Sharing Massive Biomedical Data at Magnitudes Lower Bandwidth with Implicit Neural Function

Runzhao Yang[1,†], Tingxiong Xiao[1,†], Yuxiao Cheng[1,†], Anan Li[3,4,†], Jinyuan Qu[1], Rui Liang[5], Shengda Bao[3], Xiaofeng Wang[3], Jinli Suo[1,2,*], Qingming Luo[5,*], and Qionghai Dai[1,2,*]

[1]Department of Automation, Tsinghua University, Beijing, China
[2]Institute of Brain and Cognitive Sciences, Tsinghua University, Beijing, China
[3]Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics, MoE Key Laboratory for Biomedical Photonics, Huazhong University of Science and Technology, Wuhan, China
[4]HUST-Suzhou Institute for Brainsmatics, JITRI Institute for Brainsmatics, Suzhou, China
[5]School of Biomedical Engineering, Hainan University, Haikou, China
[*]jlsuo@tsinghua.edu.cn; qluo@hainanu.edu.cn; qhdai@tsinghua.edu.cn
[†]these authors contributed equally to this work

ABSTRACT

Efficient storage and sharing of massive biomedical data would open up their wide accessibility to different institutions and disciplines. However, compressors tailored for natural photos/videos are rapidly limited for biomedical data, while emerging deep learning based methods demand huge training data and are difficult to generalize. Here we propose to conduct biomedical data compRession with Implicit nEural Function (BRIEF) by representing the original data with compact deep neural networks, which are data specific and thus have no generalization issues. Benefiting from the strong representation capability of implicit neural function, BRIEF achieves significantly higher-fidelity on diverse biomedical data than existing techniques. Besides, BRIEF is of consistent performance across the whole data volume, supports customized spatially-varying fidelity. BRIEF's multi-fold advantageous features also serve reliable downstream tasks at low bandwidth. Our approach will facilitate biomedical data sharing at low bandwidth and maintenance costs, and promote collaboration and progress in the biomedical field.

Introduction

Wide accessibility of biomedical data is fundamental for flourish of modern biomedical science[1–3]. On the one hand, advanced optical imaging technologies[4–10] and emerging computational imaging schemes[11–15] are producing a massive and rapidly growing amount of high-throughput, high-quality data. On the other hand, it is now very common for researchers with an informatics background to develop new algorithms and even benchmark new technologies. Efficient data sharing of vast biomedical data would assist interdisciplinary team working to accelerate biomedical discoveries via data reanalysis. Confronted with massive data, researchers use low-resolution acquisitions or even have to selectively keep the important one due to storage limitation[16]. However, the data sharing via either local hard disk or network is burdensome because of storage space or insufficient transmission bandwidth. Some large data sharing sites have to use servers with high bandwidth and large storage space[17,18], which alleviates the sharing challenges on server side at expenses of high maintenance costs, but is incapable of solving the downloading bandwidth and storage problems on the user side. These issues bring significant costs and greatly hamper the progress of biomedical studies.

The explosive biomedical data demands high performance data compression that removes data redundancy to fit the available storage/bandwidth. Lossless compression methods can preserve all the information in the data but at the cost of a low compression ratio[19]. Alternatively, lossy compression methods can achieve near-lossless compression quality at significantly high compression ratios[19] with small deviation from the original data, and are widely used in different fields. Visual data compression has been a decades-old topic, with a lot of commercial methods applied widely and successfully. However, these lossy compression methods such as H.264[20] and H.265[21] are carefully tuned to for natural scenes and cannot readily compress biomedical data, which is of largely different characteristics from natural scenes and would suffer from either low compression ratio or severe artifacts that would harm the downstream tasks.

To further raise performance of conventional compression methods on data other than natural photos or videos, some deep learning based approaches are gaining momentum. Recently Yang et al.[22] have proposed a compression method using bits-back coding[23] and variational autoencoders[24] which achieve superior performance to conventional lossy compression methods for the first time. The high performance is mainly attributed to the supervised model training which learns the best transform function, quantization method, and entropy model from a large amount of data. In spite of the high compression ratio and compression quality, current deep learning based compression is limited in multiple aspects: (i) The compressor is tailored for the training data, which is usually limited to a specific domain, and cannot serve as a general compression handling a large diversity of biomedical data. The difference between the target and training data might largely degenerate the performance, e.g., sample types, experimental conditions, acquisition devices, even slight changes in imaging settings. The adaption to such variations necessitates retraining a new model, which requires time consuming training data acquisition and model training. (ii) The model is tuned towards a specified compression quality, so one can not flexibly trade off between compression ratio and fidelity without retraining. Even with several models at different compression ratios, one can only choose among several discrete levels, while a flexible control at fine levels of even continuously is apparently a better option. (iii) In many biomedical studies, only part of the data is worthy of high fidelity storage/transmission or downstream tasks. Hence, a compression approach capable of providing customized compression quality based on the saliency, either defined manually or automatically, is highly useful in real applications.

Implicit neural function[25] uses a neural network to parameterize a function describing a continuous signal, and is revolutionizing the representation and processing of shape[26], video[27] and 3D scene[28]. Specifically, the network describing a data volume can yield the corresponding data intensity given

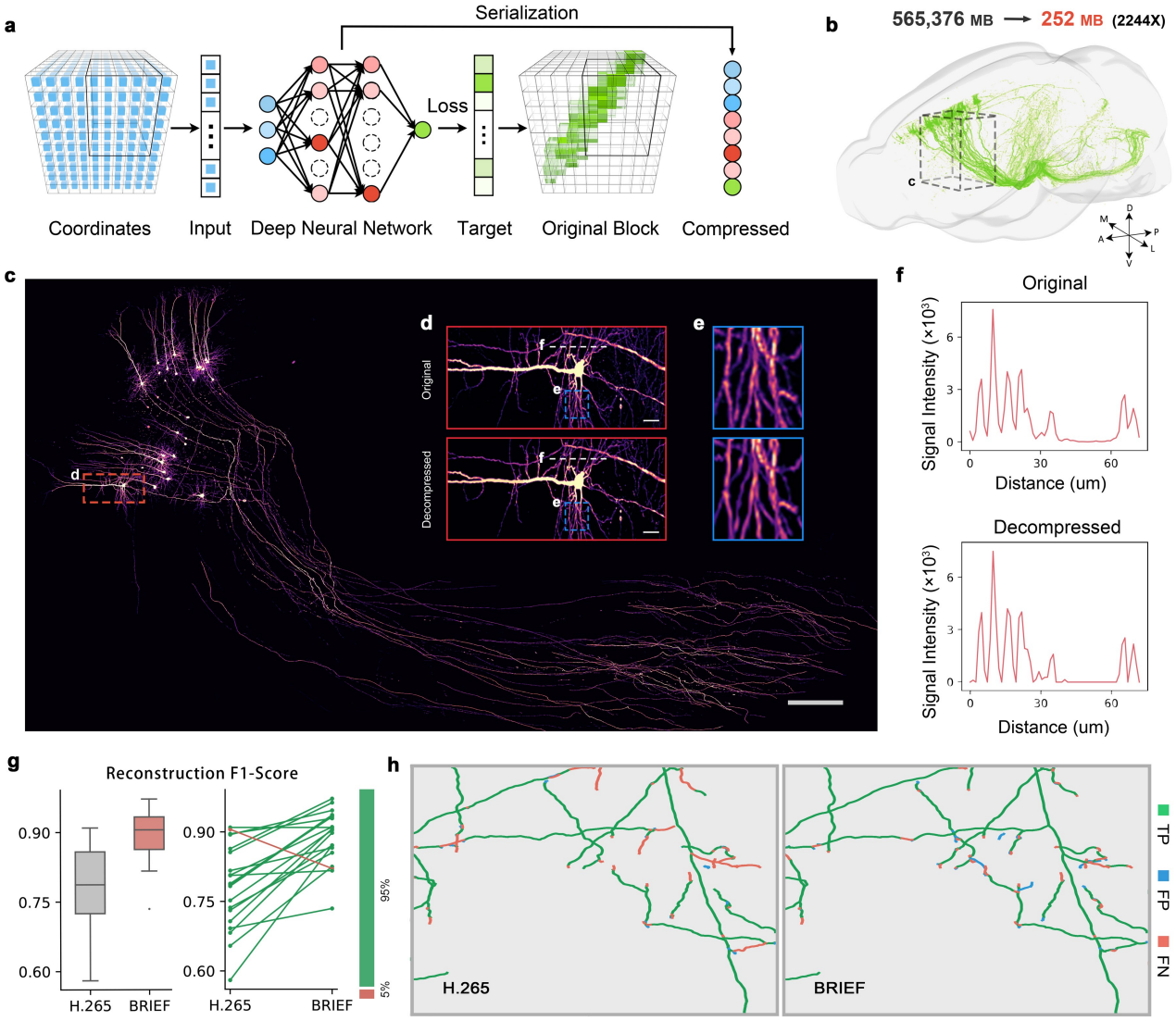


Fig. 1. The schematic of BRIEF and its compression performance on the whole-mouse-brain neuronal population. a, During BRIEF compression, the huge target data volume is firstly adaptively partitioned into blocks with similar complexity (details of partitioning strategies described in Supplementary Figure 1), and undergo separate compression via pursuing a deep neural network that maps the voxel coordinates to their intensity values. The pursued best matching network is serialized into a bit sequence for low bandwidth transmission or storage. **b**, The fluorescently labeled neurons across a whole mouse brain, including $28965 \times 20005 \times 12000$ voxels with 16 bit intensity values. BRIEF can achieve around $2244\times$ compression (from 565,376 to 252MB) at high data fidelity. **c**, Zoomed-in view of the compressed region-of-interest (ROI) labeled in **b**. Scale bar: $300\mu m$. **d,e**, Zoomed-in view of the highlighted regions in **c** and **d** respectively, in parallel with the MIP of the original data. Scale bar: $40\mu m$. **f**, Intensity profile of the original data and and its decompressed counterpart along the white dashed line labeled in **d**. **g**, The mean and standard variance of F1-score on the traced 3D morphological structure after compression by H.265 and BRIEF, and the performance difference over 20 samples, in which the BRIEF performs better and worse are highlighted with green and red lines respectively. **h**, The error map of traced 3D nerve fibers on the decompressed data. TP (True Positive): the correctly reconstructed structures; FP (False Positive): the incorrectly hallucinated structures; FN (False Negative): the missing details.

any spatial or spatio-temporal coordinate. Since most digital biomedical data is intrinsically a sampling of some field or continuous signal across a discrete grid, implicit neural function holds great potential to represent biomedical data at high fidelity. Compared to conventional representations defined over a discrete grid, implicit neural function is able to model fine details in continuous signal beyond the grid resolution but within the capacity of the neural network[25]. If we leverage the network's strong ability to find the low dimensional latent structure underlying the original data[29], we can efficiently remove the data redundancy to achieve data compression[30], i.e., fit the data with a small number of network parameters but at high representation accuracy. Moreover, such fitting is conducted in data specific manner and is thus generally applicable for diverse data, without the generalization issue confronted by previous deep learning based data compression techniques.

