developed to provide such a general and uniform user interface for compressor users, simplifying existing compressor workflows and reducing the large design space for applying different compressors. However, it lacks a general graphical user interface for users to easily configure compressors and visually evaluate compressor performance on their data.

Popular visualization tools like ParaView [1], Python Matplotlib [18], and R ggplot2 [50] are widely used for direct and post-processing scientific data visualization. However, they lack native support for lossy compressor composition and comprehensive metric comparisons. While tools such as Z-checker [46] and Foresight [16] address visual evaluation of lossy compressor performance, they overlook QoI evaluation and preservation in scientific applications. To address this gap and streamline QoI-driven lossy compression workflows, we introduce a novel framework that provides an intuitive interface for users to rapidly configure and evaluate hundreds of lossy compressor settings, visually and quantitatively compare their compression quality and performance, and make informed decisions about QoI preservation.

Our main contributions are as follows:

- We formulate QoI-aware lossy compression design as a visual analytics problem that tightly couples the *compressor design space* with the *scientific evaluation space*. This formulation highlights why existing compressor interfaces and evaluation tools are inadequate for supporting QoI-driven compression workflows.
- We introduce FZ-VIS, a **feature**-oriented lossy **compression** design and **visual** analytics framework that supports interactive compressor composition, systematic exploration of configuration series, and linked analysis across quantitative metrics, spatial comparisons, and intermediate compressor outputs within a unified environment.
- We develop QoI-aware analysis workflows for topology- and spectrum-related tasks, allowing users to assess the impact of lossy compression on topological structures and power spectra while reasoning about trade-offs between compression performance and downstream scientific fidelity.
- Through case studies involving novice users, compressor developers, and domain scientists, we demonstrate that FZ-VIS effectively supports diverse decision-making tasks, including compressor selection, pipeline diagnosis, and QoI-aware refinement under application-specific constraints.

2 Related Work

In this section, we review related work from four perspectives: error-bounded lossy compression; compressor design and visual analytics tools; topology- and power-spectrum-preserving compression; and ensemble visualization and parameter-space exploration.

2.1 Error-Bounded Lossy Compression

Lossy compression has become an essential technique for managing the vast volume of scientific data. Traditional lossless compression methods, such as GZIP [7] and Zstandard [59], often fall short in achieving the necessary compression ratios, leading to increased storage and transmission costs. As a result, researchers have turned to lossy compression methods, which sacrifice some degree of fidelity for significantly improved compression performance. Error-bounded lossy compressors can significantly reduce data size while allowing bounded pointwise error between the original and decompressed data within a user-specified error bound, and we refer the reader to [10] for a comprehensive survey.

Error-bounded lossy compressors can be broadly categorized into two types: *transform-based* and *prediction-based*. Transform-based compressors first transform the data into a different domain, such as wavelet transforms and tensor decomposition, and then compress the transformed representation. For example, ZFP [31] uses a custom orthogonal block transform to decorrelate data within blocks, producing sparsely distributed coefficients that can be efficiently encoded. SPERR [24] is a transform-based compressor that employs the CDF 9/7 discrete wavelet transform. MGARD [13] is another transform-based compressor that uses wavelet-based techniques to compress data on uniform and non-uniform structured grids, as well as unstructured grids.

In contrast, prediction-based compressors exploit data correlations by predicting values from neighboring samples and encoding the resulting prediction errors. The SZ family of compressors [8, 9, 28, 30, 45, 55, 56] exemplifies this approach. For example, SZ1.4 [45] uses the Lorenzo predictor combined with linear-scaling quantization to convert prediction residuals into integers, which are then encoded with customized Huffman coding and lossless compressors. SZ3 [30] is a modular, composable compression framework that allows for easy creation and customization of various prediction-based lossy compressors. FPZIP [32] allows a specified number of bit planes to be ignored, making the data distortion controllable on demand. AE-SZ [34] and SRNN-SZ [33] incorporate neural networks to enhance prediction accuracy and compression performance.

2.2 Compression with Feature Preservation

In addition to general-purpose lossy compression techniques, there has been growing interest in developing methods that explicitly preserve important data features. In this work, we focus specifically on features in the topological and frequency domains.

Topological features, such as critical points, persistence diagrams, and Morse-Smale (MS) segmentations, are crucial for understanding the underlying structure of scientific data. Several studies have developed lossy compression techniques that explicitly preserve these features, ensuring that essential topological characteristics are retained after compression. For scalar fields, existing studies have primarily focused on preserving critical points, contour and merge trees. LOCP [12] preserves the relative ordering of neighboring data values, referred to as local order, thereby preserving all critical points while achieving high compression ratios. Yan et al. [53] introduced TopoSZ, which extends the SZ 1.4 compression algorithm by incorporating topological constraints derived from contour-tree-induced segmentations. Gorski et al. [15] proposed a general framework that augments existing lossy compressors with topology-preserving capabilities by quantifying the adjustments required to preserve contour trees and encoding these corrections using a custom variable-precision scheme. Soler et al. [43] developed a topology-controlled compression method that adaptively quantizes data within individual topological features to preserve persistence diagrams under a given persistence simplification threshold. More recently, Xia et al. [52] designed TspSZ to preserve the topological skeleton by extending existing critical-point-preserving compressors to also retain separatrices. MSz [25, 26] represents one of the first efforts to preserve MS segmentations under error-bounded lossy compression using an edit-based paradigm. For vector fields, prior work has primarily focused on preserving critical points during compression. For example, Liang et al. [27, 29] developed a methodology for

preserving critical points in piecewise linear and bilinear vector fields. For tensor fields, TFZ [14] is a topology-preserving compression framework for 2D second-order tensor fields that preserves degenerate points in symmetric fields and eigenvector and eigenvalue graphs in asymmetric fields during lossy compression.

In addition to topology-driven analysis, many scientific applications rely on frequency-domain representations to capture key features. One particularly important Fourier-derived quantity is the *power spectrum*, which describes the power distribution across different frequency components [48]. The shape of the power spectrum curve, characterized by the wavenumber k and power spectrum $P(k)$, provides valuable insight into the underlying structure of the data. For example, peaks in the spectrum indicate dominant periodic components, whereas flat regions suggest white noise with uniformly distributed power. Despite its importance, preserving frequency-domain characteristics during lossy compression remains relatively underexplored. Jin et al. performed fine-grained rate-quality modeling to configure existing compressors [20], and the more recent FFCz [39] introduced a novel algorithm explicitly designed to preserve Fourier-domain fidelity, including the power spectrum accuracy.

2.3 Ensemble and Parameter Space Exploration

Ensemble visualization is a well-established approach for comparing collections of simulation outputs and analyzing their variability. Early work, such as Ensemble-Vis [38], introduced coordinated summary views and linked visual representations to facilitate ensemble analysis. Subsequent studies extended these concepts to uncertainty-aware workflows and multidimensional ensemble analysis [6, 49], emphasizing the importance of statistical summaries and interactive comparison mechanisms.

Parameter space exploration studies how input parameters affect output behavior, quality, and performance. Result-driven and conceptual frameworks emphasize sensitivity analysis, partitioning, optimization, and trade-off reasoning [4, 42]. Recent surveys underline that temporal/spatial outputs amplify these needs and may require compact abstractions for high-dimensional parameter spaces [37].

Several systems further bridge ensemble visualization and parameter space exploration. Schneider et al. integrated data space and model space in ensemble learning using visual analytics, enabling users to reason jointly about parameter settings, model families, and resulting outputs [41]. In scientific simulation, He et al. introduced InSituNet to support post hoc parameter space exploration of ensemble simulations through learned image synthesis, demonstrating the importance of scalable surrogates and interactive visual feedback when exhaustive inspection is infeasible [17].

2.4 Design and Visual Analytics Tools for Compression

Many lossy compressors have been designed specifically for scientific data. However, their diverse user interfaces and numerous configuration parameters pose significant challenges for users seeking to effectively utilize and compare these compression techniques. To unify the interface and streamline the adoption of these lossy compressors, LibPressio [47] provides a standardized and modular API that allows users to seamlessly integrate, configure, and evaluate multiple compression algorithms within a single framework, thereby drastically reducing the complexity of compression workflow management.

