Methods

Animal care and use. All experiments related to the use of mice in the Medical Research Council Laboratory of Molecular Biology (MRC LMB) were carried out in accordance with the UK Animals (Scientific Procedures) Act of 1986, with local ethical approval provided by the MRC LMB Animal Welfare Ethical Review Board (LMB AWERB) and overseen institutionally by designated animal welfare officers (NACWOs). *TRAP2;Ai14* mice (*TRAP2*, JAX 03032; *Ai14*, JAX 7914) were kindly shared by Liqun Luo, Stanford University, CA, US. All experiments related to the use of mice at the National Institute of Biological Sciences, Beijing (NIBS) were approved by the Animal Care and Use Committee of NIBS in accordance with the Regulations for the Administration of Affairs Concerning Experimental Animals of China. *DAT-Cre* (JAX 006660) and *Vgat-Cre* mice (JAX 028862) were obtained from the Jackson Laboratory. *Sert-Cre* mice (031028-UCD) were obtained from the University of California, Davis.

Data preparation. The average and annotation template of the Allen atlas were downloaded from the Allen Institute web portal (http://atlas.brain-map.org/). All the brain datasets, processing methods, brain names, and imaging resolutions are summarised in Table 1. *TRAP2;Ai14* mice are used to generate Somata-stained and Nuclei-stained datasets. Mice used for these two datasets received 5 mild electric foot-shocks delivered through the floor. Each foot-shock is up to 0.7 mA for up to 2 seconds. The brain samples were cleared based on modified Adipo-Clear protocol[1]. Detailed protocols of TRAP2; Ai14 staining and c-fos staining have been described previously[2]. The Sert-Stanford dataset is from the samples of Vglut3-Cre; Sert-Flp group in the study of Ren, Isakova, Friedmann and Zeng et al.[1], and raw data are kindly shared by Drew Friedmann. The DCN-Stanford dataset is from the study of Kebschull et al.[3], and the raw data are kindly shared by Justus Kebschull.

Brain samples of Sert-NIBS and GABA-NIBS are generated at NIBS. Samples of Sert-NIBS and GABA-NIBS were prepared by an unpublished labelling system, LINCS (*L*abel *I*ndividual *N*eurons with *C*hemical dyes and with controllable *S*parseness), which introduces chemical dyes (e.g., Alexa Fluor-647) as the signalling molecule for photostable and ultrabright labelling. LINCS labelling was performed via a viral genetic approach and achieved cell-type specificity using the Cre-loxP system. For Sert-NIBS brains, 50nl AAV was injected in the dorsal raphe nucleus of Sert-Cre mice. For GABA-NIBS brains, 50nl AAV was injected in the ventral tegmental area unilaterally. The brain samples were cleared based on the iDISCO+ protocol[4]. Samples were stored in dibenzyl ether until clear and imaged within one week. The cleared mouse brains were imaged in horizontal orientation with the dorsal side up on a light-sheet microscope (Ultramicroscope II, LaVision BioTec) using an sCMOS camera (Andor Neo) and a 4×/0.3 objective lens equipped with a 6 mm working distance dipping cap. Samples were scanned for 640 nm and 488 nm (auto-fluorescence) channels with dynamic focus using one-sided illumination with a step size of 3 µm.

The MRI dataset was accessed from: https://github.com/dmac-lab/mouse-brain-atlas. We used the ex vivo brain MRI dataset for validating the brain region segmentation and whole-brain registration methods. The dataset provides annotations of 21 different brain regions[5]. The image resolution is 150um per voxel. As the size and shape of MRI brains are significantly different from the Allen atlas, we resized and down-sampled the Allen atlas to the MRI resolution for whole-brain registration.