In this article, we propose a general biomedical data compression approach built on implicit neural function—BRIEF, utilizing the powerful representation capability of implicit neural function to obtain a compact encoding of biomedical data. To cover the large diversity of biomedical data, we design two adaptive strategies in BRIEF for efficient data compression, using the most appropriate neural network structure and least number of parameters to model the intrinsic structure of the original data. On the one hand, BRIEF takes advantage of the fact that implicit neural function is suitable for modeling frequency-concentrated data[31,32], and uses a divide-and-conquer strategy to decompose the original data into blocks with limited frequency distribution and conduct separate data compression. On the other hand, BRIEF also progressively adjusts the parameter allocation according to the features of different blocks to achieve the best fidelity under given budget. Benefited from the adaptive strategies, BRIEF reaches the highest compression ratio and compression quality on diverse biomedical data. Besides, the divide-and-conquer strategy offers high scalability for large data volume and allows for parallel computation.

Mathematically, BRIEF converts data compression into an optimization problem, searching for the best neural representation under constrains from the target file size (determined by the number and numerical precision of neural network parameters), which gives users the flexibility to precisely control the compression ratio according to the available storage/bandwidth and downstream tasks. The optimization objective also incorporates the representation fidelity of each location, and thus supports saliency-guided compression quality for a better utilization of given bandwidth or storage budget. In addition, most network hyperparameters in BRIEF are predefined and keep consistent across diverse data, which facilitates easy-to-use for non-expert users. The proposed BRIEF and its features largely alleviate the urgent need for sharing and storage of enormous amounts of high-quality biomedical data, and would contribute to the entire community of biomedical research.

Results

The principle of implicit neural function for compression and decompression. The schematic of BRIEF is shown in Fig. 1(a). For a large digital biomedical data volume, we regard which as a discrete sampling of a continuous signal defined over a spatial domain, and attempt to find a function with a small number of parameters to describe the mapping from the coordinate to the corresponding intensity, i.e. to serve as a compact encoding of the volumetric data. Considering its strong expression power, here we use implicit neural function as the mapping function to achieve powerful, flexible, and general data compression for the biomedical data.

Although a neural network can fit an arbitrarily complex function with infinite number of parameters, statistically strong correlations exist only locally, so it is more strategic/efficient to decompose a complex function into simple ones and use a series of small neural networks to fit these frequency-concentrated components[31,32]. Therefore, we adaptively divide the original data into blocks and pursue such a mapping function within each local region, as shown in Supplementary Figure 1(a). For each block, BRIEF first generates a coordinate system matching the original data volume, e.g., a uniform grid (2D, 3D, or even 4D) shown in Fig. 1(a) where each blue particle represents a grid point. Then, BRIEF builds a multi-layer perceptron (MLP) as a parameterized mapping function defined on the grid to fit the original data, with the number and numerical precision of network parameters in the MLP determining the size of the compressed data. Naturally, data compression is formulated as an optimization problem, where the weights and biases of the neurons in the deep network are optimized to match the network's output towards the intensity of the original biomedical data across the grid. The intensity is either scalar (single-channel data) or vector (multi-channel data), and here we show a single channel example for presentation clarity. The optimization can be conducted using either batch gradient descent or mini-batch gradient descent, depending on the available memory size. After finalizing the deep network, the network parameters (including the weights and biases of each neuron) and side information (including the hyperparameters of the neural network structure, the spatial range of the data block, bit depth, etc.) are extracted to form the compressed representation by serialization. Finally, the compressed data of different blocks are concatenated for successive storage or transmission. Experimentally, the compression ratio can achieve up to 2∼3 magnitudes at high fidelity for most biomedical data. The data decompression can be conducted simply by applying forward propagation to the spatial coordinates by corresponding implicit neural function, parsed from the compressed representation and side information. Afterwards, the decompression of different blocks is conducted separately, and the reconstructed blocks are stitched together to form the whole data volume, as shown in Supplementary Figure 1(b).

As for the block-wised compression of the large data volume, building instance specific partitioning and storage/bandwidth allocation among blocks is necessary to cover the large diversity among varying biomedical data. To address this issue we propose a two stage adaptive strategy: Firstly, an adaptive partitioning method is designed to decompose the original data into proper number of blocks, of different sizes and similar complexities (Extended Data Fig. 2). Followed by this, BRIEF allocates parameters according to the characteristics of each data block, and dynamically adjusts the parameter allocation to balance the compression quality over all the blocks during the compression of each data block (Extended Data Fig. 3).

Benefiting from the novel coding paradigm leveraging high representation power of implicit neural function and adaptive divide-and-conquer strategy, BRIEF not only achieves compression capability beyond existing compression tools, but also possesses more distinguished features, which enable high generalization ability, wide applicability, stability, flexibility, and scalability:

- BRIEF pursues the optimal neural representation tailored for the target data, and is thus totally free of generalization issues confronted by existing deep learning based compressors.
- BRIEF is a general compression approach widely applicable for diverse biomedical data for both biologic studies and clinical diagnosis, such as 2D image, 3D volume, and video.
- BRIEF produces stable compression quality along different dimensions, since the compression network is directly defined over the homogeneous spatial grid to model the anatomical structure of the target data.
- Spatially varying compression fidelity can be tailored according to the region's importance, which remains a key challenge for conventional compression methods.
- The compression ratio can be precisely regulated by pursuing a proper neural encoding (i.e., network structure, weights, and bias of a set of MLPs) under strict budget constraints from available storage or transmission bandwidth.
- Our partitioning strategy not only enhances BRIEF's representation accuracy, but also makes BRIEF scalable for the size of the original data and allows parallel computation, which enhances computation efficiency.

BRIEF compresses whole-mouse-brain neural population by 3 magnitudes with high data fidelity. A neuroanatomical study of the brain-wide neural architecture can provide rish insights into the intrinsic structure and working mechanism of brain's neural system. However, the huge number of neurons and high dynamic range causes massive data volume, which hampers the subsequent analysis and demands a compressor with high fidelity at high compression ratio. We apply BRIEF to the whole-mouse-brain fluorescent neural data captured by the HD-fMOST system[9], with $0.32 \times 0.32 \times 1.00 \mu m^3$ resolution to resolve the elaborate neural structures and 16-bit recording to cover the 5-magnitudes fluorescent intensity. The final 3D volume is of $28965 \times 20005 \times 12000$ voxels and occupies 565,376 Mega Bytes (MB), as shown in Fig. 1(b).

BRIEF can achieve 2244× compression (from 565,3769MB to 252MB) while preserving the complex thin structures and high-dynamic-range fluores-

