

Exploring Wavelet Transform Usages for Error-bounded Scientific Data Compression

Jiajun Huang,^{*} Jinyang Liu,^{*} Sheng Di,[†] Yujia Zhai,^{*} Zizhe Jian,^{*}
Shixun Wu,^{*} Kai Zhao,[§] Zizhong Chen,^{*} Yanfei Guo,[†] Franck Cappello^{†‡}

^{*}University of California, Riverside, CA, USA

[†]Argonne National Laboratory, Lemont, IL, USA

[‡] University of Illinois at Urbana-Champaign, Urbana, IL, USA

[§] Florida State University, Tallahassee, FL, USA

jhuan380@ucr.edu, jliu447@ucr.edu, sdi1@anl.gov, yzhai015@ucr.edu, zjian106@ucr.edu,
swu264@ucr.edu, kzhao@cs.fsu.edu, chen@cs.ucr.edu, yguo@anl.gov, cappello@mcs.anl.gov

Abstract—To address the challenges raised by the data management of exascale scientific data, error-bounded lossy compression has been proposed and well-researched as a prominent solution. Among the existing works, a recent trend leverages wavelet transforms in the error-bounded lossy compression task to effectively capture long-term data correlations within the inputs. Applying those transforms as data preprocessors and decorrelators, wavelet-based lossy compressors have achieved optimized compression rate-distortion on several datasets. However, certain significant limitations of wavelet-based compressors have also been observed: On one hand, attributed to the high computational cost of wavelet transforms, wavelet-based compressors suffer from relatively low computational efficiencies compared to other state-of-the-art compressors. On the other hand, one certain type of wavelet transform cannot perform well on all variations of scientific data. Consequently, to further fine-tune the wavelet-based scientific data lossy compression, more in-depth and systematic research and analysis needs to be conducted. In this paper, based on the FAZ auto-tuning-based modular compression framework, we have integrated a great number of wavelet transforms into the framework and evaluated them with various real-world scientific datasets and fields. From the analysis of those evaluations and the comparison to existing state-of-the-art wavelet-based and non-wavelet-based error-bounded lossy compressors, we conclude and present several essential takeaways for designing and optimizing the wavelet-based scientific error-bounded lossy compressor.

Index Terms—error-bounded lossy compression, wavelet transform, scientific datasets.

I. INTRODUCTION

In order to address the challenges of data storage, transmission, and analysis for modern huge-scale scientific simulation data, error-bounded lossy compression has been confirmed as the essential data reduction technique for scientific data management. Shortly speaking, error-bounded lossy compression can substantially reduce the input data size, meanwhile preserving the compression errors within pre-given constraints. Due to the significance of error-bounded lossy compression, multiple scientific error-bounded lossy compressors have been proposed by researchers and adopted by scientific communities. Among those existing compressors, various data processing techniques from diverse categories have been leveraged as the basis of data compression, including but not limited

to linear regression [1], interpolation [2], [3], [4], discrete orthogonal transform [5], singular value decomposition [6], wavelet transforms [7], [8], and so on.

In past research works, wavelet transforms (particularly multi-dimensional discrete wavelet transforms) have been regarded as an exceptionally effective data decorrelation technique for optimizing the compression ratio of error-bounded lossy compression [7], [8]. Separating high-frequency and low-frequency data features, discrete wavelet transforms have the outstanding ability to capture long-distance data correlations, therefore the transformed coefficients may have much more compressibility than the original data. Benefiting from that, existing wavelet-based error-bounded compressors have achieved state-of-the-art compression ratios and distortions under a large variety of compression tasks. SPERR [7] can have doubled or tripled compression ratios of state-of-the-art prediction-based compressor QoZ [4] under the same PSNR, and FAZ [8] even further optimized the wavelet-based compression framework, having up to 100%+ same-PSNR compression ratio improvements over SPERR in certain cases.

Nevertheless, wavelet-based error-bounded lossy compressors are still far from being a universal solution for scientific data compression tasks, with several key limitations remaining. First, the computational costs of multi-dimensional discrete wavelet transforms are relatively high, which results in the speeds of wavelet-based compressors being noticeably lower than quite a few existing efficient compressors. Secondly, according to experimental analyses in existing works, wavelet transforms can not optimize the compression ratio for all datasets, under certain cases they may cause the compression ratio to decrease. Last, even when wavelet transforms can help the compression, as there are a great number of wavelet kernels in various types, having different adaptiveness to the input data, it would be hard to identify the compressibility of the transformed coefficients with different wavelet transforms and select out the best-fit one for each input data. For example, SPERR [7] proposes a fixed compression pipeline pre-processing each input data with the CDF9/7 [9] wavelet transform, which presents sub-optimal or degraded compression ratios on quite a few data inputs.

