NeurLZ: An Online Neural Learning-based Method to Enhance Scientific Lossy Compression

Wenqi Jia
University of Texas at Arlington
Arlington, TX, USA
wenqi.jia@uta.edu

Zhewen Hu
Texas A&M University
College Station, TX, USA
zhewen@tamu.edu

Youyuan Liu
Temple University
Philadelphia, PA, USA
youyuan.liu@temple.edu

Boyuan Zhang
Indiana University
Bloomington, IN, USA
bozhan@iu.edu

Jinzhen Wang
UNC Charlotte
Charlotte, NC, USA
jwang96@charlotte.edu

Jinyang Liu
University of Houston
Houston, TX, USA
jliu217@central.uh.edu

Wei Niu
University of Georgia
Athens, GA, USA
wniu@uga.edu

Stavros Kalafatis
Texas A&M University
College Station, TX, USA
skalafatis-tamu@tamu.edu

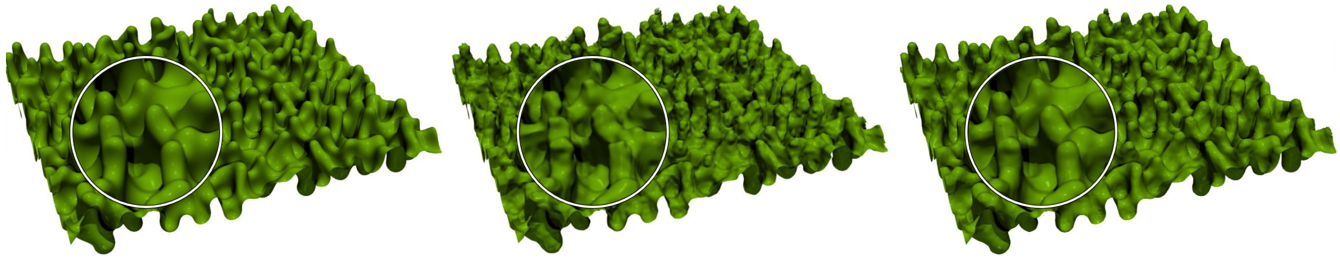
Junzhou Huang
University of Texas at Arlington
Arlington, TX, USA
junzhou.huang@uta.edu

Sian Jin
Temple University
Philadelphia, PA, USA
sian.jin@temple.edu

Daoce Wang*
Indiana University
Bloomington, IN, USA
daocwang@iu.edu

Jiannan Tian*
University of Kentucky
Lexington, KY, USA
jiannan.tian@uky.edu

Miao Yin*†
University of Texas at Arlington
Arlington, TX, USA
miao.yin@uta.edu



(a) Original Values (288MB)

(b) SZ3 (0.07MB, PSNR = 29.2)

(c) NeurLZ (0.07MB, PSNR = 39.1)

Figure 1: Renderings of an isosurface from Miranda's Density field: (a) original values, (b) SZ3, and (c) NeurLZ. The isosurface rendered by NeurLZ exhibits noticeable visual improvements compared to that produced by SZ3.

Abstract

Large-scale scientific simulations generate massive datasets, posing challenges for storage and I/O. Traditional lossy compression struggles to advance more in balancing compression ratio, data quality, and adaptability to diverse scientific data features. While deep learning-based solutions have been explored, their common practice of relying on large models and offline training limits adaptability to dynamic data characteristics and computational efficiency. To address

*Co-advising.

†Corresponding author. Email: miao.yin@uta.edu.

ICS '25, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1537-2/25/06
https://doi.org/10.1145/3721145.3725763

these challenges, we propose NeurLZ, a neural method designed to enhance lossy compression by integrating online learning, cross-field learning, and robust error regulation. Key innovations of NeurLZ include: (1) *compression-time online neural learning* with lightweight skipping DNN models, adapting to residual errors without costly offline pertaining, (2) *the error-mitigating capability*, recovering fine details from compression errors overlooked by conventional compressors, (3) 1× *and* 2× *error-regulation modes*, ensuring strict adherence to 1× user-input error bounds strictly or relaxed 2× bounds for better overall quality, and (4) *cross-field learning* leveraging inter-field correlations in scientific data to improve conventional methods. Comprehensive evaluations on representative HPC datasets, e.g., Nyx, Miranda, Hurricane, against state-of-the-art compressors show NeurLZ's effectiveness. During the first five learning epochs, NeurLZ achieves an 89% bit rate reduction, with further optimization yielding up to around 94% reduction at equivalent distortion, significantly outperforming existing methods, demonstrating NeurLZ's superior performance in enhancing scientific lossy compression as a scalable and efficient solution.

CCS Concepts

• Information systems → Data compression; **• Computing methodologies → Artificial intelligence**.

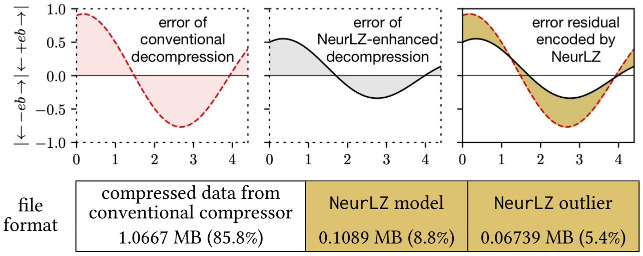
Keywords

Lossy Compression, Scientific Data, Online Learning, Cross-Field Learning, Neural Learning

ACM Reference Format:
Wenqi Jia, Zhewen Hu, Youyuan Liu, Boyuan Zhang, Jinzhen Wang, Jinyang Liu, Wei Niu, Stavros Kalafatis, Junzhou Huang, Sian Jin, Daoce Wang, Jiannan Tian, and Miao Yin. 2025. NeurLZ: An Online Neural Learning-based Method to Enhance Scientific Lossy Compression. In *2025 International Conference on Supercomputing (ICS '25), June 08–11, 2025, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3721145.3725763

1 Introduction

The exponential growth in computational power has enabled complex scientific simulations, producing massive datasets across disciplines. For instance, the CESM climate simulation [1] generates terabytes of daily data volumes for post-processing [7], highlighting the scale of data generated in modern scientific research. For warm data used in tasks like visualization and analysis [11], reducing storage overhead is crucial to minimize transmission delays and improve accessibility. Further, for cold data stored long-term, minimizing storage overhead becomes a significant concern due to the substantial expenses involved, where real-time compression is unnecessary. Services like Amazon Glacier Deep Archive [3] and Microsoft Azure Archive Storage [56], which use cost-effective storage mediums like magnetic tape [78], highlight the need for efficient compression solutions to balance accessibility, cost, and data fidelity [8, 22, 55, 60].



file format	compressed data from conventional compressor 1.0667 MB (85.8%)	NeurLZ model 0.1089 MB (8.8%)	NeurLZ outlier 0.06739 MB (5.4%)

Figure 2: (Top) compression error anatomy. (Bottom) the file format of NeurLZ compression archive.

Initially, researchers utilized and developed lossless compression algorithms [14, 16, 20, 42, 84] to mitigate data storage and transmission challenges. However, these techniques typically achieve only modest compression ratios at 1 to 3× when applied to scientific data [82]. Lossy compression algorithms [17, 37–39, 41, 46–48, 70, 71, 76, 77, 81, 83] offer higher compression ratios (e.g., 3.3× to 436× for SZ [17]) while preserving critical information within user-defined error bounds. Many lossy compressors utilize predictive techniques, such as curve-fitting [17] or spline interpolation [34], to exploit data correlations and smoothness, thereby reducing entropy and improving compression efficiency. However, spiky or irregular data, common in scientific simulations, disrupt these correlations, leading to higher prediction errors. These errors decrease the compression efficiency and increase storage requirements, motivating us to design more robust data compressors to handle data irregularities and keep improving compression efficiency.

Deep neural networks (DNNs), known as powerful predictor [27], have achieved remarkable success in tasks like image classification, object detection, and super-resolution [5, 33, 49, 62, 73, 74], and researchers have explored their potential to enhance compression quality [30, 36, 44, 45, 50]. However, core challenges exist to hinder DNN application in scientific computing at ease. 1 Static model training: Models are trained offline on fixed datasets, which poses significant limitations in the context of scientific data compression. Scientific data often originates from diverse domains across varying spatial and temporal scales [82], leading to substantial distribution shifts between training datasets and real-world scenarios. Consequently, models pre-trained on static datasets struggle to generalize on previously unseen data, resulting in poor predictive performance when deployed [57].

② Attempted mitigation of data shift: To minimize the impact of data shift, researchers pre-trained large models on diverse datasets, aiming to bridge the distribution gap. However, this approach incurs high pre-training costs and increases user-side download overhead due to the sheer size of these models (e.g., HAT with 9 million parameters [44] and Auto-Encoder with 1 million parameters [45]). Despite these efforts, data shift remains unresolved, as no model can comprehensively cover all possible distributions in scientific domains. ③ Resource constraints: The large size of these models imposes significant computational demands on resource-constrained hardware during deployment, making their use for decompression inference challenging in practical scenarios.

This paper introduces NeurLZ, a novel method to enhance the quality of error-regulating scientific lossy compression using online learning. Instead of relying on offline-trained large models toward generalization to reconstruct data, we integrate the process of finding data features into the scientific compression workflow in a finer-grain online manner, which *orthogonally* enhances a conventional compression method rather than substituting it. More specifically, NeurLZ is a quality enhancer that leverages the learned error residual features to improve the quality of the reconstructed data from the conventional method during decompression.

Effective enhancement of the conventional data reconstruction entails a reduction in the compression error, i.e., the difference between the reconstructed data and the original data. Figure 2 (top) displays the error magnitude changes before and after the enhancement, with the latter smaller in error magnitude within the error bound. Also, the error residual between the conventionally reconstructed data and its enhancement is "memorized" in a DNN model. We find that the reconstructed-enhanced error residual can be encoded using only thousands of parameters (e.g., 3,000). Despite seemingly a very small number of parameters, it efficiently captures the fine details and complex patterns in the compression error overlooked by conventional compressors. This method is much more effective than "memorizing" reconstructed-original error with full fidelity at an unknowingly high cost; it is also three orders of magnitude lower than previous DNN compression work in parameters (millions).

To do so, during compression, we online train a lightweight DNN model for each input data field, which "memorizes" the features of the residual between the reconstructed and original data. The method is at a per-block granularity, and the learned data feature is directly saved to the compressed file and is only applied to the input data. The core innovation of NeurLZ is to *handle data shift through fast and adaptive online learning of lightweight skipping DNN models*. These models learn residuals between decompressed and original data, integrating cross-field learning with error management. Unlike usually practiced DNN-based compressors [44, 45], which struggle with data shift, NeurLZ dynamically adapts to changing data characteristics, making it highly effective for diverse scientific datasets. Designed as lightweight enhancers, the skipping DNN models minimize storage and computational overhead while improving decompression quality with minimal resource impact.