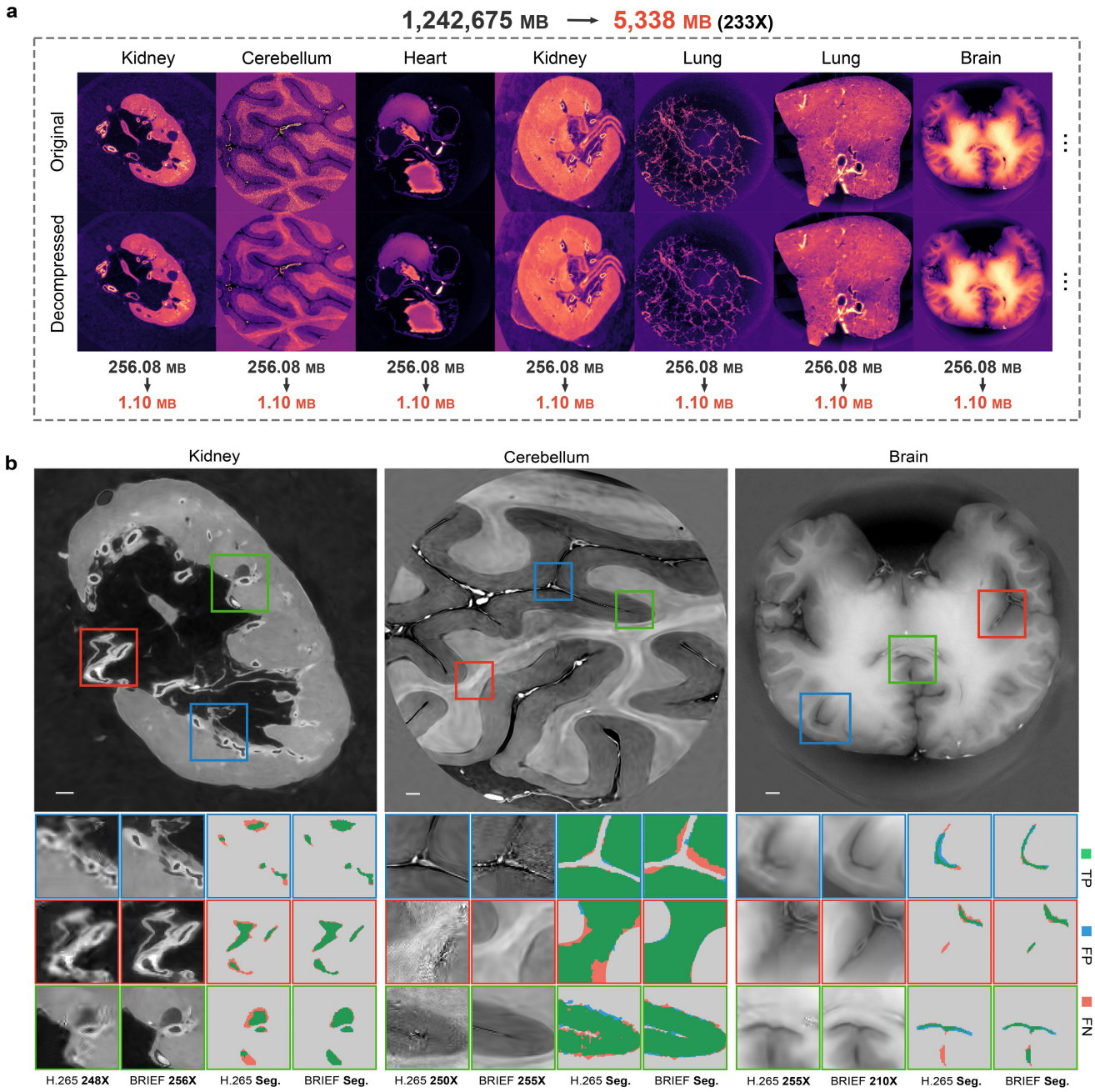


Fig. 2. The compression capability of BRIEF on organic CT data from HiP-CT dataset and the performance of downstream tasks. **a**, Exemplar 2D slices from seven data groups of different organs or organ parts (top) and their decompressed counterparts after compression by BRIEF (bottom). The whole data collection is compressed by 233× (from 1,242,675MB to 5,338MB), and the compression ratio for different groups are labeled below the exemplar images. **b**, The comparison of segmentation results on decompressed data by BRIEF and H.265 for kidney, cerebellum, and brain, at around 230× compression ratio. For each example, the zoomed-in view of decompressed data and segmentation error map of highlighted regions are shown side by side, with segmentation results color-coded by TP, FP, and FN respectively. Scale bars from left to right: $4mm, 2mm, 5mm$.

cent intensities faithfully, as shown in Fig. 1(c), the zoomed-in view showing the maximum intensity projection (MIP) of a region-of-interest (ROI) labeled in Fig. 1(b). For a better demonstration of the compression capability, we display the further zoomed-in view of a ROI in Fig. 1(c) side by side with the original version, as shown in Fig. 1(d) and (e). From the comparison one can see that BRIEF preserves the complex and faint morphological structures of whole-mouse-brain neurons with high precision despite at such high compression ratio. Besides the visual comparison, we also plot in Fig. 1(f) the profiles of fluorescence signal across a highlighted line with large intensity variations to validate BRIEF's high fidelity. We also demonstrate our performance on the the huge whole-brain neuron data in Supplementary Video 1.

Thanks to the high-fidelity compression of BRIEF, researchers can conduct subsequent down stream processing after low-band-width data sharing. For neurons, the single most important task is to precisely trace the morphological structure of the nerve fibers and soma. At a high compression ratio up to 2244×, our method still preserves the spatial location of nerve fibers. We used an open source software Gtree[1] to locate the nerve fibers with 3D tracing. Compared with H.265 we not only reconstruct the nerve fibers correctly (more TP) and ensure their continuity (less FN), but also avoid the wrong hallucination (less FP), as shown in Fig. 1(h). For numerical comparison, we use F1-score to measure the preserved morphological structures in the 3D tracing results after compression. Our algorithm significantly improves the accuracy of the 3D traces and outperforms H.265 in 95% cases (Fig. 1(g)).

BRIEF achieves effective compression on diverse clinical computed tomography (CT) data. The continuously and rapidly growing medical image data size are putting high pressure on hospitals' storage capacity and hampering the data sharing for intelligent medial data analysis. These clinical data consist of rich details crucial for diagnosis and demands high quality compression.

To demonstrate BRIEF's performance on clinical imaging data, we used the HiP-CT[10] dataset, a collection of CT data of human organs. The HiP-CT dataset consists of 54 three dimensional data volumes from different human organs or organ parts. Such clinical CT data are from tissue to cellular scales, and of large diversity in appearance, with some typical examples shown in the top show of Fig. 2(a).

The whole dataset occupies 1,242,675 MB in total, and after compression the data size decreases to 5,338 MB (i.e., 233× compression). From the comparison of the exemplar data in Fig. 2(a) top and their decompressed version in Fig. 2(a) bottom, one can see that BRIEF preserves the sharp edges and complex structures at high precision, such as the tubules in kidney, white and grey matters in cerebellum, bronchioli in lung, even under around 230× compression ratios. One can zoom in or visit Supplementary Videos 2 and 3 for a clearer comparison.

For clinical data, the compression algorithm should neither destroy the key cues important for diagnosis (e.g., fine structure and sharp edges) nor produce any artifacts that might result in misdiagnosis. At a compression ratio around 230×, BRIEF reproduces the kidney's structure including the sharp boundaries and widely distributed tiny tubules (Fig. 2(b), left), the white and grey matters in cerebellum (Fig. 2(b), middle) and the sulci in brain (Fig. 2(b), right) as well. We also evaluate the compression quality in terms of segmentation accuracy that helps diagnosis, mainly the boundaries between kidney tubules, between white and grey matters, and between cortex. Here we used marching squares[33] and intelligent scissors[34] to semi-automatically segment the ROIs, and the segmentation results are shown in the bottom row in Fig. 2(b). We also compare the segmentation accuracy with state-of-the-art compressor H.265. Clearly, after compression by H.265 the segmentation results suffer from large area false segmentation, especially near the boundaries. The incorrect segmentation is mainly attributed to the severe streak artifacts in the high frequency regions, as shown in the decompressed data placed on the side. In contrast, BRIEF preserved the details with high fidelity and few regions are mis-classified.

BRIEF allows higher-fidelity compression for more important regions. For a biomedical data volume, the details at different locations are not equally important, e.g., the precision instrument operator would preserve the faintest observations, the researchers would focus on the regions that might reveal new insights, the doctors on those help diagnosis, etc. Such requirements demand a compression approach supports customized spatially varying fidelity across the volume, while no existing compressor supports region specific quality. For BRIEF, we can simply generate a weight map from user defined ROIs to represent the priority and incorporate it into the optimization.

For example, for the nerve fibers with faint intensities, we generate a binary weight map to prioritize such neurons (intensity within range $[100, 2000]$), and conduct comparison on tens of data volumes to show the advantages of introducing prioritized fidelity. The results are shown in Fig. 3(a)-(e), with "w Weight Map" denoting the result with weight map and "w/o Weight Map" denoting that sets equal importance. Using accuracy with threshold of 200 and F1-score as the evaluation metric, the box plots and ladder plots in Fig. 3(a) show that we can get both higher accuracy, more correct 3D tracing, and smaller variance as well in almost all the cases. One typical example is shown in Fig. 3(b), from which one can see that after imposing higher priority BRIEF preserves more faint signals (top row), and thus facilitates more accurate morphology reconstruction (bottom row). We also zoom in one highlighted subregion for clearer comparison in terms of MIP (Fig. 3(c)) and intensity profile along the dashed line (Fig. 3(d)). The results clearly show that BRIEF can be tuned for higher-fidelity compression for fainter signals and thus retaining more details around smaller nerve fibers, which consists well with the statistics in Fig. 3(a).

It also potentially adds another superior feature to our approach, namely users could specify the compression quality of ROIs according to the downstream tasks. Equipped with the ability to customize compression ratio together with ROI, BRIEF can make the most extensive use the limited budget.

BRIEF produces stable compression quality along each dimension/axis. For biomedical data that describes a volume in 3D space (e.g. neurons and vessels), it is important for the compression algorithm to keep stable compression quality along each axis to facilitate accurate reconstruction of its 3D structure. We compare the fluctuation range of compression quality along z-axis for vessels between BRIEF and H.265, with the results shown in Fig. 3(e)-(g). Compared to H.265, BRIEF has a much smaller performance variation along z-axis, and the superiority is consistent over almost all the test data. As highlighted by the green arrows, one can see that H.265 loses some details in 5th slice while hallucinates some in slice #6, which causes abrupt changes between adjacent slices in the decompressed data. Such artifacts will harm the successive 3D reconstruction of vascular structure and cause incorrect merging or even breaking of vessels. The artifacts by H.265 are mainly caused by flaws in optical flow prediction, which partially explains the performance limitation of H.265 for 3D biomedical data, which hinders further processing and subsequent diagnosis. Oppositely, BRIEF obtains results almost identical to the original input. It is mainly attributed to the fact that we use an implicit neural function to model the 3D structure embedded in the data and avoid these flicker effects.

BRIEF supports precise specification of compression ratio. For effective management of massive biomedical data, it is crucially important to conduct compression to fit the large volumes into available storage or transmission bandwidth. An approach allowing users to set compression ratios freely can be helpful for making extensive use of the given budget. Supplementary Figure 2 demonstrates the ability of BRIEF to precisely control the compression ratio, which is largely different from five methods whose final achieved compression ratio might largely deviate from the target setting. In comparison, for each method, we give fixed parameter settings related to compression ratio (please refer to Supplementary Table 1 for details) and apply compression to the data randomly selected from the three aforementioned datasets. Obviously, for existing methods, even tuned for the same intended compression ratio, the achieved compression ratio varies among different data and the among-data variance is especially large for SGA and H.264. This will lead to the following shortcomings: First, repeated tuning is inevitable for users to approximate the expected compression ratio for the input data. Second, the final achieved ratio cannot exactly match the target ratio. Even worse, the compression setting reaching the expected compression ratio for data is inapplicable for others and data specific tuning is required. Oppositely, BRIEF returns a compression representation of the input data with the exact