To address those challenges of effectively leveraging wavelet transforms in scientific error-bounded lossy compression, in this paper, with the FAZ auto-tuning-based modular compression framework, we evaluated compression pipelines with diverse wavelet transforms as preprocessors on a great variety of scientific datasets, to examine, identify and analyze how each scientific dataset is adapted to wavelet transforms for error-bounded lossy compression, and which wavelet transform works best for each dataset. Our contributions include:

- We evaluated the error-bounded lossy compression of various scientific datasets with most of the existing wavelet transforms, then determined the best-fit wavelet transform for each dataset.
- We designed a new error-bounded lossy compression pipeline leveraging the best-fit wavelet transform for different input data, then tested it and compared its performance with existing wavelet-based and non-wavelet-based compressors.
- Based on the experiments and evaluations, we concluded and proposed several important takeaways for integrating and optimizing the usage of wavelet transforms in scientific error-bounded lossy compressors.

The rest of this paper is organized as: In Section II, we discuss related works. In Section III, we detail the research problem formulation. We evaluate and compare different variants of wavelet transforms in Section IV. A compression framework leveraging different wavelets is proposed and evaluated in Section V. In Section VI, we conclude our work and discuss future work.

II. RELATED WORK

Recent efforts in the domain of scientific data compression have yielded a variety of error-bounded lossy compressors to accommodate diverse scenarios [10]. These compressors are generally grouped into four main types: prediction-based, transform-based, dimension-reduction-based, and neural-network-based.

Prediction-based compressors employ algorithms such as linear regression [1] and spline interpolations [2] to estimate the values in data. They control the prediction distortion through techniques like quantization, ensuring it stays within set error thresholds. Examples of this type include FPZIP [11], SZ2 [1], and SZ3 [2], [3]. Transform-based compressors, like ZFP [5], use mathematical transforms to lessen data correlation, facilitating easier compression of the resulting coefficients through methods like exponent alignment and orthogonal transforms. Dimension-reduction-based compressors, for instance, TTHRESH [6], reduce the data volume by employing methods such as singular vector decomposition (SVD). Lastly, neural-network-based compressors [12], [13], [14], [15] leverage neural network models, including various forms of autoencoders [16], [17], [18], to encode and decode data efficiently within the required error boundaries.

Like wavelet-based image compressors such as JPEG-2000 [19], Wavelet transforms have also been applied to the scientific error-bounded lossy compression task. Besides the

early works including SSEM [20] and VAPOR [21], recently several wavelet-based compressors have been proposed which exhibit excellent compression ratio and distortion. For example, SPERR [7] adopts multi-dimensional discrete CDF9/7 wavelet [9] for effective scientific data compression, and FAZ [8] is a hybrid compression framework that combines diverse compression techniques, among which wavelet transform is a key component. There is a large variety of discrete wavelet transforms, which are potentially available for the error-bounded lossy compressor, including but not limited to Symlets [22], Coiflets [23], and Biorthogonal wavelets [24]. For the implementation of those wavelets, PyWavelets [25] is a Python-based library that contains diverse wavelet transforms including all the aforementioned ones.

III. PROBLEM FORMULATION

In this subsection, we propose several key mathematical definitions and the mathematical formulation of our research target for this paper.

1) *Compression ratio and bit rate*: Compression ratio is defined by the input data size divided by the compressed data size. Specifically, for input data X and compressed data Z , compression ratio ρ is:

$$\rho = \frac{|X|}{|Z|} \quad (1)$$

According to Eq. 1, a higher compression ratio means better (smaller) compressed size, and vice versa. In the visualization of experimental results, researchers often plot curves with another metric closely related to the compression ratio, namely the bit rate. Bit rate is defined by the average number of bytes used in the compressed data to store each data element for the input data, which can be expressed as (denote bit rate by b):

$$b = \frac{\text{sizeof}(x)}{|Z|} \quad (2)$$

in which x is an element of the input X , and $\text{sizeof}()$ returns the byte size. Since the bit rate is reciprocal to the compression ratio, a lower bit rate is better.