For compressor performance evaluation and analysis, several visualization tools have been proposed to support the design and assessment of lossy compressors. Z-checker [46] offers a standardized framework for assessing the compression quality and performance of various lossy compressors, enabling detailed analysis of error metrics and data fidelity. Foresight [16], in contrast, emphasizes extensibility and reproducibility by facilitating the integration and benchmarking of multiple compressors within customizable evaluation pipelines.

Despite these advances, existing systems still leave a gap for QoI-aware compression analysis. LibPressio standardizes access to compressors but mainly serves as an API rather than an interactive visual environment for exploring design alternatives and their effects on downstream tasks. Z-checker and Foresight support comparative evaluation, yet focus largely on aggregate compression metrics and conventional fidelity measures. In contrast, our work treats the lossy compression workflow as a visual analytics problem, enabling users to compose configuration families, inspect intermediate compressor behavior, and understand how standard metrics relate to application-specific QoIs.

3 Background

This section introduces two domain-specific QoIs used to evaluate the effects of lossy compression: topological structure and frequency-domain fidelity. We first summarize Morse-Smale segmentation, and then review Fourier-based spectral analysis and the power spectrum.

3.1 Morse-Smale Segmentation

We provide a concise overview of piecewise linear Morse-Smale segmentations (PLMSS) [35]. As depicted in Figure 2, for a piecewise linear function defined on a constant triangular or tetrahedral mesh, where each data point i has a unique value f_i, PLMSS partitions the data into distinct regions of consistent gradient behavior: all points within the same region follow the steepest ascent and descent paths to the same maximum M_k and minimum m_j, respectively.

Extrema are data points whose scalar values are higher or lower than those of all their neighbors, depicted as red spheres for maxima and blue spheres for minima in Figure 2. *Integral lines* are monotonic paths along mesh edges that follow the steepest ascent or descent direction from a given point until reaching an extremum. *Ascending* and *descending segmentations* group data points that converge to the same minimum and maximum, respectively. PLMSS combines these ascending and descending segmentations to partition the domain into segments. Each segment is defined by a pair of extrema $\langle m_j, M_k \rangle$ and is separated from neighboring segments by integral lines, as illustrated by the differently colored regions in Figure 2.

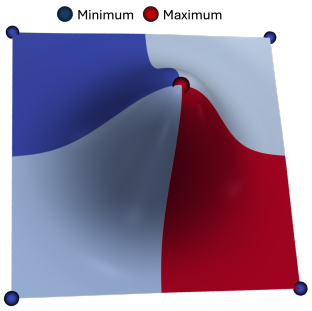


Figure 2: Example of Morse-Smale segmentation. The red and blue spheres represent local maxima and minima, respectively.

MSz [25] preserves PLMSS in error-bounded lossy compression by iteratively correcting topological distortions in decompressed data. It takes as input a 2D or 3D scalar field f and its decompressed version $\hat{f}$ obtained from an arbitrary error-bounded lossy compressor with a global error bound ξ. The algorithm outputs a set of edits $\delta = \{\delta_i\}$ applied to $\hat{f}$, yielding a corrected field $g = \hat{f} + \delta$ that preserves the PLMSS while strictly satisfying the global error bound $|f_i - g_i| \leq \xi$ for all data points. The core idea of MSz is an edit-based strategy that identifies a subset of points in the decompressed data whose values can be modified within the user-specified error bound to correct topological distortions. More specifically, the method iteratively detects topological distortions, identifies the data point i responsible for each distortion, and applies an edit δ_i to restore topological consistency.

3.2 Fourier and Spectral Analysis

Techniques based on Fourier transforms are widely used for feature identification and pattern detection. The *Discrete Fourier Transform* (DFT) [44] converts a discrete signal from the spatial domain to the frequency domain. For a 1D dataset with N data points, denoted by $\{x_n\}_{n=0}^{N-1}$, DFT computes its frequency components X_k as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}}, k \in \{0, 1, \ldots, N-1\}.$$

DFTs for higher dimensions adhere to a similar pattern, incorporating additional exponential terms for each dimension.

The *power spectrum*, denoted as $P(k)$, illustrates the allocation of power among various frequency bands. For a discrete 3D dataset with values $\{x_{m,n,p}\}$, we first normalize the variations by removing the mean background as follows:

$$x'_{m,n,p} = \frac{x_{m,n,p} - \bar{x}}{\bar{x}},$$

where $\bar{x}$ is the average value across $\{x_{m,n,p}\}$. This ensures the power spectrum reflects relative structure and variability rather than absolute magnitude. We then shift the zero-frequency component of $\{x'_{m,n,p}\}$ to the center for symmetry between positive and negative frequencies [36]. The spatial data is then mapped to the frequency domain: $X'_{u,v,w} = FFT(x'_{m,n,p})$. The power spectrum accumulates the squared magnitudes of frequencies at equivalent distances from the center:

$$P(k) = \sum_{u^2+v^2+w^2=k^2} |X'_{u,v,w}|^2,$$

where $|\cdot|$ indicates the magnitude of a complex number.

FFCz [39] preserves both spatial and frequency-domain accuracy by applying a post hoc correction to decompressed data under user-defined error bounds in both domains. It represents the error of each frequency-component as a linear combination of spatial errors, reformulating the dual-domain constraints as two intersecting hypercubes. Starting from the spatial error vector produced by an off-the-shelf compressor, FFCz iteratively projects the vector onto the two hypercubes until it converges within their intersection and satisfies both constraints. Because FFCz supports independent tolerances for individual spatial and frequency components, it can preserve the power spectrum by applying specific pointwise relative error bounds to the frequency domain.

4 Requirement Analysis

We derived the following requirements from formative discussions with colleagues who use or study lossy compression in different roles. These discussions helped us identify three primary user groups shown in Figure 3: compressor developers (*D*), novice users (*U*), and QoI-driven domain scientists (*S*). Compressor developers focus on creating, debugging, and tuning compressors. Novice users, such as trainees or students, often learn to apply compressors effectively, but may have limited knowledge of compressor internals. QoI-driven domain scientists care less about the compressor implementation itself and more about whether compression preserves the scientific structures, features, or derived quantities needed by downstream analysis.

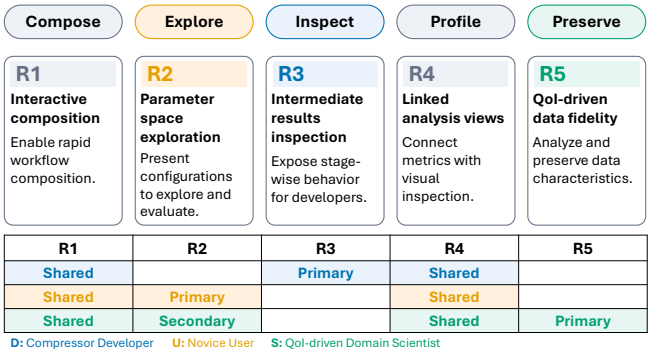


Figure 3: Design requirements derived from three user groups: compressor developers, novice users, and QoI-driven domain scientists.

R1 (*D, U, S*) — Interactively composing compressor workflows. Lossy compressors expose many interchangeable modules and parameter choices, and even small configuration changes can substantially impact compression ratio, throughput, and reconstruction quality. All three user groups therefore need an intuitive interface for composing compressor workflows, inspecting available parameters, and modifying configurations without repeated code changes. Such support shortens the iteration cycle for developers, lowers the barrier for novice users, and allows domain scientists to quickly test candidate configurations against their analysis needs.

R2 (*U, S*) — Exploring parameter spaces of compressors. This requirement is primary for novice users and also relevant to QoI-driven domain scientists. Novice users, such as trainees and students, often enter the workflow with practical learning goals, such as understanding how compressor parameters affect compression ratio, throughput, and reconstruction quality, but they may not yet know which parameters matter most or how different settings interact. QoI-driven domain scientists also need to explore parameter spaces when searching for configurations that preserve application-specific features under practical performance constraints. Therefore, the system should expose the parameter space in an interpretable manner, helping users move from educational or analytical goals to a manageable set of candidate configurations. Rather than requiring them to tune parameters blindly, it should reveal promising regions of the design space, highlight sensitive parameters and trade-offs, and support iterative exploration from coarse comparisons to focused refinement.

R3 (*D*) — Inspecting intermediate results for debugging and optimization. Compressor developers need to understand not only the final compression metrics, but also how each stage of the compressor pipeline behaves internally. They therefore require a system that can expose intermediate results, such as predictions, quantization errors, residuals, encoded outputs, or stage-wise statistics, so that failures can be localized to specific modules or parameter choices. This requirement helps developers diagnose unexpected behavior, compare alternative module designs, and reason about how local changes in the pipeline propagate to the final compression performance.

