Combining Spatial and Temporal Properties for Improvements in Data Reduction

Megan Hickman Fulp*, Ayan Biswas†, Jon C. Calhoun*
mlhickm@clemson.edu, ayan@lanl.gov, jonccal@clemson.edu
*Holcombe Department of Electrical and Computing Engineering - Clemson University, Clemson, SC 29634
†Los Alamos National Laboratory, Los Alamos, NM

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378457

***Abstract*—Due to I/O bandwidth limitations, intelligent in situ data reduction methods are needed to enable post-hoc workflows. Current state-of-the-art sampling methods save data points if they deem them spatially or temporally important. By analyzing the properties of the data values at each time-step, two consecutive steps may be very similar. This research follows the notion that if neighboring time-steps are very similar, samples from both are unnecessary, which leaves storage for adding more useful samples. Here, we present an investigation of the combination of spatial and temporal sampling to drastically reduce data size without the loss of valuable information. We demonstrate that, by reusing samples, our reconstructed data set reduces the overall data size while achieving a higher post-reconstruction quality over other reduction methods.**

***Index Terms*—Data Reduction, Data Sampling, Importance Sampling, Feature Preservation**

I. Introduction

Modern high-performance computers have increasingly high computation capabilities, being able to simulate previously intractable problems. These simulations produce petabytes worth of data [1], [2], which, due to I/O limitations, is generated faster than the system can store. The combination of massive output data and I/O bottleneck makes traditional full post-hoc analysis and visualization increasingly less viable [3]–[6].

Many researchers have aimed to solve this issue by reducing the overall data size. Lossy compression is one approach capable of achieving high compression ratios by introducing controlled error in the compressed data [7]–[9]. Data sampling is another prevalent approach to data reduction with existing efforts using simple uniform random selection techniques to determine which samples to keep [10]–[12].

Due to the limited memory present in *in situ* processes, high compression ratios are crucial to working on large data sets. Preserving regions of interest (ROI) within these data sets with high quality is critical to meet current domain scientists' demands. Both lossy compression algorithms and existing data sampling efforts reduce overall data size uniformly, consequently reducing quality within the ROI. However, biased sampling based on importance to the user and better preserves the ROI in post-reconstruction visualizations without needing prior knowledge of the data set [4], [13], [14].

In this paper, we combine concepts from existing data reduction techniques that sample data using spatial properties [13] or temporal properties [15] to improve post-reconstruction quality in both the ROI and overall data set.

Our specific contributions are as follows:

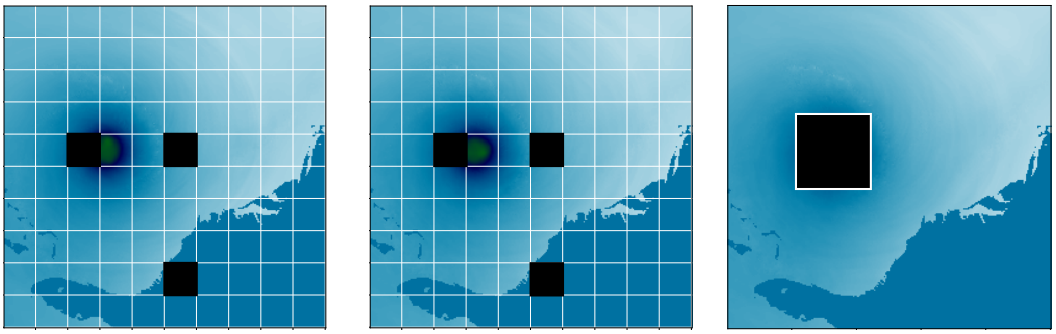
- We propose a data sampling technique that uses both spatial and temporal data properties to improve post-reconstruction data quality.
- We detail our workflow of selecting previous samples based on value histograms or error tolerance, and how we utilize them in the form of reuse or additions.
- We provide a detailed analysis of our novel approach when applied to various time-series data sets while also comparing the differences in bandwidth and quality of our approach, a state-of-the-art sampling method, and the ZFP lossy compressor. We find our approach achieves higher qualities than other data reduction schemes at low sampling rates while preserving the region of interest.

The rest of this paper is organized as follows. In Section II, we discuss related works. In Section III, we describe a few existing sampling methods that combine to form our sampling approach, which we discuss in detail in Section IV. Then we run experiments and provide a detailed evaluation in Section V. In Section VI, we discuss the different situations in which our method configurations would be applicable. We also give a comparison to compression in Section VII. Finally, in Section VIII, write our concluding statements.

II. Related Works

Basic data sampling uses simple uniform random selection techniques [10]–[12]. While these techniques accurately reflect the original data distribution, they do not consider a data point's value or importance to the user. Biasing samples based on importance enhances the visualization process by ensuring the preservation of ROIs during reconstruction [4], [13], [14]. Nouanesengsy et al. developed a basic adaptive sampling approach, using a user-defined importance function to determine the ROI [4]. A statistical data sampling method based on entropy maximization is proposed in [14] to utilize a histogram of data values to set importance factors.

Complex simulation models also often have a temporal aspect as the data changes over time. Due to the size of time-series data produced by supercomputers, offline visualization and examination of multiple time-steps can become overwhelming for users. For large-scale time-varying data sets, key time-step selection aids in reducing the number of



(a) Time-Step 11 (b) Time-Step 12 (c) Time-Step 12

Fig. 1: Hurricane Isabel Pressure Visualizations. Figures a) and b) show the data set divided into regions of dimension $25 \times 25 \times 25$. Figure c) highlights our definition of Region of Interest for this data set.

frames a user has to rebuild, visualize, and interact with. Akiba et al. developed a method that classifies the time-variant features within time-series data sets to help choose the most representative frames [16]. Zhou and Chiang developed a time-step selection method using information theory to extract the most optimal time-steps within a data series [15]. Other works develop techniques where users manually select areas of interest to make connections and visualize smaller portions of the time series by employing a time-warp function to enhance the user's ability to understand the data series [17] or by visualizing the hierarchical state transition relationships [18].

