



OPEN

GPU-accelerated connectome discovery at scale

Varsha Sreenivasan¹✉, Sawan Kumar², Franco Pestilli³, Partha Talukdar^{2,4} and Devarajan Sridharan^{1,4}✉

Diffusion magnetic resonance imaging and tractography enable the estimation of anatomical connectivity in the human brain, in vivo. Yet, without ground-truth validation, different tractography algorithms can yield widely varying connectivity estimates. Although streamline pruning techniques mitigate this challenge, slow compute times preclude their use in big-data applications. We present 'Regularized, Accelerated, Linear Fascicle Evaluation' (ReAL-LiFE), a GPU-based implementation of a state-of-the-art streamline pruning algorithm (LiFE), which achieves >100× speedups over previous CPU-based implementations. Leveraging these speedups, we overcome key limitations with LiFE's algorithm to generate sparser and more accurate connectomes. We showcase ReAL-LiFE's ability to estimate connections with superlative test-retest reliability, while outperforming competing approaches. Moreover, we predicted inter-individual variations in multiple cognitive scores with ReAL-LiFE connectome features. We propose ReAL-LiFE as a timely tool, surpassing the state of the art, for accurate discovery of individualized brain connectomes at scale. Finally, our GPU-accelerated implementation of a popular non-negative least-squares optimization algorithm is widely applicable to many real-world problems.

Intact anatomical connectivity among brain areas is critical to cognition¹. Accurate estimation of anatomical connections in vivo is critical not only for uncovering the neural underpinnings of human behavior, but also for understanding the genetic bases of neurological disorders².

Diffusion magnetic resonance imaging (dMRI), followed by tractography, enables the estimation of anatomical connectivity in the human brain, in vivo³. dMRI measures the diffusion of water molecules in the brain's white matter, then tractography algorithms estimate axonal structures based on restricted diffusion, post hoc⁴. However, dMRI and tractography algorithms are prone to challenges such as acquisition noise and redundant fiber geometries. As part of an international tractography challenge, a recent study⁵ compiled efforts by 20 teams that estimated a whole-brain connectome from a simulated dMRI scan, which was, in turn, generated with simulated 'ground-truth' fiber bundles. The diverse success rates across the different teams—in terms of their ability to match the ground truth—underscores the magnitude of this challenge. Because actual ground-truth connectivity in the brain is typically unavailable in vivo, direct validation of tractography in the human brain remains elusive.

Streamline pruning and evaluation algorithms represent a state-of-the-art, post-processing approach to address these challenges^{3,6,7}. Linear Fascicle Evaluation (LiFE) is a recent, state-of-the-art model that prunes out spurious fibers based on the quality of fit to the underlying diffusion signal³. Yet, LiFE's algorithm is implemented

on central processing units (CPUs) and suffers from both speed and memory bottlenecks³, which preclude its application for connectome evaluation at scale.

In this Brief Communication, we present an improvement to LiFE—Regularized, Accelerated, Linear Fascicle Evaluation, or ReAL-LiFE—for rapid and accurate connectome evaluation at scale. We improve the LiFE algorithm by introducing an explicit regularization (sparsity) penalty into its objective function, and present a scalable graphics processing unit (GPU) implementation that routinely achieves orders of magnitude (>100×) speedups over CPU implementations while also estimating more sparse and more consistent connectomes. Next, we show that ReAL-LiFE performs at par with, and even outperforms, other state-of-the-art approaches (for example, SIFT^{2,5} and COMMIT^{2,8}) in terms of estimating streamlines with high test-retest reliability. Finally, we apply ReAL-LiFE to identify structural connectivity correlates of behavior in a cohort of 200 participants drawn from the Human Connectome Project (HCP) database⁹. We propose ReAL-LiFE as an effective tool, surpassing the state of the art, for rapid and accurate connectome discovery at scale.

We introduced a preliminary version of the ReAL-LiFE algorithm in an earlier study¹⁰ (Fig. 1a,b); this implementation achieved 50–100× speedups over CPU implementations of LiFE. In the present study, we optimize the algorithm further (Methods) to achieve even greater speedups (>100×, up to 155×; Methods). We demonstrate these speedups with three different diffusion MRI datasets.

We tested for speedups, first, with a state-of-the-art diffusion MRI dataset (dataset *H*; $N_v = 437,495$, $N_\theta = 270$) from the HCP database⁹. We generated connectomes of seven different sizes, ranging from 50,000 to two million fibers (Methods). The streamlines in these connectomes were then pruned with the CPU implementation as well as with our GPU implementation of LiFE (Methods). The GPU implementation produced substantial speedups, ranging from 62-fold (62×; 95% confidence interval (CI), [59.8, 63.4]) for a connectome with 50,000 fibers to 129-fold (129×; 95% CI, [128.8, 129]) for a connectome with 1.5 million fibers (Fig. 1c). We evaluated these speedups on two other, independently acquired datasets: a dMRI dataset acquired in house (dataset *I*; $N_v = 116,468$, $N_\theta = 64$; Methods) and a dataset used in the original LiFE study³ (dataset *S*; $N_v = 247,969$, $N_\theta = 96$; Methods). Again, we observed maximum speedups of 124× (95% CI, [123.6, 124.4]; dataset *I*) and 155× (95% CI, [155.1, 155.2]; dataset *S*) for connectomes with 1.5 million fibers (Fig. 1c).

We also tested how speedups scaled with the number of voxels (N_v) and number of diffusion directions (N_θ). Overall, speedups scaled fastest with the number of diffusion directions, followed by the number of voxels (Supplementary Fig. 1). We also compared convergence times for ReAL-LiFE with two other pruning algorithms, SIFT⁷ and COMMIT^{2,8}. ReAL-LiFE performed comparably

¹Centre for Neuroscience, Indian Institute of Science, Bangalore, India. ²Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India. ³Department of Psychology, University of Texas at Austin, Austin, TX, USA. ⁴Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India. ✉e-mail: varshas@iisc.ac.in; sridhar@iisc.ac.in

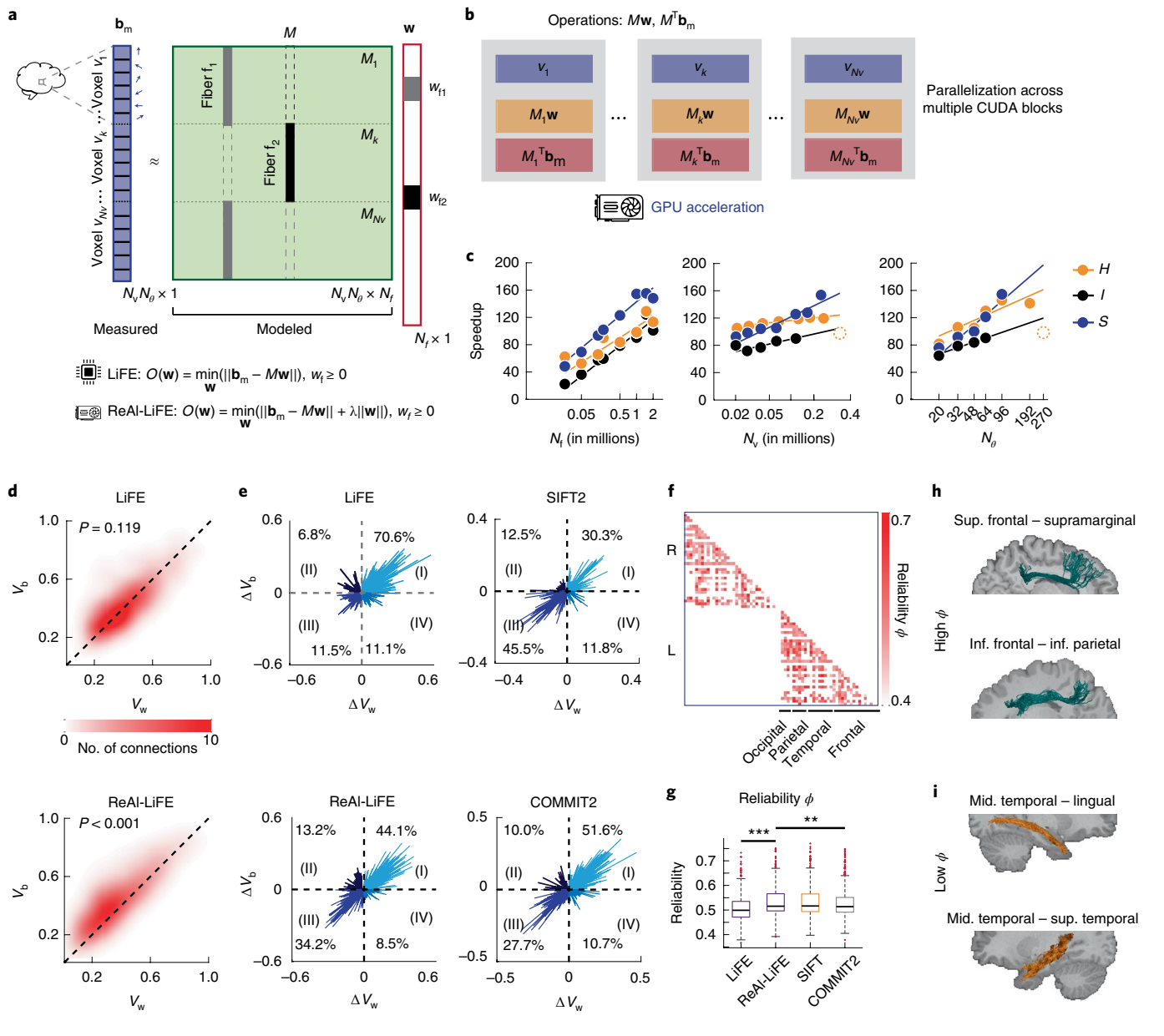


Fig. 1 | Rapid and reliable connectome evaluation with the ReAI-LiFE algorithm. **a**, Schematic of the LiFE model. $O(\mathbf{w})$, objective function value. **b**, Schematic of the GPU-accelerated LiFE algorithm. **c**, Left: speedups as a function of the number of fibers N_f for datasets H (yellow), I (black) and S (blue). The x axes are shown in log scale. Middle: speedups as a function of the number of voxels N_v in each diffusion MRI volume. Right: speedups as a function of the number of gradient directions N_θ for each diffusion MRI scan. Filled circles, connectome sizes tested. Solid lines, logarithmic fit. Dashed open circles indicate points excluded from the logarithmic fit. **d**, Heatmaps showing within-participant variability (V_w) against between-participant variability (V_b) across $n=561$ connections from five participants' test-retest data, following pruning with the LiFE algorithm (top) and the ReAI-LiFE algorithm (bottom). Dashed diagonal line, line of equality. One-sided Wilcoxon signed-rank test, $P=0.119$ (top), $***P<0.001$ (bottom). **e**, Vector plots demonstrating the improvement in reliability, relative to the unpruned connectome, following pruning with each of four different methods. Each plot shows change in within-participant variability (ΔV_w) against change in between-participant variability (ΔV_b) following pruning with LiFE (top left), ReAI-LiFE (bottom left), SIFT2 (top right) and COMMIT2 (bottom right). Each vector corresponds to one of the 561 connections evaluated. **f**, Reliability matrix, whose (i,j) th entry indicates the reliability measure ($\phi = V_b / (V_b + V_w)$) of each pair of ($n=561$) intra-hemispheric connections between regions i and j . Deeper shades indicate connections with higher test-retest reliability. R, right; L, left. **g**, Average reliability (ϕ) across all connections ($n=561$) after pruning with LiFE, ReAI-LiFE, SIFT2 and COMMIT2. Center line, median; box limits, upper and lower quartiles; whiskers, $1.5\times$ interquartile range; points, outliers. One-sided Wilcoxon signed-rank test: LiFE versus ReAI-LiFE, $***P<0.001$; ReAI-LiFE versus COMMIT2, $**P<0.01$. **h**, Exemplar tracts corresponding to two connections with high consistency. **i**, Exemplar tracts corresponding to two connections with low consistency.

to SIFT and COMMIT2, with a slight advantage for our approach, especially at larger connectome sizes (Supplementary Fig. 1 and Supplementary Information).