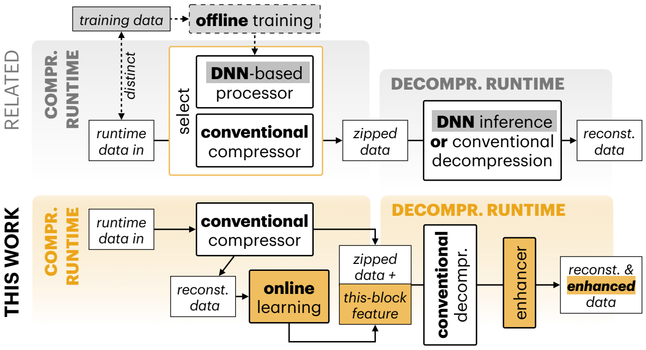


Figure 3: The NeurLZ workflow (*bottom*) compared with related work consisting of a conventional compressor and alternative DNN-based processor (*top*).

The key contributions are summarized as follows:

- A novel method, NeurLZ, to enhance the quality of scientific lossy compression with online neural learning. NeurLZ dynamically trains lightweight skipping DNN models during compression, enabling efficient adaptation to data shift. We design the method to be theoretically adaptive to any conventional lossy compressors for quality enhancement.
- Compression error mitigation: NeurLZ efficiently mitigates compression errors introduced by scientific compressors. By employing lightweight skipping DNN models, NeurLZ introduces little space and time overhead to the original compression process while significantly enhancing the quality of reconstructed data.
- Error control: our framework introduces two regulation modes, allowing users to choose from strict error control with respect to the user-input error bound for the compressor and the relaxed error control that achieves overall better quality with worst-case capped at 2× the user-input error bound.
- Cross-field learning: We leverage cross-field correlations derived from governing equations in scientific simulations to improve model predictions. By incorporating data from multiple fields, the framework captures cross-field dependencies often overlooked by traditional lossy compressors, further improving reconstruction quality.

The cross-field learning can be adopted based on data features in various scientific applications.
- Extensive evaluation: NeurLZ has been evaluated on multiple datasets and compared against state-of-the-art lossy compressors like SZ3 [40] and ZFP [41], demonstrating significant performance gains.

2 Background and Motivation

2.1 Lossy Compression

Conventional error-bounded lossy compressors fall into several types, including prediction-based, transform-based, and HOSVD-based approaches [18]. Prediction-based compressors [17, 37–39, 41, 46, 48, 70, 71, 81, 83] use predictors to estimate pointwise data based on neighboring values, followed by quantization to maintain user-defined error bounds. Data prediction is the most critical stage in these compressors, as higher accuracy reduces the burden on subsequent steps [18]. Existing methods employ predictors such as curve fitting [17], spline interpolation [34], multidimensional prediction [70], and higher-order predictors [83]. Transform-based approaches [35, 41] operate by converting the data into a coefficient space that is more amenable to compression due to its sparsity. However, maintaining control over the error within predefined bounds in this domain is often challenging. HOSVD-based techniques [9, 10] leverage Higher-Order Singular Value Decomposition (HOSVD) to decompose data into matrices and a compact core tensor, ensuring that the L2 norm error is preserved. While these methods excel in achieving high compression ratios by capturing global correlations in the data, they are significantly limited by their slow computational performance.

To push the limit of conventional lossy compression, recent efforts [36, 44, 45, 50] attempt to leverage deep neural networks (DNNs) to predict the reconstructed data adaptively, as a learnable predictor trained offline and integrated into existing prediction-based lossy compressors.

2.2 Deep Neural Networks

Deep neural networks (DNNs) are powerful machine learning models composed of hierarchical neural layers that extract increasingly complex features from input data. DNNs have achieved remarkable success in a wide range of real-world applications [19, 28, 72, 79]. However, a persistent challenge in deploying DNNs lies in their generalization problem – the gap between their performance on the training data and their prediction performance on unseen data [53, 63]. This issue is exacerbated by data shift, where the distribution of the training data diverges from that of the deployment data [57]. DNNs are fundamentally designed under the assumption that training and deployment data are drawn from the same independent and identically distributed (i.i.d.) distribution. When this assumption breaks, as is common in real-world scenarios, their learned representations often fail to adapt, leading to unreliable and suboptimal results. Addressing these challenges requires approaches that not only mitigate the impact of distribution shifts but also dynamically adapt to evolving data environments to maintain consistent performance.

2.3 Motivation

Our ultimate goal is to fully unlock the power of DNNs to push the limit of scientific lossy compression quality. In this subsection, we comprehensively rethink the relationship between the utilization of DNNs and the principles of lossy compression. Specifically, we investigate the optimal design philosophy to integrate DNN models by analyzing and answering the following two critical questions.

> *Question* 1*: What is the best learning regime for lossy compression? (Offline Learning vs. Online Learning.)*

Existing offline-learning methods [36, 44, 45, 50] feature a large DNN model pre-trained on a pre-collected static dataset and then deployed for inference. However, this approach heavily assumes that training and deployment data share the same underlying distribution. In practice, this assumption rarely holds, as scientific domains frequently involve complex systems with significant distribution variance across fields like cosmology simulations (e.g., Nyx [2]), large-scale turbulence (e.g., Miranda [15]), and weather simulations (e.g., Hurricane [29]). Each domain has distinct data distributions, and even within a single domain, the data can vary substantially across spatial or temporal contexts [82]. For instance, Nyx exhibits temporal variations corresponding to different stages of the universe's evolution indicated by redshift [2]. These temporal shifts result in fundamentally different data distributions as physical phenomena evolve and dominate at distinct epochs. Consequently, offline learning can be ill-equipped to adapt to such temporal dynamics, further amplifying the impact of data shift on model performance.

Online learning [12, 25, 26, 66] addresses this issue by allowing models to dynamically update their parameters or replace themselves as new data becomes available. This continuous updating approach ensures the model stays aligned with evolving data distributions, maintaining robust performance over time. Research in real-time psychographic imaging [6] and autonomous X-ray reflectometry experiments [61] highlights the potential of online learning to address data shift problems across scientific domains.

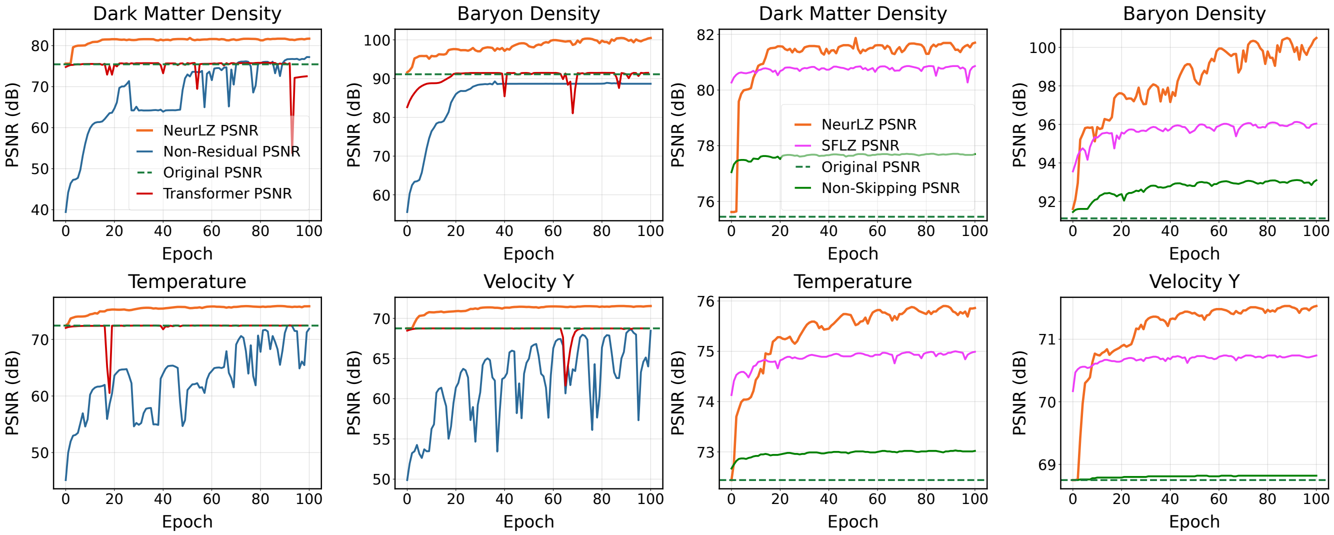


Figure 4: NeurLZ achieves higher PSNR across all Nyx fields, outperforming both the non-residual and transformer models (left), as well as the non-skipping and single-field learning (SFLZ) models (right).

Recognizing these advantages, NeurLZ adopts online learning as the core regime to tackle the challenges posed by data shift in scientific lossy compression. By leveraging the adaptability of online learning, NeurLZ ensures consistent and high-quality compression, making it well-suited for dynamic and diverse scientific scenarios. This approach represents a significant step toward overcoming the limitations of traditional offline learning and addressing the complexities of real-world datasets.

> *Question 2: What is the appropriate learning granularity for scientific data compression? (Generalization vs. Customization.)*

Generalization is a potential online learning strategy that trains a huge DNN model in the compression process, handling the entire target data. While this approach can "memorize" a wide range of features, it incurs significant computational and inference costs, requiring substantial resources for training, deployment, and user-side inference. Moreover, downloading and maintaining such a model adds significant I/O and storage overhead, leading to impracticality and unscalability in HPC applications.

On the contrary, customization employs multiple lightweight DNN models that deal with individual small data blocks. These models learn per-block data mappings, capturing the unique characteristics of each block. Compared to a general huge model, customized lightweight models benefit from lower computational costs, reduced inference resource requirements, and minimal download overhead for users. NeurLZ adopts this customization online learning strategy, ensuring high-quality compression performance and resource efficiency by updating multiple block-specific lightweight DNN models online in the lossy compression process.

3 Methodology: NeurLZ

3.1 Method Overview

As highlighted in Sec. 2.3, NeurLZ distinguishes itself from prior works by employing online neural learning and enhancing the quality of reconstructed data using learned DNN enhancer models in the post-processing stage, as shown in Figure 3. NeurLZ dynamically learns multiple lightweight DNN models for each runtime data block in a customized way. Specifically, when a runtime data block arrives, a lightweight DNN model is initialized for online learning in the compression process. NeurLZ leverages the decompressed data as input and trains the lightweight model to predict the residual error (i.e., the difference between the original and reconstructed data). This process trains DNN models to minimize the discrepancy between decompressed and original data, effectively refining the decompressed data to recover the original values.