2) *PSNR*: PSNR (Peak Signal-to-Noise Ratio) is one of the most important data distortion metrics for evaluating the quality of the decompressed data from the lossy compression. it is defined as follows:

$$PSNR = 20 \log_{10} \text{vrange}(X) - 10 \log_{10} \text{mse}(X, X'), \quad (3)$$

where X is the input data and X' is the decompressed data. $\text{vrange}()$ calculates the value range of one data array, and mse refers to the mean-squared error. Fixing the input data (and also the data range), a smaller mean-squared error will lead to higher PSNR, therefore higher PSNR means higher precision of the decompressed data.

3) *Research target*: The objective of our research is to find the best-fit wavelet preprocessor for each dataset to optimize the compression ratio under the same decompression data distortion in terms of PSNR. Mathematically speaking, given the input data X and a set of wavelet transforms $\mathcal{W} = \{W_1, W_2, \dots\}$, the compressed data Z and decompression output X' is generated from compressing/decompressing

the wavelet-transformed $W_i(X)$ with a compressor C and decompressor D . Under a certain error bound e , and the target decomposition PSNR P , we will optimize the selection of best-fit wavelet W via the following optimization problem:

$$\begin{aligned} W &= \arg \max_{\{W_i\} \subset W} \frac{|X|}{|Z|} \\ \text{s.t. } PSNR(X, X') &= P \\ Z &= C(W_i(X)), X' = D(Z) \\ |x_i - x'_i| &\leq e, \forall x_i \in X. \end{aligned} \quad (4)$$

IV. EVALUATION OF WAVELET TYPES

In this section, we feature our in-depth evaluations and analyses for determining the best-fit wavelet transforms for different scientific datasets. Shortly speaking, for each dataset, we perform compression tasks on it by preprocessing (transforming) it with multiple wavelet transform types and compressing the transformed coefficients. The compression rate-distortion with different wavelet transforms is compared and evaluated, and at last, the best-fit wavelet transform with the optimized rate-distortion is selected. In the rest of this section, we will demonstrate those evaluations in detail.

A. Wavelet transform evaluation framework

The evaluation framework for wavelet transforms is proposed in Figure 1. For each wavelet transform to be evaluated, we preprocess the input data with that wavelet. The transformed coefficients are further encoded with the SPECK encoding algorithm [26] and losslessly compressed with Zstd [27], which have been verified to be one of the most effective pipelines for wavelet-based compression [7], [8]. As SPECK is a lossy encoding method, after decoding the lossy-encoded coefficients and an inverse-transform process, we can acquire the lossy-decompressed data and compute the data distortion with metrics such as PSNR. Eventually, the compression rate-distortion for this compression task is collected. With this framework, for each input data, we collect the compression rate-distortion results by compressing it with each wavelet transform and then plot those results.

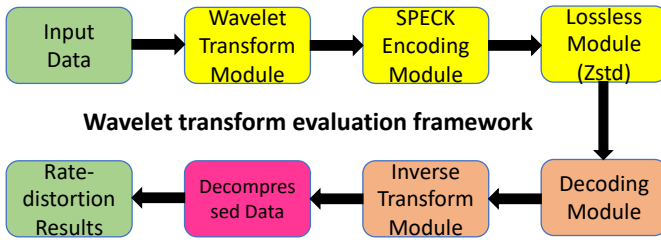


Fig. 1: The high-level pipeline of our wavelet transform evaluation framework.

B. Wavelet transform evaluation setup

In this part, we discuss the setup of the evaluation experiments for wavelet types.

1) *Evaluation Datasets*: The evaluations of wavelet types are based on the datasets listed in Table I. Those datasets are all real-world scientific datasets from diverse domains.

TABLE I: Information of the datasets in experiments

Dataset	Dimensions	Domain
RTM [28]	449×449×235	Seismic Wave
SEGSalt [29]	1008×1008×352	Geology
Miranda [30]	256×384×384	Turbulence
QMCPack [31]	288×115×69×69	Quantum
SCALE-LetKF [32]	98×1200×1200	Climate
NYX [33]	512×512×512	Cosmology
JHTDB [34]	512×512×512	Turbulence
Hurricane [35]	100×500×500	Climate

2) *Evaluation Wavelets*: The wavelets we evaluated are from 2 aspects: one is the multi-dimensional discrete CDF 9/7 wavelet transform [9] implemented by SPERR [7], and the other contains the multi-level and multi-dimensional wavelet transforms implemented in PyWavelets [25]. The latter ones are consisted of several wavelet families each having a number of variations. The information of the wavelet families is presented in Table II. The first column is the names and their abbreviations of the wavelet families, which are both to be used in the following evaluation results.