III. Background

*1) **Simple Random Sampling**:* Simple Random Sampling gives each data point an equal opportunity to be sampled. Per point, a random number ξ is generated before being compared with a user-specified sampling percentage α, where $\xi, \alpha \in [0, 1]$. If $\xi < \alpha$, the point is included in the data sample.

*2) **Importance-Based Sampling**:* Importance-Based Sampling assumes that rare data values are more important to the user and biases these values when choosing samples. The method provided by Biswas et al. gives an importance factor to each data point such that more frequent values are assigned a lower priority, while unlikely values are considered more valuable to the user [13], [14]. The importance factor is generated using the histogram distribution of values with the resulting sample set over-representing rare values without ignoring common values. Upon deciding all importance factors, a random number ξ is generated for each data point. If ξ is less than the importance factor of the data value, the point is stored, with this process repeating until reaching the sample size. The data set is sampled down to user specifications through this process while retaining high quality in important areas. We use this sampling process as the control method to compare and evaluate our method.

*3) **Time-Step Selection**:* The time-step selection process analyzes the differences between sequential time-steps to determine which steps provide a representative overview of the entire data series. For example, assuming the previous time-step (t_{k-1}) has previously been selected, we need to decide whether to select the current time-step (t_k) as well. Upon comparing the two, if t_k is similar enough to t_{k-1}, we do not need to select it as t_{k-1} is a sufficient representation.

IV. Hybrid Sampling Method

The related works we previously described explicitly study how to dynamically sample the most representative subsets of data points or the best overall time-steps for post-hoc visualization. In our work, we combine the concepts of both spatial and temporal techniques to enable improvements in data reduction.

Within many HPC simulations, large data areas transform slowly over time with only specific region of interest (ROI) changing quickly, such that there is a visual difference between two sequential time-steps. Studying Figures 1a and 1b, we find that region A changes rapidly between time-steps, region B changes slightly, and region C stays consistent. As the data in region C did not alter visually, we use samples from this region of time-step 11 when gathering samples for time-step 12. Therefore, by leveraging the data regions that remain relatively consistent, we utilize previous samples of these regions in future time-steps to save time and storage.

Our approach leverages both spatial and temporal data aspects to produce a sampling method that improves post-reconstruction data quality. To accomplish this, we first need to quantify the similarities between two corresponding regions of neighboring time-steps. When determining whether a region is similar, we first check the distribution of data values within the corresponding regions. As histograms are lightweight and add little overhead, they are a valuable way to compare two data regions. If the two distributions are similar enough, we utilize the previous samples. However, even though this approach is fast and lightweight, histograms lack spatial awareness, often resulting in lower quality than achievable with other similarity metrics.

Therefore, we introduce the concept of RMSE to quantify similarity. Using this method allows the user to set an error tolerance such that only regions with a smaller error between the current and previous time-steps are reused. While this

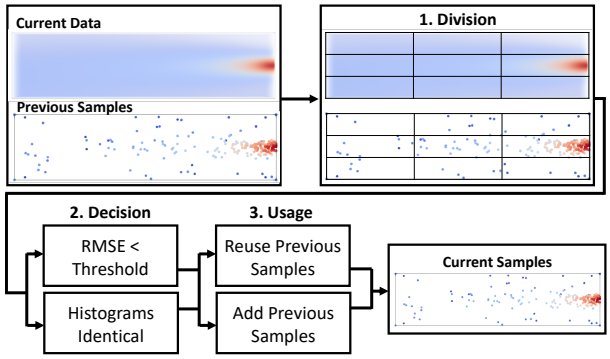


Fig. 2: A schematic workflow of Hybrid Approach.

method produces higher qualities, it does take longer to sample and requires more storage space.

Figure 2 shows an overview of our process, which consists of three steps: Division, Decision, and Usage. When sampling, we first divide each time-step of the data set into equally sized regions and examine each region's temporal aspects using an approach similar to the one seen in Section III-3.

We compare each region of t_k with the same region of t_{k-1} either by inspecting the value histograms or quantifying the error between data values to decide how similar they are. Suppose we determine the corresponding regions are too different. In that case, samples from t_k are necessary to represent the different data, and in this case, we use the Importance-Based sampling process from Section III-2 to gather samples from t_k. However, if the two regions are similar enough and samples from t_k are not necessary, we either choose to reuse the samples from t_{k-1} for t_k or use them in addition to new samples from t_k. Overall, our approach includes two internal methods for comparing regions over time-steps (2. Decision) and two for determining what to do with those regions (3. Usage). To enable a more tailored approach, the user has control over all options.

A. Division

From Figure 1, we see there are many regions within sequential time-steps where the data has changed little. Our sampling method utilizes samples from t_{k-1} in regions where the data is near identical to the same region of t_k. To select which regions are similar over time, we first define each region's specific boundaries by dividing each time-step into blocked regions, as we show in Step 1 of Figure 2. The user defines the region's size based on their data set's properties such that it is optimal, as discussed in Section V-C. As each data set is unique, setting an appropriate region size is crucial for optimal performance as a region size too small or too large affects overall efficiency.

B. Decision

With the data divided into regions, we compare each region of t_k to the corresponding region of t_{k-1} to decide when to use previous samples.