In the task of whole-brain axon segmentation (WBAS), we manually annotated six large-sized cubes for quantitative evaluation and comparison, with a size of 600×600×225 voxels, including two cubes from Sert-Stanford, two cubes from Sert-NIBS, and two cubes from GABA-NIBS. The number of automatically annotated cubes for deep model training, including "pure" axon cubes and "pure" artefact cubes, and mixed cubes after data augmentation, is

provided in Table 2. The size of automatically annotated cubes is 150×150×150 voxels. In the brain region segmentation task, we manually annotated 12 LSFM brains in the auto-fluorescence channel with six major brain regions (CTX, CB, CBX, BS, HPF, CP) for quantitative evaluation and comparison, including three somata-stained brains, three nuclei-stained brains, three Sert-Stanford brains and three Sert-NIBS brains. These annotations are also used for the evaluation of whole-brain registration. The data used for brain region segmentation and whole-brain registration are summarised in Table. 3. All training and prediction are performed using an NVIDIA GeForce RTX 3090 graphics-processing unit.

Axon Segmentation.

Automated cube annotation and data augmentation. We first binarise the selected "pure" axon cubes and extract the axons with a set of image processing techniques, including a Gaussian filter with a kernel size of 3×3×3, following the difference of Gaussians to increase the visibility of axons and thresholding to get binarised axons. The threshold can be set automatically or by users after checking the 3D cube. Subsequently, to keep the tree-topological structure of axons, we employ image dilation to connect the adjacent fragments of the above binarised axons. Then the centre lines are extracted to skeletonise the axons and are dilated to generate annotations with unified thickness, i.e., the centre voxel extends one more voxel in six directions of the 3D space (left and right, top and bottom, front and back). In doing so, the axons are annotated automatically and in 3D (Extended Data Fig.1B). For a "pure" axon cube with the size of 150×150×150 voxels, the automated annotation can be obtained within 5 minutes. We summarise the number of automatically annotated cubes in Extended Data Table 3.

We especially introduced three data augmentation strategies to simulate real scenarios of heterogeneous brain samples, i.e., CutMix[6], histogram matching[7], and local contrast enhancement[8] (Fig. 2C). In the CutMix strategy, we randomly cut part of the "pure" axon cubes and the "pure" artefact cubes respectively, and then mix the two partial cubes as a new cube which includes both axons and artefacts. In the histogram matching, we randomly selected two cubes and transferred the histogram from one to another, simultaneously preserving the annotated axons. In the local contrast enhancement, we randomly change the intensity of stochastic axons to enhance the diversity of axon morphology and connectivity patterns. Given 100 cubes of "pure" axon and "pure" artefact individually, our data augmentation strategy can generate more than 1000 cubes with annotations for deep model training after data augmentation (Extended Data Table 3).

Network for axon segmentation. We designed the axon segmentation based on the nnU-Net architecture[9]. The self-configuring parameters and settings are automatically computed according to the different training datasets. Similar to the 3D U-Net backbone[10], the network includes 6-layers of encoder and 6-layers of decoder. Moreover, we introduce the axial attention module[11] in the last decoder layer for integrating self-attention to each axis independently. The kernel size is set as 3×3×3, with the instance normalisation and the activation function of LeakyReLU. The initial learning rate is set as 0.0003. The input image size is 128×128×128 voxels, which are randomly cropped from the original training cubes with the size of 150×150×150 voxels. Binary Cross Entropy loss with a Sigmoid layer (BCEWithLogitsLoss) is employed as the loss function during the network training. The loss function $\mathcal{L}_{AS}$ for axon segmentation can be formulated as:

$$\mathcal{L}_{AS} = \frac{1}{N}\sum_{n=1}^{N}\{-w[y_n \cdot log\sigma(x_n) + (1-y_n)\cdot log\,(1-\sigma(x_n))]\} \quad (1)$$

where N indicates the number of voxels in a 3D cube. σ indicates the Sigmoid function to normalise the predictions x in [0,1]. x indicates the predictions. y indicates the ground truth annotations. We set the pos_weight w as 1 when y equals to 0 (indicates the current voxel annotation is background), and w as 3 when y equals to 1 (indicates the current voxel annotation is axon). For the sparse axon cubes, we use the skeletonised annotations for the deep model training, to keep the tree-topological structure of axons. For extremely dense axon