Once the DNN encodes the error residuals (Figure 2), the model weights (negligible size compared to compressed data) and outlier coordinates (points exceeding 1× error bound, detailed in Sec. 3.3), are packaged as the specific feature for this runtime data block along with the zipped data. This packaged format is then sent to the user. On the user side, the reconstruction process begins with conventional decompression to obtain reconstructed data. The saved lightweight DNN models and the outlier coordinates are used to enhance data quality in the post-decompression stage. Specifically, the DNN models predict the residual error, which is then added to the reconstructed data to generate enhanced data block by block. For data points in the outlier coordinates, the original reconstructed values are retained, as will be detailed in Sec. 3.3. This ensures the enhanced data achieves high fidelity while adhering to the user-defined error bound.

3.2 Lightweight Online Neural Learning

As discussed in Sec. 2.3, our NeurLZ contains multiple lightweight DNN models, which can be considered a memorizer to encode residual error information. Each memorizer is learned for a specific runtime data block, capturing its unique characteristics. The memorizer must balance feature-extracting capability and size: a weak model provides low-quality enhancement, while a large model suffers from significant overhead by outweighing the compressed data. We will introduce the detailed learning design of our NeurLZ.

3.2.1 Learning Residual Error Information. A key challenge in NeurLZ's learning process is enhancing decompressed data to recover the original values, especially given the large value ranges in scientific simulations, such as the 4.78×10^6 range in Nyx's Temperature field [82], which causes training instabilities. Instead of directly predicting the original data, NeurLZ estimates the residual error, $R = X - X'$, inspired by residual learning techniques [32, 69, 80]. By treating data slices as single-channel images, the DNN model focuses on smaller residuals, stabilizing training and improving prediction accuracy. This approach effectively enhances decompressed data to maximally recover the original data, making the DNN a strong feature-capturing memorizer for each runtime data block.

The four left subfigures in Figure 4 compare learning residual errors to learning original values for the Nyx dataset compressed with SZ3 under a relative error bound of 1E-3. PSNR (Peak Signal-to-Noise Ratio) in decibels (dB) is used to measure compression-introduced errors; higher PSNR indicates better reconstruction quality. NeurLZ (orange solid line) consistently achieves higher PSNR and improves reconstruction quality with learning residual error. In contrast, directly learning original values (green solid line) shows lower and less stable performance. These results demonstrate the effectiveness of the proposed residual-error learning strategy in improving reconstruction quality.

Upon completing the learning process during NeurLZ-compression, the trained DNN model predicts the residual error map $\hat{R}$ directly from the decompressed data X' through learned correlations. The predicted error is then added back to X', further enhancing the overall reconstruction quality as $\hat{X} = X' + \hat{R}$.

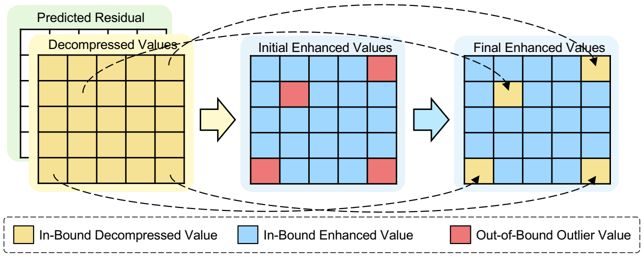


Figure 5: Outlier management: decompressed (reconstructed) values are further enhanced with predicted residuals, and outliers are replaced with in-bound decompressed values to reliably satisfy the error bound.

3.2.2 Learning Multi-Scale Patterns with Skipping Neural Networks. The skip connection structure empowers DNNs to represent intricate patterns spanning various scales [19, 24, 28, 64, 72, 79]. Accordingly, we design lightweight skipping DNNs, as shown in Figure 8. The input slices from decompressed blocks undergo down-sampling and up-sampling through convolution and de-convolution to extract multi-scale features. Skip connections concatenate low-level features with deeper layers, enriching the representation. The fused features produce a single-channel output as the predicted residual map, which, when added to the decompressed slice, yields the enhanced slice via NeurLZ.

Leveraging the skip connection structure's ability to capture multi-scale information, our skipping DNNs effectively model complex patterns while preserving high-fidelity details. As shown in Figure 4-right, the non-skipping model (green solid line) achieves lower PSNR compared to our NeurLZ model (orange solid line) with skip connections. Beyond improving reconstruction quality, the integration of convolution and de-convolution operations ensures model compactness; for instance, a 10-layer network requires only 3,000 parameters. This lightweight design makes the memorizer particularly suitable for transmission, especially when compared to billion-parameter Transformer models [44, 54].

Furthermore, as shown on the left side of Figure 4, we compare our skipping model with the Transformer-based HAT model [13], designed for super-resolution tasks. Despite its 5 million parameters—far exceeding our 3,000-parameter skipping model – HAT struggles to improve reconstruction quality, even excluding its longer training time. This is due to Transformers being more data-hungry, making them more challenging to learn on small datasets [21, 23, 51, 67, 75]. In our case, the original data block size is 512^3, equivalent to 512 images whose sizes are 512-by-512—a very small dataset by computer vision standards. In contrast, our memorizer design, which incorporates skip connections, achieves better results with significantly smaller data requirements. Its compact and effective nature perfectly aligns with NeurLZ's goals for enhancing scientific data compression quality.

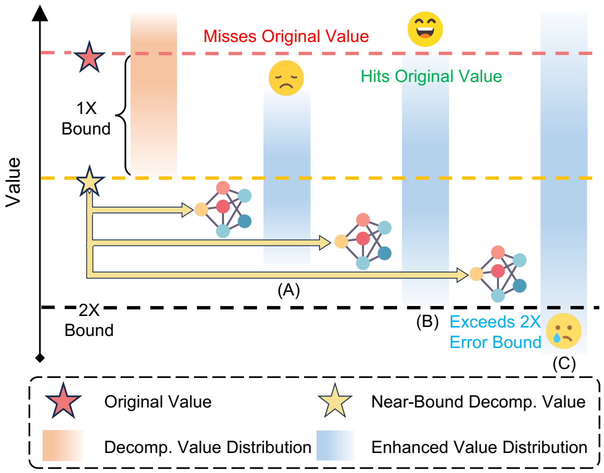


Figure 6: Regulated DNN output: The red band shows decompressed values, with the worst-case value (yellow star) processed by three DNNs—tight (A), balanced (B), and loose (C) regulation. The blue bands show that Case A misses the original, Case B reaches it accurately, and Case C exceeds the 2× bound.

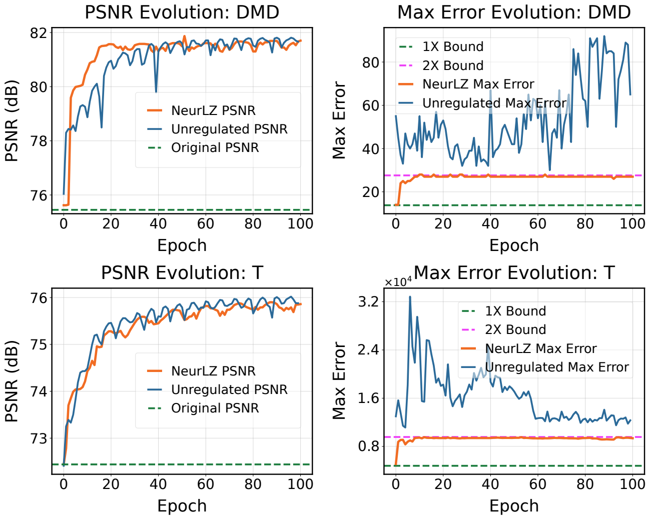


Figure 7: Training evolution of PSNR and max absolute error for two fields in the Nyx—NeurLZ (regulated) vs. unregulated cases. NeurLZ maintains stability within the 2× bound, while the unregulated case exceeds it.

3.3 Flexible and Strict Error Bounding

3.3.1 Strict Error Control. The inherent uncontrollability of DNN output causes some data points, even after enhancement with predicted residuals, to exceed the error bound. These points, referred to as outliers, require special handling to guarantee the error-bound. For the strict error-bounded reconstruction enhancement, we store the coordinates of outlier points, defined as those whose enhanced values exceed the acceptable error threshold (Figure 5-left). During reconstruction, these coordinates are used to identify and replace out-of-bound enhanced values with their corresponding decompressed values, ensuring all data points remain within the predefined error limits (Figure 5-right). This strategy leverages DNNs' ability to capture complex patterns while addressing output variability, preserving accuracy and reliability. However, storing outlier coordinates adds additional storage overhead. We denote the size of the i-th dimension as dim_i and the average number of bits required to store the N-D coordinates of a single point as $\overline{B}$. With $\overline{B} = \sum_{i=1}^{N} \log_2(\text{dim}_i)$, the coordinate overhead of storing outliers in bits is obtained by the number of outliers multiplied by $\overline{B}$. For the datasets discussed in this paper, the $\overline{B}$ values are as follows: Nyx has $\overline{B}$ values of 27.0 bits for size= 512^3, 30.0 bits for size = 1024^3, and 33.0 bits for size= 2048^3; Miranda has a $\overline{B}$ of 25.2 bits; Hurricane a $\overline{B}$ of 24.6 bits.

3.3.2 Regulating Neural Output. In some cases, storing the coordinates of outliers incurs excessive overhead, as shown in Sec. 5.1. We propose a regulating technique for application users who need an extremely high compression ratio to achieve a 2× error bound per datum without storing outlier coordinates. As depicted in Figure 6, consider a single datum: the original value (red star ★) is compressed and decompressed within a 1× bound by lossy compressors, with the decompressed value distribution shown as the red band. The worst-case decompressed value (yellow star ★) lies at the bound, 1× from the original. This value is input into the DNN, producing an enhanced value within the range shown as the blue band. While the DNN's output cannot be precisely controlled, its range can still be effectively regulated.

Figure 6 shows three scenarios with varying degrees of DNN output regulation. In Case A, tight regulation keeps the enhanced value within the 2× bound, but the maximum falls short of the original, potentially limiting performance. Case B achieves balanced regulation, with the minimum meeting the 2× bound and the maximum reaching the original, allowing the DNN to recover the original value through learning. In Case C, loose regulation allows the maximum to exceed the original, but the minimum also surpasses the 2× bound, risking degraded reconstruction quality.