Incorporating a sparsity-inducing prior (L1-norm of fiber weights; Methods) enabled ReAI-LiFE to generate sparser and more

accurate connectomes. Yet, previous studies have indicated that such a sparsity-inducing prior may increase the chances of false negatives (missed fibers), particularly when duplicate fibers occur in the connectome¹¹. We addressed this challenge precisely by constructing an artificial connectome comprised entirely of near-identical, duplicate

streamlines, and show that pruning with ReAL-LiFE largely ameliorated this challenge (Supplementary Fig. 2). In addition, we show that ReAL-LiFE reduced overfitting and produced more consistent connectomes, as compared to LiFE (Supplementary Figs. 3 and 4).

We next quantified the reliability of the ReAL-LiFE algorithm, comparing it with that of LiFE, SIFT2⁶ and COMMIT2⁸. We performed a test–retest reliability analysis¹² using a sample of $n=5$ participants drawn from the HCP database (Methods). For each participant, pairwise intra-hemispheric connectivity was computed, the total variability of which was partitioned into between-participant (V_b) and within-participant (V_w) components (Methods).

Following pruning with LiFE, within- and between-participant variabilities were comparable (V_b , CI [0.411, 0.435]; V_w , CI [0.407, 0.432]; $P=0.119$; effect size = 0.025; Fig. 1d), but, following ReAL-LiFE pruning, within-participant variability was significantly lower than between-participant variability (V_b , CI [0.374, 0.401]; V_w , CI [0.337, 0.365]; $P<0.001$, effect size = 0.217; Fig. 1d). Thus, ReAL-LiFE pruning increased test–retest reliability of the estimated connection weights. Moreover, between-participant variability following ReAL-LiFE pruning was significantly lower than that following LiFE pruning ($P<0.001$).

We further quantified the test–retest reliability with a reliability index (ϕ ; Methods). A greater reliability index is a hallmark of efficient pruning. Compared with LiFE, pruning with ReAL-LiFE yielded a significantly higher reliability index (Fig. 1g; LiFE: $\phi=0.507$, CI [0.503, 0.512], ReAL-LiFE: $\phi=0.533$, CI [0.528, 0.538]; $P<0.001$ across all connections). Pruning with SIFT2 yielded reliability indices comparable with those of ReAL-LiFE (Fig. 1g; SIFT2, $\phi=0.535$, CI [0.530, 0.540]; $P=0.812$), whereas pruning with COMMIT2 yielded a marginally lower reliability index than ReAL-LiFE (Fig. 1g; $\phi=0.526$, CI [0.521, 0.531]; $P=0.004$).

Next we quantified the improvement in reliability based on the proportion of connections in each quadrant of the (V_b , V_w) plot: a greater proportion in the II quadrant ($V_b > V_w$; ‘high-reliability’) as compared to the IV quadrant ($V_w > V_b$; ‘low-reliability’) indicates more efficient pruning. Compared to the unpruned connectome, streamline pruning with LiFE yielded fewer connections in the high-reliability (6.8%) than in the low-reliability quadrant (11.1%), although not different from chance ($P=0.994$, binomial test; Fig. 1e). On the other hand, pruning with ReAL-LiFE yielded significantly more connections in the high-reliability (13.2%) than in the low-reliability quadrant (8.5%; $P=0.011$; Fig. 1e). Pruning with SIFT2 and COMMIT2 yielded comparable proportions of connections in both the high- and low-reliability quadrants (SIFT2, $P=0.399$; COMMIT2, $P=0.390$; Fig. 1e).

Finally, we identified connections exhibiting extreme values of test–retest reliability following ReAL-LiFE pruning (Fig. 1f). The highest test–retest reliability was observed for long-range connections between the frontal and parietal lobes (Supplementary Table 3), which strongly overlap with established white-matter tracts including the superior and inferior longitudinal fasciculus (SLF/ILF), the arcuate fasciculus (AF) and the inferior fronto-occipital fasciculus (IFOF) (Fig. 1h). Conversely, test–retest reliability was least for short-range connections, especially those connecting the middle temporal gyrus with adjacent temporo-occipital regions (Fig. 1i).

As a real-world application of ReAL-LiFE, we asked if streamline pruning with ReAL-LiFE would enable identification of structural connectivity correlates of behavior. For this, we predicted 60 behavioral test scores¹³ spanning three categories—cognition, emotion and personality—of 200 participants (HCP database⁹; Supplementary Data File 1). Behavioral score prediction was performed with a support vector regression (SVR) model using recursive feature elimination (RFE)¹⁴ (Methods and Fig. 2a). Specifically, we compared predictions made with ReAL-LiFE connection weights as features against those based on the number of fibers in the unpruned connectome.

Across a range of significance thresholds ($\alpha=0.00001$ to 0.05, uncorrected), the number of behavioral scores predicted significantly by connectome features based on both the number of fibers (Fig. 2b, top, red filled circles) and ReAL-LiFE weights (Fig. 2b, top, purple filled circles) were not different ($P=0.906$; effect size = 0.011). However, predictions based on ReAL-LiFE weights were more accurate than those based on the number of fibers, as evidenced by the higher average correlations (Fig. 2b, bottom; effect size = 0.660). Similar trends were observed when each category of scores—cognition, emotion and personality—was predicted separately (Fig. 2c–f and Supplementary Fig. 5). Specifically, for cognition and personality scores, ReAL-LiFE weights yielded consistently higher prediction accuracies (average r values) than the number of fibers, across the entire range of significance thresholds (Fig. 2c,e, lower panels).

Next we asked which set of connectome features—ReAL-LiFE weights or unpruned connectome fibers—more robustly predicted behavioral scores. For this, we combined both sets of features and quantified the proportion of ReAL-LiFE features selected by the RFE algorithm for the best prediction of each behavioral score. For over 95% (58/60) of score predictions, RFE favored a higher proportion of ReAL-LiFE features, as compared to features from the unpruned connectome ($P<0.001$, Wilcoxon signed-rank test; Fig. 2g). Additional control analyses for these behavior predictions are reported in the Supplementary Information (Supplementary Fig. 5).

We also analyzed the anatomical features underlying these predictions, focusing on the ‘cognition’ scores. Briefly, reading ability correlated significantly with connectivity in the left frontal cortex, whereas picture vocabulary scores correlated significantly with connectivity within the right parietal cortex (Supplementary Fig. 5 and Supplementary Data File 2).

Our findings indicate that, in addition to yielding more accurate connectomes, ReAL-LiFE connection weights enabled more accurate predictions across a range of behavioral scores, all acquired outside the scanner environment. dMRI-based structural connectivity, quantified following connectome evaluation, may provide a reliable neuroimaging-based biomarker for key cognitive traits.

The ReAL-LiFE algorithm may be developed and improved on several key fronts to overcome its current limitations. First, the current version of the ReAL-LiFE algorithm does not take advantage of parallel computations across multiple GPUs. Moreover, ReAL-LiFE is presently not integrated with multi-CPU acceleration schemes¹⁰, although our speedups exceed (~8.7 \times) state-of-the-art numbers reported for these approaches. Combining these CPU-based schemes with our GPU implementation, or implementing parallel computations across multiple GPUs, may yield further speedups of the algorithm.

Second, ReAL-LiFE’s optimization objective, including the sparsity-inducing prior, may be further improved. A key feature of regularized pruning with ReAL-LiFE is the ability to generate connectomes at various, desired levels of sparsity using L1-norm-based regularization, a feature unavailable in the original LiFE algorithm. Yet, such a stringent regularization increases the chances of false negatives (missed fibers)^{8,11}. Although we tested for this possibility (Supplementary Fig. 2), other kinds of regularization (for example, L2-norm-based) will need to be systematically evaluated to identify those that minimize false negatives in the connectome. Nonetheless, incorporating this penalty resulted in ReAL-LiFE outperforming LiFE in terms of reducing overfitting, producing more consistent connectomes and increasing test–retest reliability. Although the test–retest reliability analyses were performed with only a limited number of participants ($n=5$), ReAL-LiFE connections with the highest test–retest reliability mapped onto established white matter tracts¹⁵. Incorporating additional features and constraints into the objective function—for example, based on brain anatomy, as with COMMIT2⁸—may enable future improvements to the ReAL-LiFE algorithm.