R4 (*D, U, S*) — Profiling compressor behavior through linked analysis views. All user groups need to evaluate the trade-offs induced by lossy compression, but they inspect them from different perspectives: developers study algorithmic behavior, novice users compare configurations to balance performance and quality, and domain scientists verify whether the reconstructed data remains valid for downstream analysis. To support these needs, the system should provide coordinated views of standard performance and fidelity metrics together with direct visual comparison between original and decompressed data. Linking these analysis views helps users move from high-level aggregate summaries to detailed inspection when selecting or rejecting configurations.

R5 (*S*) — Analyzing and preserving QoI-driven fidelity. Pointwise error metrics alone are insufficient when downstream tasks depend on preserving specific quantities of interest, such as extrema, Morse-Smale segmentations, or spectral characteristics. QoI-driven domain scientists need analysis tools that expose how compression affects these domain-specific quantities, allow them to compare candidate configurations under QoI-aware metrics, and support region- or feature-specific inspection when failures occur. More importantly, the system should assist users in reasoning about the trade-off between conventional compression objectives and QoI preservation so they can select compressors and settings that satisfy the demands of their specific scientific tasks.

5 System Design

This section describes how FZ-VIS addresses the requirements mentioned in Section 4. FZ-VIS is designed as a human-in-the-loop framework for QoI-aware lossy compression rather than as a collection of isolated interface components. In practice, users do not simply choose a compressor to run once and inspect the final compression ratio and execution time. Instead, they iteratively define an evaluation context, construct candidate configurations, execute batches of experiments, inspect quantitative and visual results, and then refine the search space based on what they learn. FZ-VIS is designed around this iterative loop so that the compressor design space and the evaluation space remain tightly connected throughout the workflow.

At a high level, FZ-VIS serves three complementary user groups. For novice users, it exposes the parameter space of compressors through a guided user interface and supports comparison among candidate configurations that satisfy target performance or fidelity requirements.

For compressor developers, it provides a structured workspace for pipeline composition, variant management, and inspection of intermediate results. For domain scientists focusing on QoI in scientific applications, it integrates different feature analysis modules into the compression workflow so that QoI preservation becomes a proactive decision instead of an external post hoc validation step.

5.1 System Overview

Figure 4 summarizes the architecture of FZ-VIS. The system is organized into three major stages: *setup*, *composition and execution*, and *analysis and refinement*.

- The *setup* stage establishes the experimental context by defining datasets (e.g., climate simulation data, X-ray CT images, cosmology datasets) with necessary pre-processing operations, standard compression metrics (e.g., bit rate, PSNR, DSSIM), and key QoIs (e.g., local maxima, Morse-Smale segmentations, power spectra). This ensures that experiments are grounded in specific scientific and performance requirements.
- In the *composition and execution* stage, users construct and configure compressor pipelines (e.g., SZ3, ZFP) through a guided interface, exploring module options and parameter combinations. This includes specifying different error bounds and managing large families of configurations. Experiments are then executed, recording not only final results but also essential intermediate artifacts for debugging and understanding compressor behavior.
- The *analysis and refinement* stage presents linked quantitative views (e.g., rate-distortion curves, frequency error plots) and visual views (e.g., error maps, distributions of critical points, Morse-Smale segmentation boundaries). These views enable users to compare, filter, and refine the design space, making informed decisions on compressor selection to meet specific application needs. This adaptive feedback loop ensures a systematic and iterative refinement of compressor configurations, directly supporting scientifically grounded decisions.

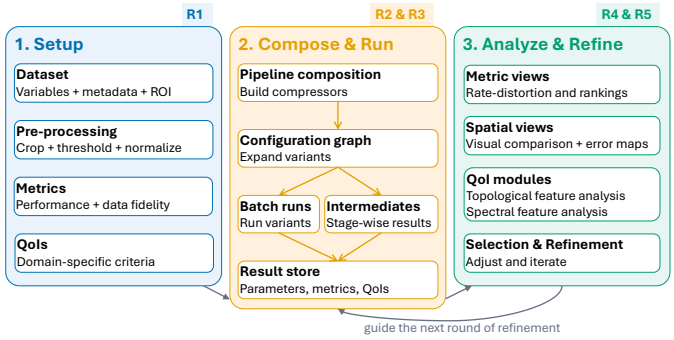


Figure 4: Workflow-oriented architecture of FZ-VIS. The workflow consists of three stages: setup, composition and execution, and analysis and refinement. Each stage produces structured outputs that serve as inputs to the next, while analysis results inform subsequent rounds of configuration refinement.

This workflow-centered organization is important because it bridges two spaces that are often disconnected in existing tools. On one side is the *compressor design space*, which consists of module choices, parameter values, and execution variants. On the other side is the *scientific evaluation space*, which consists of performance metrics, visual fidelity, and downstream QoIs. In conventional workflows, users often move between command-line scripts, benchmark tables, plotting tools, and domain-specific analysis codes. FZ-VIS unifies these activities in a single environment so that decisions made in one space can be immediately interpreted in the other.

5.2 Design Space Setup

FZ-VIS frames compression tuning as a structured design-space exploration task rather than as a sequence of disconnected trial-and-error runs. At the beginning of a study, users define an experiment specification that fixes the dataset, required pre-processing, optional region-of-interest (ROI), evaluation metrics, and any QoI analysis to be applied. This specification establishes a stable analytical context in which many related compressor variants can be generated and compared without repeatedly reconstructing the surrounding workflow.

The purpose of this setup stage is to make the exploration process explicit and reproducible. Instead of executing one compressor configuration at a time, users start from a default pipeline and branch into related alternatives by varying modules, parameters, or shared settings. The resulting variants remain part of the same study definition, so each configuration is interpreted as one point in a larger design space rather than as an isolated benchmark result.

FZ-VIS also preserves provenance for every explored configuration, including its dataset, pipeline structure, parameter values, execution outputs, derived metrics, and QoI summaries. This provenance is critical for iterative refinement. As illustrated in Figure 5, the workflow graph is translated into a generated JSON specification that records the same case study definition, enhancing the reproducibility and traceability of the experiment.

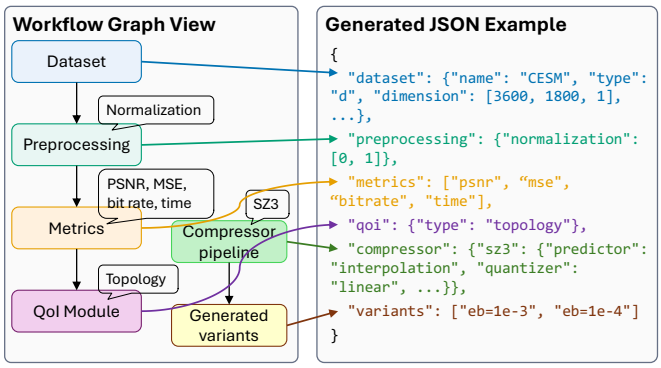


Figure 5: Workflow graph and generated JSON specification in FZ-VIS. The workflow graph defines the experiment structure, including data preparation, evaluation metrics, QoI modules, and compressor variants, while the generated JSON records the same context as an experiment specification. The correspondence between them makes the interactive workflow design reproducible.

5.3 Compressor Configuration Generation

A core component of FZ-VIS is the composition of compressor pipelines. It exposes the available modules and parameters of a compressor through a unified interface provided by LibPressio [47], allowing users to configure internal pipeline modules, such as the predictor, quantizer, encoder, and lossless stage in the SZ3 compressor, without directly editing code or scripts. From the perspective of **R1**, this interface shortens the iteration cycle by turning low-level compressor configuration into direct interactive manipulation.

FZ-VIS is designed to support both single-configuration composition and large-scale configuration generation. A user may begin with a reference compressor pipeline and then derive multiple alternatives by varying modules and parameter ranges. As illustrated in Figure 6, the reference pipeline acts as a starting point, while parallel coordinates summarize the candidate choices for different modules. The generated configurations are then organized as parent nodes in a force-directed graph. Each parent configuration can further expand into sub-nodes by propagating parameter values (e.g., error bound) or common execution options (e.g., single-thread, multi-thread, or GPU-accelerated).