In NeurLZ, balanced regulation normalizes residuals with the error bound and uses a Sigmoid layer in the skipping DNN to constrain outputs to [0, 1], equivalent to a 2× bound. Figure 7 shows the training evolution of PSNR and maximum absolute error for two Nyx fields, comparing regulated NeurLZ and the unregulated case. Both enhanced SZ3 with a 1E-3 bound and achieved similar PSNR, but

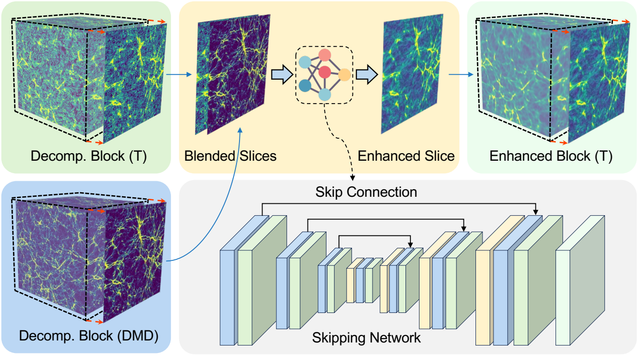


Figure 8: Skipping DNN model and cross-field learning. 'Decomp.' denotes the decompressed data, while 'T' and 'DMD' denote two fields where the former is enhanced.

regulated NeurLZ ensures stable training. It consistently keeps errors within the 2× bound. By default, outlier coordinates are saved to enforce strict bounds, but omitting them improves compression ratios with minimal impact (Sec. 5.1).

3.4 Cross-Field Learning

Scientific applications model multiple physical fields in a spatiotemporal domain, such as temperature, velocity, and density in the Nyx simulation, governed by complex equations like *Dark Matter Evolution* and *Self-gravity Equations* [2, 52, 59]. However, existing lossy compressors [17, 40, 43, 44, 70, 83] are typically limited to take single-field inputs, thereby overlooking inherent physical connections among multiple fields and timesteps.

Using skipping DNNs to model complex inter-field patterns, we implement cross-field learning, as shown in Figure 9. For instance, predicting a single pixel (red cube in the Temperature field) leverages local values and information from other fields. These patterns are learned during training, enabling predictions informed by correlations across fields. By capturing such relationships, the skipping model achieves a more accurate prediction of the data source, thereby enhancing compression ratios.

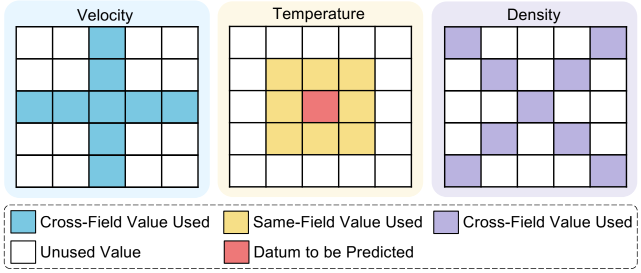


Figure 9: Cross-field learning: A datum's prediction leverages values from its own field as well as other related fields through learnable patterns.

Figure 8 illustrates the training process of a skipping model for cross-field learning. Decompressed slices from Temperature (T) and Dark Matter Density (DMD) serve as a 2-channel input to predict the residual map for Temperature. After fusing features across channels, the model outputs the residual map, which is added to the decompressed slice to produce an enhanced slice via NeurLZ. The right side in Figure 4 highlights the advantages of cross-field learning. While single-field learning (purple solid line) provides some improvement over the original PSNR (green dashed line), it consistently underperforms compared to cross-field learning (orange solid line). This underscores the superior ability of cross-field learning to capture complex inter-field dependencies and enhance model performance.

4 Evaluation

In this section, we present the evaluation results of the proposed NeurLZ on typical scientific applications compared to existing lossy compressors for scientific data.

4.1 Experimental Setup

■ **Testbed.** Experiments were conducted on the servers with eight NVIDIA RTX 6000 Ada GPUs, two 24-core AMD EPYC Genoa 9254 CPUs, and 1.5 TB of DRAM each.

■ **Datasets.** Table 1 summarizes the three datasets used in our experiments: Nyx [2], Miranda [15], and Hurricane [29]. Sample blocks for Miranda and Hurricane were obtained via SDRBench [82]. Sample dataset of Nyx was sourced from SDRBench and NERSC open-data portal [58] for scalability-related experiments.

■ **Learning configurations.** Each skipping DNN model comprises four down- and up-sampling operations with skip connections, totaling around 3,000 parameters. The training was conducted over 100 epochs with a batch size of 10, an initial learning rate of 1E-2, and a cosine annealing schedule. Once trained, the model weights are embedded into the final compressed data format in FP32 or FP64 precision, aligned with the dataset's precision.

■ **Lossy compressor.** NeurLZ works seamlessly with various lossy compressors. We select two popular ones: the prediction-based SZ3 [40] and the transform-based ZFP [41]. In prediction-based compressors like SZ3, superior predictors effectively capture correlations. NeurLZ enhances this process by capturing previously overlooked correlations. For ZFP which uses orthogonal transforms to decorrelate data [18], NeurLZ evaluates its performance with transform-based compressors.

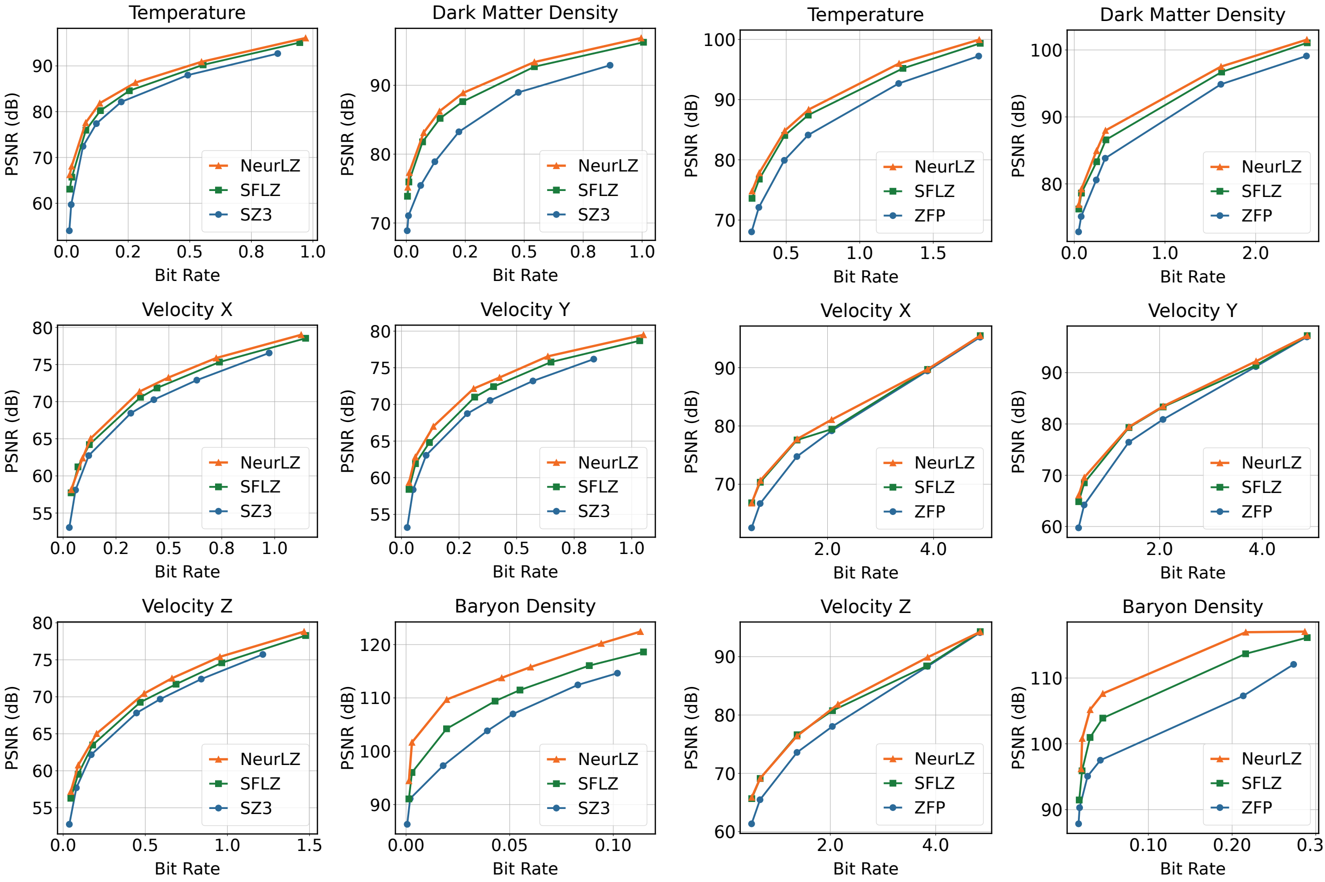


Figure 10: Bit rate comparison for the Nyx dataset: (Left) SZ3-based methods—original SZ3, SFLZ (single-field learning), and NeurLZ (cross-field learning). (Right) ZFP-based methods—original ZFP, SFLZ, and NeurLZ.

■ **Performance evaluation criteria.** Using compressor-specified error bounds, NeurLZ enhances decompressed values. In SZ3, we use a value-range-based relative error bound (denoted as e), which is equivalent to the absolute error bound ϵ used in ZFP. The relationship between them is given by $e = \epsilon \cdot value\_range$, where $value\_range$ represents the specific data block's value range being compressed. For simplicity, all error bounds are referred to as e. Experiments are defined by the compressor type, error bound, dataset, and field, and NeurLZ is evaluated using the criteria described below:

- Error validation: verifying that NeurLZ's error is strictly bounded by the user-defined error threshold.
- Data visualization: assessing the visual quality of enhanced data relative to the original decompressed data.
- Bit rate vs. PSNR: comparing PSNR across bit rates.
- Bit rate reduction: evaluating NeurLZ's reduction relative to SZ3 and ZFP at equal PSNR.
- Scalability testing: testing NeurLZ on larger datasets.

Table 1: Experiment-used datasets [82].

Dataset	Domain	Per Block Size	Data Type
Nyx [2, 58]	Cosmology	$512^3/1024^3/2048^3$	FP32
Miranda [15]	Large Turbulence	(256, 384, 384)	FP64
Hurricane [29]	Weather	(100, 500, 500)	FP32

4.2 Experimental Results

■ **Error validation.** To validate the error distribution, we first analyze training dynamics. Figure 12 shows experiments on two Nyx fields, where NeurLZ enhances SZ3 with a 1E-3 bound. PSNR improves rapidly, then stabilizes around 100 epochs, and significantly exceeds the original. The Outlier Rate (OLR) (%), representing the percentage of outliers, decreases sharply to 0.01%, reducing the coordinate overhead for storing outlier locations. These improvements yield bit rate reductions of 66.4% and 36.3% at the same PSNR, respectively (Table 2).

After learning, outlier coordinates are saved in the compressed archive. During decompression, outliers are replaced by in-bound decompressed values, ensuring that strict error bounds are maintained. Figure 11 clearly demonstrates the enhancement of SZ3 compression under a stricter targeted 5E-5 error bound, where NeurLZ confines all errors within this limit, with a more concentrated error distribution than that of SZ3.

