A Details on Experimental Datasets

The Heated Flow dataset comes from the simulation of a 2D flow generated by a heated cylinder using the Boussinesq approximation [33, 47]. We convert one time instance of the flow into a scalar field using the magnitude of the velocity vector. The dataset is available via the Computer Graphics Laboratory [1].

The E3SM Wind dataset is a 2D scalar field processed using a HiResMIP-v1.0 (1950-Control) dataset [14] from the Energy Exascale Earth System Model (E3SM) [29] project [2]. We use the magnitude of *UBOT* and *VBOT* parameters as scalar fields, which correspond to the lowest model level zonal and meridional wind, respectively.

The Viscous Fingers dataset is a snapshot of a simulation run capturing the viscous fingers, that is, areas of high concentration during diffusion. During the simulation, a cylinder is filled with water with an unlimited supply of salt at the top of the cylinder. The simulation captures the diffusion of the salt as higher density salt solution sinks down in the cylinder. This dataset originates from the IEEE Scientific Visualization Contest 2016 [5].

The Tornado dataset is a 3D synthetic model of a tornado created by Roger Crawfis [21]. The flow is scaled to a larger domain and sampled onto a regular grid. It is also available via [1].

The Tangaroa dataset contains one instance from the simulation of an incompressible 3D flow around a CAD model of the Research Vessel Tangaroa [48]. We use the magnitude of the velocity vector as the 3D scalar field.

The Isabel dataset originates from the IEEE Scientific Visualization Contest 2004 [4]. It is a simulation of a hurricane from the National Center for Atmospheric Research in the United States. We use the wind speed field and truncate $500 \times 500 \times 90$ from the original $500 \times 500 \times 100$ volume to avoid "no data" values on land.

The NYX dataset in Appendix D is a post analysis cosmological simulation dataset composed of 3D arrays in space [11]. It is based on the Lawrence Berkeley National Laboratory (LBNL) compressible cosmological hydrodynamics simulation code *Nyx* [8] that solves equations of compressible hydrodynamics flows in an expanding universe. We use dark matter density as the scalar field with the original 512^3 volume, together with truncated 128^3 and 256^3 volumes for a performance analysis.

B Evaluation Metrics

We review several metrics used for evaluating the compression results.

Number of false cases. Our method can eliminate all false cases in the decompressed data. We report the number of false cases in the decompressed data when comparing TopoSZ with off-the-shelf topology-agnostic lossy compressors.

Data compression ratio. The data compression ratio is defined to be the ratio between the uncompressed size and compressed size of the input data.

PSNR. Let f and f' denote the original and the decompressed scalar fields. The *Peak Signal to Noise Ratio* (PSNR) is defined as

$$PSNR = 20 \times \log_{10}\left(\frac{\max(f)}{\sqrt{MSE}}\right), \tag{2}$$

$$MSE = \frac{1}{N}\sum_{i=0}^{N-1} ||f(x_i) - f'(x_i)||^2. \tag{3}$$

where $||\cdot||$ denotes the L^2-norm.

Bottleneck and Wasserstein distances. We use two topology-based metrics to evaluate how much topology is preserved between f and f'. Let D and D' denote 0-dimensional persistence diagrams of f and f', respectively. Let η denote a bijection $\eta : D \to D'$. The *bottleneck distance* between D and D' is defined as [20]

$$W_\infty(D, D') = \inf_{\eta: D \to D'} \sup_{p \in D} ||p - \eta(p)||_\infty. \tag{4}$$

The *q-Wasserstein distance* [24, page 183] is

$$W_q(D, D') = \left[\inf_{\eta: D \to D'} \sum_{p \in D} ||p - \eta(p)||_\infty^q\right]^{1/q}. \tag{5}$$

We set $q = 2$ and quantify the topological differences between f and f' using

$$d_B(f, f') = W_\infty(D, D'), \tag{6}$$

$$d_W(f, f') = W_2(D, D'). \tag{7}$$

C Compression Across Persistence Thresholds

We test various lossy compressors with the Viscous Fingers dataset across multiple persistence thresholds, in addition to the results shown in Fig. 10. As shown in Fig. 15, we reran the experiments in Sec. 5.1 with persistence threshold $\varepsilon = 0.02$ (A-C), $\varepsilon = 0.06$ (D-F), and $\varepsilon = 0.18$ (G-I). We obtain the same observations from Sec. 5.1. First, TopoSZ outperforms all the error-bounded compressors on preserving topology. Second, TopoSZ has a slightly worse rate distortion in terms of PSNR, compared to SZ3 and ZFP.