TABLE II: Wavelet families

Family (Abbr.)	Description	# of variations
bior (b)	Biorthogonal Wavelet	15
coif (c)	Coiiflets	17
db (d)	Daubechies Wavelet	38
rbio (r)	Reverse biorthogonal	15
sym (s)	Symlets	19
CDF 9/7 (C9/7)	CDF9/7 [9]	1

3) *Evaluation Configurations*: For a fair comparison, when encoding the wavelet coefficients with the lossy SPECK algorithm, for each wavelet transform we encode the corresponding coefficients under the same error threshold, which is 0.1% of the input data range. For the padding during the wavelet transform, we apply the "periodical" mode in PyWavelets.

4) *Evaluation Metric*: We evaluate the effectiveness of wavelet-based compression by calculating the bit rate of compressed data and the PSNR [36] between input and decompressed data. A lower bit rate or a higher PSNR is better.

C. Wavelet transform evaluation results and analysis

In this section, we present the compression results of different scientific datasets using various types of wavelets. Due to the page limitation, we only present the compression results of several datasets but we have done the same evaluation and analysis to all the mentioned datasets in Table I.

1) *Hurricane dataset*: In this paragraph, we analyze the compression result of the Hurricane dataset using two different fields. From Figure 2, we notice that CDF9/7 has the best compression quality(PSNR) as well as the compression ratio among all the evaluated wavelets for the u field. For the qvapor field, CDF9/7 also has the best overall compression result, however, it can not beat the bior3.3 in terms of the compression

ratio. Therefore, CDF9/7 serves as the most suitable wavelet for compressing both the qvapor and u fields.

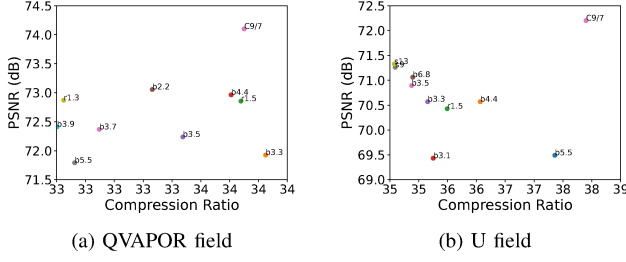


Fig. 2: Compression quality (PSNR) and compression ratio for various wavelets when compressing the Hurricane dataset.

2) *Miranda dataset*: In this paragraph, we compare the compression result of the Miranda dataset utilizing two different fields. From Figure 3, we can see that the wavelet-coif7 has the best PSNR and compression ratio compared with other wavelets in the pressure field. However, in the vx field of the Miranda dataset, coif5 has the best compression quality (PSNR) while bior6.8 has the best compression ratio. Thus, unlike the Hurricane dataset, we can find that different fields have their own suitable wavelets for the Miranda dataset.

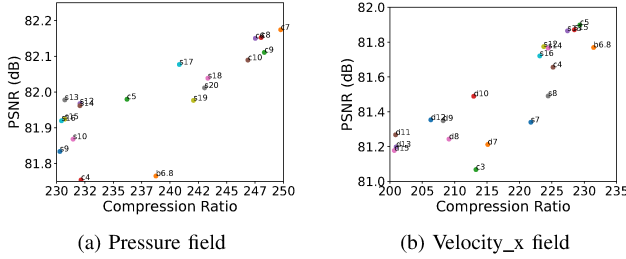


Fig. 3: Compression quality (PSNR) and compression ratio for various wavelets when compressing the Miranda dataset.

3) *RTM/SegSalt dataset*: For the RTM and SegSalt datasets, we illustrate the compression results of two different snapshots with different wavelets. In Figure 4, we notice that sym14 is the overall best pick in terms of both compression quality and compression ratio for the RTM #1800 snapshot. On the contrary, coif13 presents the best data accuracy and sym20 outperforms other wavelets regarding compression ratio for the SegSalt #2000 snapshot. This finding suggests that different wavelets should be used for different snapshots to obtain optimal compression results in the RTM/SegSalt datasets.