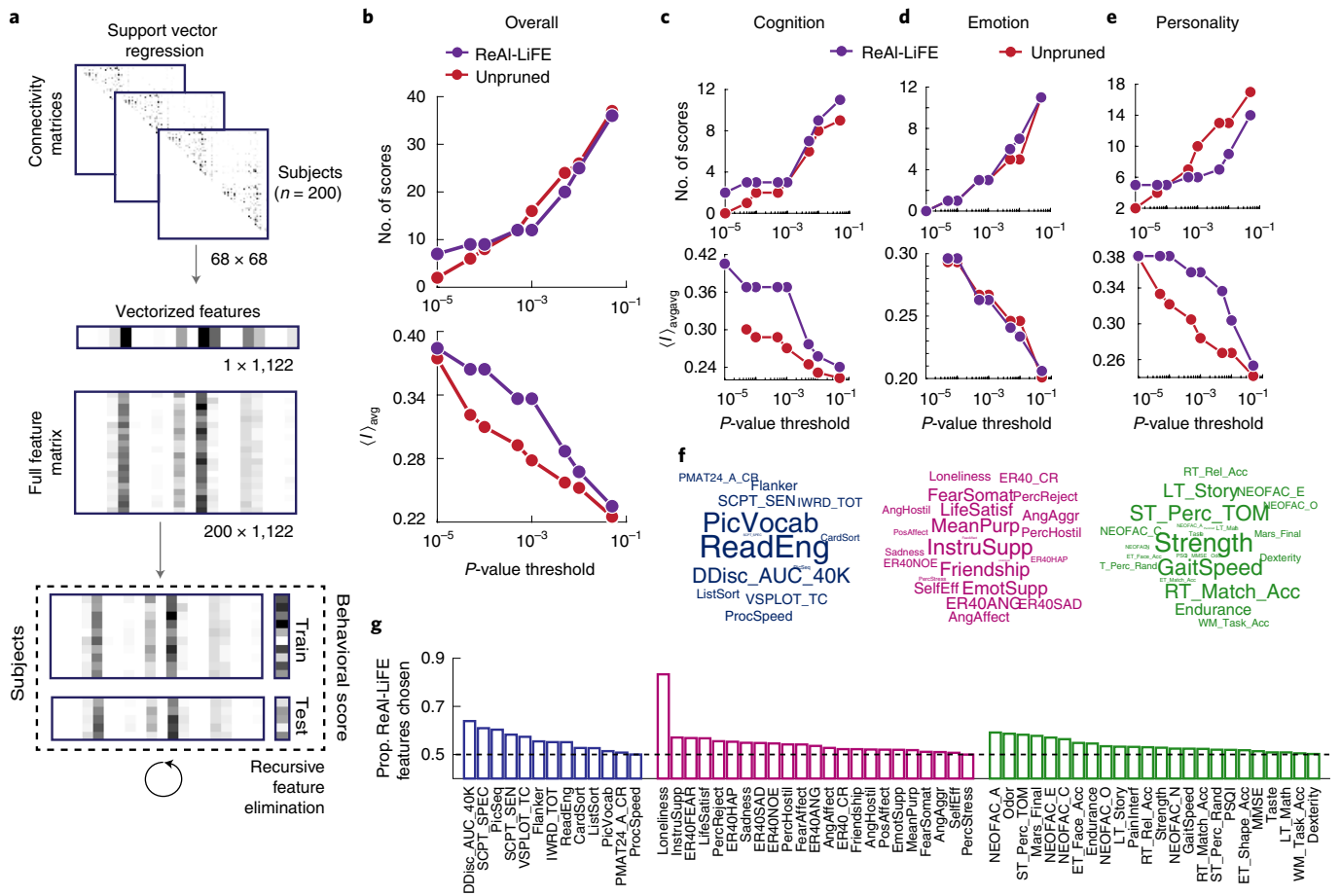


Fig. 2 | Predicting key cognitive scores using ReAl-LiFE connection weights. **a**, Schematic of the SVR-RFE prediction model. For each participant ($n = 200$), the 68×68 whole-brain connectivity matrix was vectorized to give 1,122 connectivity features. Feature vectors from all participants were collated to form a feature matrix of size $200 \times 1,122$, which was used to predict 60 different behavioral and cognitive scores. Data were divided into training and testing folds, and the prediction model was trained on the train fold using SVR (dashed box). Feature selection was implemented using RFE. **b**, Top: number of scores significantly predicted as a function of the uncorrected P -value threshold for predictions based on the number of fibers in the unpruned connectome (red circles) and connection weights in the ReAl-LiFE-pruned connectome (purple circles). Bottom: average correlations between the observed and predicted scores as a function of the uncorrected P -value threshold. Other conventions are the same as in the top panel. **c-e**, As in **b**, but for scores from the cognition (**c**), emotion (**d**) and personality (**e**) categories. Other conventions are the same as in **b**. **f**, Word clouds showing the different behavioral scores from the cognition (left), emotion (middle) and personality (right) categories, sized based on their prediction accuracy values using ReAl-LiFE connection weights. Larger words indicate better predicted scores. **g**, Proportion of ReAl-LiFE features chosen by the SVR-RFE model for predicting each of the 60 behavioral and cognitive scores using a combined feature set, including both the number of fibers in the unpruned connectome (1,122 features), as well as ReAl-LiFE connection weights (1,122 features). The proportions of features are shown in descending order, separately, for each of the three categories of scores—cognition (blue, $n = 13$), emotion (magenta, $n = 23$) and personality (green, $n = 24$).

Third, a principled evaluation of the reasons underlying the differences between ReAl-LiFE and other state-of-the-art algorithms—SIFT/SIFT2 and COMMIT2⁶⁻⁸—remains to be carried out. ReAl-LiFE pruning times were largely comparable with the other approaches, but demonstrated a marginal advantage for larger connectome sizes. Moreover, ReAl-LiFE differed from the other approaches in the proportion of high-reliability connections retained following pruning. The reasons for these differences need to be investigated carefully. Finally, ReAl-LiFE will need to be directly compared against these competing approaches, in terms of their respective success rates with mapping structural connectivity to behavior. Such a principled comparison is essential for identifying robust structural connectivity bases of higher-order cognitive functions, such as attention, learning and decision-making^{15,16}.

More generally, the Subspace Barzilai–Borwein Non-Negative Least-Squares (SBB-NNLS) algorithm, at the heart of ReAl-LiFE, is widely applicable to optimization problems in many real-world

applications, including healthcare¹⁷. Our GPU-accelerated implementation of the SBB-NNLS algorithm has the potential for wide application in diverse domains that go beyond connectome pruning.

Methods

All experiments were conducted according to protocols approved by the Institute Human Ethics Committee, Indian Institute of Science, Bangalore. Informed written consent was obtained from each participant before the study.

ReAl-LiFE. Description of the LiFE algorithm. The LiFE algorithm models a diffusion signal from the reconstructed whole-brain connectome. LiFE’s optimization algorithm minimizes the error between the modeled and the measured diffusion signal, while eliminating fibers from the connectome that do not contribute to the underlying diffusion signal. The diffusion signal is typically measured along multiple, uniformly sampled gradient directions in space, N_g . For each voxel and gradient direction, this signal is then encoded into a vector $\mathbf{b} \in \mathbb{R}^{N_g \times N_v}$, where N_v is the number of voxels in the data. In a given whole-brain connectome, fibers traverse multiple voxels and each voxel may contain many different fibers. The contribution of each fiber f , traversing a voxel v , measured

along the gradient direction θ , is encoded into a matrix $M \in \mathbb{R}^{N_v \times N_f \times N_f}$, where N_f is the number of fibers in the connectome. The diffusion signal in each voxel is modeled as a weighted sum of the individual fibers traversing the voxel. This can be written as $\mathbf{b} = M\mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^{N_f}$ signifies the contribution (or weight, w_i) of each fiber f to the diffusion signal \mathbf{b} .

LiFE minimizes the error between the modeled and measured diffusion signal by assigning a non-negative weight to each fiber. This objective is posed as a non-negative least-squares optimization problem:

$$\min_{\mathbf{w}} (O(\mathbf{w})), \quad O(\mathbf{w}) = \frac{1}{2} \|\mathbf{b} - M\mathbf{w}\|^2, \quad \mathbf{w} \geq 0 \quad (1)$$

Solving this problem imposes substantial memory demands due to the large size of M , which is typically 2 GB for a typical connectome with $N_v = 100,000$, $N_f = 1$ million and $N_\theta = 64$ (refs. 3,18). A recent study¹⁸ overcame this limitation by adopting a more efficient, sparse tensorial representation of M . The demeaned diffusion signal $M_v \in \mathbb{R}^{N_\theta \times N_f}$ in each voxel v was represented, using sparse Tucker decomposition (STD), as $M_v = S_0(v)D\Phi_v$, where $S_0(v)$ is the 'baseline' diffusion signal measured in the absence of a diffusion gradient, $D \in \mathbb{R}^{N_\theta \times N_a}$ is a dictionary matrix quantifying the contribution of canonical diffusion 'atoms' (N_a atoms) toward each diffusion direction, and $\Phi_v = \mathbb{R}^{N_a \times N_f}$ is a sparse, binary matrix whose columns indicate the contribution of each atom to each fiber, in that voxel. Collating Φ_v for all voxels v into a sparse 3D tensor Φ , the modeled (or predicted) diffusion signal may be written as

$$Y = \Phi \times_1 D \times_2 S_0 \times_3 \mathbf{w}^T \quad (2)$$

where \mathbf{w} is the vector of all the individual fiber weights and \times_k represents a matrix product in mode k . M in equation (1) is now given by $M = \Phi \times_1 D \times_2 S_0$.

With this representation, the optimization problem is solved using an SBB-NNLS algorithm¹⁰. Briefly, given \mathbf{w}^0 as an initial weight vector, the weight updates occur as follows:

$$\mathbf{w}^{(i+1)} = [\mathbf{w}^{(i)} - \alpha^{(i)} \nabla g(\mathbf{w}^{(i)})]^+$$

where the gradient term is given by

$$\nabla g(\mathbf{w}) = M^T(M\mathbf{w} - \mathbf{b}) \quad (3)$$

and $\alpha^{(i)}$, the step value at each iteration, is given by

$$\alpha_{\text{odd}}^{(i)} = \frac{\langle \nabla \tilde{g}^{(i-1)}, \nabla \tilde{g}^{(i-1)} \rangle}{\langle M \nabla \tilde{g}^{(i-1)}, M \nabla \tilde{g}^{(i-1)} \rangle}$$

for the odd iterations and

$$\alpha_{\text{even}}^{(i)} = \frac{\langle M \nabla \tilde{g}^{(i-1)}, M \nabla \tilde{g}^{(i-1)} \rangle}{\langle M^T M \nabla \tilde{g}^{(i-1)}, M^T M \nabla \tilde{g}^{(i-1)} \rangle}$$

for the even iterations. The tilde denotes the projection of the gradient into the positive space at each iteration and $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes an inner product of the vectors \mathbf{a} and \mathbf{b} .

In its original form, the LiFE algorithm suffered from a few key drawbacks, which needed to be overcome to enable connectome evaluation at scale. First, a CPU implementation of the memory-efficient implementation of the LiFE algorithm still suffered from computational bottlenecks, because every iteration of the algorithm requires a large number ($O(N_v N_f N_\theta)$) of multiplications, typically, with matrices comprising 10^{13} – 10^{14} elements. This requirement produced a considerable speed bottleneck when evaluating large connectomes. Second, LiFE (and related pruning algorithms) suffer from another key limitation: there is no explicit provision in LiFE or in other popular pruning algorithms, such as SIFT⁷ or SIFT2⁶, for directly eliminating redundant fibers in the connectome.