cubes, the skeletonised annotations are also quite dense and cannot reflect the real structure of axons. Hence, we use binarised annotations for the deep model to recognise axons and artefacts. The network is trained within 550 epochs. The number of cubes used for training is summarised in Extended Data Table 3. In general, based on 1000 cubes after data augmentation, the network training can be finished within 12 hours. The prediction of a whole-brain (e.g., with the size of 2000×2500×2000 voxels) for all stained axons can be finished within 6 hours.

In the testing phase, six large-sized cubes with the size of 600×600×225 voxels are used for quantitative evaluation, including two cubes from Sert-Stanford, two cubes from Sert-NIBS, and two cubes from GABA-NIBS (Extended Data Fig.1C). For the evaluation metric, we first employ the Dice score, which has been widely used for the evaluation of most image segmentation and registration tasks. As the Dice score indicates the voxel-wise volumetric scores which cannot well evaluate the connectivity of axon tubular structures, we also introduce the ClDice based on the intersection of centrelines and axon volumes[12], which has also been employed for the evaluation of mouse brain vasculature segmentation[13]. Additionally, we also reported the Precision and ClPrecision to reflect whether methods can well identify axons and artefacts.

Brain style transfer. We develop a deep model of brain style transfer based on the CycleGAN[14]. Different from the original version of CycleGAN, we introduce a brain outline segmentation sub-network, namely CEA-Net (Extended Data Fig.3A), in the CycleGAN framework, for the segmentation and preservation of the brain outline between the input brain and the output style transferred brain. As an example, we employ a somata-stained LSFM brain as the experimental brain, and the Allen atlas as the atlas brain (Fig. 3A). In the brain style transfer framework, Generator A includes three convolutional layers and several residual blocks, which is trained to generate images from the LSFM brain style to Allen style. In the meantime, a CEA-Net is also trained for the brain outline segmentation of synthetic Allen images, with ground truth annotations of the brain outline from the original LSFM outline. This CEA-Net can keep the brain outline consistent between the synthetic Allen and the original LSFM. Generator B has the same architecture, which is trained to generate images from the Allen style to the LSFM brain style. The CEA-Net for brain outline segmentation is also embedded in Generator B, which can keep the brain outline consistent between the synthetic LSFM and the original Allen. Discriminator A includes five fully convolutional layers, which are trained to differentiate LSFM brain style images that are original or synthetic. Discriminator B has the same architecture, which is trained to differentiate Allen-style images that are original or synthetic. Accordingly, the overall loss function $\mathcal{L}_{BST}$ for brain style transfer includes the CycleGAN loss and the CEA-Net segmentation loss, which can be formulated as:

$$\mathcal{L}_{BST} = \mathcal{L}_{GAN}(G_A, D_B, X, Y) + \mathcal{L}_{GAN}(G_B, D_A, Y, X) + \mathcal{L}_{cyc}(G_A, G_B) + \mathcal{L}_{BCE}(X) + \mathcal{L}_{BCE}(Y) \quad (2)$$

$$\mathcal{L}_{GAN}(G_A, D_B, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[log(D_B(y))] + \mathbb{E}_{x \sim p_{data}(x)}[log(1 - D_B(G_A(x)))] \quad (3)$$

$$\mathcal{L}_{GAN}(G_B, D_A, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[log(D_A(x))] + \mathbb{E}_{y \sim p_{data}(y)}[log(1 - D_A(G_B(y)))] \quad (4)$$

$$\mathcal{L}_{cyc}(G_A, G_B) = \mathbb{E}_{x \sim p_{data}(x)}[\|G_B(G_A(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G_A(G_B(y)) - y\|_1] \quad (5)$$