[1] https://github.com/GTreeSoftware/GTree

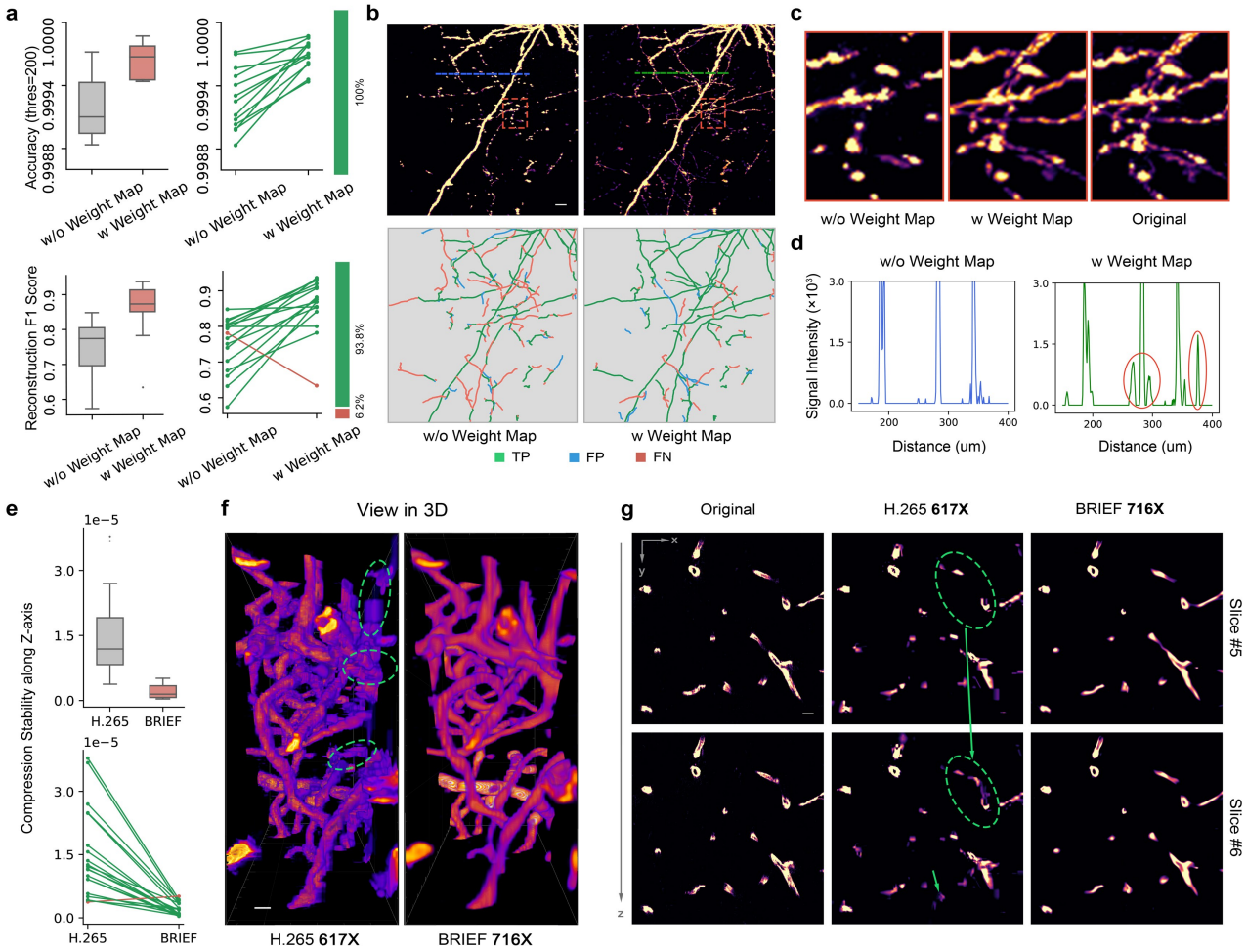


Fig. 3. BRIEF's features in supporting spatially varying quality specification and homogeneous compression quality along different dimensions. a, The mean and standard variance of the data accuracy with threshold of 200 (top) and F1-score on the traced neuronal morphology (bottom), calculated from 16 data volumes decompressed by BRIEF with and without weight map. Here the BRIEF with weight map performs better and worse are highlighted with green and red lines respectively. **b**, The visual comparison of MIP (top) and tracing result (bottom) on one exemplar decompressed data in **a**, with the color coded error map computed with respect to the morphology retrived from the original data. Scale bar: $30\mu m$. **c**, Zoomed-in view of the highlighted ROIs in **b**, in parallel with the original data. **d**, Intensity profiles along the dashed lines labeled in **b**. **e**, The mean and standard deviation of BRIEF's and H.265's performance fluctuations along axial slices, calculated from 20 test samples and displayed with the same layout and legend as in **a**. **f**, 3D vascular structures reconstructed from one of decompressed data in **e** by H.265 (left) and BRIEF (right), shown in *xz*- top view. Green dashed circles highlight the incorrectly broken and merged vessels caused by H.265's unstable compression quality. Scale bar: $9\mu m$. **g**, The visual comparison among original data, BRIEF's and H.265's decompressed version, on two adjacent slices from **f**, with compression ratio and slice index labeled at top and right respectively. The green dashed circle highlights H.265's flicker artifacts along *z*-axis. Scale bar: $9\mu m$.

specified compression ratio (Extended Data Fig. 4). This virtue facilitates users choosing a compression model according to the available transmission bandwidth or storage. Besides, we can also search a proper compressor achieving given quality by searching within a range of compression ratios.

BRIEF achieves high-fidelity compression for a diverse of biomedical data. In addition to the neurons and clinical CT data described above, we have validated BRIEF's wide applicability on diverse biomedical data, such as fluorescence micro-optical sectioning tomography of brain vessels[35], X-ray holographic nano-tomographic (XNH)[8], MRI[36], Fundus image[37], and videos of calcium imaging[38].

The vessel data were obtained by the HD-fMOST system[9], with different fluorescently labeling from neurons. To precisely reconstruct 3D vascular structure from the decompressed data, the compression methods need to accurately preserve the 3D vascular structure, including the location, continuity, and thickness of the vessels. The reconstructed 3D vascular structure is displayed in Supplementary Figure 3 (right), which indicates that BRIEF can successfully preserve the intricate spatial vascular structures and enormous branches with uneven thicknesses under high compression ratio. On the contrary, some thin capillary vessel are lost after compression by H.265. Moreover, BRIEF exhibits a superior noise suppression effect, where noise come from the capture of vessels data, indicating its high suppression of background noise while preserving a great amount of target signal (see Supplementary Note 3 for more experiments).

The XNH and MRI data were released from two published articles[8,36] respectively. Similar to the CT data, these data also include rich details describing the fine structures that crucial for successive medical diagnosis. BRIEF can achieve 2 magnitudes compression on these data while preserving fine structures, as shown in Supplementary Figures 5 and 6 and Supplementary Video 4.

As a general compressor, BRIEF can theoretically compress visual information other than 3D volumetric data. Hence, we also validated BRIEF's applicability on 2D images and videos. For 2D image, we compare BRIEF with JPEG on Fundus[37] data and show that BRIEF clearly preserves the continuous blood vessels and avoids color distortion, as shown in Supplementary Figure 7. For videos, we apply BRIEF on calcium imaging of neural activities in mouse brain[38], and results are shown in Supplementary Figure 8 and Supplementary Video 5. One can see that BRIEF shows impressive denoising ability while retaining the weak calcium signals. The calcium activity traces of individual neurons resolved from the decompressed data by BRIEF are almost identical to those from the original data, which indicates that BRIEF facilitates reliable neural signal analysis at high compression ratio.

BRIEF outperforms the state-of-the-art (SOTA) compressors on biomedical data. To demonstrate the advantageous performance of our approach, here we conduct comparison on three groups of biomedical data (neurons, vessels, and organic CT) with five methods with state-of-the-art compression quality: three widely used commercial compression techniques—JPEG[39], H.264[20], H.265[21] and two representative machine learning schemes in the literature—SGA[22] and DVC[40]. For quantitative performance comparison, we adopt three evaluation metrics for the decompressed data: Accuracy (after thresholding) for neurons and vessels, while PSNR and SSIM for organic CT. The performance statistics of six methods at different BPVs (bits per voxel) are illustrated in Fig. 4(a). Generally, BRIEF shows superior performance to the SOTAs over a wide range of BPVs, across three groups of data and in terms of all the evaluation metrics. The superiority is more distinct at higher compression ratios (lower BPV). Moreover, compared with SOTAs, the deterioration of the data fidelity along with the growth of compression ratio is less observable in BRIEF, which surprisingly indicates less contradiction between BPV and quality metrics in our method. This is mainly attributed to the optimization based encoding scheme, which searches for the optimal representation under storage/bandwidth constraints and thus can achieve decent reconstruction even at very high compression ratios.

To show more specific details in the comparison results, we compare with the second best method (H.265) on three different data (neurons, vessels and organic CT) around specific BPVs and the results are demonstrated in Fig. 4(b). In general, the average performance of BRIEF is higher than the second best competitor, with a close or less deviation in all the cases. Specifically, BRIEF outperforms H.265 in (i) 100% cases in neurons, in terms of Accuracy with thresholds of 200 and 500, (ii) 100% cases in vessels, in terms of accuracy with thresholds of 200 and 500, (iii) 84.4% cases in organic CT, considering the metric of PSNR and SSIM. The results not only prove the superior performance of BRIEF in most cases, but a better generalization ability on different groups of biomedical data.

Discussion

We propose BRIEF, an efficient and easy-to-use compression approach for biomedical data compression based on deep neural network, to facilitate efficient storage of massive biomedical data, and timely sharing across disciplinary and institutional borders. BRIEF outperforms widely used compression methods which are tailored for natural photos or videos. BRIEF intrinsically searches for a compact neural network representation fitting the original data under available budget, and is also advantageous over deep learning based compression in that achieving higher-fidelity compression, requiring no training data and totally out of generalization problems.

In addition, BRIEF is built on implicit neural network defined over homogeneous spatial grid and thus achieves comparable fidelity along both lateral and axial dimensions. One can also specify a saliency map to tailor the fidelity across the volume, which is especially useful for the huge data with only limited regions of interest, or spatially varying contributions to the downstream tasks. Implemented in an optimization manner, BRIEF supports trade off between fidelity and at arbitrary compression ratios instead of roughly choosing across coarse levels as in off-the-shelf compression software/tools.

Our approach is widely applicable and of markedly performance for diverse biomedical data. We test BRIEF's performance on seven types of biomedical data (fluorescence image of brain neurons and vessels, CT data of human organs, XNH data of drosophila leg, knee MRI, fundus image, and calcium imaging of neural activities) and achieve 2∼3 magnitudes compression while keeping at high fidelity sufficient for down stream studies. The universality holds potential for easy sharing of a large amount of valuable data in the whole community.

To give full play of intrinsic neural representation in data compression and cover large diversity in biomedical data, BRIEF proposed adaptive partitioning and adaptive budget (number of parameters) allocation. Such strategies enable high fidelity at high compression ratios and also allows parallel compression, which can be helpful due to the wide availability of multi-core processor and maturity of multi-thread programming.