We could use any persistence threshold with TopoSZ. In practice, we recommend setting $\xi > \varepsilon$, since a smaller global error bound ξ typically leads to smaller topological regions that need fine-grained controls. This would lead to less iterations and a higher compression ratio.

TopoSZ produces results with low compression ratios when we set $\xi = 0.1$ and $\varepsilon = 0.02$, as pointed by an arrow in Fig. 15 (B). TopoSZ performs similarly at $\xi = 0.1$ and $\varepsilon = 0.06$, see an arrow in Fig. 15 (E). This phenomenon happens because a larger global error bound ξ is more lenient toward false cases, whose persistence are larger than ε and less than ξ. These false cases require finer control thus more iterations, whereas more interactions decrease the compression ratio.

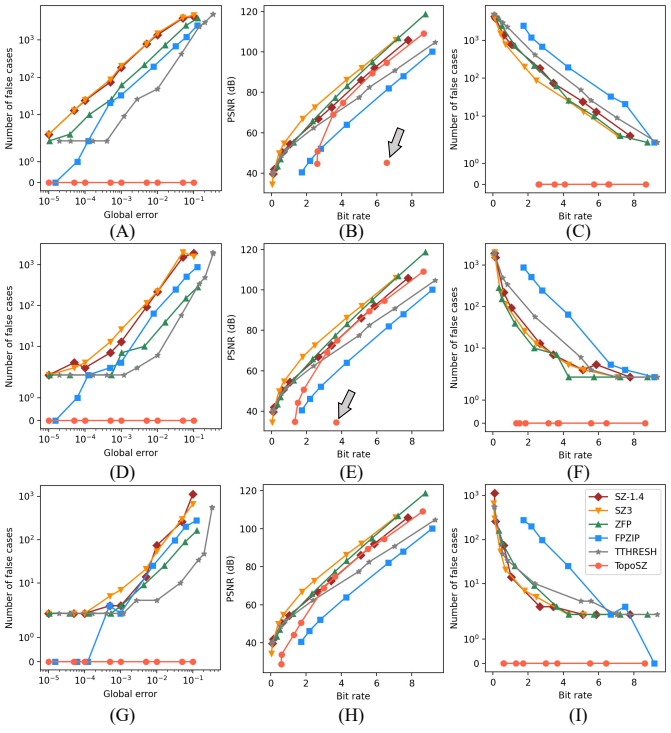

Fig. 15: Test lossy compressors with the Viscous Fingers dataset across three persistence thresholds. From left to right: the number of false cases w.r.t. global error bound; the PSNR; and the number of false cases w.r.t. bit rate (i.e., average bits per compressed data sample). Lossy compressors include TopoSZ, SZ3, ZFP, FPZIP, and TTHRESH. Persistence threshold $\varepsilon = 0.02$ (A-C), $\varepsilon = 0.06$ (D-F), and $\varepsilon = 0.18$ (G-I). All figures use the same color encoding in (I).