1) ***Histogram Intersection****:* A data value histogram is a low-storage approach that shows the distribution of data values within a particular region. By quantifying the distance between corresponding histograms of the data values within a region in t_k and t_{k-1}, using Histogram Intersection (see Eq. 1), we find if they are similar enough to utilize previous samples. Here, p_i is the number of elements in the ith bin of the value histogram of t_{k-1} and q_i is the number of elements in the corresponding ith bin of the histogram of t_k, and n is the total number of bins. We normalize the intersection value by dividing by q_i, such that the results are always between 0 (no intersection) and 1 (identical distributions).

$$\frac{\sum_{i=1}^{n} min(p_i, q_i)}{\sum_{j=1}^{n}(q_i)} \tag{1}$$

In the following experiments, we use histogram intersection to quantify the difference between value histograms; however, other probability distribution comparison methods were implemented, including KL Divergence, Bhattacharyya Distance, and Chi-Squared, and all show near-identical results.

Figure 3a shows the histogram intersection that is calculated between time-steps 11 and 12 of the Hurricane Isabel Pressure data set. In the regions where the value histograms fully intersect (i.e. 1.00), the data value distribution of that region has not changed; thus, we use the samples from t_{k-1}.

By utilizing histograms, we face the question of what number of bins to use, as more bins create a more specific distribution of the data, yielding higher dissimilarity probability. In our experiments, we use 16 bins for all experiments, as further discussed in Section V-B.

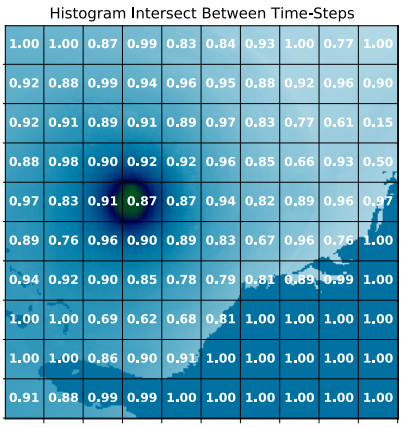
2) ***Error Based****:* The disadvantage of the histogram similarity method is the loss of spatial information. Just comparing the value distributions leads to the potential introduction of error when reusing incorrectly placed samples. If too much error is added, the reconstructed data set's quality may be lower than anticipated.

To resolve this, we look at the error between data points. This method considers the data values and locations of previous samples for a specific region and compares them to the current value at that location. If the root mean squared error (RMSE) (see Eq. 2) for all previous samples within a region is greater than the user-specified error, we do not reuse threshold samples as they introduce more error than wanted. Here, p_i is the ith element in the region in t_{k-1}, q_i is the corresponding element in t_k, and n is the number of elements within the region. Figure 3b shows the RMSE between corresponding regions of time-steps 11 and 12.

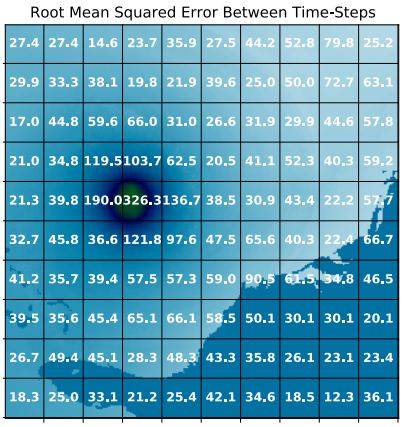
$$\sqrt{\frac{\sum_{i=1}^{n}(p_i - q_i)^2}{n}} \tag{2}$$

C. Usage

We utilize samples from a previous region if either the region's value histograms are identical or if the error introduced is low, but we need to determine what to do with these previous



(a) Histogram Intersect



(b) RMSE

Fig. 3: Intersect and RMSE of time-steps 11 and 12 for Hurricane Isabel Pressure data set.

samples: reuse them instead of taking current samples or use them in addition.

*1) **Reuse Previous Samples Instead**:* The reuse methods use regions of previous samples instead of gathering new samples. Originally, we collected sample values and their locations for all regions in each time-step. Using the reuse methods, if a region in t_k is similar to the same region in t_{k-1}, we record a flag symbol and reuse samples for this region from t_k. By only storing a flag value instead of unnecessary samples, the resulting raw files of information are smaller than the methods that do not reuse samples. From the information gathered in Figure 4, we find that we save 25% - 31% of the original storage amount by reusing samples for some regions instead of using more space on taking new samples as we vary sample rate.

Paraview [19] is a common tool for visualizing large data sets; among its Visualization ToolKit file formats is the Visualization Toolkit Polygonal Data (VTP) file, that aids in visualizing unstructured polygonal data sets of combinations of vertices. Upon time for the user to view the sampled data, we create a VTP file by locating all of the samples in flagged regions and are conglomerated. The VTP file containing all

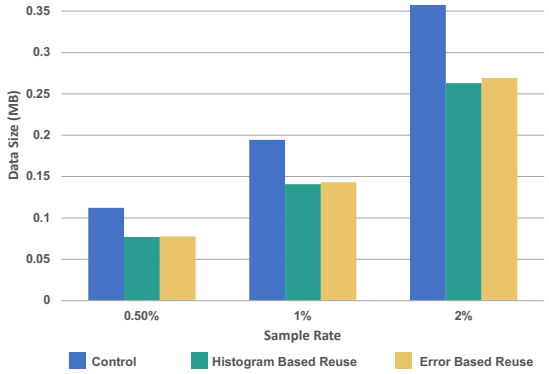


Fig. 4: The amount of storage each method needs to save samples, using the Hurricane Isabel Pressure data set.

samples for this time-step is only dependent on the sample files of the previous time-step, as regions from t_{k-1} are only considered for reuse if it did not reuse samples from t_{k-2}. This method alleviates any domino-like dependencies, thus to visualize and analyze the data for t_k, we only need access to samples from the two time-steps t_k and t_{k-1}. Once we conglomerate current samples with previous samples in flagged regions, we use this VTP file to reconstruct the data set.