This design is particularly important for novice users addressed by **R2**. Such users may have desired compression requirements, e.g., a minimum compression ratio, a throughput threshold, or a target quality measure, but they often do not know which parameter combinations are worth testing. FZ-VIS therefore does not force them to tune one parameter at a time in isolation. Instead, it presents the parameter space in a form that can be explored through branching, propagation, and

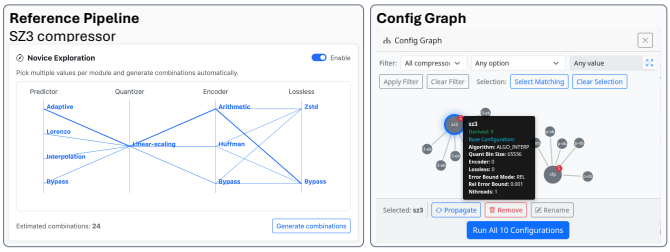


Figure 6: Compressor configuration generation in FZ-VIS. Starting from a reference compressor pipeline, users can combine candidate module options using parallel coordinates. The resulting configurations are organized as a force-directed graph, where each parent configuration can further expand into sub-nodes by propagating parameter values or common execution options.

side-by-side comparison. The resulting configuration space is explicit and inspectable, making it easier for users to search for candidate settings that match their requirements.

5.4 Intermediate Inspection and Execution Management

Once a set of configurations has been defined, FZ-VIS executes the corresponding compression pipelines and records the resulting outputs. The execution layer manages the mapping from a logical compressor configuration to a concrete experiment instance, including the input dataset and the metrics that need to be collected. This stage transforms interactive design decisions into executable experiments that can be compared consistently.

In addition to final compression statistics, FZ-VIS also supports the capture of intermediate results and stage-wise summaries. This capability is essential to **R3**. For compressor developers, final metrics alone are often insufficient to explain why two related configurations behave differently. Taking SZ3 as an example, a change in the predictor, quantization behavior or entropy characteristics may propagate through the compressor in ways that are not visible from the final compression metrics alone.

This stage therefore serves both experimental execution and debugging. Developers can inspect whether changes in the compressor structure alter residual distributions, whether a stage becomes a performance bottleneck, or whether a configuration behaves anomalously compared with other variants. In this sense, FZ-VIS is not just a benchmark dashboard; it is also a debugging and optimization platform for compressor development.

5.5 Linked Quantitative and Spatial Analysis

After experiments are executed, FZ-VIS presents the results through linked quantitative and spatial comparison views. The quantitative comparison view enables users to compare candidate configurations across user-selected metrics, such as bit rate, PSNR, compression ratio, and runtime. It summarizes the tested configurations using plots such as rate-distortion diagrams, trade-off curves, and ranked candidate lists. Users can choose which quantitative measures to compare so that the same result set can support different tasks. For novice users, these views help identify configurations that satisfy target requirements. For developers, they reveal algorithmic trends across compressor variants and expose the trade-offs among fidelity, rate, and execution cost.

These quantitative summaries are tightly coupled with direct visual inspection of the data. FZ-VIS supports side-by-side comparison between original and decompressed data, error-map visualization, and ROI-focused inspection. Linking these visual and quantitative views directly addresses **R4**.

As illustrated in Figure 7, a user can first select a candidate in the quantitative view, inspect its corresponding original, decompressed, and error-map views, and then compare ROI-focused details across multiple candidates before returning to the configuration space to refine the search. This coordination between high-level summaries and localized visual comparisons is essential because numerical measures alone may fail to reveal spatially concentrated artifacts, while visual inspection alone does not scale to large configuration spaces.

Another important aspect of the linked analysis design is selective comparison. Rather than treating all configurations equally at all times, FZ-VIS lets users focus on subsets of configurations that are relevant to the user requirements. For example, a user may examine a region in the quantitative plot, and then compare only those candidates in the ROI-focused view. This filtering-and-linking mechanism progressively narrows the design space and helps users move from coarse exploration to targeted decision-making.

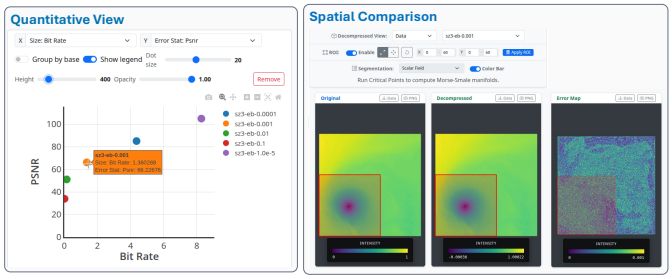


Figure 7: Linked quantitative comparison and spatial analysis in FZ-VIS. The quantitative view on the left shows the metric plot with customizable x- and y-axes, allowing users to select promising configurations from the tested design space. The spatial comparison view on the right shows the original data, the reconstructed data, its error map, and ROI-focused comparison, supporting localized inspection and iterative refinement of the candidate set.

5.6 QoI-Specific Analysis Modules

To support domain scientists, FZ-VIS extends beyond generic compression metrics and includes dedicated QoI-specific modules. In the current design, three complementary capabilities are emphasized: topological analysis, spectral analysis, and QoI correction. These modules are integrated into the same environment as the standard performance and fidelity views, so users do not need to export results into separate tools before determining whether a compressor is acceptable for downstream analysis.

The topological analysis module evaluates structures such as critical points and Morse-Smale segmentations. It enables users to compare how different compressor configurations change topological structures, such as the number of false critical points introduced, and which settings preserve the relevant topological structures. This is especially important in applications where downstream analysis depends more on structural correctness than on global pointwise error.

The spectral analysis module supports applications in which the frequency-domain representation is a primary scientific quantity. By computing and comparing power spectra and related spectral relative errors, FZ-VIS allows users to inspect how compression affects frequency-domain representations in addition to spatial fidelity. Users can therefore examine whether a configuration that appears acceptable under spatial error bounds still introduces unacceptable distortion in the frequency domain.

Beyond analysis, FZ-VIS also provides QoI correction modules to assist users when a candidate configuration fails to preserve the target QoI. Instead of treating QoI evaluation as the end of the workflow, these modules use the detected discrepancy to suggest corrective actions, such as applying MSz [25] to fix Morse-Smale segmentation and FFCz [39] to retain the power spectrum. These modules therefore extend FZ-VIS from an evaluation system into an iterative refinement framework, enabling users to retain desirable candidate configurations by applying QoI-specific corrections.

Above all, these QoI-oriented modules address **R5** by making QoI preservation an active part of the decision process for domain scientists. Rather than selecting a compressor based only on compression metrics and then validating QoI preservation afterward, users can evaluate candidate configurations under QoI-aware criteria, inspect where and why QoI failures occur, and apply QoI-guided corrections. This

unified design supports scientifically grounded compressor selection and enables FZ-VIS to be extended with additional application-specific QoI modules in the future.

6 Case Study

The key advantage of FZ-VIS is that its linked views guide users through compression design. Rather than assuming that users already know which compressors, parameters, or QoI criteria to prioritize, the system supports an iterative workflow: generating candidates, comparing quantitative and spatial evidence, inspecting failures, and refining the search space. In this section, we demonstrate how FZ-VIS satisfies different requirements through several case studies. These case studies are intended to illustrate integrated workflow support, while achieving the same sequence of configuration, comparison, QoI analysis, and refinement would otherwise require substantially more manual scripting and movement across separate tools.

6.1 Compressor Selection for Novice Users

Novice users, such as trainees and students who are new to lossy compression, may be unfamiliar with compressor parameter spaces and how these parameters affect performance. However, they often still have concrete goals for compression behavior, such as target compression ratio and throughput. In this subsection, we use a climate application as a case study to demonstrate how FZ-VIS supports these users in selecting appropriate compressor configurations.

Requirements. For trainees and students who are new to lossy compression, an important early task is to understand how compressor choices relate to application-specific goals. We illustrate this learning process through a climate application. Climate simulation models have grown increasingly complex as computing resources have advanced, producing far more data than scientists can afford to store in full. As a result, climate users often need lossy compression solutions that substantially reduce storage cost while preserving data quality for downstream analysis. In this case study, we use the CLDHGH field from the CESM [19] dataset. A key quality metric for this application is the Data Structural Similarity Index Measure (DSSIM) [3], and a practical goal is to achieve a compression ratio roughly 2–3× higher than lossless compression [5]. These requirements define the decision context for novice-oriented compressor selection.