The compression process of BRIEF need to solve optimization problems to fit the data best under constraints from the given budget, including searching for the best partitioning strategy and progressive adjustment of the network parameters. Hence, it is slower than conventional ready tools (Supplementary Table 2). To address this issue, we can try to find better initialization or conduct optimization of blocks in parallel by implementing the whole neural network using fully-fused CUDA kernels[41]. However, the decompression speed of BRIEF is almost one-magnitude faster than H.265 since we conduct decompression by simple forward propagation through neural network. It is worth noting that biomedical data need to be compressed only once before saving or transmission, but decompressed frequently for processing and analysis. Therefore, BRIEF is more efficient when used in practice in the biomedical field.

To facilitate easy-to-use over the whole community, we distribute the network as an open source software package with a detailed manual on the settings. We believe that it will provide the researchers with a powerful tool for data sharing, which would allow reliable verification of experimental results and innovative research built on the existing huge amount of biomedical data.

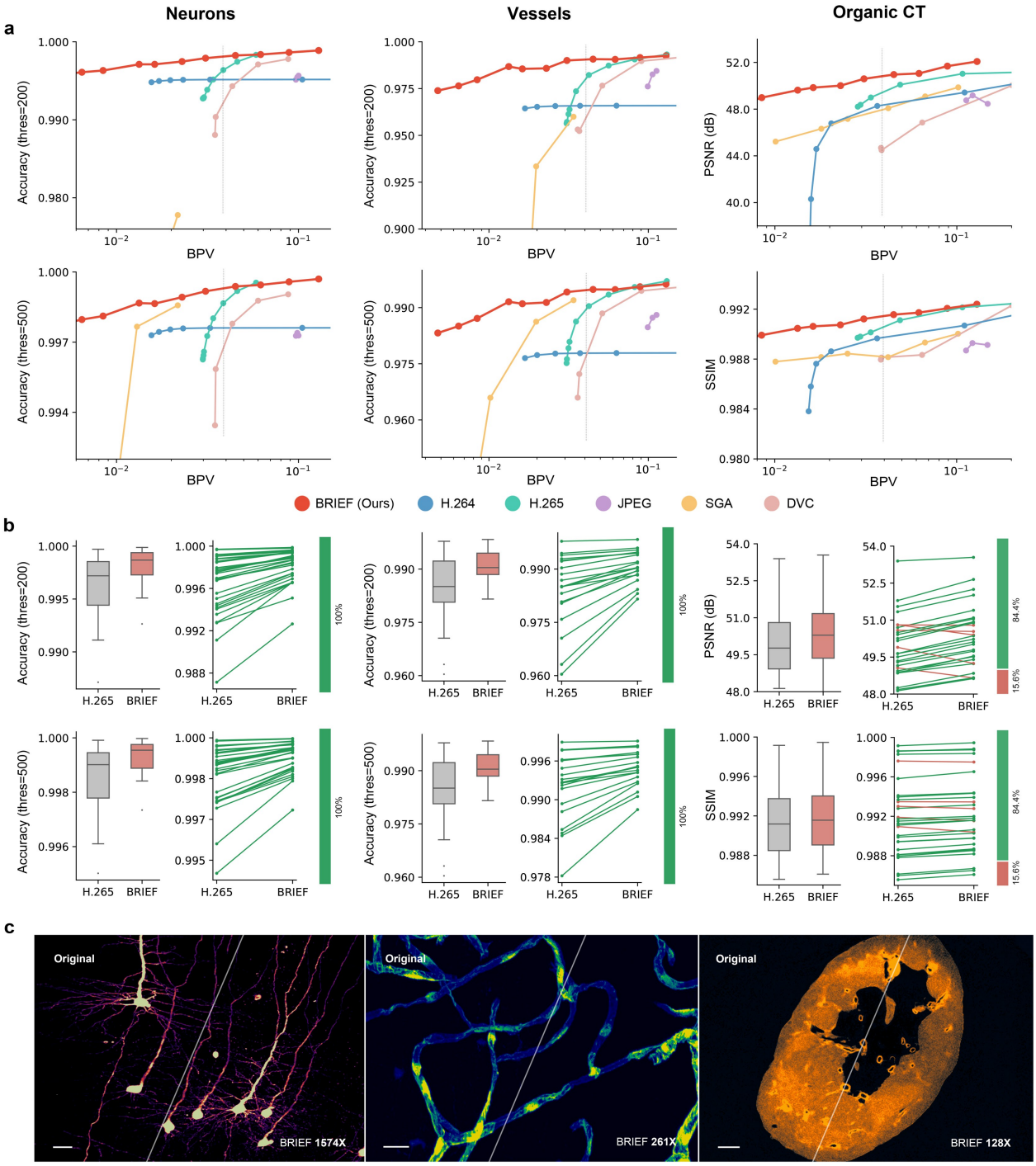


Fig. 4. Performance comparison between BRIEF and state-of-the-art compressors. a, The performance of BRIEF and five SOTAs across three groups of biomedical data at varying compression ratios. The SOTAs include three widely used commercial compression tools (JPEG, H.264, H.265) and two emerging machine learning based compressors (SGA and DVC). For quantitative analysis, we use Accuracy with thresholds of 200 and 500 as evaluation metrics for neurons and vessels with sparse structures, while PSNR and SSIM for organic CT including rich details. We control the BPV within range $[10^{-2}, 10^{-1}]$ and test at multiple discrete BPVs. **b**, The comparison between BRIEF and the second most competitor, around a given BPV annotated by grey vertical lines in **a**. Left: the mean and standard variance of compression quality over twenty test data. Right: the performance difference on each instance, with the ones BRIEF performs better and worse are respectively highlighted with green and red lines. **c**, The visual comparison of the original data and BRIEF's decompressed version, with compression ratio labeled at right bottom. Scale bars from left to right: $100\mu m, 12\mu m, 5mm$.

Methods

Network structure of this paper. We use a SIREN[25]-like multi-layer perceptron (MLP), consisting of an input layer, several hidden layers and an output layer, with the network structure shown in Extended Data Fig. 1. The input layer embeds the spatial coordinates of the original data to a higher dimensional space using a sinusoidal function with learnable parameters (frequency and phase). The sinusoidal function helps map the coordinates to the sine wave frequency components of biomedical data, which tend to lie on low dimensional structures. The hidden layers compose a fully-connected neural network with activation functions to model the intrinsic structure of data in the higher dimensional embedded space. While the output layer, a weighted linear combination with bias, i.e, without activation function, fits the mapping from the embedded coordinates to the final intensity. One can flexibly tune the number of hidden layers, activation function and the number of neurons in these layers. In our implementation, the network adopts the sinusoidal activation function and includes 5 hidden layers with equal number of neurons. The number of neurons is determined by the compression ratio and the size of the original data (Extended Data Fig. 4).

Denoting $\mathbf{v}$ as the normalized coordinate vector (2D, 3D or even 4D) and L as the number of hidden layers, the output of the input layer $\mathbf{z}^{(0)}$, of the lth hidden layer $\mathbf{z}^{(l)}$, and the final output $f_\theta(\mathbf{v})$ are respectively defined as

$$\mathbf{z}^{(0)} = \gamma(\mathbf{W}^{(\mathbf{0})}\mathbf{v} + \mathbf{b}^{(\mathbf{0})}) \tag{1}$$

$$\mathbf{z}^{(l)} = \rho^{(l)}(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}), l = 1, 2, ..., L \tag{2}$$

$$f_\theta(\mathbf{v}) = \mathbf{z}^{(L+1)} = \mathbf{W}^{(L+1)}\mathbf{z}^{(L)} + \mathbf{b}^{(L+1)}. \tag{3}$$

Here $\gamma(\cdot)$ is the sinusoidal mapping function; $\mathbf{W}^{(l)}$ is the weight, $\mathbf{b}^{(l)}$ is the bias, and $\rho^{(l)}(\cdot)$ is the activation function; $f_\theta(\mathbf{v})$ is the final mapping from coordinate $\mathbf{v}$ to the corresponding intensity, with $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)} \mid l = 0, 1, \ldots, L+1\}$ representing the learnable parameters of neural network.

Network optimization. We formulate the compression task as an optimization problem

$$\min_{\theta} \sum_{\mathbf{v}\in\mathbb{V}} \mathscr{L}(f_\theta(\mathbf{v}), \mathbf{d}(\mathbf{v})), \tag{4}$$

where $\mathbb{V}$ is the grid coordinate system matching the input data volume; $\mathbf{d}(\mathbf{v})$ is the intensity value of original data at vertex $\mathbf{v}$; $\mathscr{L}$ measures the difference between decompressed data and the original version before compression. In implementation, we adopted L2 norm loss function in all experiments if not specified. During optimization, we use Adaptive Moment Estimation Algorithm (Adam) to update the network parameters in an iterative manner, and apply mini-batch gradient descent to fit the limited memory by randomly sampling a batch of $\mathbf{v}$ from $\mathbb{V}$.

Adaptive partitioning. Since strong data correlation only exists locally, it is impractical to learn a unified implicit neural function to represent the whole data volume. Instead, here we decompose the whole data volume into blocks, and learn their own implicit neural function separately. To accommodate diverse biomedical data, we design an adaptive partitioning strategy, which formulates data partitioning as an Integer Linear Programming (ILP) problem to find a proper tree structure organization of the original data volume.