$$\mathcal{L}_{BCE}(X) = -\sum_{i=1}^{N} x_n \cdot log(\bar{x}_n) + (1 - x_n) \cdot log\,(1 - \bar{x}_n) \quad (6)$$

$$\mathcal{L}_{BCE}(Y) = -\sum_{i=1}^{N} y_n \cdot log(\bar{y}_n) + (1 - y_n) \cdot log\,(1 - \bar{y}_n) \quad (7)$$

where the $\mathcal{L}_{GAN}$ indicates the adversarial losses, L_{cyc} indicates the cycle consistency loss, following the settings of the CycleGAN. L_{BCE} indicates the segmentation loss for brain outline segmentation and preservation. G_A, G_B, D_A, D_B indicates Generator A, Generator B, Discriminator A and Discriminator B respectively. X and Y indicate the LSFM brain image datasets and Allen brain image datasets respectively. $\bar{x}_n$ indicates the predictions X, where x_n indicates the ground truth. $\bar{y}_n$ indicates the predictions Y, where y_n indicates the ground truth.

In our implementation, the brain style transfer is completed for each 2D image slice, e.g., transferring hundreds of brain slices to Allen atlas style. The slices for the style transfer can

be either from coronal, horizontal, or sagittal views. In our experiments, all style-transferred models are trained on brain images in coronal view, with a fixed image size of 320×448 pixels. We trained style transfer models based on different experimental brain datasets and brain atlas. During the style transfer training, the learning rate is set as 0.0002, with a batch size of 1. The network is trained within 200 epochs. One experimental brain with around 500 slices in coronal view is enough to train a style transfer model with a brain atlas. The deep model training for the brain style transfer can be finished within 12 hours.

Brain region segmentation.

Network for brain region segmentation. The backbone of the brain region and outline segmentation is CEA-Net (Extended Data Fig.3A). CEA-Net is originally from CE-Net[15], which is a recently developed semantic segmentation model including two additional modules in comparison with the U-Net architecture, i.e., dense atrous convolution (DAC), and residual multi-kernel pooling (RMP), which can better investigate local to global context cues for brain region segmentation. Based on the CE-Net, we introduce the attention gates[16] following the RMP module, which can better learn to segment brain regions with varying shapes and sizes. Considering the limited access of brain images with ground truth manual annotations, we extend CEA-Net in a semi-supervised manner, namely Semi-CEA (Extended Data Fig.3B), based on the most commonly used semi-supervised benchmarks, i.e., Mean Teachers[17]. Semi-CEA includes two models, i.e., the student model and the teacher model, where the parameters in the teacher model are first obtained from the student model trained in annotated data, by the exponential moving average (EMA) weights. Then a consistency loss (i.e., mean square error, MSE) is computed based on the original prediction from the student model and the noisy prediction from the teacher model (i.e., prediction for the image with π angle rotation). Moreover, as some brain regions cannot be well identified in a single view (e.g., HPF, CP, CBX), we further develop a novel Multi-view Semi-CEA framework for more accurate brain region segmentation (Fig. 3C). In our experiment, we train the Multi-view Semi-CEA in coronal and horizontal views. For a specific brain region, the predictions in coronal and horizontal views are combined in 3D for the computation of multi-view MSE loss. Accordingly, the overall loss function is the combination of the MSE loss and the semi-supervised segmentation loss (i.e., BCEWithLogitsLoss) in two views:

$$\mathcal{L}_{BRS} = \mathcal{L}_{cor} + \mathcal{L}_{hor} + \mathcal{L}_{3D} \quad (8)$$

$$\mathcal{L}_{cor} = \mathcal{L}_{hor} = \mathcal{L}_{sup} + \mathcal{L}_{qua} \quad (9)$$

$$\mathcal{L}_{sup} = -\sum_{i=1}^{N}(x_i log\, y_i + (1 - x_i) log(1 - y_i) \quad (10)$$