Design Phase. A novice user typically does not know which compressor family or predictor to try first. FZ-VIS addresses this uncertainty by exposing candidate SZ3 module choices and ZFP settings through a guided composition interface, producing a small starting set instead of many scripted runs. The user begins with three SZ3 predictors (adaptive, Lorenzo, and interpolation-based predictors) and compares them with a ZFP baseline under the same default 1% relative error bound. The linked rate-distortion and spatial views in Figure 8 show not only that the SZ3 variants achieve lower bit rates while ZFP attains higher PSNR, but also how these differences appear spatially in the reconstructed field. This first comparison gives the user a basis for deciding which configurations merit broader exploration.

Evaluation Phase. After this initial comparison, the user still does not know which error-bound range best balances quality and compression. FZ-VIS supports the next step by expanding the study across a broader range of error bounds (i.e., from 10^{-2} to 10^{-5}) and aggregating the results in linked evaluation views. The user can switch among metric pairs, inspect where candidate points cluster, and use the ranked view to focus on promising regions of the design space. When PSNR is plotted against bit rate in Figure 9, the rate-distortion view confirms that SZ3 generally achieves lower bit rates, while ZFP tends to produce higher PSNR under similar error-bound values. The view thus supports targeted selection rather than ad hoc trial and error.

For climate applications, DSSIM and compression ratio are also important, so the workflow continues by switching the quantitative comparison from PSNR against bit rate to DSSIM against error bound. This interaction lets the user filter the candidates that satisfy DSSIM greater than 0.95 and then compare compression ratios only among the admissible configurations. Under this requirement, SZ3

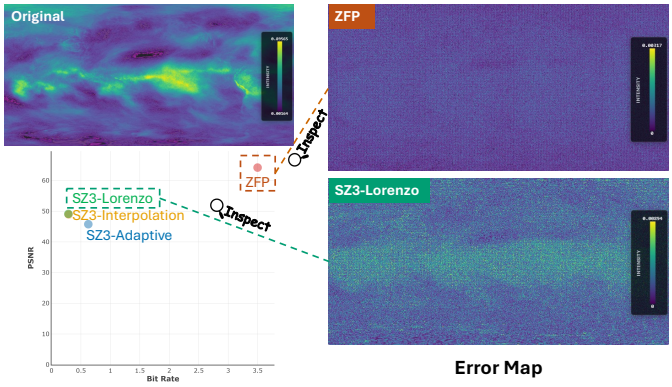


Figure 8: Comparison of SZ3 using different predictors and ZFP on the CESM dataset. Left: the original field visualization and rate-distortion scatter plot of the tested compressor configurations. Right: error maps of decompressed data from SZ3 and ZFP at a 1% relative error bound.

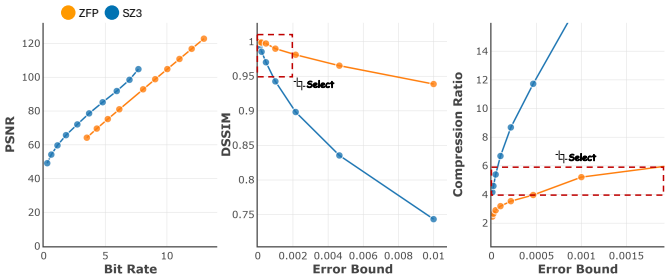


Figure 9: Performance comparison of SZ3 and ZFP on the CESM dataset across multiple error bound values. The figure summarizes the evaluated compressor instances using linked metric views, enabling users to compare trade-offs among bit rate, PSNR, DSSIM, and compression ratio.

achieves a higher compression ratio than ZFP. For example, if the target compression ratio is around 4×, the user may choose SZ3 with a relative error bound of 2.15×10^{-5}, which yields a DSSIM of 0.99848 and a compression ratio of 4.6.

6.2 Compressor Analysis for Developers

Compressor developers need to understand the internal behavior of a compressor in order to improve its pipeline, rather than simply choose among existing options. In this case study, we show how FZ-VIS supports SZ3 developers by exposing intermediate outputs such as prediction residuals, quantization indices, and stage-wise runtime. These views help developers diagnose why a new module succeeds or fails and identify the next implementation steps.

Requirements. Consider a realistic development scenario in which the SZ3 team evaluates an alternative regression-based predictor against the baseline Lorenzo predictor using the Hurricane Isabel dataset. To evaluate this new algorithm, developers need to determine whether the regression predictor improves residual structure, whether any such benefit survives quantization and entropy coding, and where it fails spatially. More specifically, they need tooling that reveals whether performance gains are confined to locally smooth regions, whether sharp transitions create large residual and quantization outliers, and whether the resulting compression-throughput trade-off justifies adoption or further optimization.

Design Phase. Developers begin with the hypothesis that the regression-based predictor offers advantages, but they lack detailed insight into where it succeeds or fails within the pipeline. FZ-VIS supports this workflow by enabling them to evaluate the Lorenzo and regression-based predictors under identical settings, including error bound, quantizer, encoder, and lossless backend, while exposing intermediate artifacts for direct comparison. By inspecting residual

and quantization-index distributions alongside stage-wise runtime breakdowns, developers can seamlessly trace behavior from prediction quality to downstream coding efficiency and runtime cost, rather than analyzing each metric in isolation.

Evaluation Phase. The linked views guide the developers through a staged diagnosis of the pipeline's performance. The residual histogram in Figure 10 shows that while Lorenzo produces a noticeably higher concentration of near-zero residuals, the regression predictor exhibits a heavier residual tail. The quantization-index distributions then confirm that this weaker residual behavior persists downstream, translating directly into a wider index spread and poorer compression effectiveness. Finally, the stage-wise runtime breakdown in Figure 11 demonstrates that the regression-based predictor does not offer a sufficient throughput advantage to offset its lower compression ratio. By moving from initial residual evidence to downstream coding behavior and final runtime costs, this workflow clearly illustrates why the baseline Lorenzo predictor remains the better choice for this particular dataset and pipeline configuration.

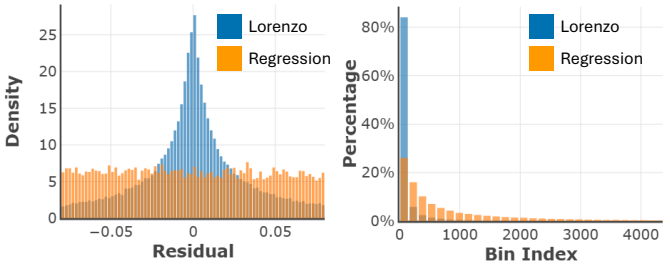
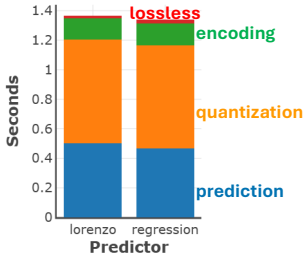


Figure 10: Comparison of the regression-based predictor and Lorenzo predictor in SZ3. Left: distribution of prediction residuals. Right: distribution of quantization indices.



	Lorenzo	Regression
Compression Ratio	3.6199	2.4146
Bits/value	8.8400	13.2528
Prediction time (s)	0.5002	0.4684
Throughput (values/s)	9.195e+6	9.062e+6

Figure 11: Stage-wise runtime comparison and additional metrics of the regression-based and Lorenzo predictors in SZ3 compressor.

6.3 Topological Feature Preservation

Lossy compressors facilitate substantial data reduction while maintaining a predefined accuracy threshold. However, they may fail to preserve topological features, such as critical points, Morse-Smale (MS) complexes, and merge trees in decompressed data, even with bounded error [29, 53]. This case study shows how FZ-VIS guides domain scientists from diagnosing QoI failure to selecting a correction strategy. Specifically, we demonstrate how the framework enables users to choose a compressor that preserves specific topological features, including extrema and MS segmentations, based on their requirements.

Requirements. In the analysis of Atmospheric Rivers (ARs), elongated bands of water vapor transport that originate from the tropics to North America and cause flooding, scientists characterize the skeleton of ARs with MS complexes and MS segmentations to better understand the formation and development of ARs [22]. Distortions in such topological features could significantly impact scientists' understanding of AR formation and development, potentially leading to inaccurate evaluations of ARs' impact on precipitation and flooding in North America. Scientists characterize AR skeletons by the boundaries of descending segmentations of the Integrated Vapor Transport (IVT) field. Therefore, scientists need to choose a compressor that preserves MS segmentations for downstream data analysis tasks.