■ **Data visualization.** Figure 13 demonstrates the performance of SZ3, NeurLZ , NeurLZ (Relaxed), and NeurLZ (Unregulated) on Nyx field T at the same bit rate, evaluated using PSNR, MAE, DSSIM [65], and FLIP [4]. While PSNR is commonly used to assess reconstruction quality, it does not fully account for human perceptual sensitivity to visual differences [4, 65]. To address this, additional metrics are considered: MAE (Mean Absolute Error) quantifies pixel-wise reconstruction errors, DSSIM (Structural Dissimilarity Index) evaluates structural fidelity, and FLIP, originally proposed by NVIDIA, measures perceptual differences aligned with human vision. Higher PSNR values indicate better quality, and lower values for MAE, DSSIM, and FLIP indicate better performance overall.

Compared to SZ3, NeurLZ achieves significantly higher PSNR and lower MAE, DSSIM, and FLIP, highlighting its ability to produce reconstructions that are not only quantitatively superior but also more perceptually accurate. Between NeurLZ (Relaxed) and NeurLZ (Unregulated), the detailed analysis in Sec. 5.1 provides a further discussion on their trade-offs. However, as shown here, NeurLZ (Relaxed) consistently delivers the highest PSNR and the lowest MAE, DSSIM, and FLIP among all methods, making it an appealing choice for users seeking objective accuracy and perceptual quality at the same bit rate.

■ **Bit rate vs. PSNR.** Bit rate reflects the average number of bits required to represent one data point, calculated from

$$bitrate = \frac{\texttt{size}(Z) + supplementary}{num\ points\ of\ original\ data},$$

where $\texttt{size}(Z)$ is the size of compressed data Z. The *supplementary* includes storage for outlier coordinates and model parameters. Thus, the bit rate reflects the compressed data size and the storage costs of outliers and model parameters, providing a comprehensive measure of average storage cost. Figure 10 (left) presents the Bit rate vs. PSNR curves for SZ3, SFLZ, and NeurLZ across different Nyx fields, comparing their compression performance under varying error bounds. SFLZ applies single-field learning, while NeurLZ incorporates cross-field learning.

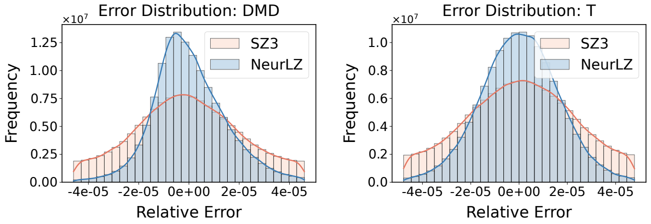


Figure 11: Error distribution for two Nyx fields under a 5E-5 relative error bound, comparing SZ3 decompressed and NeurLZ-enhanced values to the original.

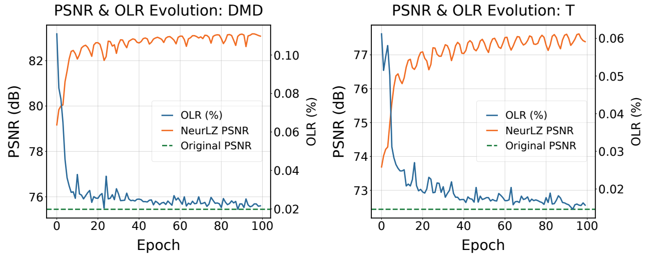


Figure 12: Training evolution of PSNR and outlier rate (OLR) (%) for two fields in the Nyx, using NeurLZ to enhance SZ3 with a 1E-3 relative error bound.

NeurLZ and SFLZ curves consistently shift to the upper right compared to SZ3 across all fields, indicating higher PSNR at the same bit rate or lower bit rate for the same PSNR. This improvement is particularly pronounced in fields like Dark Matter Density and Baryon Density. Interestingly, NeurLZ does not always mirror SFLZ's movement. For example, in the Baryon Density field, NeurLZ's final point lies above and to the right of SFLZ, achieving a lower bit rate and higher PSNR. This advantage stems from NeurLZ's use of cross-field information, which reduces the proportion of outliers and thereby decreases overhead.

Beyond SZ3, NeurLZ's impact on ZFP is also evaluated. Figure 10 (right) presents Bit rate vs. PSNR curves for ZFP, SFLZ, and NeurLZ across Nyx fields. For velocity fields, SFLZ shows no improvement over ZFP at higher bit rates, and NeurLZ's gains diminish as the bit rate increases. For fields such as Temperature, Dark Matter Density, and Baryon Density, NeurLZ consistently outperforms SFLZ, and both significantly surpass the baseline ZFP. These results highlight NeurLZ's ability to capture dependencies overlooked by ZFP, particularly through leveraging cross-field relationships. However, as ZFP converts data into a coefficient space, achieving further improvements may require unique architectural designs tailored to transform-based compressors. Nevertheless, the significant enhancements observed in these fields validate NeurLZ's potential to improve compression efficiency in such frameworks.

■ **Bit rate reduction.** To quantify NeurLZ's improvements, three datasets (Table 1) are tested with varying relative error bounds on SZ3 and ZFP. SZ3 uses bounds from 1E-2 to 1E-4, while ZFP uses 1E-1 to 1E-3 to maintain comparable PSNR ranges. ZFP's conservative error distribution results in higher PSNR under the same bound [70], necessitating

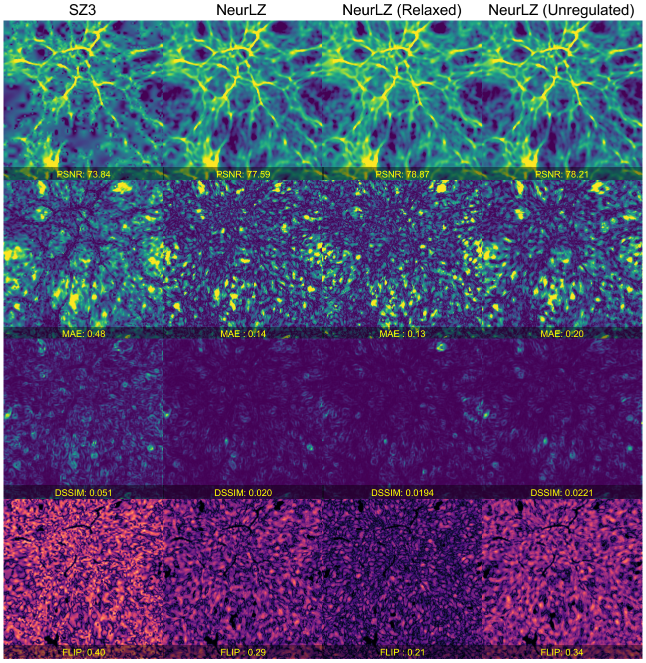


Figure 13: Comparison among SZ3, NeurLZ, NeurLZ (Relaxed), and NeurLZ (Unregulated) on Nyx field T at the same bit rate. Higher PSNR and lower MAE, DSSIM, and FLIP are preferred. "Relaxed" relaxes strict error control with a regulated 2× error bound, while "Unregulated" has no enforced error bound.

more relaxed bounds. Table 2 presents the relative bit rate reduction (%) at equal PSNR.

NeurLZ demonstrates significant improvements with SZ3, particularly for the Nyx dataset. 1 For Baryon Density, the bit rate reduction reaches 91.3% at a 1E-3 error bound, with both it and Dark Matter Density exceeding over 40% reductions across all bounds. Temperature achieves 62.9% at 1E-2, but reductions decrease for stricter bounds, except for Baryon Density, which improves from 50.1% at 1E-2 to 69.0% at 1E-4, suggesting NeurLZ captures more relationships under stricter bounds. 2 For Miranda, most fields see 20% reductions at 1E-4, following the trend of lower reductions with stricter bounds. Although less pronounced than Nyx, the improvements remain consistent. 3 For Hurricane, reductions are weaker, under 10% at 1E-4, with a maximum of 36.8% for W at 1E-2. Hurricane's unique characteristics as a climate dataset may account for the smaller gains compared to Nyx and Miranda.

As for ZFP, NeurLZ achieves the largest bit rate reductions of 81.8%, 29.3%, and 28.1% for Nyx, Miranda, and Hurricane, respectively. While lower than SZ3's 91.3%, 37.4%, and 36.8%, they remain significant. In Nyx, Temperature and Baryon Density see the largest reductions at 1E-2, not the loosest 1E-1 bound. Miranda shows stable reductions above 10% across all tested bounds without SZ3's dynamic variations. For Hurricane, reductions are smallest (<5%) at stricter bounds and largest (20%) at looser bounds. These results collectively show NeurLZ's effectiveness on ZFP, despite its transform-based nature posing more significant challenges than prediction-based SZ3.

Table 2: NeurLZ-achieved relative reduction (%) in bit rate at equal PSNR across three datasets for SZ3 and ZFP; higher values indicate better overall performance.

Compressor		SZ3					ZFP				
Error Bound		1E-2	5E-3	1E-3	5E-4	1E-4	1E-1	5E-2	1E-2	5E-3	1E-3
Nyx	1. Temperature	62.9	51.2	36.3	37.0	21.8	25.5	25.6	28.0	27.0	21.8
	2. Dark Matter Density	89.1	86.4	66.4	57.2	40.5	53.4	54.9	35.4	36.7	24.7
	3. Velocity Y	44.5	41.5	36.5	24.0	1.17	28.6	29.5	21.6	15.1	3.67
	4. Baryon Density	50.1	76.8	91.3	91.1	69.0	40.1	75.3	81.8	79.0	38.4
Miranda	1. Diffusivity	37.4	36.2	32.8	30.3	21.9	12.4	29.2	22.0	19.3	11.4
	2. Velocity Z	29.9	28.1	28.3	25.1	23.0	10.6	26.1	26.0	23.9	14.8
	3. Viscocity	36.5	36.0	31.6	30.7	20.8	11.8	29.3	21.8	18.0	13.6
Hurricane	1. Cloud	27.7	16.9	4.54	5.07	4.34	15.3	13.7	8.79	6.30	4.43
	2. Precip	24.2	30.9	19.3	14.2	4.85	16.3	14.5	10.5	9.03	5.26
	3. W	36.8	30.6	24.9	18.1	6.50	28.1	25.3	16.7	13.3	6.46

■ **Scalability testing.** Table 3 demonstrates NeurLZ's scalability on a Nyx field across dimensions and training epochs. Larger dimensions achieve higher bit rate reductions, up to 93.9% for 2,048 at 10 epochs. This cross-epoch trend suggests our memorizer captures broader spatiotemporal and cross-field patterns compared to the conventional local feature search in SZ3 [40], as discussed in Sec. 5.2. With larger data dimensions, our memorizer naturally attends to a wider data range in a single pass, offering a broader attention scope. This gives it a distinct advantage over SZ3, whose focus is constrained to local features.