For an N-dimensional visual data, let $S_1 \times \cdots \times S_N$ denote its spatial coordinates and c denote the channels (e.g., color, spectrum, etc.), we can define an H-level tree data structure. At the first level, the root node can be described as $x \in \mathbb{R}^{S_1\times\cdots\times S_N\times c}$. At finer levels, $x^{(h)} = \{x_i^{(h)} \in \mathbb{R}^{S_1^{(h)}\times\cdots\times S_N^{(h)}\times c}, i = 1, 2, ..., 2^{N(h-1)}\}$ represents a set of blocks with $x_i^{(h)}$ being the ith node at the hth level, and the dimension of the block is

$$S_n^{(h)} = S_n / 2^{N(h-1)}, n = 1, 2, ..., N. \tag{5}$$

A partitioning scheme can be described as a tree with binary node values: $a_i^{(h)} = 0$ indicates the ith node at hth level being a leaf node with sufficiently high within-block correlation or $a_i^{(h)} = 1$ represents a non-leaf node needs further partitioning. Further, we define an objective as Eq. (6) to explicitly model the within-block correlation for pursuing a good partitioning scheme

$$\min_{\mathscr{A}} \sum_{a_i^{(h)}\in\mathscr{A}} a_i^{(h)} \cdot \frac{\mathrm{corr}\left(x_i^{(h)}\right)}{2^{Nh}}, \tag{6}$$

where $\mathrm{corr}(\cdot)$, the within-block correlation (please refer to Supplementary Note 1 for details), is calculated by the proportion of $x_i^{(h)}$'s top Γ amplitude spectrum

$$\mathrm{corr}\left(x_i^{(h)}\right) = \frac{\sum_{\gamma=1}^{\Gamma} \max[\gamma]\left(\left|\mathscr{F}(x_i^{(h)})\right|\right)}{\sum\left|\mathscr{F}(x_i^{(h)})\right|}, \tag{7}$$

with function $\max[\gamma](\cdot)$ returning the γth largest entry and $\mathscr{A}$ denotes all possible partitioning strategies

$$\mathscr{A} := \bigcup_{h=1}^{H} \left\{a_i^{(h)}, i = 1, \ldots, 2^{N(h-1)}\right\}. \tag{8}$$

The feasible set of our partitioning strategy is given in Eqs. (9) and (10).

$$\sum_{a\in\mathscr{A}} a \le a_{max} \tag{9}$$

$$\sum_{a\in\mathscr{P}(a_j^{(H)})} a = 1, j = 1, \ldots, 2^{N(H-1)}, \tag{10}$$

with $\mathscr{P}(\cdot)$ denoting a path to the root node. Here Eq. (9) defines the upper bound of the number of blocks which can be limited by a theoretical minimum number of parameters in a single implicit neural function to maintain representation capability[42], and Eq. (10) is introduced according to the rules of the tree data structure. Based on the above formulation, searching for the best partitioning strategy is formulated into an ILP problem that can be solved efficiently by state-of-the-art solvers.

Adaptive parameter allocation. Under the derived partitioning scheme, we further develop a heuristic parameter allocation strategy to allocate the given parameter budget across the blocks (Extended Data Fig. 3). Initially, the number of parameters allocated to each small block is proportional to $\frac{|x_i^{(h)}|}{\mathrm{corr}(x_i^{(h)})}$, where $|x_i^{(h)}|$ denotes the size of block $x_i^{(h)}$ and $\mathrm{corr}(x_i^{(h)})$ measures the within-block correlation. This fraction roughly reflects the complexity (or representation difficulty) of the block. Then, we alternatively optimize the implicit neural function of each block and adjust the parameter allocation according to the fitting loss on those blocks. On the one hand, we allocate more parameters to enhance the accuracy of the under-performing blocks with larger fitting loss. To be more specific, we assign the implicit neural function of under-performing blocks more neurons, and zero initialize the newly added weight and bias to secure consistency with the previous function,

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & 0 \\ 0 & 0 \end{bmatrix}, \quad \widetilde{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \tag{11}$$

where $\mathbf{W}$ and $\mathbf{b}$ are respectively the weights and bias of the original model, and $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{b}}$ are the updated version. On the other hand, the out-performing blocks with low fitting loss tend to be pruned and quantized for higher sparsity (i.e., containing more zero values) by Condensa[43]. Such a strategy would facilitate higher decompression quality and more consistent fidelity over the whole data.

Regulating the compression ratio. For our network architecture (Extended Data Fig. 1), the number of neurons in the input layer M_{in} is determined by the spatial dimensions of the original data, and that in the output layer M_{out} by the channels, so the total number of parameters in the neural network P is determined by the number of hidden layers L and the number of neurons contained in each hidden layer m as

$$P = M_{in}m + m + L(m^2 + m) + mM_{out} + M_{out}. \tag{12}$$

Then the size of the compressed data P_s can be calculated as $P_s = P \times bits$, where $bits$ is related to the numerical precision of neurons, e.g., $bits = 32$ for single-precision and $bits = 16$ for half-precision.

To regulate the compression ratio r, BRIEF first calculates the file size of compressed data (i.e., the neural network representation) as $P_s = F_s/r$, where F_s is the size of original data. Correspondingly, the number of parameters can be derived as $P = P_s/bits$. Given an L-layer network structure, BRIEF can rewrite Eq. 12 into a univariate quadratic equation

$$Lm^2 + (M_{in} + L + M_{out} + 1)m + (M_{out} - P) = 0, \tag{13}$$

whose closed-form integral solution is

$$m = \left\lfloor \frac{-(M_{in}+L+M_{out}+1)+\sqrt{(M_{in}+L+M_{out}+1)^2-4L(M_{out}-P)}}{2L} \right\rfloor, \tag{14}$$

where $\lfloor\cdot\rfloor$ means rounding down operation. Finally, BRIEF, following this workflow, will approximately achieve the given compression ratio r, with only slight deviation coming from the facts that the number of neurons can only be integral and some extra bytes are required for the side information. A detailed schematic diagram is shown in Extended Data Fig. 4.

Prioritizing fidelity of user defined ROIs. Our approach supports prioritizing compression quality for different regions. This advantageous feature is attributed to the fact that we formulate the compression task as an optimization problem defined in Eq. (4), which can easily incorporate region specific importance to guide the compression process. Specifically, one can generate a weight $\omega(\mathbf{v})$ over the grid spatial coordinate $\mathbb{V}$, with a greater value indicating higher fidelity in compression. Then the original optimization problem can be modified into

$$\min_{\theta} \quad \sum_{\mathbf{v}\in\mathbb{V}} \omega(\mathbf{v})\mathscr{L}(f_\theta(\mathbf{v}), d(\mathbf{v})). \tag{15}$$

Such user-friendly compression is quite useful but not limited to the following cases: (i) Avoiding global performance degeneration caused by a small amount of extreme values (e.g., with magnitudes higher of lower intensity) by excluding data fidelity at these positions. Otherwise, ensuring fidelity of these extreme portions might harm the overall compression quality largely; (ii) Making extensive use of the budget, by giving high priority to the ROIs, which is especially important when facing a severe bandwidth or storage shortage.

Quantitative fidelity criteria. For a 2D image of size $N_W \times N_H$, suppose $\Phi(x,y)$ and $\Psi(x,y)$ represent the original data and the decompressed version respectively, the fidelity of the decompressed data are evaluated in terms of widely used criteria, including peak signal-to-noise ratio

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{16}$$

with MAX being the dynamic range of the original data, MSE being the mean squared error

$$MSE = \frac{1}{N_W N_H}\sum_{x=1}^{N_W}\sum_{y=1}^{N_H}[\Phi(x,y)-\Psi(x,y)]^2, \tag{17}$$

and structural similarity index

$$SSIM = \left(\frac{2\mu_\Phi\mu_\Psi + C_1}{\mu_\Phi^2+\mu_\Psi^2+C_1}\right)\left(\frac{2\sigma_\Phi\sigma_\Psi + C_2}{\sigma_\Phi^2+\sigma_\Psi^2+C_2}\right)\left(\frac{\sigma_{\Phi\Psi}+C_3}{\sigma_\Phi\sigma_\Psi+C_3}\right), \tag{18}$$

with μ_Φ and μ_Ψ denoting the mean of Φ and Ψ respectively, ${\sigma_\Phi}^2$ and ${\sigma_\Psi}^2$ denoting the variance of Φ and Ψ respectively and $\sigma_{\Phi\Psi}$ denoting the covariance between Φ and Ψ. For the data the morphological structure other than actual intensity values matters, such as the neurons and vessels, we adopt the indicator from front-background segmentation task to evaluate the decompressed data structure

$$Accuracy(\tau) = \frac{1}{N_W N_H}\sum_{x=1}^{N_W}\sum_{y=1}^{N_H}\delta(\Phi_\tau(x,y)-\Psi_\tau(x,y)), \tag{19}$$

where $\Phi_\tau(x,y) = \begin{cases} 1, & \Phi(x,y) \geq \tau \\ 0, & \Phi(x,y) < \tau \end{cases}$ and the same for $\Psi_\tau(x,y)$; $\delta(\cdot)$ is the delta function.

[2] https://human-organ-atlas.esrf.eu/

[3] https://github.com/GTreeSoftware/GTree

For 3D volume or video, the fidelity score is calculated by taking average over all the slices (2D images).

Collection of biomedical data. Neurons and vessels data were imaged by our home-made HD-fMOST system[9] . Please refer to Supplementary Note for details on sample preparation and HD-fMOST system. Organic CT data were downloaded from HiP-CT project[2]. XNH, MRI, Fundus, and Calcium imaging data were released from the published articles[8,36–38]

Down-stream vision tasks. For neurons, we used an open source software Gtree[3] to locate the nerve fibers with 3D tracing. We adjusted the threshold parameter for each data to make the tracking results more continuous. For the tracing points of decompressed data, the ones having a counterpart within $4\mu m$ in the tracing result of original data are counted as TP, otherwise counted as FP. Conversely, the tracing points in original data without correspondence within its $4\mu m$ neighborhood in the decompressed counterpart are regarded as FN. For quantitative evaluation, $F1$-score is used to numerically measure the differences in tracing results before and after compression, where

$$F1\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{20}$$

with precision $= TP/(TP+FP)$, and recall $= TP/(TP+FN)$.

For vessels, we used Imaris 9.0 software to reconstruct their 3D vascular structures with intensity ranging from 20 to 2000. For rendering, we choose the Blend mode which uses all the intensities along the viewing direction together with their transparency to produce the final 3D visual demonstration.

For organic CT, the ROIs were segmented using marching squares[33] and intelligent scissors[34] semi-automatically by experts with anatomical knowledge and expertise in diagnosis.

For the retrieval of calcium activity traces of individual neurons, we simply took average over the intensities within a window around each neuron.

Workstation configuration All of the mentioned experiments are conducted on a workstation with AMD Ryzen Threadripper 3970X Processor and NVIDIA GeForce RTX3090 GPU. We use PyTorch and CUDA to implement our compression algorithm, and Gurobi to solve related ILP problems.