These reuse methods take less storage by sacrificing post-reconstruction quality. In some scenarios, it is beneficial to accept the slight drop in quality from reusing samples in trade for less storage space; however, we design our methods behind the concept that the user has specified a storage constraint in an *in situ* situation. Since the user specifies a budget for storage, it reasons that they want to utilize all of that space in order to yield higher post-reconstruction quality. To fill this unused storage budget, we add a layer of simple random samples on top of the samples gathered thus far in regions that are not reusing previous samples.

Each method now generates enough new samples to fill the storage constraint, without going over. The reuse methods have access to more samples overall than the control method. This means that the reuse methods yield higher qualities on average than methods that do not reuse samples, because more samples generally means higher quality, as we have more true values of the data, and fewer data points need to be reconstructed.

*2) **Use Previous Samples in Addition**:* The reuse methods use simple random sampling to gather more samples, while the addition-based methods choose samples solely using advanced methods. Here, we sample according to the data set's value histogram as described in Section III-2 for every time-step, then we use samples from t_{k-1} in addition to what we sample for t_k. We go through the same process of determining which regions are similar enough to utilize the samples from their previous time-step and flag those regions. Upon construction of the VTP file, these previous samples are gathered together.

D. Method Combinations

Below, we specify the four combinations of our method, based on when and how to utilize previous samples as "HR," "ER," "HA," and "EA." We also define the baseline for evaluating our methods as "control."

1) ***"HR"*** *- Histogram Based Reuse:* Reuses samples from t_{k-1} instead of taking samples from t_k, if the value histograms between corresponding regions are identical.

2) ***"ER"*** *- Error Based Reuse:* Reuses samples from t_{k-1} instead of taking samples from t_k, if the RMSE between corresponding regions is less than a user specified error threshold.

3) ***"HA"*** *- Histogram Based Additions:* Appends samples from t_{k-1} to the samples gathered for the same region in t_k, if the value histograms of the corresponding regions are identical.

4) ***"EA"*** *- Error Based Additions:* Appends samples from t_{k-1} to the samples gathered for the same region in t_k, if the RMSE is tolerable.

5) ***"Control"*** *- Spatial, Time Independent Method:* We use the value-based importance method implemented by Biswas et. al [13] as the base comparison for our methods, as it is the state-of-the-art algorithm that we found our methods upon.

V. Experiments

All experiments are run on Clemson University's Palmetto cluster using phase 8c nodes which have a 16 core Intel Xeon E5-2665 CPU and 64GB of DDR3 RAM.

A. Data Sets

1) ***Hurricane Isabel****:* The Hurricane Isabel Data models the 2003 hurricane in the west Atlantic region [20]. This data was produced by the Weather Research and Forecast model, courtesy of NCAR, and the U.S. National Science Foundation. In the following experiments, we use the pressure variable, as it provides a distinct representation of the eye of the hurricane, the ROI of this data set (Fig. 1c). We use 48 time-steps with a down-sampled spatial resolution of $250 \times 250 \times 50$.

2) ***Exascale Additive Manufacturing Project****:* (ExaAM) uses exascale simulations to design Additive Manufacturing components [21], [22]. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S.Department of Energy Office of Science and the National Nuclear Security Administration. For the following experiments, we use 108 time-steps with its full spatial resolution of $20 \times 200 \times 50$. We experiment with this data set primarily to show the difference in results when using a smaller data set that has more time-steps. Figure 5 shows time-step 64 of this data set, with highlighted ROI.

B. Determining Number of Bins

To better understand the effects number of bins has on the amount of data intersection between two corresponding regions over two time-steps and to determine the optimal number of bins to use, we evaluate the effects different numbers of bins have on our results. This evaluation uses the three regions specified in Figures 1a and 1b from the Hurricane Isabel Pressure data set and plots the histogram intersection between time-steps 11 and 12 for each region.

Fig. 5: ExaAM time-step 64 with highlighted ROI.

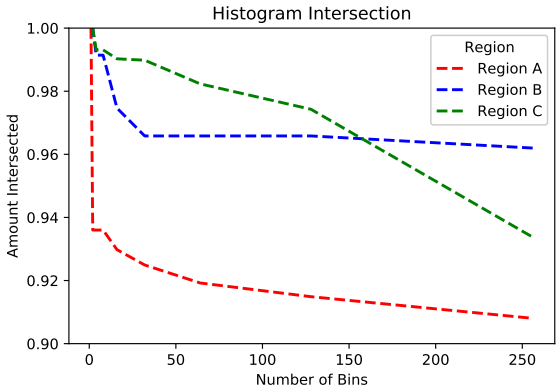
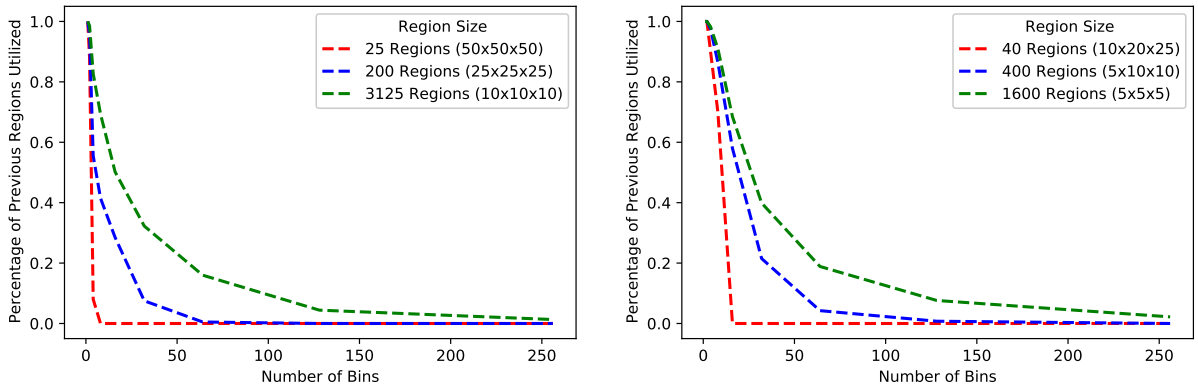


Fig. 6: Amount of distribution intersection with varying number of histogram bins at three regions with varying entropy within the Hurricane Isabel Pressure data set, seen in Figure 1.