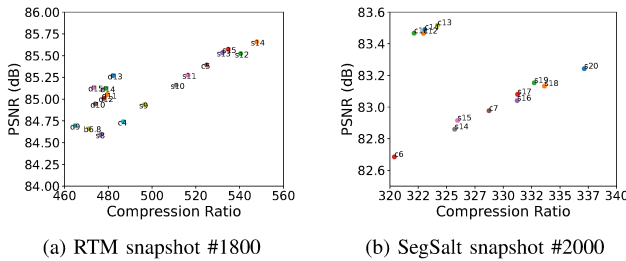


Fig. 4: Compression quality (PSNR) and compression ratio for various wavelets when compressing the RTM dataset.

4) *Nyx dataset*: We demonstrate the variant compression results of the Nyx dataset using two distinct fields. We can see that coif8, coif9, and coif10 have similar overall compression results and are generally outperforming other wavelets for the bdlog field. Specifically, coif8 has the best PSNR among these three wavelets, while coif10 has the finest compression ratio (20.69). Besides, coif9 is in the middle of the three wavelets with regard to both compression ratio and compression quality. As a result, we can choose any one from these three wavelets as the best-fit wavelet for the bdlog field. On the other hand, the bior6.8 wavelet has the top compression quality and the bior4.4 wavelet showcases the best compression ratio for the vx field. However, as the PSNR difference between these two wavelets is comparably small compared with the compression ratio difference, bior4.4 is the overall winner for the vx field.

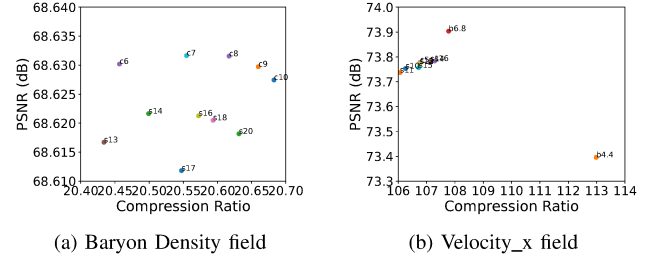


Fig. 5: Compression quality (PSNR) and compression ratio for various wavelets when compressing the Nyx dataset.

5) *Conclusion*: From the previous experimental analysis, we can conclude that different datasets or even distinct fields/snapshots within the same dataset require various wavelets to achieve the best compression quality and compression ratio. This motivates us to find and utilize the best-fit wavelet for a specific snapshot to obtain the finest compression results.

V. MULTI-WAVE DESIGN AND EVALUATION

In this section, we present the design and performance evaluation of our proposed wavelet-based error-bounded compressor—MultiWave.

A. High-level design of MultiWave compressor

Based on our previous analysis, we conclude that different wavelets should be utilized to compress various datasets, fields, and snapshots to optimize the compression results. Thus, we design and implement the MultiWave compressor that always utilizes the best-fit compressor to compress the input data. Figure 6 represents the compression pipeline of our MultiWave compressor. It first auto-tune the point-wise error threshold of the SPECK encoding based on the input data and then carries out the wavelet transform using the best-fit wavelet, followed by the SPECK Encoding Module. Because of the lossy nature of the SPECK encoding, an error-controlling module is required to ensure the error-bounded compressed results. After that, the lossless compressor—Zstd is utilized to compress the error-controlled data and we can obtain the final compressed data.

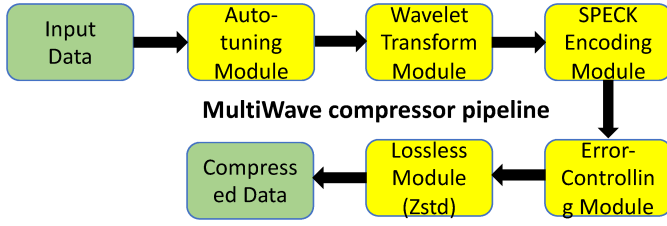


Fig. 6: The high-level pipeline of our MultiWave compressor.

B. Experimental setup

1) *Experimental environment*: We conducted all the evaluation experiments on the Purdue Anvil supercomputer. On the Anvil supercomputer, each computing node features two AMD EPYC 7763 CPUs with 64 cores having a 2.45GHz clock rate and 256 GB DDR4-3200 RAM.