Design Phase. At the outset, the scientist cannot predict whether simply tightening the error bound will recover the MS segmentation. FZ-VIS supports this diagnosis by coupling a quantitative QoI view with a linked spatial view. Through this interface, the user explores compression configurations across error bounds ranging from 10^{-2} to 10^{-5}, computing critical points and MS segmentation on the decompressed data. As shown in Figure 12, ZFP generally preserves MS segmentations better than SZ3; however, even at the tightest tested error bound, more than 20,000 false segmentation labels remain. The linked spatial view explicitly reveals where distorted boundaries occur, which can affect the structural integrity of the AR features. Together, these views demonstrate that tightening error bounds alone is insufficient, thereby motivating a QoI-correction step.

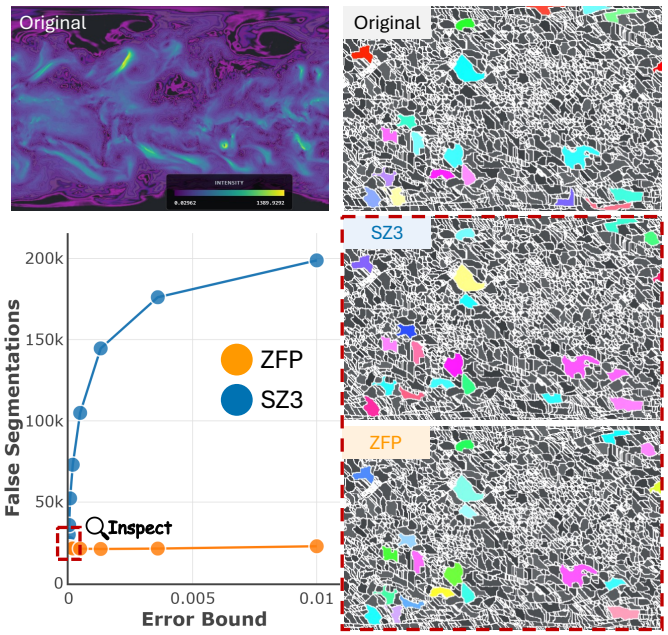


Figure 12: Impact of lossy compression on MS segmentations of the AR dataset. Left: visualization of the original dataset and the number of incorrect MS segmentation labels under varying error bounds for SZ3 and ZFP. Right: examples of distorted boundaries in the top 20 descending segmentations from decompressed SZ3 and ZFP data at a relative error bound of 10^{-5}.

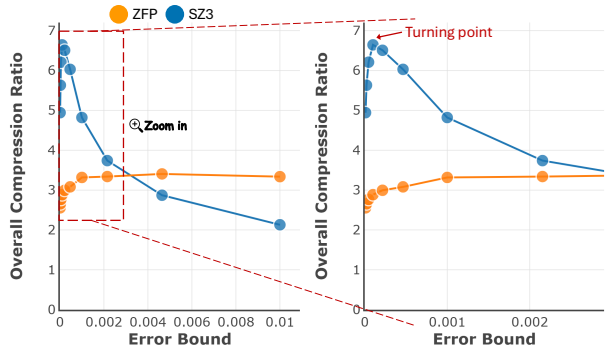


Figure 13: Overall compression ratio vs. error bound when applying MSz to preserve Morse-Smale segmentations of the AR dataset using SZ3 and ZFP compressors.

Evaluation Phase. Once the initial diagnosis reveals that standard error-bounded compression alone is insufficient, FZ-VIS guides the

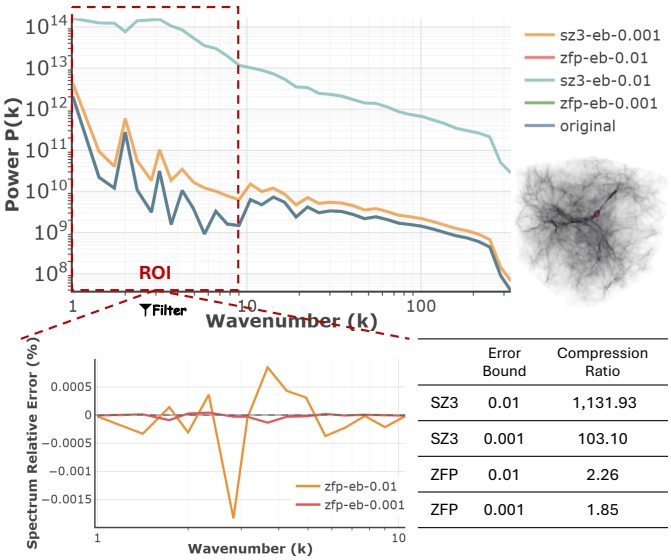
user to the MSz correction module. The user evaluates the corrected candidates using the *Overall Compression Ratio (OCR)* plot, where OCR is defined as the ratio of the original data size to the combined size of the compressed data and MSz edits [25]. This view exposes the trade-off between exact MS-segmentation recovery and compression efficiency. Figure 13 shows that SZ3 consistently attains a higher OCR than ZFP for error-bound values below 0.002, with a favorable operating point near a relative error bound of 10^{-4}.

6.4 Spectral Feature Preservation

While pointwise error control is important, the frequency-domain representation of data is also crucial for many scientific applications, as it provides complementary scientific insights. This case study shows how FZ-VIS extends a conventional spatial-error workflow with spectral evidence for compressor selection and refinement. We use the dark matter field from the NYX dataset to demonstrate this spectral workflow.

Requirements. In cosmology applications, many analysis tasks rely on frequency-domain representations to capture key features. For example, the power spectrum of cosmology data is the key to analyzing matter and energy distribution across spatial scales. Therefore, scientists need to ensure that compressed data satisfies requirements in both the spatial and frequency domains.

Design Phase. A scientist evaluating compressors solely through spatial error bounds may not know whether a configuration that satisfies pointwise thresholds also preserves the underlying data power spectrum. FZ-VIS bridges this gap by integrating dedicated spectral views into the workflow. When comparing SZ3 and ZFP at relative error bounds of 1% and 0.1%, the user can simultaneously evaluate the power spectrum curves and their associated spectral relative errors. As shown in Figure 14, SZ3 introduces substantial spectral distortion at both tested error bounds, whereas ZFP better preserves the trend of the original power spectrum and keeps the spectral relative error below 0.0020% within the selected frequency ROI. However, this improved spectral fidelity comes at the cost of compression performance: at a 1% relative error bound, ZFP achieves a compression ratio of only 2.26, compared to SZ3's significantly higher ratio of 103.10.



	Error Bound	Compression Ratio
SZ3	0.01	1,131.93
SZ3	0.001	103.10
ZFP	0.01	2.26
ZFP	0.001	1.85

Figure 14: Top: power spectra of different compressor instances on the NYX dataset. Bottom: spectral relative error of ZFP in the selected frequency range, together with the corresponding compression ratios.

Evaluation Phase. Recognizing that direct compressor selection forces a difficult trade-off between spectral fidelity and data reduction, the user next turns to refinement rather than abandoning the high-ratio SZ3 candidate. FZ-VIS supports this step by integrating FFCz [39] and visualizing the trade-off between compression ratio and spectral error

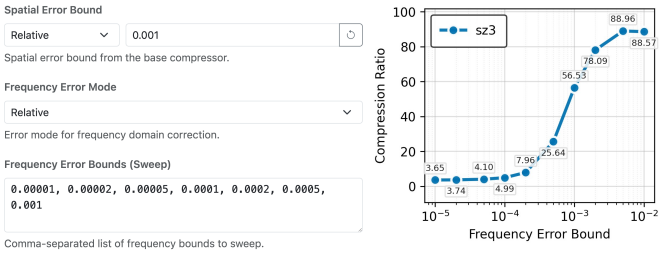


Figure 15: Left: the FFCz correction module allows users to sweep the frequency error bound. Right: compression ratio against frequency error bound for SZ3 with a spatial error bound of 0.1% on the NYX dataset.

bound under a fixed spatial tolerance. Figure 15 lets the user sweep the frequency-domain constraint and evaluate how aggressively it can be tightened before the compression ratio drops unacceptably. When the user sets the frequency error bound to 0.1% with a spatial error bound of 0.1%, the resulting compression ratio is still around 56, far higher than directly choosing ZFP.