From Figure 6, we find fewer histogram intersections between the two histograms as we increase the number of bins used to construct them. Using more bins, we parse values out to more specific bins, reducing the areas where both histograms can overlap. This trend is especially true within regions of high entropy, like region A. We find this concerning as, without enough intersections, our method is left unoptimized and will rarely utilize previous samples. Conversely, using a lower number of bins enables our method to group more items. Still, too few bins lead to excess intersections, resulting in high levels of error in the data. We confirm these results by running similar experiments using the ExaAM data in which we found similar results.

In order to better understand the correlations between the number of bins we use and the region size we use, we evaluate different combinations of each. From Figures 7a and 7b, we likewise find varying the number of bins has similar effects, independent of region size. Specifically, we find using a higher number of bins leads to much lower levels of reuse for all region sizes we test. Therefore, based on these results, we use 16 bins for all our experiments as we find this number of bins enables an adequate amount of sample reuse while also retaining high levels of quality in the data.

When determining whether to utilize previous samples from a particular region, we only do so when both histograms are identical, enabling us to maintain high data quality. From Figure 6, we find the number of bins directly affects what



(a) Hurricane Isabel.

(b) ExaAM.

Fig. 7: Percentage of previous regions utilized, varying number of bins and region sizes.

percentage of regions are reused, with more bins leading to less reuse. This is due to the higher number of bins leading to more specific bins, making identical histograms less likely.

C. Determining Region Size

As the number of data regions affects sampling results, we evaluate the effects of multiple different region sizes and quantify their impact. Specifically, in Figure 8, we assess region sizes that split the time-steps 11 and 12 of the Hurricane Isabel Pressure data set into regions ranging from 0 to 25,000 regions. In this assessment, we compare the percentage of regions utilized from time-step 11 when gathering samples for time-step 12, using 16 bins in our histograms. We repeat this process in Figure 8 where we assess region sizes that split the time-steps 64 and 65 of the ExaAM data set into regions ranging from 0 to 25,000 regions.

Analyzing both these figures, we find dividing the data into more regions enables our method to reuse more samples than when dividing the data into fewer regions. With more regions, our method utilizes more data from the previous time-step, as we further separate regions of high entropy from those of low entropy. However, adding additional regions also increases the number of similarity comparison computations, which drastically slows down the algorithm, as shown in Figures 9a and 9b. From these two figures, we find increasing the number of regions the data is split into generally leads to reduced bandwidth. The process slows down with more regions because we calculate the similarity between every corresponding region between two neighboring time-steps; thus the more regions we have, the more calculations that have to be made. Therefore, when determining the number of regions to divide the data set into, we choose a middle-ground number to have a more general amount of previous time-step utilization.

Based on our findings, we divide the Hurricane Isabel data set into 200 (sized $25 \times 25 \times 25$) regions and ExaAM into 400 (sized $5 \times 10 \times 10$) regions for the following experiments.

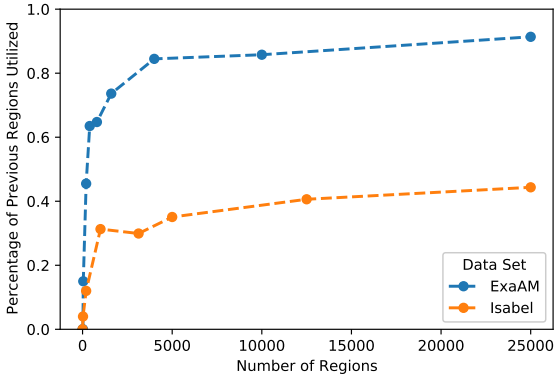
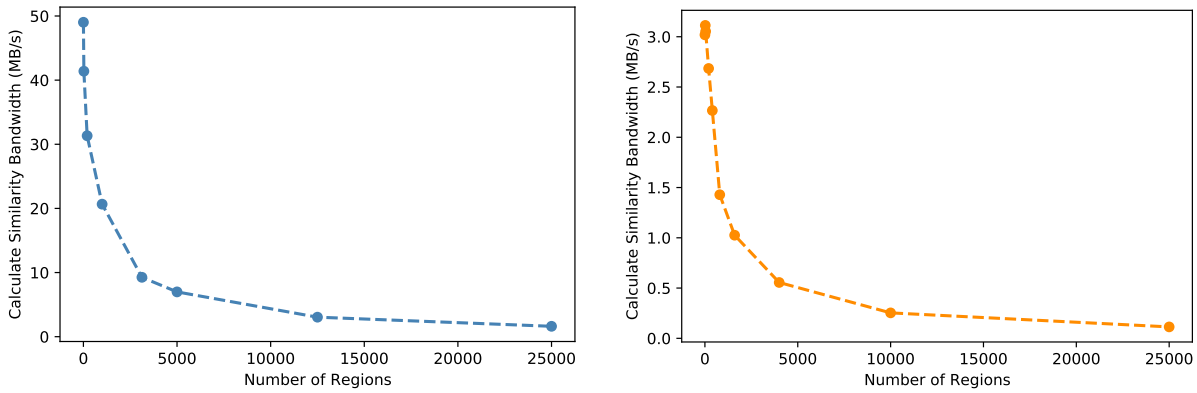


Fig. 8: Percentage of previous regions utilized, varying number of regions with 16 bins for histograms.