2) *Experimental datasets and configurations*: The datasets used are already presented in Table I. For each dataset, the best-fit wavelet found in Section IV is utilized in MultiWave and will be detailed later in the experimental results.

3) *Comparison of lossy compressors in evaluation*: We involve 3 other state-of-the-art scientific error-bounded lossy compressors for the comparison with MultiWave. They are the representatives of different wavelet-usage strategies, including:

- **QoZ1.1** [4]: A non-wavelet-based but interpolation-based error-bounded lossy compressor;
- **SPERR** [7]: A wavelet-based lossy compression with fixed-design of compression pipeline, integrating CDF9/7 [9] wavelet and SPECK [26] encoding.
- **FAZ** [8]: A lossy compressor with auto-tuned compression pipeline. Depending on different inputs, different wavelet transforms (no transform, CDF9/7 or sym13) may be applied.

Through comparing with those state-of-the-art error-bounded lossy compressors, we can determine whether the flexibility of wavelet usage in MultiWave is the optimized wavelet usage strategy for scientific data compression.

C. Rate-distortion evaluation

In this subsection, we present the rate-distortion evaluations of the compressors. We have employed both PSNR and SSIM as distortion metrics to assess whether MultiWave can adaptively optimize different quality metrics throughout the compression process. Our baseline includes state-of-the-art error-bounded compressors such as FAZ, SPERR, and QoZ1.1.

1) *RTM/SegSalt datasets*: Figure 7 showcases the rate-distortion comparison of our MultiWave compressor with other baselines for the RTM/SegSalt datasets. For the RTM dataset, we could notice that MultiWave has a slightly better PSNR compared with the second-best compressor—FAZ when the bit rate is the same. This is because MultiWave is able to utilize the best-fit wavelet (sym14) while FAZ is limited to CDF9/7 and Sym13 wavelets. The same phenomenon can be witnessed in the SegSalt dataset, where MultiWave with sym20 wavelet has a notable improvement compared with the FAZ and both of the compressors are significantly better in terms of rate-distortion results compared with SPERR and QoZ1.1.

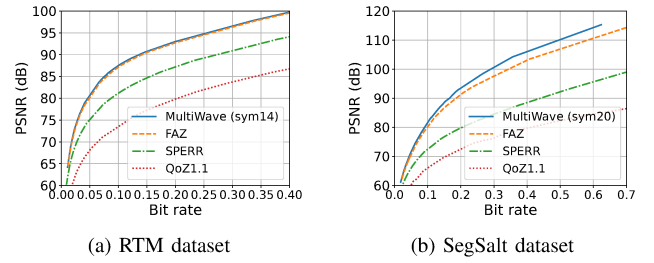


Fig. 7: The rate-distortion comparison between our MultiWave and multiple baseline compressors when compressing the RTM/SegSalt datasets.

2) *Hurricane/Scale datasets*: Figure 8 illustrates the rate-distortion analysis comparing our MultiWave compressor against other baseline compressors using the Hurricane/Scale datasets. From the results in the Hurricane dataset, we know that FAZ is showing the overall best rate-distortion result while QoZ1.1 can obtain a great PSNR when the bit rate is low. When the bit rate is increasing, the PSNR of MultiWave and SPERR is gradually exceeding QoZ1.1 and MultiWave is outperforming SPERR for all bit rates. However, because MultiWave always uses the wavelet-based compression, it can not reach the same PSNR as FAZ for some bit rates where the wavelet-based compression is not a good choice. Besides, as the FAZ is also capable of using the best-fit wavelet (CDF9/7) of the Hurricane dataset and it can adaptively choose from wavelet-based compression and FAZ-like compression pipeline, the FAZ can obtain a better PSNR than our MultiWave throughout all the bit rates. A similar trend can be observed in the Scale dataset, where FAZ is surpassing MultiWave for the same reason. However, we can see that the Scale dataset is more suitable to be compressed with wavelet-based method compared with the Hurricane dataset, because MultiWave can have a similar PSNR compared with FAZ when the bit rate is large.