For size 2,048, NeurLZ achieves 80.0% reduction in 1 epoch, with training time at 41.9% of SZ3's compression time. Inference time remains constant across epochs, maintaining a 46.3% decompression time. As dimensions grow, the reduced ratios highlight NeurLZ's efficiency for large data, dropping from 95.7% and 93.6% for 512 to 41.9% and 46.3% for 2048, enabled by convolutional networks' ability to process large data regions in a single pass.

The training-time overhead compared to SZ3 is justified by 1) the long-term storage benefits [3, 56] and 2) the future elimination by parallel optimization. NeurLZ's strong scalability creates opportunities to explore its performance on even larger datasets in the future.

5 Discussions

5.1 Doubling the Error Bound

The central question surrounding error regulation is

Table 3: The NeurLZ-achieved relative reduction (%) in bit rate at equal PSNR on a Nyx field across varying dimensions and training epochs for SZ3.

Size	Metrics (↓) / Epochs (→)	1	2	5	10
512^3	Rate Reduction (%)	8.3	41.0	51.5	72.9
	Train/Comp. Time (%)	95.7	168.	539.	930.
	Inf./Decomp. Time (%)	93.6	93.6	93.6	93.6
1024^3	Rate Reduction (%)	45.4	61.3	75.9	82.6
	Train/Comp. Time (%)	48.	104.	240.	475.
	Inf./Decomp. Time (%)	59.2	59.2	59.2	59.2
2048^3	Rate Reduction (%)	80.0	84.2	88.6	93.9
	Train/Comp. Time (%)	41.9	83.6	209.	418.
	Inf./Decomp. Time (%)	46.3	46.3	46.3	46.3

> *whether doubling the error bound can eliminate the need for storing outlier.*

As discussed in Section 3.3, NeurLZ achieves strict 1× error bounds by storing outlier coordinates. Figure 14 shows the error distribution, along with the percentage of outliers under a 5E-5 bound for two fields, comparing SZ3 decompressed values with NeurLZ initial enhanced values. To maintain strict error control, final enhanced values are obtained by replacing outliers with decompressed values (Figure 5), which necessitates storing outlier coordinates. This results in bit rate savings of 20.9% and 15.6% for these two fields, respectively.

By skipping the storage of outlier coordinates, however, compression efficiency improves significantly, with bit rate savings increasing to 37.6% and 25.7%, providing over 10% additional storage reduction in both cases. More importantly, relaxing strict error control does not noticeably compromise reconstruction quality. As discussed in Sec. 3.3, the regulation mechanism in NeurLZ ensures that errors remain within a 2× bound. Figure 13 further illustrates that NeurLZ (Relaxed) leverages this mechanism to achieve higher reconstruction quality than strict error control (NeurLZ) at the same bit rate.

Regarding the importance of regulation, we test NeurLZ (Unregulated), which removes both strict error control and regulation mechanisms. While it achieves a PSNR of 78.21 (even higher than NeurLZ), its lack of constraints results in significantly worse MAE, DSSIM, and FLIP scores. This aligns with the observations in Figure 7, where a controlled experiment demonstrated that Unregulated and Regulated approaches achieve similar final PSNR values. However, the absence of regulation in the Unregulated approach leads to substantially higher overall error magnitudes. These findings highlight that our regulation is essential for providing a flexible and practical trade-off between compression efficiency and reconstruction quality.

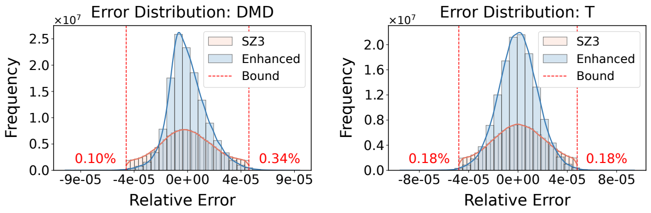


Figure 14: Error distribution for two Nyx fields—SZ3 decompressed vs. NeurLZ-initial-enhanced values. Outlier rates are 0.10%–0.34% for DMD and 0.18% for T.

5.2 Uncovering Cross-Field Contributions

The central question surrounding cross-field learning is

> *how different fields contribute to predictions.*

Figure 4 shows that cross-field learning achieves higher PSNR than single-field learning. Information from multiple fields enables more accurate predictions, enhancing PSNR. Understanding how different fields contribute to this improvement is crucial – how much does each field contribute? Are all fields equally helpful? We analyze the Temperature field using SZ3 with a 5E-3 bound. Incorporating two other Nyx fields into cross-field learning, we train the model for 100 epochs and use Captum with integrated gradients for attribution analysis [31, 68].

Using decompressed slices (512 × 512) from these three fields, the model predicts the residual error for Temperature. Attribution analysis targets the datum at position (256, 256) in the Temperature slice. Captum computes scores quantifying each field's contribution to this prediction. Figure 15 shows the results: the left slice displays the target marked by a red star, while significant attribution regions are highlighted with red rectangles across all slices for clarity.

Each field shows a prominent red rectangle around (256, 256), corresponding to the predicted datum, clearly highlighting NeurLZ's identification of local patterns near the target across fields. Additionally, smaller rectangles distributed across the fields indicate global patterns contributing to the prediction. The variation in local and global patterns among fields suggests differing levels of contributions from each field. This visualization effectively demonstrates that NeurLZ leverages adaptable, learnable cross-field patterns, enhancing the reconstruction quality.

5.3 Analyzing Performance Limitations

The central question surrounding performance limitations is

> *how conflicts affect NeurLZ at stricter error bounds.*

As discussed in Section 4.2, stricter error bounds reduce NeurLZ's effectiveness. For example, in the Nyx Velocity Y

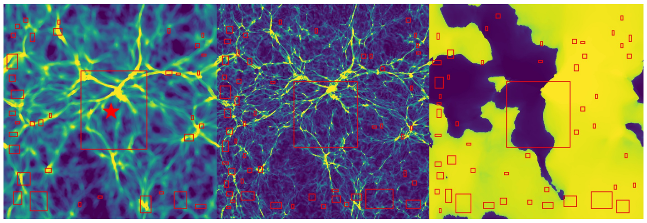


Figure 15: Significant attribution regions (red rectangles) are overlaid on slices from three Nyx fields—T, DMD, and Vx—highlighting the areas that contribute to the prediction at the center of the T field (red star).

field with a 1E-4 error bound, only a 1.17% bit rate reduction is achieved by `NeurLZ` (Table 2), and with smaller bounds like 5E-5, no improvement may occur. Figure 16 shows the PSNR and outlier rate (OLR) evolution during training for Velocity Y with a 5E-5 error bound. PSNR plateaus for approximately 20 epochs, then briefly rises before stagnating, while OLR increases alongside PSNR, leading to a negative bit rate reduction, indicating a potentially detrimental effect. This contrasts with cases in Figure 12, where PSNR rises, OLR decreases, and bit rate reductions of 66.4% and 36.3% are achieved. These results suggest that stricter error bounds leave `NeurLZ` with less room for optimization, raising the question: why is there a performance limitation with stricter error bounds?

We hypothesize that sample conflict introduces noise and further hinders `NeurLZ` performance at stricter error bounds. This occurs when input data x_1 and x_2 are very similar or identical ($x_1 \approx x_2$ or $x_1 = x_2$), but their desired outputs y_1 and y_2 differ significantly ($y_1 \neq y_2$). In such cases, the model struggles to learn effectively, as it must map nearly identical inputs to distinct outputs ($f(x_1) = y_1$, $f(x_2) = y_2$). Instead, the model naturally tends to map similar inputs to similar outputs ($f(x_1) \approx f(x_2)$), conflicting with the objective of $y_1 \neq y_2$. This inherent challenge makes it difficult to fit the training data properly.

In our case, x represents the decompressed slice, and y represents the target field’s residual slice. To illustrate, we analyze two fields from the Nyx dataset—Temperature and Velocity Y—under SZ3, focusing on single-field learning. This approach can naturally be extended to cross-field learning scenarios. These fields were selected due to their contrasting behaviors: Temperature shows notable improvements even under stricter error bounds, while Velocity Y lacks similar gains. Each decompressed slice, with a shape of 512 × 512, corresponds to an input x, while each residual slice, also with a shape of 512 × 512, corresponds to the output y. We can compute the similarity between different x slices and between different y slices using the absolute value of cosine similarity as the metric. If the similarity between two decompressed slices, x_1 and x_2, exceeds 0.95 while the similarity between their corresponding residual slices, y_1 and y_2, is less than 0.05, we consider (x_1, y_1) and (x_2, y_2) to be in a conflict. Otherwise, they are not considered as being in conflict under this criterion.

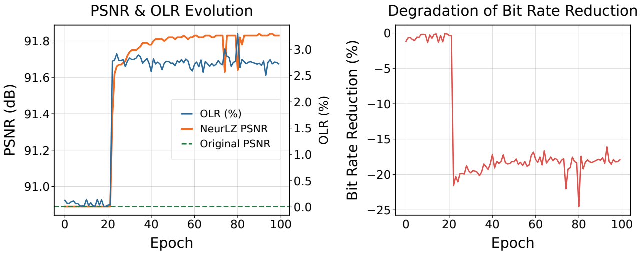


Figure 16: Training evolution of PSNR and outlier rate (OLR, %) for one Nyx field. The sharp PSNR rise, high OLR, and eventual degradation of bit rate reduction indicate challenges for `NeurLZ` in balancing PSNR improvement and maintaining a low OLR.

The upper part of Figure 17 shows the sample conflict adjacency matrix. Since there are 512 (x_i, y_i) pairs, the matrix has a shape of 512 × 512, representing a total of 262,144 pairwise comparisons for conflict calculation. The matrices depict the occurrence of conflicts between these pairs: white areas (value 0) indicate non-conflicting pairs, while dark blue areas (value 1) represent conflicting pairs. The subplot titles display the proportion of conflicts, quantifying the percentage of pairs exhibiting conflicts. In the left figure for Temperature, fewer conflicts are observed, with only 3.09% of the total pairs showing conflicts, compared to 15.3% for Velocity Y. This aligns with the fact that Velocity Y is more challenging to enhance. Interestingly, for both cases, most conflicts occur between neighboring slices, such as (x_i, y_i) and (x_{i+1}, y_{i+1}), which is visible as a dark blue region along the line $y = x$. This may be due to the decompressed values being smoother than the residual values, as the residuals tend to be more random [40]. As a result, similar x_i and x_{i+1} values but different y_i and y_{i+1} values lead to conflicts.