D. Sampling Bandwidth

As reducing the overhead incurred while gathering samples is critical to optimizing our approach, we analyze the sampling overhead independent of I/O and reconstruction overhead. Figure 11 shows the average bandwidth of each sampling method over the 48 time-steps of the Hurricane Isabel Pressure data set. The control method performs with the highest bandwidth, as all of the other methods are based upon it, but spend extra time checking every region to determine to use previous samples or not. While the control and HR methods maintain a reasonably consistent bandwidth as the sample rate increases, the ER linearly decreases because the higher the sampling rate, the more samples that are kept for each time-step. Since we keep more samples, more work is needed to calculate RMSE, which linearly increases the amount of time it takes to calculate which previous regions to use and affects the overall sampling bandwidth. HA and EA both decrease as sample rate increases because we have to check the location of samples taken for t_{k-1} to make sure they do not overlap any of the samples taken for t_k. If there is any overlap, we remove those



(a) Hurricane Isabel.

(b) ExaAM.

Fig. 9: Bandwidth (MB/s) of calculating which previous regions to utilize.

samples from t_k and pick new samples for that time-step. When evaluating the ExaAM data set, we find comparable results.

E. Samples Gathered

Each of our sampling configurations generates a different number of total samples, based on which and how many regions they determine to utilize. Table I shows the average percentage of utilized previous regions and the total number of samples gathered when sampling the Hurricane Isabel Pressure data set with varying sample rate. The ER method has the least number of samples because it is the most selective of which regions to reuse. When we reuse fewer regions, fewer random samples are added, therefore lowering the overall number of additional samples. The HA method has the most number of samples because using histogram intersection to determine which regions to reuse allows more regions to be reused.

The time series data set's unchanging regions are most likely not part of the ROI, as the interesting data values usually move and change over time. Since the HR and ER methods reuse samples in regions that do not change much between time-steps, they have more storage available to add more samples from these interesting features randomly. Figures 10a and 10b, demonstrate this notion, as when sampling the ExaAM data set at a 0.1% sample rate, the HR and ER methods have more samples clustered around the ROI. The HA and EA methods, however, use the previous samples in addition to samples taken for the current time-step; thus, Figures 10c and 10d show more samples outside of the ROI.

Our sampling methods do not explicitly find the boundaries of the ROI, nor do they take in an extra parameter to specify these bounds. By their nature, they either have more samples in the static regions (HA and EA) or in the dynamic regions (HR and ER), which usually corresponds to regions outside and inside the ROI, respectively.

F. Post-Reconstruction Quality

We use linear interpolation using a Delaunay triangulation to reconstruct the data set from our samples, then compare the quality of our new visualization of the data with the original by calculating the signal-to-noise (SNR), defined as

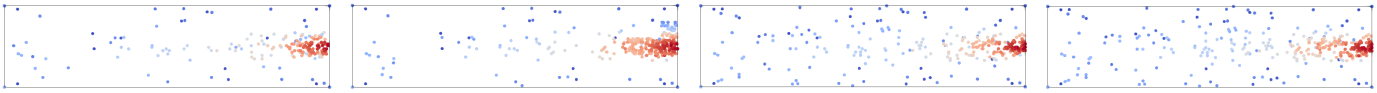
$$SNR = 20 * log_{10} \frac{\sigma_{raw}}{\sigma_{noise}} \tag{3}$$

where σ_{raw} is the standard deviation of the original data and σ_{noise} is the standard deviation of the error of the reconstruction (calculated as the difference between original and reconstructed data values). Error bars of the standard deviation are included to show the average range of each of the methods as well.

We first experiment with the Hurricane Isabel Pressure data set. Since this data set has a relatively small number of time-steps, we are able to manually specify an independent ROI boundary for each of the 48 time-steps after we have gathered our samples and reconstructed the data set. We do so to calculate the average SNR of both the overall data set and the ROI across time.

Figure 12a shows that over the entire region, as sample rate increases, the SNR of all methods scale linearly and the ER method yields the highest quality. Specifically looking at the quality of the ROI, Figure 12b shows that HR and ER yield the highest SNR.

When experimenting with the ExaAM data set, we measure quality only for the overall data set, as manually defining the ROI for over 100 time-steps becomes too time-consuming and not practical for the typical end-user. Even without being given specific ROI dimension and location, our method innately yields higher quality in the vastly changing data set regions, which are usually part of the ROI. Figure 12c shows that overall, the reuse methods again choose samples that yield the highest SNR. We achieve higher overall SNR values for the ExaAM data set, because our methods work best with data sets that evolve smoothly over time. The Hurricane Isabel data set has an identifiable ROI to track over time, but the data in other regions, like clouds, are moving more sporadically while ExaAM consists of a ROI and static surrounding areas.



(a) HR samples (b) ER samples (c) HA samples (d) EA samples

Fig. 10: Samples in Our Four Method Combinations, Using the ExaAM data set and a sample ratio of 0.1%

	Ratio	Control	HR	ER	HA	EA
Regions Reused	0.5%	0	53%	35%	53%	63%
	1%	0	53%	35%	53%	63%
	2%	0	53%	35%	53%	63%
Samples	0.5%	15.6k	19.2k	18.7k	19.2k	19.2k
	1%	31.3k	38.8k	37.8k	39.0k	38.8k
	2%	62.5k	79.8k	77.1k	80.0k	79.4k

TABLE I: Hurricane Isabel Pressure Data set comparison of samples gathered per method (16 bins; 200 regions).

	Ratio	HR	ER	HA	EA	ZFP
ROI	0.1%	21.13	20.16	18.89	19.08	-
	0.5%	29.23	28.06	25.51	25.79	14.96
	1%	31.39	30.60	28.80	29.00	27.44
	2%	33.07	32.57	31.28	31.35	38.64
Overall	0.1%	14.34	14.75	14.65	14.83	-
	0.5%	16.66	17.08	16.81	16.77	14.41
	1%	17.86	18.26	17.90	17.88	27.44
	2%	18.67	19.19	19.15	19.07	39.96

TABLE II: Hurricane Isabel data set SNR comparison to ZFP and methods, with varying sample/compression ratio.