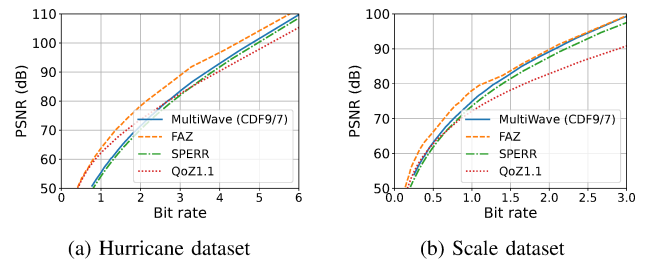


Fig. 8: The rate-distortion comparison between our MultiWave and multiple baseline compressors when compressing the Hurricane/Scale datasets.

3) *Miranda/Nyx datasets*: Figure 9 presents a comparative rate-distortion analysis of the MultiWave compressor and other baseline compressors using the Pressure field from the Miranda dataset and the Baryon Density field from the Nyx dataset. The reason why we only show the compression results of a single field for each dataset is that the best-fit wavelets vary for different fields in these two datasets. For the Pressure

field of the Miranda dataset, the most suitable wavelet is *coif7* and the MultiWave compressor shows the best compression result compared with all other baselines because it is able to compress using the bestfit wavelet while other compressors can not. Additionally, the non-wavelet-based compressor QoZ1.1 is falling behind all other counterparts because the Pressure field of Miranda data is more suitable for wavelet-based compression. For the Baryon Density field of the Nyx dataset, we can witness similar compression results as shown in the sub-figure of the Miranda dataset. The MultiWave with the *bior6.8* wavelet presents the best compression performance for the same reason.

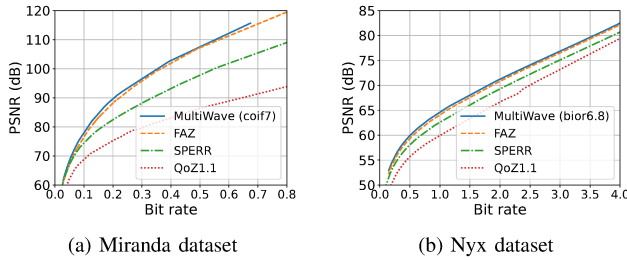


Fig. 9: The rate-distortion comparison between our Multiwave and multiple baseline compressors when compressing the Miranda/Nyx datasets.

VI. CONCLUSION AND FUTURE WORK

This paper delves into the utilization of wavelet transforms for compressing scientific datasets and introduces the MultiWave compressor. This approach consistently selects the optimal wavelet for each input, as determined by extensive testing and analysis. Our findings demonstrate that varying datasets and even different fields or snapshots within the same dataset necessitate specific wavelets to attain the highest quality and compression ratio. Besides, by comparing MultiWave with multiple state-of-the-art lossy compressors, we find that MultiWave is able to demonstrate the finest rate-distortion results for various datasets. However, not all the datasets are suitable for wavelet-based compression, and MultiWave with the bestfit wavelet can not always have the best compression result compared with non-wavelet-based compressor (QoZ1.1) or compressor with hybrid approaches (FAZ). In the future, we plan to extend our MutiWave compressor to enable a more flexible compression for different scientific datasets.

ACKNOWLEDGMENTS

This research was 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. 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 Grant OAC-2003709, OAC-2104023, OAC-2311875, OAC-2311877, and OAC-2153451. We acknowledge the computing resources provided on Bebop (operated by Laboratory Computing Resource Center at Argonne) and on Theta and JLSE (operated by Argonne Leadership Computing Facility).