7 Limitations and Discussion

FZ-VIS is designed to make QoI-aware lossy compression exploration more accessible. However, it does not eliminate all practical challenges in compressor design and evaluation.

First, its current composition model primarily supports existing compressors and predefined pipelines exposed through LibPressio. Although users can compare module options and parameter variants, they cannot yet freely construct a new lossy compressor by arbitrarily composing modules. Enabling this capability would require careful consideration of module compatibility and execution constraints.

Second, the current QoI support in FZ-VIS focuses on specific features, such as critical points, Morse-Smale segmentation, and power spectrum, whereas many applications require additional task-specific quantities and derived measures.

Finally, the current web-based interactive framework may not scale well to very large datasets because data movement, rendering, and repeated analysis can degrade responsiveness during exploration.

8 Conclusion

In this paper, we presented FZ-VIS, a human-in-the-loop visual analytics framework for QoI-aware lossy compression. FZ-VIS unifies compressor configuration, batch evaluation, and QoI analysis within a single workflow, enabling users to explore compressor variants, examine trade-offs between compression performance and data fidelity, and evaluate feature preservation for downstream analysis. By integrating an intuitive web-based interface, configuration generation via a force-directed graph, linked visual and quantitative analysis views, and QoI-oriented modules for topological and spectral analysis, FZ-VIS supports novice users, compressor developers, and domain scientists within a unified environment.

More broadly, FZ-VIS connects the compressor design space to the scientific evaluation space, allowing users to reason jointly about configuration choices, conventional fidelity metrics, and application-specific QoIs during the decision-making process. Our case studies show that FZ-VIS helps users navigate complex design spaces and make informed compression decisions across diverse scientific tasks.

We view FZ-VIS as a foundation for more flexible and scalable QoI-aware compression analysis, and as a step toward making compression design more accessible, interpretable, and scientifically grounded across diverse applications. Looking forward, possible directions include supporting more flexible compressor composition, broadening QoI coverage to additional scientific tasks, and improving scalability for larger datasets and more demanding interactive analysis tasks.

Supplemental Materials

All supplemental materials are available on OSF at `https://osf.io/7cspq`, released under a CC BY 4.0 license. They include (1) a video

demonstration of the proposed system, (2) a formative user study for requirements analysis, and (3) an extended version of this paper that includes the appendix. The source code for the implementation is also available at the same link under the MIT license.

Acknowledgment

The material was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357, and supported by the National Science Foundation under grants OAC-2311875, OAC-2311878, OAC-2514036, OAC-2514034, OAC-2609480, OAC-2513768, OAC-2313122, OAC-2313123, OAC-2313124, and OAC-2628473.

References

[1] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., Clifton Park, NY, USA, 2015. 2

[2] A. H. Baker, D. M. Hammerling, S. A. Mickelson, H. Xu, M. B. Stolpe, P. Naveau et al. Evaluating lossy data compression on climate simulation data within a large ensemble. *Geoscientific Model Development*, 9(12):4381–4403, Dec. 2016. doi: 10.5194/gmd-9-4381-2016 2

[3] A. H. Baker, A. Pinard, and D. M. Hammerling. On a Structural Similarity Index Approach for Floating-Point Data. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6261–6274, 2024. doi: 10.1109/TVCG.2023.3332843 7

[4] S. Bruckner and T. Möller. Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1468–1476, 2010. doi: 10.1109/TVCG.2010.190 3

[5] F. Cappello, R. Underwood, Y. Alexeev, A. Baker, E. Bozdağ, M. Burtscher et al. What to Support When You're Compressing: The State of Practice Gaps and Opportunities for Scientific Data Compression. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1966–1979. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3712285.3759856 7

[6] H. Chen, S. Zhang, W. Chen, H. Mei, J. Zhang, A. Mercer et al. Uncertainty-Aware Multidimensional Ensemble Data Visualization and Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1072–1086, 2015. doi: 10.1109/TVCG.2015.2410278 3

[7] L. P. Deutsch. GZIP file format specification version 4.3. Request for Comments RFC 1952, Internet Engineering Task Force, May 1996. doi: 10.17487/RFC1952 2

[8] S. Di and F. Cappello. Fast Error-Bounded Lossy HPC Data Compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 730–739, May 2016. doi: 10.1109/IPDPS.2016.11 2

[9] S. Di and F. Cappello. Optimization of Error-Bounded Lossy Compression for Hard-to-Compress HPC Data. *IEEE Transactions on Parallel and Distributed Systems*, 29(1):129–143, Jan. 2018. doi: 10.1109/TPDS.2017.2749300 2

[10] S. Di, J. Liu, K. Zhao, X. Liang, R. Underwood, Z. Zhang et al. A Survey on Error-Bounded Lossy Compression for Scientific Datasets, Apr. 2025. doi: 10.48550/arXiv.2404.02840 2

[11] D. E3SM Project. Energy exascale earth system model v2.1.0. [Computer Software] `https://doi.org/10.11578/E3SM/dc.20230110.5`, Jan. 2023. 1

[12] A. Fallin, N. Gorski, T. Agarwal, B. Wang, G. Gopalakrishnan, and M. Burtscher. Fast Topology-Aware Lossy Data Compression with Full Preservation of Critical Points and Local Order. *IEEE Transactions on Big Data*, pp. 1–14, 2026. doi: 10.1109/TBDATA.2026.3705355 2

[13] Q. Gong, J. Chen, B. Whitney, X. Liang, V. Reshniak, T. Banerjee et al. MGARD: A multigrid framework for high-performance, error-controlled data compression and refactoring. *SoftwareX*, 24:101590, 2023. doi: 10.1016/j.softx.2023.101590 2

[14] N. Gorski, X. Liang, H. Guo, and B. Wang. TFZ: Topology-Preserving Compression of 2D Symmetric and Asymmetric Second-Order Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):527–537, 2026. doi: 10.1109/TVCG.2025.3634844 3

[15] N. Gorski, X. Liang, H. Guo, L. Yan, and B. Wang. A General Framework for Augmenting Lossy Compressors With Topological Guarantees. *IEEE Transactions on Visualization and Computer Graphics*, 31(6):3693–3705, 2025. doi: 10.1109/TVCG.2025.3567054 2

[16] P. Grosset, C. M. Biwer, J. Pulido, A. T. Mohan, A. Biswas, J. Patchett et al. Foresight: Analysis That Matters for Data Reduction. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, Nov. 2020. doi: 10.1109/SC41405.2020.00087 2, 3

[17] W. He, J. Wang, H. Guo, K.-C. Wang, H.-W. Shen, M. Raj et al. InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):23–33, 2020. doi: 10.1109/TVCG.2019.2934312 3

[18] J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55 2

[19] J. W. Hurrell, M. M. Holland, P. R. Gent, S. Ghan, J. E. Kay, P. J. Kushner et al. The Community Earth System Model: A Framework for Collaborative Research. *Bulletin of the American Meteorological Society*, 94(9):1339 – 1360, 2013. doi: 10.1175/BAMS-D-12-00121.1 7

[20] S. Jin, J. Pulido, P. Grosset, J. Tian, D. Tao, and J. Ahrens. Adaptive Configuration of In Situ Lossy Compression for Cosmology Simulations via Fine-Grained Rate-Quality Modeling. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '21, pp. 45–56. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3431379.3460653 3

[21] W. Köpp and T. Weinkauf. Temporal Merge Tree Maps: A Topology-Based Static Visualization for Temporal Scalar Data. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1157–1167, 2023. doi: 10.1109/TVCG.2022.3209387 2

[22] F. Lan, B. Gamelin, L. Yan, J. Wang, B. Wang, and H. Guo. Topological Characterization and Uncertainty Visualization of Atmospheric Rivers. *Computer Graphics Forum*, 43(3):e15084, 2024. doi: 10.1111/cgf.15084 8

[23] S. Li, S. Di, K. Zhao, X. Liang, Z. Chen, and F. Cappello. Resilient Error-Bounded Lossy Compressor for Data Transfer. In *SC21: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14, Nov. 2021. doi: 10.1145/3458817.3476195 2

[24] S. Li, P. Lindstrom, and J. Clyne. Lossy Scientific Data Compression With SPERR. In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1007–1017, 2023. doi: 10.1109/IPDPS54959.2023.00104 2