Conflicts among samples can cause conflicting gradient updates during training. Models were trained for 100 epochs, and gradients for all 3,000 parameters were extracted per sample, forming gradient vectors of size 3,000 × 1. Gradient conflict adjacency matrices were calculated based on these gradient vectors. The lower part of Figure 17 shows these matrices for Temperature and Velocity Y. Temperature has a low conflict rate of 2.38%, indicating similar gradient directions and consistent updates. In contrast, Velocity Y shows a high conflict rate of 41.6%, with samples disagreeing on update directions. These differences align with the training

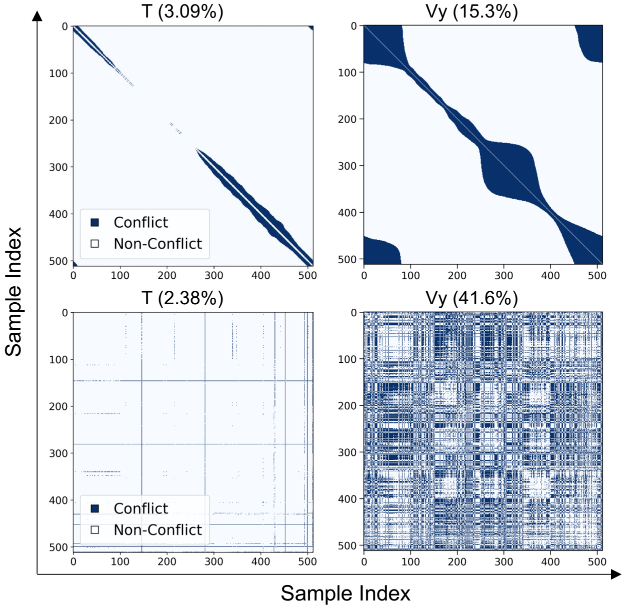


Figure 17: Sample (top) and gradient (bottom) conflict matrices for two Nyx fields, each with 512 samples. The matrices show conflicts across 262,144 pairs, where white (0) indicates no conflict, dark blue (1) indicates conflict, with titles showing conflict proportions.

dynamics shown in Figures 12 and 16. For Temperature, consistent gradients result in smooth and steady PSNR increases, and OLR decreases. For Velocity Y, conflicting gradients lead to unstable PSNR and OLR increases, severely hindering optimization. Overall, sample and gradient conflicts inherently limit model performance, causing reduced PSNR gains and higher OLR, leading to diminished or even negative overall performance improvements.

6 Conclusion

In this paper, we propose NeurLZ, an online neural learning-based method designed to enhance lossy compression quality in handling large-scale scientific data, fully leveraging the power of DNNs. NeurLZ dynamically trains multiple lightweight skipping DNN models during compression for specific data blocks, adapting efficiently to residual errors and evolving data characteristics. Evaluations on diverse datasets such as Nyx, Miranda, and Hurricane show NeurLZ's superior performance, achieving up to 94% reduction in bit rate while maintaining equivalent levels of data distortion and significantly improving upon existing state-of-the-art conventional compression methods.

References

[1] [n. d.]. Community Earth Simulation Model (CESM). https://www2.cesm.ucar.edu/.

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

[3] Amazon Web Services. n.d.. Amazon S3 Glacier. https://aws.amazon.com/glacier/. Accessed: 2025-01-07.

[4] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. FLIP: A Difference Evaluator for Alternating Images. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 2, Article 15 (Aug. 2020), 23 pages. https://doi.org/10.1145/3406183

[5] Saeed Anwar, Salman Khan, and Nick Barnes. 2020. A deep journey into super-resolution: A survey. *ACM computing surveys (CSUR)* 53, 3 (2020), 1–34.

[6] Anakha V Babu, Tao Zhou, Saugat Kandel, Tekin Bicer, Zhengchun Liu, William Judge, Daniel J Ching, Yi Jiang, Sinisa Veseli, Steven Henke, et al. 2023. Deep learning at the edge enables real-time streaming ptychographic imaging. *Nature Communications* 14, 1 (2023), 7059.

[7] Allison H. Baker, Haiying Xu, John M. Dennis, Michael N. Levy, Doug Nychka, Sheri A. Mickelson, Jim Edwards, Mariana Vertenstein, and Al Wegener. 2014. A methodology for evaluating the impact of data compression on climate simulation data. In *Proceedings of the 23rd International Symposium on High-Performance Parallel and Distributed Computing* (Vancouver, BC, Canada) *(HPDC '14)*. Association for Computing Machinery, New York, NY, USA, 203–214. https://doi.org/10.1145/2600212.2600217

[8] Shobana Balakrishnan, Richard Black, Austin Donnelly, Paul England, Adam Glass, Dave Harper, Sergey Legtchenko, Aaron Ogus, Eric Peterson, and Antony Rowstron. 2014. Pelican: A Building Block for Exascale Cold Data Storage. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*. USENIX Association, Broomfield, CO, 351–365. https://www.usenix.org/conference/osdi14/technical-sessions/presentation/balakrishnan

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

[10] Rafael Ballester-Ripoll, Susanne K Suter, and Renato Pajarola. 2015. Analysis of tensor approximation for compression-domain volume visualization. *Computers & Graphics* 47 (2015), 34–47.

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

[12] Nicolo Cesa-Bianchi and Gabor Lugosi. 2006. *Prediction, Learning, and Games.* Cambridge University Press, Cambridge.

[13] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. 2023. Activating More Pixels in Image Super-Resolution Transformer. arXiv:2205.04437 [eess.IV] https://arxiv.org/abs/2205.04437

[14] Yann Collet and EM Kucherawy. 2015. Zstandard-real-time data compression algorithm.

[15] Andrew Cook and William Cabot. 2024. Miranda Simulation Code. https://wci.llnl.gov/simulation/computer-codes/miranda. Accessed: [Date you accessed the site].

[16] P. Deutsch. 1996. RFC1952: GZIP file format specification version 4.3.

[17] Sheng Di and Franck Cappello. 2016. Fast Error-Bounded Lossy HPC Data Compression with SZ . In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society, Los Alamitos, CA, USA, 730–739. https://doi.org/10.1109/IPDPS.2016.11

[18] Sheng Di, Jinyang Liu, Kai Zhao, Xin Liang, Robert Underwood, Zhaorui Zhang, Milan Shah, Yafan Huang, Jiajun Huang, Xiaodong Yu, Congrong Ren, Hanqi Guo, Grant Wilkins, Dingwen Tao, Jiannan

Tian, Sian Jin, Zizhe Jian, Daoce Wang, MD Hasanur Rahman, Boyuan Zhang, Jon C. Calhoun, Guanpeng Li, Kazutomo Yoshii, Khalid Ayed Alharthi, and Franck Cappello. 2024. A Survey on Error-Bounded Lossy Compression for Scientific Datasets. arXiv:2404.02840 [cs.DC] https://arxiv.org/abs/2404.02840

[19] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. 2022. SUNet: Swin Transformer UNet for Image Denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, Austin, Texas, USA, 2333–2337. https://doi.org/10.1109/iscas48785.2022.9937486

[20] Jean-loup Gailly and Mark Adler. 2004. zlib Compression Library. http://www.dspace.cam.ac.uk/handle/1810/3486 Accessed: 2025-01-08.

[21] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. 2022. How to Train Vision Transformer on Small-scale Datasets? arXiv:2210.07240 [cs.CV] https://arxiv.org/abs/2210.07240

[22] Matthias Grawinkel, Lars Nagel, Markus Mäsker, Federico Padua, Andre Brinkmann, and Lennart Sorth. 2015. Analysis of the ECMWF Storage Landscape. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*. USENIX Association, Santa Clara, CA, 15–27. https://www.usenix.org/conference/fast15/technical-sessions/presentation/grawinkel

[23] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. 2022. Escaping the Big Data Paradigm with Compact Transformers. arXiv:2104.05704 [cs.CV] https://arxiv.org/abs/2104.05704

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV] https://arxiv.org/abs/1512.03385

[25] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online learning: A comprehensive survey. *Neurocomputing* 459 (2021), 249–289.

[26] Steven C.H. Hoi, Jialei Wang, and Peilin Zhao. 2014. LIBOL: A Library for Online Learning Algorithms. *Journal of Machine Learning Research* 15, 15 (2014), 495–499. http://jmlr.org/papers/v15/hoi14a.html

[27] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[28] Xiaodan Hu, Mohamed A. Naiel, Alexander Wong, Mark Lamm, and Paul Fieguth. 2019. RUNet: A Robust UNet Architecture for Image Super-Resolution . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, Los Alamitos, CA, USA, 505–507. https://doi.org/10.1109/CVPRW.2019.00073

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

[30] Wenqi Jia, Sian Jin, Jinzhen Wang, Wei Niu, Dingwen Tao, and Miao Yin. 2024. GWLZ: A Group-wise Learning-based Lossy Compression Framework for Scientific Data. In *Proceedings of the 14th Workshop on AI and Scientific Computing at Scale Using Flexible Computing Infrastructures* (Pisa, Italy) *(FlexScience'24)*. Association for Computing Machinery, New York, NY, USA, 34–41. https://doi.org/10.1145/3659995.3660041

[31] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896 [cs.LG]

[32] Filippos Kokkinos and Stamatios Lefkimmiatis. 2018. Deep Image Demosaicking Using a Cascade of Convolutional Residual Denoising Networks. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 317–333.

[33] Yu Kong and Yun Fu. 2022. Human action recognition and prediction: A survey. *International Journal of Computer Vision* 130, 5 (2022), 1366–1401.

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

[35] Shaomeng Li, Peter Lindstrom, and John Clyne. 2023. Lossy scientific data compression with SPERR. In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 1007–1017.

[36] Xiao Li, Jaemoon Lee, Anand Rangarajan, and Sanjay Ranka. 2024. Attention Based Machine Learning Methods for Data Reduction with Guaranteed Error Bounds. arXiv:2409.05357 [cs.LG] https://arxiv.org/abs/2409.05357

[37] Xin Liang, Sheng Di, Sihuan Li, Dingwen Tao, Bogdan Nicolae, Zizhong Chen, and Franck Cappello. 2019. Significantly improving lossy compression quality based on an optimized hybrid prediction model. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '19)*. Association for Computing Machinery, New York, NY, USA, Article 33, 26 pages. https://doi.org/10.1145/3295500.3356193

[38] Xin Liang, Sheng Di, Dingwen Tao, Sihuan Li, Shaomeng Li, Hanqi Guo, Zizhong Chen, and Franck Cappello. 2018. Error-Controlled Lossy Compression Optimized for High Compression Ratios of Scientific Datasets . In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE Computer Society, Los Alamitos, CA, USA, 438–447. https://doi.org/10.1109/BigData.2018.8622520

[39] Xin Liang, Ben Whitney, Jieyang Chen, Lipeng Wan, Qing Liu, Dingwen Tao, James Kress, David Pugmire, Matthew Wolf, Norbert Podhorszki, et al. 2021. Mgard+: Optimizing multilevel methods for error-bounded scientific data reduction. *IEEE Trans. Comput.* 71, 7 (2021), 1522–1536.