REFERENCES

- [1] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello, “Error-controlled lossy compression optimized for high compression ratios of scientific datasets,” in *2018 IEEE International Conference on Big Data*. IEEE, 2018.
- [2] 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)*, 2021, pp. 1643–1654.
- [3] X. Liang, K. Zhao, S. Di, S. Li, R. Underwood, A. M. Gok, J. Tian, J. Deng, J. C. Calhoun, D. Tao *et al.*, “SZ3: A modular framework for composing prediction-based error-bounded lossy compressors,” *IEEE Transactions on Big Data*, 2022.
- [4] J. Liu, S. Di, K. Zhao, X. Liang, Z. Chen, and F. Cappello, “Dynamic quality metric oriented error bounded lossy compression for scientific datasets,” in *2022 SC22: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE Computer Society, 2022, pp. 892–906.
- [5] P. Lindstrom, “Fixed-rate compressed floating-point arrays,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.
- [6] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, “TTHRESH: Tensor compression for multidimensional visual data,” *IEEE transactions on visualization and computer graphics*, vol. 26, no. 9, pp. 2891–2903, 2019.
- [7] S. Li, P. Lindstrom, and J. Clyne, “Lossy scientific data compression with sperr,” in *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2023, pp. 1007–1017.
- [8] J. Liu, S. Di, K. Zhao, X. Liang, Z. Chen, and F. Cappello, “Faz: A flexible auto-tuned modular error-bounded compression framework for scientific data,” in *Proceedings of the 37th International Conference on Supercomputing*, 2023, pp. 1–13.
- [9] A. Cohen, I. Daubechies, and J.-C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Communications on pure and applied mathematics*, vol. 45, no. 5, pp. 485–560, 1992.
- [10] F. Cappello, S. Di, S. Li, X. Liang, G. M. Ali, D. Tao, C. Yoon Hong, X.-c. Wu, Y. Alexeev, and T. F. Chong, “Use cases of lossy compression for floating-point data in scientific datasets,” *International Journal of High Performance Computing Applications (IJHPCA)*, vol. 33, pp. 1201–1220, 2019.
- [11] P. G. Lindstrom *et al.*, “Fpzip,” Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2017.
- [12] J. Liu, S. Di, K. Zhao, S. Jin, D. Tao, X. Liang, Z. Chen, and F. Cappello, “Exploring autoencoder-based error-bounded compression for scientific data,” in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021, pp. 294–306.
- [13] A. Glaws, R. King, and M. Sprague, “Deep learning for in situ data compression of large turbulent flow simulations,” *Physical Review Fluids*, vol. 5, no. 11, p. 114602, 2020.
- [14] T. Liu, J. Wang, Q. Liu, S. Alibhai, T. Lu, and X. He, “High-ratio lossy compression: Exploring the autoencoder to compress scientific data,” *IEEE Transactions on Big Data*, 2021.
- [15] L. Hayne, J. Clyne, and S. Li, “Using neural networks for two dimensional scientific data compression,” in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 2956–2965.
- [16] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *arXiv preprint arXiv:2003.05991*, 2020.
- [17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [18] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, “Sliced Wasserstein auto-encoders,” in *International Conference on Learning Representations*, 2018.
- [19] D. S. Taubman and M. W. Marcellin, “Jpeg2000: Standard for interactive imaging,” *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, 2002.

- [20] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of lossy compression for application-level checkpoint/restart," in *IPDPS 2015*, 2015, pp. 914–922.
- [21] J. Clyne, P. Mininni, A. Norton, and M. Rast, "Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation," *New Journal of Physics*, vol. 9, no. 8, p. 301, 2007.
- [22] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.
- [23] —, *Ten lectures on wavelets*. SIAM, 1992.
- [24] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.
- [25] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O’Leary, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.
- [26] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE transactions on circuits and systems for video technology*, vol. 14, no. 11, pp. 1219–1235, 2004.
- [27] Y. Collet, "Zstandard – real-time data compression algorithm," <http://facebook.github.io/zstd/>, 2015.
- [28] S. Kayum *et al.*, "GeoDRIVE – a high performance computing flexible platform for seismic applications," *First Break*, vol. 38, no. 2, pp. 97–100, 2020.
- [29] "SEGSalt," https://wiki.seg.org/wiki/SEG/EAGE_Salt_and_Overthrust_Models.
- [30] Miranda application. [Online]. Available: <https://wci.llnl.gov/simulation/computer-codes/miranda>
- [31] J. Kim *et al.*, "QMCPACK: an open source ab initio quantum monte carlo package for the electronic structure of atoms, molecules and solids," *Journal of Physics: Condensed Matter*, vol. 30, no. 19, p. 195901, 2018.
- [32] "Scalable computing for advanced library and environment (scale) – letkf," <https://github.com/gyllien/scale-letkf>.
- [33] NYX simulation, <https://amrex-astro.github.io/Nyx>, 2019, online.
- [34] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink, "A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence," *Journal of Turbulence*, no. 9, p. N31, 2008.
- [35] Hurricane ISABEL simulation data, <http://vis.computer.org/vis2004contest/data.html>, 2004, online.
- [36] 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*, vol. 33, no. 2, pp. 285–303, 2019.