Serialization of neural networks. For a given data, we need to save its partitioning information and organize the compressed file by creating a folder for each block, named by its position and size, e.g., for a $16 \times 128 \times 128$ block starting from$(0,256,0)$ is named "0_15-256_383-0_12". After compressing a block (i.e., optimizing its implicit neural function), we serialize both its parameters and side information, and save them into a folder. As for the network parameters, we only store the weight and bias of each layer as a binary file named with its layer index. The side information, including descriptions of the network structure, bit depth, and size of the original data, inverse normalization coefficients, are saved as a YAML file. This file can be very small since all the items are discrete data, i.e., being countable and can only take certain values, and we can simply build a table listing all possible values and record the index. In addition, the length of the above file name is much smaller than the limit of modern file systems.

Data preprocessing. Before optimization, we normalize the coordinate vector $\mathbf{v}$ within range $[-1,1]$ and the intensity $\mathbf{d}(\mathbf{v})$ within range $[0,100]$ by Min-Max normalization. The original range and the normalization method are recorded in side information which will be used in decompression.

Data postprocessing. During decompression, we need to combine the separately decompressed blocks into whole data. As a block-wise compressor, there might exist artifacts along the partitioning boundary, especially at high compression ratios. To address this issue, we adopt an adaptive deblocking filter[44] to adaptively identify the boundaries that need to be smoothed and perform smoothing based on the values near the boundaries (please refer to Supplementary Note 2 for details).

Code availability We promise to release the codes of BRIEF once manuscript accepted.

References

1. Vines, T. H. Journals must boost data sharing. *Nature* **508**, 44–44 (2014).
2. Lambert, D., Millar, C. & Willerslev, E. Data sharing: do scientists know best? *Nature* **548**, 281–281 (2017).
3. Moore, J. *et al.* OME-NGFF: A next-generation file format for expanding bioimaging data-access strategies. *Nat. Methods* **18**, 1496–1498 (2021).
4. Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of the mouse brain. *Science* **330**, 1404–1408 (2010).
5. Hama, H. *et al.* Scale: A chemical approach for fluorescence imaging and reconstruction of transparent mouse brain. *Nat. Neurosci.* **14**, 1481–1488 (2011).
6. McConnell, G. *et al.* A novel optical microscope for imaging large embryos and tissue volumes with sub-cellular resolution throughout. *Elife* **5**, e18659 (2016).
7. Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *Elife* **5**, e14472 (2016).
8. Kuan, A. T. *et al.* Dense neuronal reconstruction through x-ray holographic nano-tomography. *Nat. Neurosci.* **23**, 1637–1643 (2020).
9. Zhong, Q. *et al.* High-definition imaging using line-illumination modulation microscopy. *Nat. Methods* **18**, 309–315 (2021).
10. Walsh, C. *et al.* Imaging intact human organs with local resolution of cellular structures using hierarchical phase-contrast tomography. *Nat. Methods* **18**, 1532–1541 (2021).
11. Betzig, E. *et al.* Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**, 1642–1645 (2006).
12. Rust, M. J., Bates, M. & Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat. Methods* **3**, 793–796 (2006).
13. Chen, B.-C. *et al.* Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution. *Science* **346**, 1257998 (2014).
14. Fan, J. *et al.* Video-rate imaging of biological dynamics at centimetre scale and micrometre resolution. *Nat. Photonics* **13**, 809–816 (2019).
15. Wu, J. *et al.* Iterative tomography with digital adaptive optics permits hour-long intravital observation of 3D subcellular dynamics at millisecond scale. *Cell* **184**, 3318–3332 (2021).
16. McDole, K. *et al.* In toto imaging and reconstruction of post-implantation mouse development at the single-cell level. *Cell* **175**, 859–876.e33 (2018).
17. Boergens, K. M. *et al.* webKnossos: Efficient online 3D data annotation for connectomics. *Nat. Methods* **14**, 691–694 (2017).
18. Wiseman, S. A FAIR platform for data-sharing. *Nat. Neurosci.* **24**, 1640–1640 (2021).
19. Said, A. & Pearlman, W. A. An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on Image Process.* **5**, 1303–1310 (1996).
20. Wiegand, T., Sullivan, G. J., Bjontegaard, G. & Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits Syst. for Video Technol.* **13**, 560–576 (2003).
21. Sullivan, G. J., Ohm, J.-R., Han, W.-J. & Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits Syst. for Video Technol.* **22**, 1649–1668 (2012).
22. Yang, Y., Bamler, R. & Mandt, S. Improving inference for neural image compression. In *Neural Information Processing Systems*, vol. 33, 573–584 (2020).
23. Hinton, G. E. & Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Annual Conference on Computational Learning Theory*, 5–13 (1993).
24. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
25. Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B. & Wetzstein, G. Implicit neural representations with periodic activation functions. In *Neural Information Processing Systems*, vol. 33, 7462–7473 (2020).
26. Genova, K. *et al.* Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7154–7164 (2019).
27. Chen, H. *et al.* NeRV: Neural representations for videos. In *Neural Information Processing Systems*, vol. 34, 21557–21568 (2021).
28. Mildenhall, B. *et al.* Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 405–421 (2020).
29. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366 (1989).
30. Lu, Y., Jiang, K., Levine, J. A. & Berger, M. Compressive neural representations of volumetric scalar fields. *Comput. Graph. Forum* **40**, 135–146 (2021).
31. Mehta, I. *et al.* Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14214–14223 (2021).
32. Kadarvish, M. S. *et al.* Ensemble neural representation networks. *arXiv preprint arXiv:2110.04124* (2021).
33. Maple, C. Geometric design and space planning using the marching squares and marching cube algorithms. In *International Conference on Geometric Modeling and Graphics*, 90–95 (2003).
34. Mortensen, E. N. & Barrett, W. A. Intelligent scissors for image composition. In *Annual Conference on Computer Graphics and Interactive Techniques*, 191–198 (1995).
35. Li, Y. *et al.* VBNet: An end-to-end 3D neural network for vessel bifurcation point detection in mesoscopic brain images. *Comput. Methods Programs Biomed.* **214**, 106567 (2022).
36. Bien, N. *et al.* Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet. *PLoS medicine* **15**, e1002699 (2018).
37. Hernandez-Matas, C. *et al.* FIRE: fundus image registration dataset. *Model. Artif. Intell. Ophthalmol.* **1**, 16–28 (2017).

38. Li, X. *et al.* Reinforcing neuron extraction and spike inference in calcium imaging using deep self-supervised denoising. *Nat. Methods* **18**, 1395–1400 (2021).

39. Wallace, G. K. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electron.* **38**, xviii–xxxiv (1992).

40. Lu, G. *et al.* DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11006–11015 (2019).

41. Müller, T. Tiny CUDA Neural Network Framework (2021). https://github.com/nvlabs/tiny-cuda-nn.

42. Trenn, S. Multilayer perceptrons: Approximation order and necessary number of hidden units. *IEEE Transactions on Neural Networks* **19**, 836–844 (2008).

43. Joseph, V., Gopalakrishnan, G. L., Muralidharan, S., Garland, M. & Garg, A. A programmable approach to neural network compression. *IEEE Micro* **40**, 17–25 (2020).

44. List, P., Joch, A., Lainema, J., Bjontegaard, G. & Karczewicz, M. Adaptive deblocking filter. *IEEE Transactions on Circuits Syst. for Video Technol.* **13**, 614–619 (2003).

Acknowledgements

We acknowledge F. Bao for the valuable comments on the manuscript. This work is jointly supported by Ministry of Science and Technology of China (Grant No. 2020AAA0108202), Beijing Natural Science Foundation (Grant No. Z200021) and the National Natural Science Foundation of China (Grant Nos. 61931012, 62088102).

Author contributions

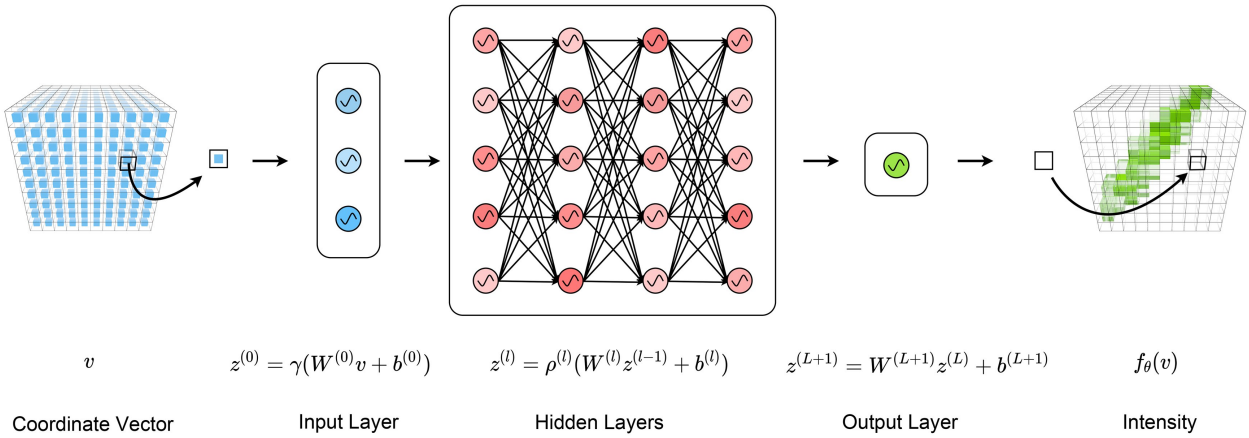
J. S., R. Y., T. X., and Y. C conceived this project. Q. D., Q. L., and J. S. supervised this research. R. Y. and T. X. designed the BRIEF architecture. T. X., R. Y., and Y. C. designed the adaptive partitioning and allocation algorithms. Y. C., T. X., A. L., and J. Q. conducted the experiments and data analysis. A. L., X. W., R. L., S. B., Y. C., and T. X. collected the biomedical data. All the authors participated in the writing of this paper.