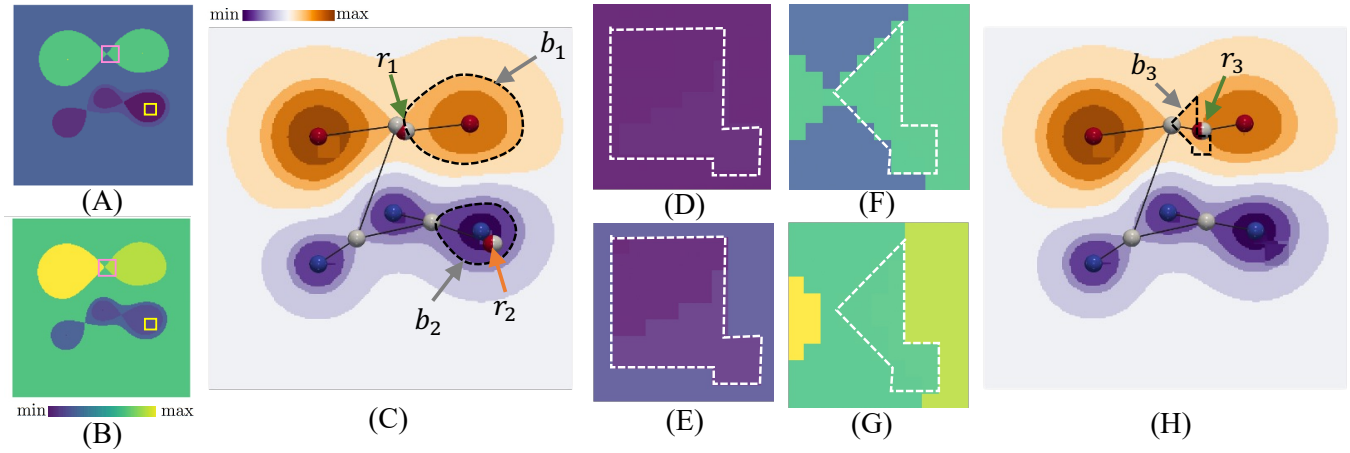

Fig. 16: Lower (A) and upper (B) bounds of TopoSZ in the initialization step for the dataset in Fig. 4. (C) shows the decompressed scalar field after initialization. (D-E): Parts of the updated lower (D) and upper (E) bounds to eliminate false positive case r_2. (D) and (E) are zoomed-in views of the yellow boxes in (A) and (B), respectively. (F-G): Parts of updated lower (F) and upper (G) bounds to eliminate the false positive case r_1. (F) and (G) are zoomed-in views of pink boxes in (A) and (B), respectively. (H) shows the decompressed scalar field after the 1st iteration.

D Additional Experiments

We perform additional experiments with the NYX dataset. We use the NYX dataset with the original (512^3) and two truncated volumes (128^3 and 256^3), respectively, as shown in Fig. 17(A-C). We investigate the relationship between the compression quality, the run time, and the size of data. Each truncated volume is a subset of the original volume with high feature density; see critical points in Fig. 17(D-F). Table 5 provides run time and compression quality (PSNR and compression ratio) of TopoSZ with a persistence threshold $\varepsilon = 0.01$ and a global error bound $\xi = 0.005$. We observe that the larger volume needs more iterations to eliminate all false cases, and more run time for each iteration and the initialization. The PSNR does not change much with the increasing data size, whereas the compression ratio increases with the data size under a uniform parameter setting.

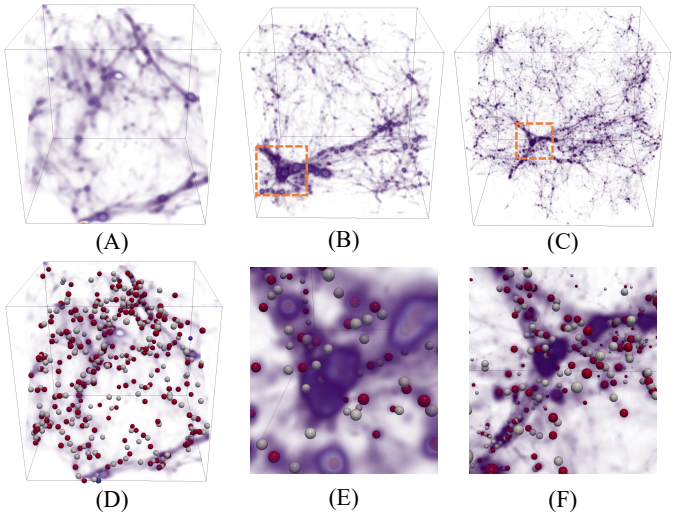

Fig. 17: 3D visualization of the NYX dataset with two truncated volumes (A) 128^3 and (B) 256^3, and the original volume (C) 512^3. (D) is (A) overlaid with critical points. (E) and (F) are the zoomed-in views of the orange boxes in (B) and (C), respectively, overlaid with critical points.

E A Worst-Case Analysis of TopoSZ

Given the nature of the expanding k-layers (see Sec. 4.1.2), it may be possible (in a worst-case scenario) that k expands to the entire dataset, forcing lossless encoding of a significant fraction of the data.

Table 5: Run time and performance analysis of TopoSZ using the NYX dataset with varying dimensions. Dim., #CP, and RO represent the dimension (the size of each volume), the number of critical points, and the compression ratio, respectively. Other notations follow the Table 1. All times are in seconds.