VI. Discussion

The user's constraints dictate which method is best to use. For users with strict time constraints, the original, time-independent method is the better option, as it spends no extra time making comparisons between time-steps, while our method introduces an overhead. The second best method would be to use Histogram Based Reuse, as it will yield slightly higher quality while taking slightly more time. Our other three methods may take too much overhead in speed for the improvements in quality they bring.

For users with a strict cache constraint, like an *in situ* situation where large simulation data is taking the majority of storage, both the original time independent method and the HR method are viable options, as they use very little to no extra space.

Lastly, if the user has a strict quality constraint, the ER method becomes the best option, because it, on average, yields the highest quality.

VII. Comparison with Compression

Lossy compression achieves higher compression ratios than standard lossless compression through the addition of some inaccuracies within the data [9], [23]–[28]. Both lossy compression and data sampling aim to reduce the overall data size while introducing error within the data, but their approaches are fundamentally different. In this section, we compare the quality overall and feature regions of our novel methods to the industry standard lossy compressor ZFP [9].

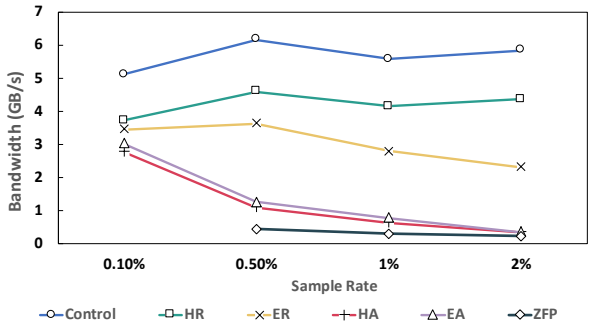


Fig. 11: Average sampling bandwidth (GB/s) of 48 time-steps of Hurricane Isabel data set (16 bins; 200 regions).

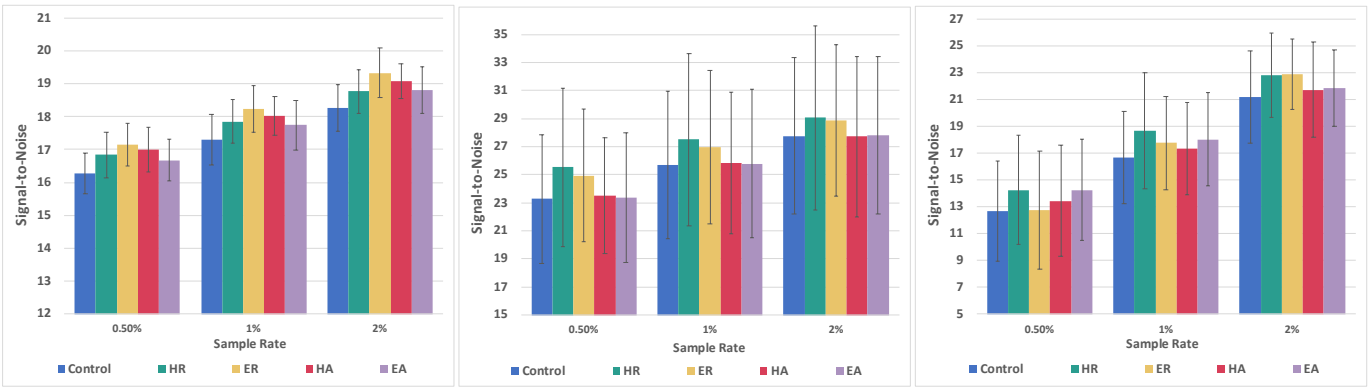
We compare our results against ZFP's Fixed-Accuracy mode, where the user provides an absolute error bound, which ensures all data is kept with similar accuracy. We choose this mode as it is the ZFP configuration that yields the highest compression ratios while also retaining high levels of data quality. While ZFP has a Fixed-Rate mode where the user sets a fixed compression ratio, we set the configuration to yield the highest compression ratio possible and were not able to reach the high compression ratios needed for our comparison.

To accurately compare ZFP's Fixed-Accuracy mode against our four sampling methods, input parameters must be set in all such that similar compression ratios are the result. This process is straightforward when using our four sampling methods as they are capable of reducing data size to a specified compression ratio. However, this is not the case with ZFP, as a trial-and-error process or a tool such as FRaZ [28] is needed to determine an error bound that results in a specific compression ratio, but at a high cost.

A. Evaluation

When comparing ZFP against our four methods, we use the average of 48 consecutive time steps of the Hurricane Isabel Pressure data set. When looking at the bandwidth of all trials in Figure 11, we show that all of our sampling methods reduce data size faster than ZFP; however, as sampling rate increases, HA and EA begin to reach the speeds of ZFP.

In Table II, we list the SNR of the ROI and the overall region. We design our sampling method to work at low sampling rates ($\leq 1\%$); thus each of our method configurations achieve greater quality than ZFP in both the ROI and overall



(a) Overall Isabel Pressure: 48 time-steps (b) ROI Isabel Pressure: 48 time-steps (c) Overall ExaAM: 108 time-steps

Fig. 12: Average SNR over Varying Sample Rates.

when the sample ratio is extremely low. In fact, ZFP was unable to produce decent representations of the data after decompression for a sample ratio of 0.1% (a compression ratio of 1000:1); however, as the sample ratio increases, ZFP begins to outperform the level of quality data sampling can achieve. Across sampling rates, ZFP shows an even SNR across the data set, while our methods consistently yield higher quality within the ROI, which is specifically intended in our design.