[25] Y. Li, X. Liang, B. Wang, Y. Qiu, L. Yan, and H. Guo. MSz: An Efficient Parallel Algorithm for Correcting Morse-Smale Segmentations in Error-Bounded Lossy Compressors. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):130–140, Jan. 2025. doi: 10.1109/TVCG.2024.3456337 2, 3, 6, 9

[26] Y. Li, M. Xia, X. Liang, B. Wang, R. Underwood, S. Di et al. pMSz: A Distributed Parallel Algorithm for Correcting Morse-Smale Segmentations for Lossy Compression. In *Proceedings of the 40th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2026. Accepted. doi: 10.48550/arXiv.2601.01787 2

[27] X. Liang, S. Di, F. Cappello, M. Raj, C. Liu, K. Ono et al. Toward Feature-Preserving Vector Field Compression. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5434–5450, 2023. doi: 10.1109/TVCG.2022.3214821 2

[28] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo et al. Error-Controlled Lossy Compression Optimized for High Compression Ratios of Scientific Datasets. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 438–447, Dec. 2018. doi: 10.1109/BigData.2018.8622520 2

[29] X. Liang, H. Guo, S. Di, F. Cappello, M. Raj, C. Liu et al. Toward Feature-Preserving 2D and 3D Vector Field Compression. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 81–90, 2020. doi: 10.1109/PacificVis48177.2020.6431 2, 8

[30] X. Liang, K. Zhao, S. Di, S. Li, R. Underwood, A. M. Gok et al. SZ3: A Modular Framework for Composing Prediction-Based Error-Bounded Lossy Compressors. *IEEE Transactions on Big Data*, 9(2):485–498, Apr. 2023. doi: 10.1109/TBDATA.2022.3201176 2

[31] P. Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, Dec. 2014. doi: 10.1109/TVCG.2014.2346458 2

[32] P. Lindstrom and M. Isenburg. Fast and Efficient Compression of Floating-Point Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1245–1250, Sept. 2006. doi: 10.1109/TVCG.2006.143 2

[33] J. Liu, S. Di, S. Jin, K. Zhao, X. Liang, Z. Chen et al. Scientific Error-bounded Lossy Compression with Super-resolution Neural Networks. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 229–236,

Dec. 2023. doi: 10.1109/BigData59044.2023.10386682 2

[34] J. Liu, S. Di, K. Zhao, S. Jin, D. Tao, X. Liang et al. Exploring Autoencoder-based Error-bounded Compression for Scientific Data. In *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 294–306, Sept. 2021. ISSN: 2168-9253. doi: 10.1109/Cluster48925.2021.00034 2

[35] R. G. C. Maack, J. Lukasczyk, J. Tierny, H. Hagen, R. Maciejewski, and C. Garth. Parallel Computation of Piecewise Linear Morse-Smale Segmentations. *IEEE Transactions on Visualization and Computer Graphics*, 30(4):1942–1955, 2024. doi: 10.1109/TVCG.2023.3261981 3

[36] NumPy Developers. numpy.fft.fftshift. `https://numpy.org/doc/stable/reference/generated/numpy.fft.fftshift.html`, 2026. 4

[37] N. Piccolotto, M. Bögl, and S. Miksch. Visual Parameter Space Exploration in Time and Space. *Computer Graphics Forum*, 42(6):e14785, 2023. doi: 10.1111/cgf.14785 3

[38] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci et al. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 233–240, 2009. doi: 10.1109/ICDMW.2009.55 3

[39] C. Ren, R. Underwood, S. Di, E. Kutay, Z. Lukic, A. Yener et al. FFCz: Fast Fourier Correction for Spectrum-Preserving Lossy Compression of Scientific Data. In *Proceedings of the 40th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2026. Accepted. doi: 10.48550/arXiv.2601.01596 3, 4, 6, 9

[40] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka. Exploration of Lossy Compression for Application-Level Checkpoint/Restart. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pp. 914–922, May 2015. doi: 10.1109/IPDPS.2015.67 2

[41] B. Schneider, D. Jäckle, F. Stoffel, A. Diehl, J. Fuchs, and D. Keim. Integrating Data and Model Space in Ensemble Learning by Visual Analytics. *IEEE Transactions on Big Data*, 7(3):483–496, 2021. doi: 10.1109/TBDATA.2018.2877350 3

[42] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014. doi: 10.1109/TVCG.2014.2346321 3

[43] M. Soler, M. Plainchault, B. Conche, and J. Tierny. Topologically Controlled Lossy Compression. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 46–55, 2018. doi: 10.1109/PacificVis.2018.00015 2

[44] D. Sundararajan. *The Discrete Fourier Transform: Theory, Algorithms and Applications*. World Scientific, 2001. 3

[45] D. Tao, S. Di, Z. Chen, and F. Cappello. Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1129–1139, May 2017. ISSN: 1530-2075. doi: 10.1109/IPDPS.2017.115 2

[46] D. Tao, S. Di, H. Guo, Z. Chen, and F. Cappello. Z-checker: A framework for assessing lossy compression of scientific data. *The International Journal of High Performance Computing Applications*, 33(2):285–303, Mar. 2019. doi: 10.1177/1094342017737147 2, 3

[47] R. Underwood, V. Malvoso, J. C. Calhoun, S. Di, and F. Cappello. Productive and Performant Generic Lossy Data Compression with LibPressio. In *2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7)*, pp. 1–10, Nov. 2021. doi: 10.1109/DRBSD754563.2021.00005 2, 3, 5

[48] M. P. van Daalen, J. Schaye, C. M. Booth, and C. Dalla Vecchia. The effects of galaxy formation on the matter power spectrum: a challenge for precision cosmology. *Monthly Notices of the Royal Astronomical Society*, 415(4):3649–3665, 2011. doi: 10.1111/j.1365-2966.2011.18981.x 3

[49] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. Visualization and Visual Analysis of Ensemble Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2853–2872, 2019. doi: 10.1109/TVCG.2018.2853721 3

[50] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. 2

[51] M. Xia, S. Di, F. Cappello, P. Jiao, K. Zhao, J. Liu et al. Preserving Topological Feature with Sign-of-Determinant Predicates in Lossy Compression: A Case Study of Vector Field Critical Points. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 4979–4992, 2024. doi: 10.1109/ICDE60146.2024.00378 2

[52] M. Xia, B. Wang, Y. Li, P. Jiao, X. Liang, and H. Guo. TspSZ: An Efficient Parallel Error-Bounded Lossy Compressor for Topological Skeleton Preservation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 3682–3695, 2025. doi: 10.1109/ICDE65448.2025.00275 2

[53] L. Yan, X. Liang, H. Guo, and B. Wang. TopoSZ: Preserving Topology in Error-Bounded Lossy Compression. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1302–1312, 2024. doi: 10.1109/TVCG.2023.3326920 2, 8

[54] X. Yu, S. Di, K. Zhao, J. Tian, D. Tao, X. Liang et al. Ultrafast Error-bounded Lossy Compression for Scientific Datasets. In *Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '22, pp. 159–171. Association for Computing Machinery, June 2022. doi: 10.1145/3502181.3531473 2

[55] K. Zhao, S. Di, M. Dmitriev, T.-L. D. Tonellot, Z. Chen, and F. Cappello. Optimizing Error-Bounded Lossy Compression for Scientific Data by Dynamic Spline Interpolation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1643–1654, Apr. 2021. doi: 10.1109/ICDE51399.2021.00145 2

[56] K. Zhao, S. Di, X. Liang, S. Li, D. Tao, Z. Chen et al. Significantly Improving Lossy Compression for HPC Datasets with Second-Order Prediction and Parameter Optimization. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '20, pp. 89–100. Association for Computing Machinery, June 2020. doi: 10.1145/3369583.3392688 2

[57] K. Zhao, S. Di, D. Perez, X. Liang, Z. Chen, and F. Cappello. MDZ: An Efficient Error-bounded Lossy Compressor for Molecular Dynamics. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 27–40, May 2022. doi: 10.1109/ICDE53745.2022.00007 2

[58] J. Zhou and M. Takatsuka. Automatic Transfer Function Generation Using Contour Tree Controlled Residue Flow Model and Color Harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1481–1488, 2009. doi: 10.1109/TVCG.2009.120 2

[59] Zstandard. `http://www.zstd.net`, 2024. 2