Dim.	Initialization			Iteration			#	#CP	PSNR	RO
	CT	UBLB	CSZ-1.4	CT	UBLB	CSZ-1.4				
128^3	27.64	2.59	0.20	-	-	-	0	400	76.54	16.7
256^3	144.89	24.02	1.19	90.25	40.27	1.13	3	444	71.8	63.9
512^3	2425.99	200.31	11.18	826.16	806.74	10.24	5	2,474	72.6	78.8

Theoretically, TopoSZ does not formally guarantee that such a worst-case scenario would not occur. However, we can study when iterations and the expansion of k-layer neighborhoods are needed, using the example dataset from Fig. 4. To force TopoSZ to run multiple iterations, we set the persistence threshold ε and the global error bound ξ to be relatively high, that is, $\varepsilon = 0.1$ and $\xi = 0.2$, respectively.

Fig. 16 (A) and (B) visualize the lower and upper bounds during the initialization step. Fig. 16 (C) shows that there are two false positive cases, marked as critical points r_1 and r_2 after the initialization. r_2 is located within a topological region (i.e., a contour-tree-induced segment) that is bounded by the curve b_2 in Fig. 16 (C). False cases such as r_2 are easy to eliminate and usually disappear when we give finer-grained bounds for their corresponding regions. Indeed, r_2 disappears after the 1st iteration, when we update the lower and upper bounds within regions Fig. 16 (D) and (E), which are zoomed views of the regions bounded by yellow boxes in Fig. 16 (A) and (B), respectively.

False cases located at the boundary of topological regions are harder to eliminate. They are usually the reason behind multiple iterations. Such false cases occur because each topological region has its own lower and upper bounds and, therefore, its own "local" compression and decompression process. When two decompressed topological regions are "glued" with each other, some false cases may occur on their shared boundary. These false cases are harder to eliminate because a new iteration might create a new, but smaller, topological region and false cases may occur on the its boundary.

For example, r_1 in Fig. 16 (C) is on the boundary of a topological region bounded by the curve b_1. In order to eliminate r_1, TopoSZ runs the 1st iteration with an updated lower and upper bounds (Fig. 16 (F) and (G)). However, a new critical point r_3 in Fig. 16 (H) appears on this new boundary (bounded by the curve b_3) after the 1st iteration.

In practice, the good news is that false cases that appear on the boundary of topological regions are usually located in tiny regions of the domain and correspond to tiny branches of the contour tree. Therefore, we only need to update the lower and upper bounds of a

tiny region in the domain to eliminate them. We also observe that in practice, these cases may disappear and appear pixel by pixel (or voxel by voxel) during iterations, if we use a 1-layer neighborhood expansion. Therefore, we use expanding k-layer neighborhood to reduce the number of iterations while sacrificing some compression ratio. Since these false cases occur in tiny regions of the domain, they are easy to eliminate with a small number of iterations. For example, in the experiment of Fig. 16, all false cases are eliminated in the 2nd iteration. Across all our experiments, we never encountered the worst-case scenario.

Finally, since the performance of TopoSZ (in terms of topology preservation) improves during iterations (see Sec. 5.2), we may terminate the compression process after a fixed number of iterations or arriving at a fixed number of false cases, to tolerate the rare worst-case scenario.

For example, when we run TopoSZ with the NYX dataset, the number of false cases after initialization is around 400, which decreases to 4 after 8 iterations. In this case, if TopoSZ is terminated after 8 iterations, almost all topological information is preserved with less compression time and a larger compression ratio, compared with removing all false cases.

F Pointwise Error Control of TopoQZ and TopoSZ

Fig. 18 demonstrates the evolution of the maximum pointwise error between the original and the decompressed data, averaged over all datasets in Table 2, for an increasing persistence threshold with different global error bounds ξ. Fig. 18 (left) indicates that TopoSZ has a strict control on pointwise error, whereas TopoQZ does not, as shown in Fig. 18 (right). Therefore, to the best of our knowledge, TopoSZ is the first lossy compressor that combines pointwise error control and topological guarantee during compression.

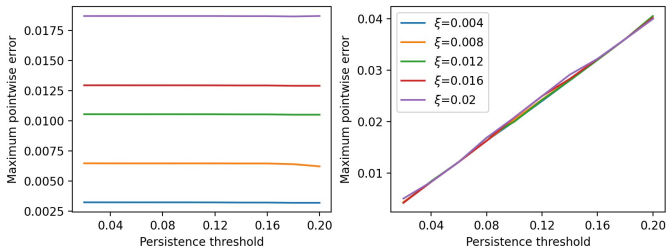

Fig. 18: The average maximum pointwise difference between the original and the decompressed data using TopoSZ (left) and TopoQZ (right).