GPU acceleration. To address the issue of speed, we sought to identify key bottlenecks in LiFE's algorithm that could be optimized on GPUs. Briefly, the SBB-NNLS optimization algorithm requires several multiplications of the form Mx or $M^T y$, where x and y are generic notations of matrices that are used in various steps of the optimization. We sought to speed up these multiplications with efficient GPU-based computations. The detailed steps are presented as pseudocode in algorithms 1 and 2 in the Supplementary Information. Here we describe these steps briefly.

A key ingredient of our GPU acceleration approach is splitting the computation among voxels, with each CUDA block handling data associated with one voxel. For storage efficiency, the matrix M was stored in a sparse tensor (Coordinate list, COO format) with indices into the dictionary matrix D , as it was not feasible to use standard sparse matrix multiplication packages. Following STD of M (equation (2)), computing Mx requires computing linear combinations of columns from D while $M^T y$ computation requires computation of inner products with columns of D (ref. 18). The former has a high memory write bandwidth requirement, whereas the latter has a high memory read bandwidth requirement as well as a reduction operation. To address this issue we sorted the Φ tensor, stored in the COO format,

along the voxel dimension, enabling faster per-voxel execution of both Mx and $M^T y$ by reducing memory write and read requests, respectively.

We fixed the block size to the warp size of the GPU, which corresponds to the number of threads processing each voxel. Each thread handled one or more diffusion directions, depending on the total number of diffusion directions. Data along the diffusion direction dimension were padded such that its size was a multiple of the warp size, to avoid branching in the kernel code that is to be run on the GPUs. This also permitted maximizing the usage of warp shuffle instructions and reducing shared memory usage. We used shared memory only for storing the final results. In each block, we read up to warp size entries from the sparse tensor in parallel, to leverage memory coalescing advantages, and stored them in thread local memory. The threads in a block then computed on different diffusion directions for the read entries sequentially. We used warp broadcast instructions to share data from thread local memories to all threads in a block. In the case of $M^T y$ computation, we used warp shuffle instructions for computing inner products. This freed up resources, potentially allowing more blocks to be scheduled at any given time.

We implemented these algorithms (algorithms 1 and 2 in the Supplementary Information) with the CUDA language for use with NVIDIA GPUs. The results reported in the main text reflect speedups with the CUDA implementation. In addition, we implemented these same algorithms on AMD GPUs with the HIP (Heterogenous Compute Interface for Portability) language, using the HIPIFY package (<https://github.com/ROCm-Developer-Tools/HIPIFY>). We found that execution times for algorithms 1 and 2 on the NVIDIA GeForce GTX 1080 Ti GPU and the AMD Radeon RX 580 GPU are comparable, and several orders of magnitude faster than their corresponding CPU implementations in Matlab (Supplementary Fig. 1).

Regularized evaluation. To address the second issue of redundant fibers, we developed a regularized pruning algorithm, extending that of LiFE¹⁰. We modified LiFE's least-squares error minimization objective function to incorporate a regularization term for the weights, such that an L1-norm penalty was incorporated into the objective function: $O(\mathbf{w}) + \lambda (\|\mathbf{w}\|_1)$, $\mathbf{w} \geq 0$, where λ is the regularization constant. The gradient calculation in equation (3) now changes to $g(\mathbf{w}) = M^T(M\mathbf{w} - \mathbf{b}) + \lambda \mathbf{1}$ where $\mathbf{1}$ is a vector of all 1s. Similarly, we also implemented L2 regularization for the weights, by adding an L2-norm penalty to the objective function: $O(\mathbf{w}) + \lambda (\|\mathbf{w}\|_2)$, $\mathbf{w} \geq 0$. The gradient calculation in equation (3) now changes to $g(\mathbf{w}) = M^T(M\mathbf{w} - \mathbf{b}) + \lambda \mathbf{w}$. We tested several values of the penalty λ , for both the L1 and L2 regularization (Supplementary Fig. 3).

The estimated fiber weights vector \mathbf{w} is encouraged to be sparse in LiFE through the non-negativity constraint. Additionally, in ReAL-LiFE, more sparsity is induced, through regularization, on the weights vector.

Although a preliminary version of the method has been published previously¹⁰, in the present study we advance the ReAL-LiFE algorithm by leveraging the sparsity of \mathbf{w} to further speed up the GPU implementation of Mx . Specifically, whenever a weight vector element x is zero, the entire matrix multiplication computation for that step can be skipped, in each thread (step 5 in Supplementary algorithm 1). This yielded a substantial improvement in speedups for the present ReAL-LiFE algorithm (>100–150× for the largest connectome sizes tested) over the version of the algorithm published previously¹⁰ (~50–100×).

Diffusion MRI acquisition and preprocessing. Dataset I. Structural and dMRI scans were acquired on a Siemens Skyra, 3T scanner with a 32-channel head coil, at the HealthCare Global Hospital, Bangalore. A T1-weighted MPRAGE structural scan was acquired before the diffusion scan (1-mm spatial resolution; echo time (TE) = 2.32 ms, repetition time (TR) = 2,300 ms, field of view (FoV) = 240 mm, flip angle = 8°, 256-voxel matrix size, parallel acquisition technique (PAT) with in-plane acceleration factor 2 (GRAPPA)). Diffusion scans were acquired along 64 non-collinear directions with a b value of 1,000 s mm^{-2} (2-mm isotropic voxels, TE = 90 ms, TR = 8,900 ms, FoV = 256 mm, 128-voxel matrix size, 68 transversal slices with interleaved slice acquisition; parallel acquisition technique (PAT) with in-plane acceleration factor 2 (GRAPPA), phase encoding direction: A>>P). Two non-diffusion weighted images ($b = 0 \text{ s mm}^{-2}$) were acquired, one at the beginning and one at the end of each scan, respectively. Preprocessing of dMRI images followed previously published protocols³. Briefly, the T1 image was first manually aligned to the participant's anterior commissure–posterior commissure (AC–PC) axis coordinates. Following this, scans were preprocessed to correct for head motion, eddy current-related distortions were corrected using a rigid-body alignment algorithm³, followed by alignment to the AC-PC aligned T1 image using the VISTA LAB (Stanford Vision and Imaging Science and Technology) diffusion MRI software package, as part of the Vistasoft suite 2017 (<https://github.com/vistalab/vistasoft/>). This dataset is available on a Figshare repository¹⁹.

Dataset S. We acquired a publicly available, preprocessed dataset as used in the evaluation of the LiFE algorithm³. Data were acquired on a General Electric Discover 750 (GE Healthcare), 3T scanner with a 32-channel head coil. Diffusion scans were acquired along 96 non-collinear directions with a b value of 2,000 s mm^{-2} (1.5-mm isotropic voxels, TE = 96.8 ms). Ten non-diffusion weighted images ($b = 0 \text{ s mm}^{-2}$) were acquired at the beginning of the scan. Preprocessing steps included correcting distortions arising out of B0 field inhomogeneities as well as participant head motion

correction using a rigid-body alignment method³. Further details regarding the acquisition and preprocessing steps are available in refs. ³ and ¹⁸.

Dataset H. We acquired a publicly available dataset from the HCP database⁹. Structural and dMRI scans were acquired on a customized Siemens Connectome Skyra, 3T scanner with a 32-channel head coil. The T1-weighted MPRAGE structural scan was acquired at 0.7-mm spatial resolution, TE = 2.14 ms, TR = 2,400 ms, FoV = 224 × 224 mm², flip angle = 8°, PAT with in-plane acceleration factor 2. Diffusion scans were acquired along 270 non-collinear directions using multi-shell imaging with *b* values of 1,000, 2,000 and 3,000 s mm⁻² (1.25 mm isotropic voxels, TE = 89.5 ms, TR = 5,520 ms, FoV = 210 × 180 mm², 164 × 144 matrix size, multiband-factor = 3, phase encoding direction: L>>R and R>>L). A total of 18 non-diffusion weighted images (*b* = 0 s mm⁻²) were acquired, interspersed throughout the scan. Preprocessing steps included B0 intensity normalization, susceptibility-induced distortion correction, eddy current and participant head motion correction, and gradient nonlinearity correction. Finally, the diffusion images were registered to the structural (T1-weighted) image. For all our analyses, we used HCP's minimally preprocessed data⁹.

Dataset M. This dataset was provided as part of an international Tractography Challenge⁵. Briefly, the authors used one dataset from the HCP database⁹ to manually delineate 25 known fiber bundles and their corresponding termini regions (regions of interest, or ROIs) in the human brain⁹; these were termed 'ground-truth' bundles. Next, using these ground-truth bundles, diffusion MRI data were simulated corresponding to a *b* value of 1,000 s mm⁻² and 32 gradient directions using the Fiberfox software. A T1-weighted image was also simulated. The authors provide two datasets—one artifact-free dataset with no noise and one dataset with added artifacts such as head motion, susceptibility-induced distortion, eddy currents, spiking noise, ghosting and ringing artifacts, as well as Gaussian noise.

In addition to the dataset already provided by the authors, we simulated a second, independent dataset with the same parameters, using Fiberfox⁵ version 2018.09.99. Briefly, we generated one diffusion MRI dataset with *b* = 1,000 s mm⁻² and 32 gradient directions (2-mm isotropic voxels) in the A>>P (anterior to posterior) phase encoding direction. For each dataset, we used a four-compartment model to simulate the (1) inter-axonal, (2) intra-axonal, (3) gray matter (GM) and (4) cerebrospinal fluid (CSF) tissue response profiles. Parameter values corresponding to each compartment model are listed in Supplementary Table 1. All other parameters were set to their respective, default values. We also simulated an additional non-diffusion weighted image (*b* = 0 s mm⁻²) with the phase encoding direction reversed (P>>A). Individual masks used for each compartment as well as the anatomical T1 image were the same as those provided in the original dataset. Preprocessing the simulated data involved denoising with MRtrix3²⁰ followed by motion correction and susceptibility-induced distortion correction²⁰. Finally, the diffusion MRI scan was aligned to the T1 image.

HCP datasets for studying structure–behavior relationships. For predicting behavioral scores using structural connectivity, we used minimally preprocessed diffusion MRI datasets from the HCP database⁹. We utilized data from *n* = 200 participants with an equal number of males and females (*n* = 100 each) for whom both dMRI data as well as behavior data was available (Supplementary Table 2; participant IDs). These participants included 60 participants who were matched for age, gender and handedness from a previous study²¹. The remaining 140 participants were drawn in chronological order from the HCP database to ensure gender and age parity (*n* = 100 females: mean = 29.2 years, s.d. = 3.7 years; *n* = 100 males: mean = 28.5 years, s.d. = 3.9 years). We confirmed that there was no significant difference between the average age of male and female participants (*P* = 0.223, two-sample *t*-test).

Tractography and generating whole-brain connectomes from dMRI data.