[40] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M Gok, Jiannan Tian, Junjing Deng, Jon C Calhoun, Dingwen Tao, et al. 2022. Sz3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data* 9, 2 (2022), 485–498.

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

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

[43] Peter G Lindstrom. 2017. *Fpzip*. Technical Report. Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States).

[44] Jinyang Liu, Sheng Di, Sian Jin, Kai Zhao, Xin Liang, Zizhong Chen, and Franck Cappello. 2023. Scientific Error-bounded Lossy Compression with Super-resolution Neural Networks. In *Proceedings - 2023 IEEE International Conference on Big Data, BigData 2023 (Proceedings - 2023 IEEE International Conference on Big Data, BigData 2023)*, Jingrui He, Themis Palpanas, Xiaohua Hu, Alfredo Cuzzocrea, Dejing Dou, Dominik Slezak, Wei Wang, Aleksandra Gruca, Jerry Chun-Wei Lin, and Rakesh Agrawal (Eds.). Institute of Electrical and Electronics Engineers Inc., United States, 229–236. https://doi.org/10.1109/BigData59044.2023.10386682 Publisher Copyright: © 2023 IEEE.; 2023 IEEE International Conference on Big Data, BigData 2023 ; Conference date: 15-12-2023 Through 18-12-2023.

[45] Jinyang Liu, Sheng Di, Kai Zhao, Sian Jin, Dingwen Tao, Xin Liang, Zizhong Chen, and Franck Cappello. 2021. Exploring Autoencoder-based Error-bounded Compression for Scientific Data. In *Proceedings - 2021 IEEE International Conference on Cluster Computing, Cluster*

2021 (Proceedings - IEEE International Conference on Cluster Computing, ICCC). Institute of Electrical and Electronics Engineers Inc., United States, 294–306. https://doi.org/10.1109/Cluster48925.2021.00034 Publisher Copyright: ©2021 IEEE.; 2021 IEEE International Conference on Cluster Computing, Cluster 2021 ; Conference date: 07-09-2021 Through 10-09-2021.

[46] Jinyang Liu, Sheng Di, Kai Zhao, Xin Liang, Zizhong Chen, and Franck Cappello. 2022. Dynamic Quality Metric Oriented Error Bounded Lossy Compression for Scientific Datasets . In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society, Los Alamitos, CA, USA, 1–15. https://doi.org/10.1109/SC41404.2022.00067

[47] Jinyang Liu, Sheng Di, Kai Zhao, Xin Liang, Zizhong Chen, and Franck Cappello. 2023. Faz: A flexible auto-tuned modular error-bounded compression framework for scientific data. In *Proceedings of the 37th International Conference on Supercomputing*. 1–13.

[48] Jinyang Liu, Sheng Di, Kai Zhao, Xin Liang, Sian Jin, Zizhe Jian, Jiajun Huang, Shixun Wu, Zizhong Chen, and Franck Cappello. 2024. High-performance Effective Scientific Error-bounded Lossy Compression with Auto-tuned Multi-component Interpolation. *Proc. ACM Manag. Data* 2, 1, Article 4 (March 2024), 27 pages. https://doi.org/10.1145/3639259

[49] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2020. Deep learning for generic object detection: A survey. *International journal of computer vision* 128 (2020), 261–318.

[50] Youyuan Liu, Wenqi Jia, Taolue Yang, Miao Yin, and Sian Jin. 2024. Enhancing Lossy Compression Through Cross-Field Information for Scientific Applications. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 300–308. https://doi.org/10.1109/SCW63240.2024.00046

[51] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. 2021. Efficient Training of Visual Transformers with Small Datasets. arXiv:2106.03746 [cs.CV] https://arxiv.org/abs/2106.03746

[52] Zarija Lukić, Casey W Stark, Peter Nugent, Martin White, Avery A Meiksin, and Ann Almgren. 2015. The Lyman α forest in optically thin hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society* 446, 4 (2015), 3697–3724.

[53] Julia Lust and Alexandru Condurache. 2020. A Survey on Assessing the Generalization Envelope of Deep Neural Networks at Inference Time for Image Classification. *ArXiv* abs/2008.09381 (2020). https://api.semanticscholar.org/CorpusID:221246352

[54] Yu Mao, Yufei Cui, Tei-Wei Kuo, and Chun Jason Xue. 2022. TRACE: A Fast Transformer-based General-Purpose Lossless Compressor. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 1829–1838. https://doi.org/10.1145/3485447.3511987

[55] Bunjamin Memishi, Raja Appuswamy, and Marcus Paradies. 2019. Cold Storage Data Archives: More Than Just a Bunch of Tapes. In *Proceedings of the 15th International Workshop on Data Management on New Hardware* (Amsterdam, Netherlands) *(DaMoN'19)*. Association for Computing Machinery, New York, NY, USA, Article 1, 7 pages. https://doi.org/10.1145/3329785.3329921

[56] Microsoft Azure. n.d.. Azure Archive Storage. https://azure.microsoft.com/en-us/services/storage/archive/. Accessed: 2025-01-07.

[57] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45, 1 (2012), 521–530.

[58] NERSC High-Dimension Nyx Dataset. [n. d.]. https://portal.nersc.gov/cfs/nyx/highz/. Online.

[59] Jose Oñorbe, FB Davies, Z Lukić, JF Hennawi, and D Sorini. 2019. Inhomogeneous reionization models in cosmological hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society* 486, 3 (2019), 4075–4097.

[60] Cyril Pernet, Claus Svarer, Ross Blair, John D Van Horn, and Russell A Poldrack. 2023. On the long-term archiving of research data. *Neuroinformatics* 21, 2 (2023), 243–246.

[61] Linus Pithan, Vladimir Starostin, David Mareček, Lukas Petersdorf, Constantin Völter, Valentin Munteanu, Maciej Jankowski, Oleg Konovalov, Alexander Gerlach, Alexander Hinderhofer, Bridget Murphy, Stefan Kowarik, and Frank Schreiber. 2023. Closing the loop: Autonomous experiments enabled by machine-learning-based online data analysis in synchrotron beamline environments. arXiv:2306.11899 [physics.data-an] https://arxiv.org/abs/2306.11899

[62] Waseem Rawat and Zenghui Wang. 2017. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* 29, 9 (2017), 2352–2449.

[63] Chris Rohlfs. 2025. Generalization in neural networks: A broad survey. *Neurocomputing* 611 (2025), 128701. https://doi.org/10.1016/j.neucom.2024.128701

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241.

[65] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. 2019. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications* (2019). https://api.semanticscholar.org/CorpusID:104425037

[66] Shai Shalev-Shwartz et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* 4, 2 (2012), 107–194.

[67] Ran Shao and Xiao-Jun Bi. 2022. Transformers Meet Small Datasets. *IEEE Access* 10 (2022), 118454–118464. https://doi.org/10.1109/ACCESS.2022.3221138

[68] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[69] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. 2017. MemNet: A Persistent Memory Network for Image Restoration . In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 4549–4557. https://doi.org/10.1109/ICCV.2017.486

[70] Dingwen Tao, Sheng Di, Zizhong Chen, and Franck Cappello. 2017. Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization . In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society, Los Alamitos, CA, USA, 1129–1139. https://doi.org/10.1109/IPDPS.2017.115

[71] Dingwen Tao, Sheng Di, Xin Liang, Zizhong Chen, and Franck Cappello. 2019. Optimizing lossy compression rate-distortion from automatic online selection between SZ and ZFP. *IEEE Transactions on Parallel and Distributed Systems* 30, 8 (2019), 1857–1871.

[72] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. 2023. UVCGAN: UNet Vision Transformer cycle-consistent GAN for unpaired image-to-image translation . In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE Computer Society, Los Alamitos, CA, USA, 702–712. https://doi.org/10.1109/WACV56688.2023.00077

[73] Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A. Fox. 2021. Natural Language Processing Advancements By Deep Learning: A Survey. arXiv:2003.01200 [cs.CL] https://arxiv.

org/abs/2003.01200

[74] Zhihao Wang, Jian Chen, and Steven CH Hoi. 2020. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3365–3387.

[75] Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021. Optimizing Deeper Transformers on Small Datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 2089–2102. https://doi.org/10.18653/v1/2021.acl-long.163

[76] Taolue Yang, Youyuan Liu, Bo Jiang, and Sian Jin. 2024. GPUFASTQLZ: An Ultra Fast Compression Methodology for Fastq Sequence Data on GPUs. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 317–325.

[77] Taolue Yang, Youyuan Liu, Chong Li, Xinhua Mindy Shi, and Sian Jin. 2024. SeqBench: A Benchmark Suite for Lossless and Lossy Compression of Sequence Data *(BCB '24)*. Association for Computing Machinery, New York, NY, USA, Article 58, 7 pages. https://doi.org/10.1145/3698587.3701386

[78] Taketoshi Yoshida, Ken Iizawa, and Masahisa Tamura. 2020. Mass Data Archiving Acceleration Technology for Magnetic Tape Storage. *Fujitsu Technical Review* 5 (2020).

[79] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. 2018. Semantic Image Inpainting with Progressive Generative Networks. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM '18)*. Association for Computing Machinery, New York, NY, USA, 1939–1947. https://doi.org/10.1145/3240508.3240625

[80] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing* 26, 7 (2017), 3142–3155.

[81] Kai Zhao, Sheng Di, Maxim Dmitriev, Thierry-Laurent D. Tonellot, Zizhong Chen, and Franck Cappello. 2021. Optimizing Error-Bounded Lossy Compression for Scientific Data by Dynamic Spline Interpolation . In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 1643–1654. https://doi.org/10.1109/ICDE51399.2021.00145

[82] Kai Zhao, Sheng Di, Xin Lian, Sihuan Li, Dingwen Tao, Julie Bessac, Zizhong Chen, and Franck Cappello. 2020. SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors . In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE Computer Society, Los Alamitos, CA, USA, 2716–2724. https://doi.org/10.1109/BigData50022.2020.9378449

[83] Kai Zhao, Sheng Di, Xin Liang, Sihuan Li, Dingwen Tao, Zizhong Chen, and Franck Cappello. 2020. 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* (Stockholm, Sweden) *(HPDC '20)*. Association for Computing Machinery, New York, NY, USA, 89–100. https://doi.org/10.1145/3369583.3392688

[84] Jacob Ziv and Abraham Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on information theory* 23, 3 (1977), 337–343.