Competing interests

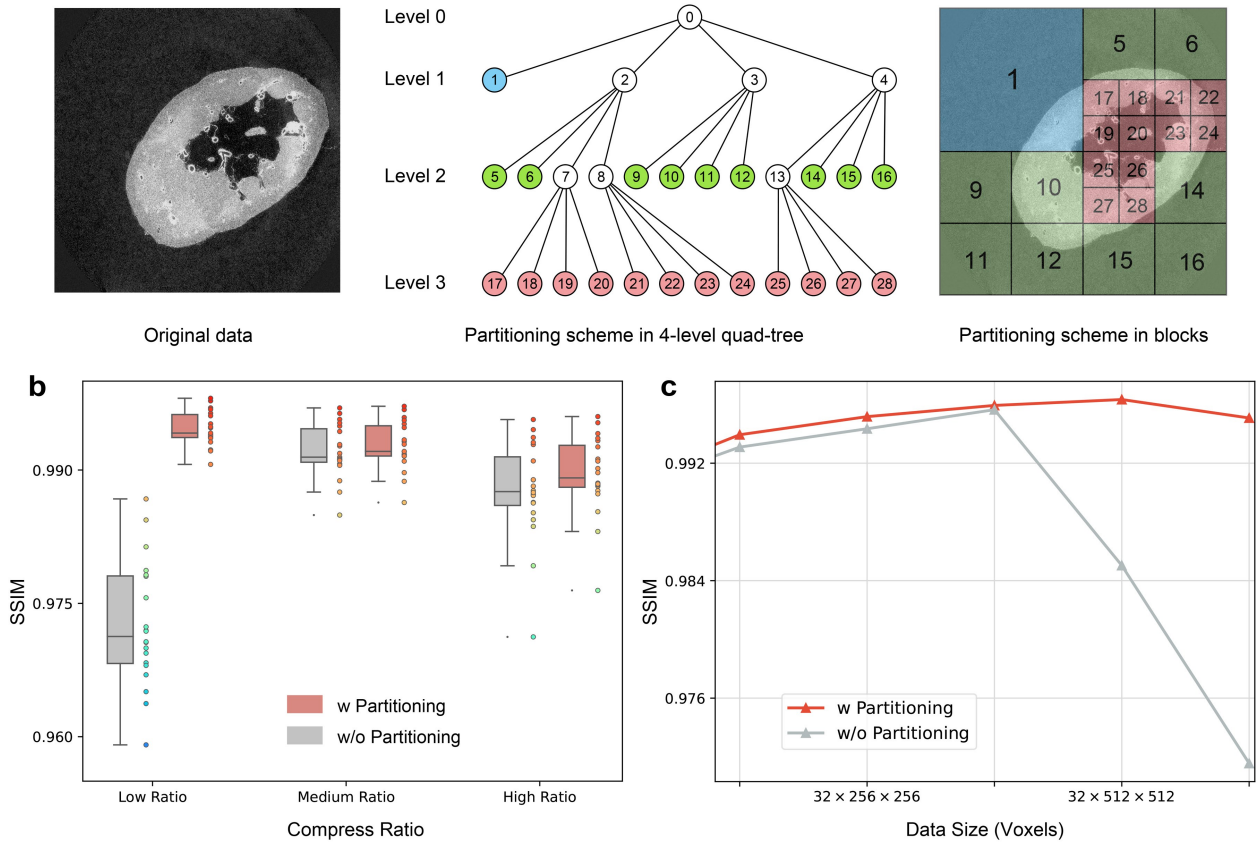
The authors declare no competing financial interests.

Materials and Correspondence

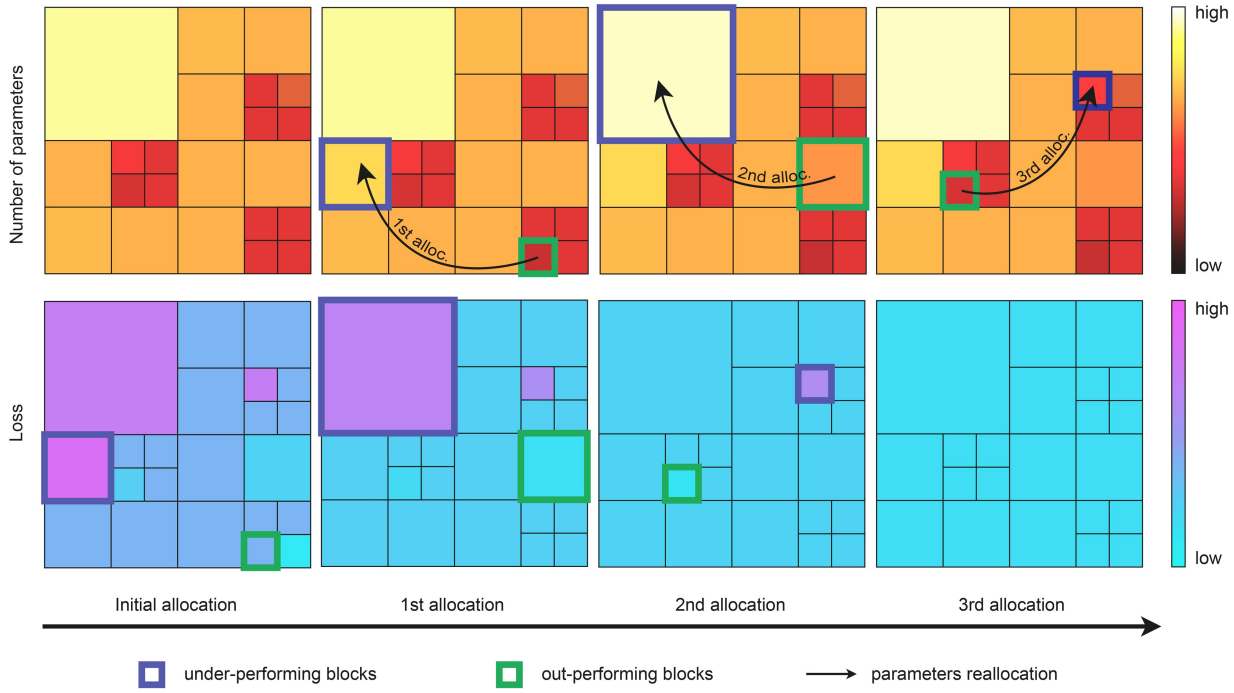
Correspondence and material requests should be addressed to Jinli Suo.



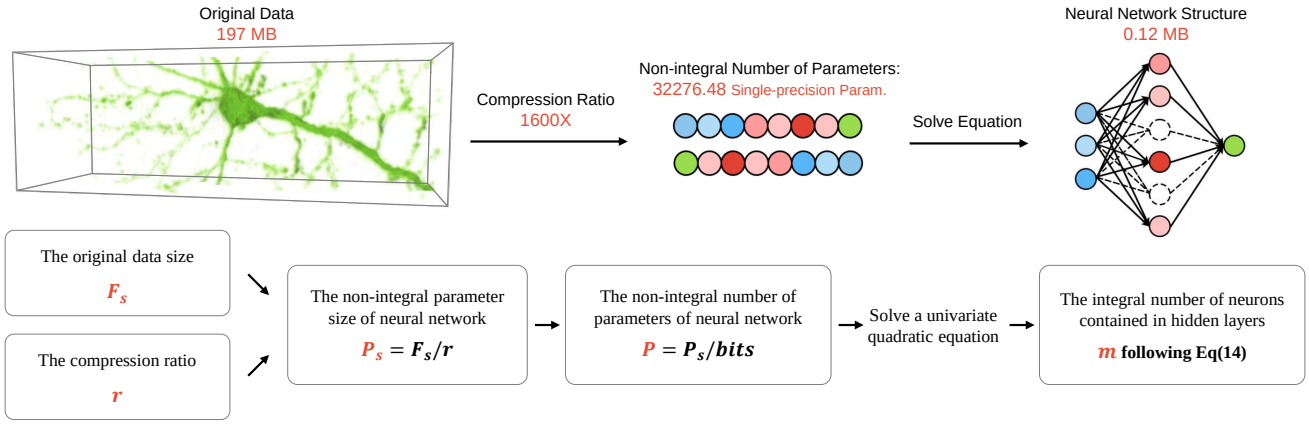
Extended Data Fig. 1. A visualization of our neural network architecture. We use a SIREN-like MLP as our neural network. The input v is the coordinate vector of each voxel in the original data. Then v is embedded into a higher dimensional space using a sinusoidal function γ at input layer. The hidden layers, a fully-connected neural network with activation function ρ, model the underlining data structure $z^{(L)}$ with the embedding $z^{(0)}$. The output layer finally maps $z^{(L)}$ into corresponding voxel's intensity, $f_\theta(v)$, written as the function of coordinate vector v. The numbers of neurons in input layer and output layer signify the coordinate's dimension and the number of channels of original data respectively. The number of neurons in hidden layers can be adjusted freely according to the need.



Extended Data Fig. 2. The adaptive partitioning. a, Schematic diagram of adaptive partitioning for a two-dimensional data (left) based on 4-level quad-tree. The optimal solution of Eq. (6) by ILP solver is shown both in tree-view (middle) and blocks-view(right). The leaf nodes with sufficiently high within-block correlation are filled with different colors according to their levels, while the non-leaf nodes line to their further partitioned nodes. The numbers and colors on the blocks (right) correspond to the ones on the nodes (middle). **b,c**, The ablation study of adaptive partitioning at different compression ratios and data sizes. We conduct experiments under three compression ratio (0.25, 0.13, 0.06 measured by BPV, respectively) and five data size ($16 \times 128 \times 128$, $32 \times 256 \times 256$, $32 \times 256 \times 512$, $32 \times 512 \times 512$, $64 \times 512 \times 512$, respectively) on tens of vessels data. The compression quality is measured by SSIM. Results show that partitioning strategy improves performance especially when compression ratio is relatively low and data size is relatively large, since the neural network representing the whole data (i.e., without partitioning) has too many parameters to be optimized.



Extended Data Fig. 3. The adaptive parameter allocation. BRIEF initially allocates the number of parameters according to each block's size and within-block correlation (top left). After certain steps of fitting, the fitting loss of each block appears to vary due to different fitting difficulties (bottom left). Then BRIEF reallocates the number of parameters from the out-performing blocks (highlighted in blue box) to under-performing blocks (highlighted in green box) and fits those blocks again. BRIEF repeats reallocation and fitting until certain times or the loss are evenly distributed. The numbers of parameters and fitting loss are color-coded.



Extended Data Fig. 4. Flow chart of regulating compression ratio. For a compression task, given the original data size F_s and compression ratio r, the non-integral parameter size of neural network can be calculated as $P_s = F_s/r$. And the non-integral number of parameters will be $P = P_s/bits$. The $bits$ of a single-precision parameter is 32 and of a half-precision one is 16. By solving the univariate quadratic equation in Eq. (13), BRIEF can derive the feasible integral number of neurons contained in hidden layers through the closed-form solution given in Eq. (14).