Datasets I, S, H and M. For each of these datasets, we used a standard tractography pipeline available with MRtrix3²⁰. This pipeline comprises the following steps. We first performed a five-tissue-type segmentation on the T1 image to separate out the (1) cortical GM, (2) subcortical GM, (3) white matter, (4) CSF and (5) other pathological tissue. Next, a constrained spherical deconvolution (CSD) algorithm was employed to estimate the fiber orientation distribution (FOD) in each voxel²⁰, with the maximum harmonic order at 8 (default value). Finally, anatomically constrained probabilistic tractography was performed using dynamic seeding⁷. The maximum fiber length cutoff and FOD amplitude threshold for datasets I, S and M were set to default values (200 mm and 0.1, respectively). For dataset H, the maximum length cutoff and FOD amplitude threshold were set to 250 mm and 0.06, respectively. Following this, we constructed whole-brain connectomes with specific fiber counts, as indicated in the respective sections in the main text. For the behavioral score predictions with 200 participants' dMRI data from the HCP database, and for the test–retest reliability analyses with five participants' data from the HCP Retest database, dMRI preprocessing and connectome generation followed the same protocol as indicated for dataset H.

Ensemble tractography (dataset S(ET)). Tractography algorithms include multiple parameter settings with many degrees of freedom. Ensemble tractography seeks to overcome biases associated with specific parameter choices by estimating several

connectomes, one for each choice of parameter value, and subsequently combining them into an 'ensemble' connectome¹⁰. For ensemble tractography, we used dataset S to generate five whole-brain connectomes, with the maximum radius of curvature of fibers set to one of five parameter values (0.25 mm, 0.5 mm, 1 mm, 2 mm and 4 mm). Each connectome was generated by seeding fibers at the GM–white matter interface. Next, we combined these individual connectomes to form the ensemble connectome. We generated two connectomes: a smaller, 0.8-million-fiber connectome for streamline pruning with LiFE and a larger 1.6-million-fiber connectome for pruning with ReAL-LiFE (for details see the section Estimating the regularization parameter λ).

Quantifying ReAL-LiFE performance as speedups and fits to data. *Quantifying speedups.* For each of the datasets I, S and H and for every connectome size (Fig. 1c), we first pruned streamlines with the original LiFE optimization algorithm³ for 500 iterations. Next, we pruned streamlines with the same connectome using the GPU-accelerated version of LiFE (with no regularization), again for 500 iterations. Speedup of the GPU-accelerated pruning as compared to CPU pruning was computed as

$$\text{Speedup} = \frac{t(\text{LiFE}_{\text{CPU}})}{t(\text{LiFE}_{\text{GPU}}) + t(\text{Overhead}_{\text{GPU}})}$$

where $t(\text{LiFE}_{\text{CPU}})$ is the time taken for 500 iterations of the LiFE algorithm on the CPU, $t(\text{LiFE}_{\text{GPU}})$ is the time taken for 500 iterations of the GPU-accelerated LiFE and $t(\text{Overhead}_{\text{GPU}})$ is the GPU overhead time corresponding to the time taken for data transfer between the CPU and GPU memory and other preprocessing steps, such as sorting along the voxels dimension.

To calculate the speedup scaling factor, we fit a function of the form $y = a + b \log(x)$ to the speedups, where y is the speedup, x is the parameter (N_s , N_p , N_f), a is the intercept and b is the slope. We quantified the scaling factor (q) as change in speedup from the lowest to the highest value for each of the parameters N_s , N_p and N_f .

We compared the speedups of GPU-accelerated LiFE over a second state-of-the-art algorithm, SIFT⁷. We tested for speedups for each of the three datasets—*I*, *H* and *S*—on a connectome of one million fibers ($N_f = 10^6$), using the respective default values for parameters N_s and N_p . To ensure a fair comparison, we first pruned streamlines from each unpruned connectome with the GPU-accelerated LiFE, systematically running the optimization algorithm from 50 to 500 iterations (steps of 50 iterations). Next, we pruned the same (respective) unpruned connectome with SIFT, albeit to the exact same size as the corresponding LiFE-pruned connectome. In other words, the termination criterion for SIFT was specified to match the number of fibers in the LiFE-pruned connectome. We then defined the convergence time t_c as the iteration at which the change in the objective function $O(t)$ over the last ten iterations was less than 0.1% of its initial value $O(t_c + 10) - O(t_c) < \Delta \cdot O(t_0)$, where $O(t_0)$ is the initial value of the objective function and $\Delta = 0.001$. For each dataset, we then computed the speedup of GPU-accelerated LiFE over SIFT at convergence as the average of the speedups across the nearest two multiples of 50 iterations. Speedups were compared based on total execution times of each algorithm from start to finish, including all overheads associated with loading data into memory.

We also compared the speedups of GPU-accelerated LiFE over yet another state-of-the-art algorithm, COMMIT2, which incorporates anatomically informed priors into its objective⁶. As before, for each of the three datasets (*I*, *H* and *S*), we computed speedups of a connectome of one million fibers ($N_f = 10^6$), with default parameter values for N_s and N_p . To enable a fair comparison, in this case, we first pruned the connectome with COMMIT2 until convergence, with all default parameters. Next, we pruned the same (respective) unpruned connectome with GPU-accelerated LiFE to within 1% of the initial size of the corresponding COMMIT2-pruned connectome. In this case, we pruned the ReAL-LiFE streamlines to the size of the COMMIT2-pruned connectome, because the converse approach—pruning the COMMIT2 streamlines to match the ReAL-LiFE-pruned connectome, at convergence—required over 500 iterations of the COMMIT2 algorithm. As with SIFT, speedups were compared based on total execution times of each algorithm from start to finish, including all overheads associated with loading data into memory.

Finally, we tested a dockerized version of the LiFE algorithm (<https://doi.org/10.25663/bl.app.104>) to optimize a one-million fiber connectome for each of the three datasets (500 iterations). We found that the run times were comparable to those of CPU-LiFE, as reported in Supplementary Fig. 1 (dataset H: CPU-LiFE 17.9 h, dockerized LiFE, 23.1 h; dataset I: CPU-LiFE 3.0 h, dockerized LiFE 3.3 h; dataset S: CPU-LiFE 8.2 h, dockerized LiFE 8.4 h).

Quantifying duplication of fiber weights. We tested ReAL-LiFE's ability to prune away duplicate (redundant) fibers and compared its performance with the LiFE and SIFT2 algorithms⁶. For this we used two approaches. In the first approach, we used dataset M to generate a whole-brain connectome comprising 0.5 million fibers. Next, we used this connectome to simulate a noise-free diffusion signal³. We then created two, randomly 'jittered', near-identical versions of this connectome (C1 and C2), by randomly perturbing the spatial coordinates of 10% of the nodes in each

fiber, randomly by $\pm 0.01\%$. We combined these perturbed connectomes to create a single connectome comprising one million fibers. Finally, we pruned streamlines in the combined connectome with LiFE and ReAL-LiFE, using the noise-free diffusion dataset simulated in the previous step. For pruning with SIFT2, we followed a slightly different approach. SIFT2, unlike LiFE or ReAL-LiFE, does not prune out (eliminate) any fiber. To facilitate a fair comparison with LiFE and ReAL-LiFE, we first pruned the combined connectome with SIFT followed by pruning with SIFT2 to obtain fiber weights of the unpruned fibers retained by SIFT. We sought to perturb the fibers across the two connectomes before pooling them, rather than combining two connectomes with identical, duplicated fibers; in the latter case, each algorithm yielded identical weights across each pair of duplicate fibers because none of the algorithms could break ties across identical fibers.

Following pruning, we computed a ‘uniqueness’ index quantifying the normalized difference in weights between copies of corresponding fibers as

$$\zeta_{\text{uniq}} = \frac{|w_1 - w_2|}{(w_1 + w_2)}$$

A higher value of ζ_{uniq} indicates a higher tendency to prune out redundant fibers and to retain only one copy of the two near-identical fibers.

In the second approach, we created a ‘trimmed’ version of the whole-brain connectome from dataset M (same as estimated in the previous approach) by trimming out 5% of the nodes from each terminus of each fiber. We then combined the original whole-brain connectome with this jittered copy, followed by pruning with LiFE, ReAL-LiFE and SIFT2, as before.

Estimating the regularization parameter λ . We sought to compare the performance of LiFE with ReAL-LiFE in terms of model fit. Because the L1-regularization penalty in ReAL-LiFE yields systematically sparser connectomes as compared to LiFE, to enable fair comparisons of the model fit we matched the summed weights (L1-norm) of the fibers, following pruning with each approach (Supplementary Fig. 3). To permit this, we generated a larger initial connectome, with $2\times$ the number of fibers for pruning with ReAL-LiFE, as compared to LiFE, and sampled λ , in the range of $[10^{-8}, 1]$ in logarithmically spaced steps (Supplementary Fig. 3). For datasets I and M , we compared the cross-validated root-mean-square error (r.m.s.e.) by generating a connectome with one million fibers, followed by pruning with LiFE (unregularized), against the accuracy of a connectome generated with two million fibers, followed by pruning with ReAL-LiFE. For dataset S(ET), we generated two ensemble connectomes: one with 0.8 million fibers (for pruning with LiFE) and another with 1.6 million fibers (for pruning with ReAL-LiFE). These ensemble connectomes were assembled from five smaller whole-brain connectomes generated with 160,000 and 320,000 fibers respectively. We then chose the λ that matched the L1-norm of weights across both pruning approaches. For datasets I , M and S(ET), λ values of 0.006, 0.01 and 0.01, respectively, provided this match (Supplementary Fig. 3); unless otherwise specified, the same configurations and regularization parameter values were used for all subsequent analyses (for example, Fig. 2 and Supplementary Figs. 2–4).

Quantifying the model fit. For each dataset (datasets I , S(ET) and M), we evaluated the performance of the pruning algorithm in two ways—by testing for overfitting and consistency. To test for overfitting, we first generated a whole-brain connectome (C1) with one diffusion dataset (D1; Supplementary Fig. 3). Next, we pruned C1 with LiFE and ReAL-LiFE using dataset D1 as ground truth to obtain a predicted diffusion signal (P1; Supplementary Fig. 3). Finally, we computed the voxel-wise, cross-validated r.m.s.e. between the predicted diffusion signal P1 and a second, independently acquired diffusion dataset (D2; Supplementary Fig. 3) from the same participant. For dataset M , D2 was independently simulated from the same underlying ground-truth connectome (see section Diffusion MRI acquisition and preprocessing). We then computed the distribution of voxel-wise r.m.s.e.s, following pruning with LiFE and ReAL-LiFE, and also computed their pairwise differences (ReAL-LiFE – LiFE; Supplementary Fig. 3).