VIII. Conclusion

In this paper, we combine spatial and temporal data reduction techniques to enable a higher post-reconstruction quality than existing reduction methods. We show that by utilizing samples from certain regions in the previous time-step, we achieve an improvement in quality both overall and in the region of interest. Our method's process depends on user constraints, which dictate how to determine which regions to utilize and how to use those samples. This user input and method flexibility enable us to have the better method in several categories.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. SHF-1910197. This work was funded by Los Alamos National Laboratory under Information Science and Technology Student Fellowship program. The publication has been assigned the LANL identifier LA-UR-20-27478. We would like to thank our ECP collaborators on the ExaAM project. Clemson University is acknowledged for generous allotment of compute time on Palmetto cluster.

References

[1] G. Strand, "The cesm workflow re-engineering project," *AGUFM*, vol. 2015, pp. IN11C–1791, 2015.

[2] S. Habib, V. Morozov, N. Frontiere, H. Finkel, A. Pope, and K. Heitmann, "Hacc: extreme scaling and performance across diverse architectures," in *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2013, pp. 1–10.

[3] A. Tikhonova, C. D. Correa, and K.-L. Ma, "Explorable images for visualizing volume data." in *PacificVis*. Citeseer, 2010, pp. 177–184.

[4] B. Nouanesengsy, J. Woodring, J. Patchett, K. Myers, and J. Ahrens, "Adr visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement," in *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, 2014, pp. 43–50.

[5] S. Dutta, C. Chen, G. Heinlein, H. Shen, and J. Chen, "In situ distribution guided analysis and visualization of transonic jet engine simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 811–820, 2017.

[6] J. Ahrens, S. Jourdain, P. OLeary, J. Patchett, D. H. Rogers, and M. Petersen, "An image-based approach to extreme scale in situ visualization and analysis," in *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 424–434.

[7] S. Di and F. Cappello, "Fast error-bounded lossy hpc data compression with sz," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, p. 730–739.

[8] 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 (Big Data)*, Dec 2018, p. 438–447.

[9] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec 2014.

[10] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann, "In-situ sampling of a large-scale particle simulation for interactive visualization and analysis," *Computer Graphics Forum*, vol. 30, no. 3, p. 1151–1160, 2011.

[11] H. Childs, "Data exploration at the exascale," *Supercomputing frontiers and innovations*, vol. 2, no. 3, pp. 5–13, 2015.

[12] T.-H. Wei, S. Dutta, and H.-W. Shen, "Information guided data sampling and recovery using bitmap indexing," in *2018 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2018, pp. 56–65.

[13] A. Biswas, S. Dutta, E. Lawrence, J. Patchett, J. C. Calhoun, and J. Ahrens, "Probabilistic data-driven sampling via multi-criteria importance analysis," in Submission to IEEE Transactions on Visualization and Computer Graphics.

[14] A. Biswas, S. Dutta, J. Pulido, and J. Ahrens, "In situ data-driven adaptive sampling for large-scale simulation data summarization," in *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization - ISAV '18*. ACM Press, 2018, p. 13–18. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3281464.3281467

[15] B. Zhou and Y.-J. Chiang, "Key time steps selection for large-scale time-varying volume datasets using an information-theoretic storyboard," *Computer Graphics Forum*, vol. 37, no. 3, p. 37–49, 2018.

[16] H. Akiba, N. Fout, and K.-L. Ma, "Simultaneous classification of time-varying volume data based on the time histogram." in *EuroVis*, vol. 6, 2006, pp. 1–8.

[17] V. Solteszova, N. N. Smit, S. Stoppel, R. Grüner, and S. Bruckner, "Memento: Localized time-warping for spatio-temporal selection," in

Computer Graphics Forum, vol. 39, no. 1. Wiley Online Library, 2020, pp. 231–243.
[18] Y. Gu and C. Wang, "Transgraph: Hierarchical exploration of transition relationships in time-varying volumetric data," vol. 17, p. 2015–2024, Dec 2011.
[19] U. Ayachit, *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.
[20] "Hurricane isabel simulation data." [Online]. This data was produced by the Weather Research and Forecast model, courtesy of NCAR, and the U.S. National Science Foundation. Available at http://vis.computer.org/vis2004contest/data.html.
[21] J. Belak, J. Turner, and E. T. Team, "Exaam: Additive manufacturing process modeling at the fidelity of the microstructure," *APS*, vol. 2019, pp. C22–010, 2019.
[22] Z. Jibben, "truchas-pbf," https://gitlab.com/truchas/truchas-pbf/, 2020.
[23] X. Liang, S. Di, D. Tao, S. Li, B. Nicolae, Z. Chen, and F. Cappello, "Improving performance of data dumping with lossy compression for scientific simulation," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*, 2019, pp. 1–11.
[24] F. Cappello, S. Di, S. Li, X. Liang, A. M. Gok, D. Tao, C. H. Yoon, X.-C. Wu, Y. Alexeev, and F. T. Chong, "Use cases of lossy compression for floating-point data in scientific data sets," *The International Journal of High Performance Computing Applications*, vol. 33, no. 6, pp. 1201–1220, 2019.
[25] S. Di and F. Cappello, "Fast error-bounded lossy hpc data compression with sz," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, pp. 730–739.
[26] J. Calhoun, F. Cappello, L. N. Olson, M. Snir, and W. D. Gropp, "Exploring the feasibility of lossy compression for pde simulations," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 397–410, 2019. [Online]. Available: https://doi.org/10.1177/1094342018762036
[27] L. Fischer, S. Götschel, and M. Weiser, "Lossy data compression reduces communication time in hybrid time-parallel integrators," *Computing and Visualization in Science*, vol. 19, no. 1, pp. 19–30, Jun 2018. [Online]. Available: https://doi.org/10.1007/s00791-018-0293-2
[28] R. Underwood, S. Di, J. C. Calhoun, and F. Cappello, "Fraz: A generic high-fidelity fixed-ratio lossy compression framework for scientific floating-point data," 2020.