To test for consistency (Supplementary Fig. 4), we generated two whole-brain connectomes (C1 and C2) from two independently acquired diffusion datasets (D1 and D2, respectively), both from the same participant (Supplementary Fig. 4). Next, we pruned C1 (with LiFE or ReAL-LiFE) with dataset D2 as ground truth and, similarly, pruned C2 with dataset D1 as ground truth (Supplementary Fig. 4). We then chose the fibers assigned the top 50th percentile of the weights, to predict the diffusion signals P1 and P2 (Supplementary Fig. 4). Finally, we computed the voxel-wise, r.m.s.e. between the predicted diffusion signals P1 and P2 (Supplementary Fig. 4). As before, we computed the distribution of voxel-wise r.m.s.e.s, following pruning with LiFE and ReAL-LiFE, and also computed their pairwise differences (ReAL-LiFE – LiFE; Supplementary Fig. 4).

Test–retest reliability analysis, HCP retest data. *Estimating structural connectivity features.* For each participant in the HCP retest dataset ($n = 5$; blue indices in Supplementary Table 2), we estimated two whole-brain connectomes, one comprising one million fibers and a second two million fibers (see ‘Tractography and generating whole-brain connectomes from dMRI data’). This was done both for the original dataset (dataset 1) and the retest dataset (dataset 2). For each visit’s

data, we employed FreeSurfer’s anatomical parcellation based on the Desikan–Killiany atlas comprising 34 regions on each hemisphere to construct two 34×34 structural connectivity matrices (one per hemisphere)²¹. We computed five kinds of structural connectivity matrix: (1) unpruned, (2) ReAL-LiFE pruned ($\lambda = 0.01$, two-million-fiber connectome), (3) LiFE pruned (one-million-fiber connectome), (4) SIFT2 pruned (one-million-fiber connectome) and (5) COMMIT2 pruned (one-million-fiber connectome). For each connectivity matrix, the (i, j) th entry indicates the number of fibers, or the sum of pruned fiber weights (after pruning with either ReAL-LiFE, LiFE, SIFT2 or COMMIT2), across all fibers connecting regions i and j , respectively. Because diffusion MRI does not provide information regarding the direction of connectivity, each matrix was symmetric about the diagonal. For these analyses we considered only connections between pairs of intra-hemispheric regions and ignored diagonal elements of the connectivity matrix, that is, connections that originate and terminate within the same ROI. As a result, the total number of connectivity features across both hemispheres was 1,122 ($^{34}C_2$ connections per hemisphere \times 2 hemispheres). To limit noisy estimates of test–retest reliability metrics, we chose only connections with the top 50th percentile of fibers (561 connections), for the test–retest reliability analysis.

Within-participant variability (V_w). To compute within-participant variability, for each participant, and for each connection in the connectivity matrix, we computed a normalized difference index between the connectivity metrics of the first dataset and the second dataset, that is

$$V_w^i = \frac{|c_1^i - c_2^i|}{(c_1^i + c_2^i)_k}$$

where k denotes the participant, i denotes the connection, $i = 1, 2, \dots, 561$, and c_1 and c_2 denote the connectivity strengths based on datasets 1 and 2 (two visits), respectively. We then computed average within-participant variability for each connection as

$$V_w^i = \langle V_w^i \rangle_k$$

where the angle brackets $\langle \cdot \rangle_k$ denote the average across the participants.

Between-participant variability (V_b). To compute between-participant variability, for each connection in the connectivity matrix, we first computed a normalized difference index between the connectivity metrics of every participant’s dataset m and every other participant’s dataset l ($m \neq l$), that is

$$V_b^i = \frac{|c_{1l}^i - c_{2m}^i|}{(c_{1l}^i + c_{2m}^i)_{lm}}$$

where i denotes the connection, $i = 1, 2, \dots, 561$, and c_1 and c_2 denote the connectivity strengths based on datasets 1 and 2. We then computed average between-participant variability as

$$V_b^i = \langle V_b^i \rangle_{l, m}, l \neq m$$

where the angle brackets $\langle \cdot \rangle_{l, m}$ denote the average across each pair of participants.

Finally, for each connection i , we computed a difference in variability between pruned and unpruned connectivity as

$$\Delta V_w = V_w^{\text{pruned}} - V_w^{\text{unpruned}} \text{ and } \Delta V_b = V_b^{\text{pruned}} - V_b^{\text{unpruned}}$$

Reliability. Finally, for each connection i , we computed a reliability metric ϕ as

$$\phi_i = \frac{V_b^i}{V_b^i + V_w^i}$$

That is, the reliability of connection i was taken as the ratio of the between-participant variability to the total variability, including between- and within-participant components, following ref.¹². Using this reliability metric computed using the HCP retest data, we identified connections with the highest and lowest reliability metrics.

Analysis of structure–behavior relationships. We tested whether structural connectivity could explain inter-individual variations in various cognitive scores, with data from $n = 200$ participants from the HCP database. Specifically, we performed SVR-based prediction of cognitive scores with structural connectivity features estimated either from the unpruned connectome (number of fibers) or by pruning streamlines with ReAL-LiFE or SIFT (connection weights).

Behavioral scores. For each of the $n = 200$ participants drawn from the HCP database, we predicted 60 behavioral scores (Supplementary Data 1) from dMRI connectivity⁹. These 60 scores were chosen based on selection criteria employed

in previous studies that sought to predict these scores from functional MRI connectivity¹³. In addition to selecting the 58 scores employed in these studies, we included the in-scanner score related to performance accuracy in the shape-matching subtask of the emotion task (Emotion_Task_Shape_Acc, Supplementary Data 1) and replaced the overall accuracy in the relational task with accuracies in each of its subtasks (object matching/Relational_Task_Match_Acc and object relation/Relational_Task_Rel_Acc). Both state and trait scores were included, and age-adjusted scores were used, wherever available. Importantly, behavioral score selection was agnostic to the results of our prediction analyses. Broadly, these behavioral scores fell into three major categories: cognition ($n=13$), emotion ($n=23$) or personality ($n=24$)²². These scores also included nine behavioral performance scores recorded during the task-functional MRI sessions. These included accuracy scores in all subtasks of each task, except for the working-memory task, for which we only chose the overall performance accuracy across all subtasks. Further details are provided in Supplementary Data 1.

Estimating structural connectivity features. For each participant in the HCP dataset ($n=200$), we estimated a whole-brain connectome comprising one million fibers (see ‘Tractography and generating whole-brain connectomes from dMRI data’). As with the test–retest reliability analysis, we used the FreeSurfer anatomical parcellation based on the Desikan–Killiany atlas comprising 34 regions on each hemisphere to construct a 68×68 structural connectivity matrix²¹. We computed two kinds of structural connectivity matrix—(1) unpruned and (2) ReAL-LiFE-pruned ($\lambda=0.01$)—where the (i,j) th entry in each matrix indicates the number of fibers, or the sum of ReAL-LiFE-pruned fiber weights, for fibers connecting regions i and j , respectively. We also carried out predictions with a combination of features (1) and (2) (Fig. 2g). Because diffusion MRI does not provide information regarding the direction of connectivity, each matrix was symmetric about the diagonal. In addition, for these analyses we considered only connections between pairs of intra-hemispheric regions and also set all diagonal elements of the connectivity matrix (connections that originate and terminate within the same ROI) to zero. As a result, the total number of connectivity features across both hemispheres was 1,122 ($^{24}C_2$ connections per hemisphere \times 2 hemispheres).

Prediction model. For predicting behavioral scores using structural connectivity as features, we used an SVM-based regression model with a linear kernel (‘fitsvm’ function in MATLAB), along with RFE¹⁴. Behavioral scores were standardized by z -scoring, prior to model fitting. The 1,122 connection features were organized into a feature matrix having dimensions $200 \times 1,122$, where each row corresponds to a participant and each column corresponds to one of the 1,122 connections (features). We employed RFE to identify features with the largest weight magnitudes in the linear prediction model that provided the highest generalized cross-validation accuracy. We describe the algorithm briefly below (a more detailed description is provided in ref. ¹⁴).

In the first stage of the RFE, the data were divided into N folds. During each iteration i , $N-1$ of these were used for training and the N th fold was reserved for testing. In the second stage, for each training set ($N-1$ folds), the training data were further divided into K folds. Subsequently, each of these K folds was left out exactly once and the SVR model was trained on the $K-1$ folds. In each of the K iterations, the estimated regression coefficients were used to predict the behavioral scores of the left-out fold (N th fold) from the first stage, and a correlation between the predicted scores and the observed scores of the test fold was computed. At the end of the second stage (K iterations), the β weights (regression coefficients) and correlation values were averaged across the K iterations to obtain robust estimates. Next, based on these average β weights, the bottom 10% of the features were discarded. This marks the end of the second stage. The second stage of the RFE was repeated until all features were eliminated. Finally, we chose the set of features, say F_i , that yielded the maximum correlation between the observed and predicted scores.

The above procedure was repeated N times, leaving one fold for testing each time. At the end of the N iterations, we averaged (across the N folds), the final set of estimated β weights ($\langle F_i \rangle$). We repeated the entire RFE procedure (first and second stage) for 100 runs, to robustly estimate the top features and predict behavioral scores. For these analyses, we chose $N=10$ and $K=5$.

Comparing predictions based on the number of fibers and ReAL-LiFE connection weights. We compared the predictions based on ReAL-LiFE weights with those based on the number of fibers. We computed the number of significant predictions (based on the correlation between the observed and predicted scores) for each feature set (ReAL-LiFE weights and number of fibers) for a range of different levels of significance α , ranging from $\alpha=0.00001$ to $\alpha=0.05$. For each value of α we computed the number of scores for which the P value for the correlation between the observed and predicted score was less than α . We also computed average correlation coefficient values across all significant scores at each α level. The number of significant scores and the average correlation coefficients, at each α level, for predictions based on each set of features, and for the different score categories, are plotted in Fig. 2.

Univariate correlations between connectivity and behavioral scores. To understand the anatomical relevance of the ReAL-LiFE connections predicting behavioral

scores, we chose the five best predicted scores in the ‘cognition’ category. For each score, we identified the top connections that contributed most to the prediction, based on the β weight assigned to the connection by the SVR-RFE prediction model. For each such connection, we then computed three additional connectivity metrics: (1) the number of fibers post pruning with ReAL-LiFE, (2) streamline volume and (3) streamline length (voxels intersected). For each connection, we then computed univariate correlations between each of these connectivity metrics and the corresponding behavioral score. Univariate correlations (r and P values) are reported based on robust correlations. This procedure allowed us to identify connectivity metrics, strongly predictive of behavioral scores, that were most strongly correlated with the respective behavioral score.

Control analyses. We performed three control analyses for behavioral score predictions, with ReAL-LiFE connection weights as features.

Predicting five minimally correlated scores. To account for correlations among the 60 behavioral scores, we selected a subset of five minimally correlated scores, following the same procedure as in ref. ¹³. We recapitulate their approach as follows. First, a pair of scores with an absolute correlation of less than 0.1 was chosen at random. Subsequently, three behavioral scores were selected, one at a time, so that each new score correlated minimally with the existing set of scores ($|r| < 0.01$). This procedure was repeated 100 times, resulting in 100 such sets of five minimally correlated scores. Finally, the subset of five scores with the least maximum absolute mutual correlation was selected. These scores corresponded to personality extroversion (HCP field: NEOFAC_E), emotion recognition (ER40HAP and ER40NOE), picture vocabulary (PicVocab) and processing speed (ProcSpeed); these five scores are not identical to those in ref. ¹³, possibly because, unlike their study, we employed age-adjusted scores, wherever these were available. In our analyses, these scores exhibited a non-significant maximum absolute correlation of $r=0.059$ ($P=0.403$).

Controlling for head motion confounds. To account for confounding of head motion on behavioral score predictions, we repeated predictions with the SVR-RFE model after regressing out the effects of motion parameters from the behavioral scores, again following a procedure closely similar to that of ref. ¹³. From each participant’s dMRI scan, six head-motion-related parameters were extracted: three parameters corresponding to rotations (about the x , y and z axes, respectively) and three parameters corresponding to translations (along the x , y and z axes, respectively). We fit a multiple linear regression model with the motion parameters as the independent variables to fit each behavioral score. Subsequently, the residual corresponding to each prediction was fit with the SVR-RFE model with ReAL-LiFE weights as features. If the residual could not be fit well, this would indicate that the inter-participant behavioral score variations could not be explained by structural connection features over and above what could be explained with the motion parameters alone.

Permutation test for RFE predictions. Finally, we corrected for a potential bias of the RFE algorithm toward positive prediction accuracies (r values). Because, at each iteration, the RFE algorithm selects features with the highest numerical value of the generalized cross-validation accuracy, average prediction accuracies across iterations could be positively biased. To control for this bias, we performed a random permutation test in which we shuffled participant labels 100 times across each behavioral score and structural connectivity feature and generated a null distribution of prediction accuracies (r values). Because generating the null distribution for each behavioral score is computationally expensive, each r value in the null distribution (100 permutations) was computed by averaging across, at most, ten iterations of the RFE algorithm. The P value was computed as the proportion of observations in the null distribution that were greater than the actual prediction accuracy (actual r value; Supplementary Fig. 5).

Statistical tests. Unless otherwise stated, all pairwise statistical comparisons were performed with the nonparametric Wilcoxon signed-rank test (for example, Fig. 1 and Supplementary Fig. 2). Voxel-wise r.m.s.e.s post pruning with LiFE and ReAL-LiFE (Supplementary Figs. 3 and 4) were compared using a Kolmogorov–Smirnov test. All correlations were computed based on ‘robust’ correlations: we report values of the ‘bend correlation’ (Fig. 2b–f, h and Supplementary Fig. 5), an approach that accounts for univariate outliers in the data²³. Unless otherwise specified, multiple comparisons corrections were carried out with the Benjamini–Hochberg approach, at a significance level of $P=0.05$. To compare the execution times of the GPU implementation of LiFE with those of the CPU implementation of LiFE as a function of the number of fibers N_f (Supplementary Fig. 1), we employed a two-way analysis of variance with the implementation (CPU/GPU) and the number of fibers as factors. Effect sizes were quantified as Cohen’s d . Statistical significance values are reported as *** $P < 0.001$, ** $P < 0.01$ and * $P < 0.05$.

Hardware and software specifications. All analyses described in this study were performed on a desktop computer with the following hardware and software specifications:

- CPU: eight cores; Intel(R) Xeon(R) CPU E5-2623 v3 @ 3.00 GHz
- GPU: one NVIDIA GeForce GTX 1080 Ti (or) AMD Radeon RX 580

- RAM: DDR4, 4 × 16 GB 1,866 MHz (total 64 GB)
- Hard disk space: 447 GB (64 GB configured as swap)
- Operating system: Ubuntu 16.04
- Software: MATLAB R2017b (64 bit); CUDA toolkit 9.0.

We performed all our benchmarking experiments with only one CPU core and one GPU. GPU code binaries were compiled using the 'nvcc' compiler with the '-ptx' flag. Because the original LiFE package was implemented in Matlab, integrating the CUDA or HIP implementation with the ReAL-LiFE package requires Matlab support for GPU computation. Matlab support is available for NVIDIA GPUs, but is currently unavailable for AMD GPUs (<https://www.mathworks.com/help/parallel-computing/gpu-support-by-release.html>). The end-to-end version of the code, integrated with ReAL-LiFE, is currently available for NVIDIA GPUs (Code availability section), and will be released for AMD GPUs as soon as Matlab support for the latter hardware becomes available.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this Article.

Data availability

Dataset *I* has been deposited into a figshare repository¹⁹. Dataset *S* is a part of the original LiFE algorithm and can be accessed from the associated repository¹⁸. Dataset *H* and all HCP data can be accessed from the HCP database⁹. Dataset *M* is available on the Zenodo repository⁵. Source data are provided with this paper.

Code availability

The custom code for reproducing all figures is available on figshare¹⁹. The current version of the ReAL-LiFE algorithm is available on Code Ocean²⁴. In addition, to streamline pruning, the code pipeline integrates with a standard atlas (HCP-MMP atlas⁹) to label brain regions connecting each pair of streamlines. The pipeline also integrates with the TractSeg tool²⁵ for automated tract labeling.

Received: 6 July 2021; Accepted: 21 April 2022;

Published online: 30 May 2022

References

- Chanraud, S., Zahr, N., Sullivan, E. V. & Pfefferbaum, A. MR diffusion tensor imaging: a window into white matter integrity of the working brain. *Neuropsychol. Rev.* **20**, 209–225 (2010).
- Damoiseaux, J. S. et al. White matter tract integrity in aging and Alzheimer's disease. *Hum. Brain Mapp.* **30**, 1051–1059 (2009).
- Pestilli, F., Yeatman, J. D., Rokem, A., Kay, K. N. & Wandell, B. A. Evaluation and statistical inference for human connectomes. *Nat. Methods* **11**, 1058–1063 (2014).
- Jeurissen, B., Descoteaux, M., Mori, S. & Leemans, A. Diffusion MRI fiber tractography of the brain. *NMR Biomed.* **32**, e3785 (2019).
- Maier-Hein, K. H. et al. The challenge of mapping the human connectome based on diffusion tractography. *Nat. Commun.* **8**, 1349 (2017).
- Smith, R. E., Tournier, J. D., Calamante, F. & Connelly, A. SIFT2: enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *Neuroimage* **119**, 338–351 (2015).
- Smith, R. E., Tournier, J. D., Calamante, F. & Connelly, A. SIFT: spherical-deconvolution informed filtering of tractograms. *Neuroimage* **67**, 298–312 (2013).
- Schiavi, S. et al. A new method for accurate in vivo mapping of human brain connections using microstructural and anatomical information. *Sci. Adv.* **6**, eaba8245 (2020).
- Van Essen, D. C. et al. The Human Connectome Project: a data acquisition perspective. *NeuroImage* **62**, 2222–2231 (2012).
- Kumar, S., Sreenivasan, V., Talukdar, P., Pestilli, F. & Sridharan, D. ReAL-life: accelerating the discovery of individualized brain connectomes on GPUs. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 33, 630–638 (AAAI, 2019).
- Daducci, A., Dal Palú, A., Descoteaux, M. & Thiran, J. P. Microstructure informed tractography: pitfalls and open challenges. *Front. Neurosci.* **10**, 247 (2016).
- Jiang, C., Betzel, R., He, Y. & Zuo, X.-N. Toward reliable network neuroscience for mapping individual differences. Preprint at *bioRxiv* <https://doi.org/10.1101/2021.05.06.442886> (2021).
- Kong, R. et al. Spatial topography of individual-specific cortical networks predicts human cognition, personality, and emotion. *Cereb. Cortex* **29**, 2533–2551 (2019).
- De Martino, F. et al. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. *Neuroimage* **43**, 44–58 (2008).
- de Schotten, M. T. et al. A lateralized brain network for visuospatial attention. *Nat. Neurosci.* **14**, 1245–1246 (2011).
- Brodt, S. et al. Fast track to the neocortex: a memory engram in the posterior parietal cortex. *Science* **362**, 1045–1048 (2018).
- Ho, J. C., Ghosh, J. & Sun, J. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 115–124 (ACM, 2014); <https://doi.org/10.1145/2623330.2623658>
- Caiafa, C. F. & Pestilli, F. Multidimensional encoding of brain connectomes. *Sci. Rep.* **7**, 11491 (2017).
- Sreenivasan, V. et al. ReAL-LiFE. *figshare* <https://doi.org/10.6084/m9.figshare.13491024.v1> (2002).
- Tournier, J. D., Calamante, F. & Connelly, A. MRtrix: diffusion tractography in crossing fiber regions. *Int. J. Imaging Syst. Technol.* **22**, 53–66 (2012).
- Sreenivasan, V. & Sridharan, D. Subcortical connectivity correlates selectively with attention's effects on spatial choice bias. *Proc. Natl Acad. Sci. USA* **116**, 19711–19716 (2019).
- Kashyap, R. et al. Individual-specific fMRI-subspaces improve functional connectivity prediction of behavior. *Neuroimage* **189**, 804–812 (2019).
- Pernet, C. R., Wilcox, R. & Rousselet, G. A. Robust correlation analyses: false positive and power validation using a new open source Matlab toolbox. *Front. Psychol.* **3**, 606 (2013).
- Sreenivasan, V. et al. ReAL-LiFE. *Code Ocean* <https://doi.org/10.24433/CO.5578739.v1> (2022).
- Wasserthal, J., Neher, P. & Maier-Hein, K. H. TractSeg—fast and accurate white matter tract segmentation. *Neuroimage* **183**, 239–253 (2018).

Acknowledgements

We thank A. Nagaraj, R. Bhuthesh, A. Ashok Nayak, A. Panwar, S. Pandey and A. Basu for help and resources with porting ReAL-LiFE to AMD GPUs. We also thank P. Gupta, S. Gupta and G. T. Anandan for their feedback on an earlier version of this manuscript. We also thank Health Care Global (HCG) Hospital for access to their MRI scanning facility. This research was supported by a Department of Biotechnology-Wellcome Trust India Alliance Intermediate fellowship (no. IA/1/15/2/502089), a Science and Engineering Research Board Early Career award (no. ECR/2016/000403), a Department of Biotechnology-Indian Institute of Science Partnership Program grant, an India Trento Partnership Program grant (to D.S.), a Pratiksha Trust Intramural grant (to D.S. and P.T.), NSF OAC-1916518, NSF IIS-1912270, NSF IIS-1636893, NSF BCS-1734853, NIH NIMH R01MH126699, NIH NIBIB R01EB030896, NIH NIBIB R01EB029272, Microsoft Investigator Fellowship, a gift by the Kavli Foundation (to F.P.), and the Ministry of Human Resource Development, Government of India (to V.S. and S.K.).

Author contributions

D.S. and P.T. designed the research. V.S. performed the research. S.K. and F.P. contributed unpublished reagents/analytic tools. V.S. analyzed data and V.S. and D.S. wrote the manuscript.

Competing interests

D.S. is a research consultant at Google. P.T. is a research scientist at Google. He is also the founder of Kenome, an enterprise AI company. V.S., S.K. and F.P. declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s43588-022-00250-z>.

Correspondence and requests for materials should be addressed to Varsha Sreenivasan or Devarajan Sridharan.

Peer review information *Nature Computational Science* thanks Xi-Nian Zuo and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Ananya Rastogi, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- | | |
|-----|-----------|
| n/a | Confirmed |
|-----|-----------|
- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
 - A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
 - The statistical test(s) used AND whether they are one- or two-sided
Only common tests should be described solely by name; describe more complex techniques in the Methods section.
 - A description of all covariates tested
 - A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
 - A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
 - For null hypothesis testing, the test statistic (e.g. F , t , r) with confidence intervals, effect sizes, degrees of freedom and P value noted
Give P values as exact values whenever suitable.
 - For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
 - For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
 - Estimates of effect sizes (e.g. Cohen's d , Pearson's r), indicating how they were calculated

Our web collection on [statistics for biologists](#) contains articles on many of the points above.

Software and code

Policy information about [availability of computer code](#)

Data collection Dataset I was acquired at the HealthCare Global Hospital, Bangalore.

Data analysis Dataset I was preprocessed using Vistasoft 2017 (<https://github.com/vistalab/vistasoft>).
Dataset S was available in preprocessed format. Preprocessing was performed using Vistasoft 2017
Dataset M was preprocessed using Mrtrix3 and FSL.
Dataset H and all data used for behavioral predictions, we used minimally preprocessed data as part of the HCP S1200 release.
For all datasets, tractography was performed using MRtrix3 and cortical parcellations were obtained using FreeSurfer. All analyses were carried out using custom code written in MATLAB2017b. Code for the ReAI-LiFE algorithm is available at 10.24433/CO.5578739.v1.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research [guidelines for submitting code & software](#) for further information.

Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Dataset I was acquired at the HealthCare Global Hospital, Bangalore.

Dataset S was acquired at Stanford's Center for Cognitive and Neurobiological Imaging, and is publicly available at doi:10.5967/K8X63JTX.

Dataset M is a synthetic dataset generated as part of the ISMRM Tractography Challenge 2015, and is available at http://tractometer.org/ismrm_2015_challenge/data.

Dataset H and all data used for behavioral predictions are part of the WU-Minn HCP Consortium S1200 Release, and are publicly accessible at <https://www.humanconnectome.org>.

Data availability

The datasets generated during and/or analyzed during the current study are available in the repositories linked below:

Dataset I: [10.6084/m9.figshare.13491024](https://doi.org/10.6084/m9.figshare.13491024)

Dataset S/S(ET): [doi:10.5967/K8X63JTX](https://doi.org/10.5967/K8X63JTX).

Dataset H: <https://www.humanconnectome.org/study/hcp-young-adult>

Dataset M: http://www.tractometer.org/ismrm_2015_challenge/

Source Data for Figure 1 and Figure 2 are available with this manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

Life sciences Behavioural & social sciences Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://www.nature.com/documents/nr-reporting-summary-flat.pdf)

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Power analysis for sample size estimation was not performed. Datasets I, H, S and M are drawn from diverse databases and correspond each to scans from individual subjects; these were meant to illustrate ReAI-LiFE speedups and features. For behavioral predictions, data for 200 participants were drawn from the HCP database in chronological order, after matching for age and gender. Sample sizes (>100) are typical for studies in this domain and similar sample sizes have been routinely employed in previous studies for establishing brain-behavior relationships. A systematic analysis of the effect of sample sizes on brain-wide association studies is available at: https://doi.org/10.1038/s41586-022-04492-9
Data exclusions	Because tractography is a time consuming step, dMRI scans for 200 subjects in the HCP database were analyzed for behavioral predictions. These were drawn in chronological order from among the 1200 subjects' data available in the database, matching for age and gender, in a manner agnostic to prediction results.
Replication	We confirmed that all key results pertaining to GPU acceleration and regularization were replicated across three independently acquired datasets (Datasets I, S and H).
Randomization	No experimental groupings were used in this study.
Blinding	Blinding is not relevant because no grouping was performed.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies
<input checked="" type="checkbox"/>	<input type="checkbox"/> Eukaryotic cell lines
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology
<input checked="" type="checkbox"/>	<input type="checkbox"/> Animals and other organisms
<input type="checkbox"/>	<input checked="" type="checkbox"/> Human research participants
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern

Methods

n/a	Involved in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input type="checkbox"/>	<input checked="" type="checkbox"/> MRI-based neuroimaging

Human research participants

Policy information about [studies involving human research participants](#)

Population characteristics

Dataset I: Female, 26-29 yrs.

Population characteristics	Dataset S: Male, 37-39 yrs. Dataset H: Male, 26-30 yrs. Dataset M: synthetic data generated from one HCP dataset, male, 26-30 yrs. Data for behavioral predictions: 100 females, 22-37 yrs.
Recruitment	Dataset I: Acquired as part of a previous study (Sreenivasan and Sridharan, PNAS, 2019). Dataset S: Information can be found at doi:10.5967/K8X63JTX. Dataset M: Information can be found at http://tractometer.org/ismrm_2015_challenge/data . Dataset H and all data used for behavioral predictions are part of WU-Minn HCP Consortium S1200 Release. Information for these datasets can be found at https://www.humanconnectome.org/study/hcp-young-adult/project-protocol/recruitment .
Ethics oversight	Institute Human Ethics Committee, Indian Institute of Science, Bangalore.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Magnetic resonance imaging

Experimental design

Design type	Not applicable for diffusion imaging data.
Design specifications	Not applicable for diffusion imaging data.
Behavioral performance measures	60 different behavioral scores spanning cognition, emotion and personality, from the HCP database were used for behavioral predictions. Information on behavioral scores can be found at https://www.humanconnectome.org/study/hcp-young-adult/project-protocol/behavioral-testing .

Acquisition

Imaging type(s)	Structural, Diffusion
Field strength	3 Tesla
Sequence & imaging parameters	Dataset I: Spin echo sequence, EPI, FoV=256mm, matrix 128 slice thickness=2mm, orientation:T>C-7.3>S-0.7, TE=90ms, TR:8900ms, flip angle=8 degrees. Dataset S: Details of scan protocols can be found in Pestilli et al., 2014 (https://www.nature.com/articles/nmeth.3098). Dataset M: Synthetic data generated using a single subject dataset from the HCP database, details for which can be found at https://www.humanconnectome.org/storage/app/media/documentation/s1200/HCP_S1200_Release_Reference_Manual.pdf . Dataset H and data for behavioral predictions: Details of scan protocols can be found at https://www.humanconnectome.org/storage/app/media/documentation/s1200/HCP_S1200_Release_Reference_Manual.pdf .
Area of acquisition	Whole brain
Diffusion MRI	<input checked="" type="checkbox"/> Used <input type="checkbox"/> Not used
Parameters	Dataset I: 64 directions (plus 2 b=0 acquisitions), single-shell (b-value=1000 s/mm ²). No cardiac gating was used. Dataset S: 96 directions (plus 10 b=0 acquisitions), single-shell (b-value=2000 s/mm ²). No cardiac gating was used. Dataset M: Synthetic data simulated from single subject from HCP database: 270 directions (plus 18 b=0 acquisitions), multi-shell (b-values=1000, 2000, 3000 s/mm ²). Dataset H and data for behavioral predictions: 270 directions (plus 18 b=0 acquisitions), multi-shell (b-values=1000, 2000, 3000 s/mm ²).

Preprocessing

Preprocessing software	MRtrix3, FSL, version 5.0, FreeSurfer version 6.
Normalization	Datasets I, S, and M were not normalized. Dataset H and all scans used for behavioral prediction were normalized to the MNI template.
Normalization template	MNI template.
Noise and artifact removal	Datasets I and S: Denoising, motion, and eddy correction. Dataset M: Denoising, Gibbs ringing artifact removal, motion and eddy correction, and susceptibility-induced distortion correction. Dataset H and all data used for behavioral predictions: We used the minimally preprocessed data available in the HCP database. Preprocessing steps included B0 intensity normalization, susceptibility-induced distortion correction, motion and eddy correction, and gradient non-linearity correction.
Volume censoring	No volume censoring was performed in any dataset.

Statistical modeling & inference

Model type and settings For connectome estimation using MRtrix3, the Constrained Spherical Deconvolution (CSD) model was used to estimate Fiber Orientation Distributions (FODs), followed by probabilistic tractography.
For connectome evaluation using LiFE and ReAI-LiFE the Stejskal-Tanner model was used for estimating the diffusion signal based on a structural connectome, b-values and gradient directions.

Effect(s) tested To quantify LiFE/ReAI-LiFE model accuracy, voxelwise root mean squared error (RMSE) between the measured and predicted diffusion signals were computed.
For behavioral score prediction analyses, Bend correlation was computed between predicted and actual scores.

Specify type of analysis: Whole brain ROI-based Both

Anatomical location(s) Anatomical segmentation was performed using FreeSurfer. The Desikan-Killiany atlas was used for all analyses.

Statistic type for inference (See [Eklund et al. 2016](#)) The Kolmogorov-Smirnov test and Wilcoxon signed-rand test was used to compare the voxelwise RMSE post pruning with SIFT/SIFT2, LiFE and ReAI-LiFE.
For behavioral score predictions, significantly predicted scores were identified for a range of significance levels and also using a permutation test.

Correction For behavioral score predictions, corrections for multiple comparisons was carried out using the Benjamini-Hochberg correction.

Models & analysis

n/a | Involved in the study

Functional and/or effective connectivity

Graph analysis

Multivariate modeling or predictive analysis

Multivariate modeling and predictive analysis Behavioral score predictions using structural connectivity features were carried out using Support Vector Machine (SVM) based regression model with a linear kernel. Correlation between observed and predicted scores were carried out using the Bend correlation, correcting for univariate outliers (details provided in the Methods).