$$\mathcal{L}_{qua} = \sum_{i=N}^{N+P}(\hat{y}_i - \hat{y}_i')^2 \quad (11)$$

$$\mathcal{L}_{3D} = \sum_{i=1}^{N}(y_i^c - y_i^{ht})^2 + \sum_{j=1}^{P}(\hat{y}_j^c - \hat{y}_j^{ht})^2 \quad (12)$$

where $\mathcal{L}_{cor}$ and $\mathcal{L}_{hor}$ indicate the segmentation loss in coronal and horizontal views respectively. $\mathcal{L}_{cor}$ and $\mathcal{L}_{hor}$ are computed in same way (i.e., Semi-Loss), which includes both the $\mathcal{L}_{sup}$ and $\mathcal{L}_{qua}$. The training set consists of *N* annotated brain image slices and *P* unannotated brain images. The $\mathcal{L}_{sup}$ indicates the supervised loss for annotated brain image slices (BCEWithLogitsLoss), where $Loss_{qua}$ indicates the quadratic loss function for unannotated brain image slices. x_i indicates the ground truth annotation, where y_i indicates the prediction of annotated brain region. $\hat{y}_i$ indicates the unannotated image predictions without π angle rotation, where $\hat{y}_i'$ indicates the unannotated image predictions after π angle rotation. Additionally, $\mathcal{L}_{3D}$ indicates the MSE loss of two views' prediction in 3D for annotated and unannotated images, where y_i^c, $\hat{y}_j^c$ indicate the annotated and unannotated image predictions in coronal view. y_i^{ht} and $\hat{y}_j^{ht}$ indicate the annotated and unannotated image predictions in coronal view which are transformed from the horizontal view.

When training the multi-view Semi-CEA network, the network input image size is unified as 448×320 pixels in the coronal view, and 512×448 pixels in the horizontal view. The initial learning rate is 0.0001, with a batch size of 16. RMSProp is adopted as the optimiser. The network is trained within 100 epochs. To facilitate the training of multiple brain regions, pairs

of brain regions are trained together, such as CB and BS, CTX and CBX, CP and HPF. The brain segmentation model can be trained within 8 hours. After the model training, one brain region with 400 slices in a whole brain can be quickly predicted within 1 minute.

In the experiment of LSFM autofluorescence brain region segmentation, the Allen atlas brain slices in coronal view are first used for training the multi-view Semi-CEA model. Then the 12 LSFM brains are used for evaluation, where D-LMBmap transfers the 12 LSFM brains with Allen style for a more accurate prediction (Fig. 3E). We also train the multi-view Semi-CEA model based on four LSFM autofluorescence brains, with one from somata-stained, one from nuclei-stained, one from Sert-Stanford, and one from Sert-NIBS. The remaining eight brains are used for evaluation (Fig. 3F, Extended Data Fig.5B). In the experiment of MRI brain region segmentation, we use only one MRI brain for the multi-view Semi-CEA training. Then seven MRI brains are used for evaluation (Extended Data Fig. 6A & B). As the MRI dataset doesn't annotate the brain region of CBX, we only report the Dice score on the other five brain regions. In the experiment of LSFM stained-specific brain region segmentation, we use the deep model trained on Allen atlas for evaluation, where three LSFM stained-specific brains are transferred to Allen style before brain region segmentation (Extended Data Fig. 7A & B).

Whole-brain registration.

Network architecture for whole-brain registration. We designed a multi-scale and multi-constraints deep neural network for the whole-brain 3D registration (Fig. 4A). As an example, we employ a somata-stained brain as the source brain, and the Allen atlas as the reference brain. The network inputs include the original LSFM brain, the LSFM brain with Allen style, the segmented major brain regions of the LSFM brain, the Allen atlas, and corresponding brain regions in the Allen atlas. The initial brain size is unified as 320×456×528 voxels, which is consistent with the original size of Allen atlas. The network down-samples all the inputs twice to 160×228×264 voxels and 80×114×132 voxels. Then the inputs in the minimum resolution (80×114×132 voxels) are first trained with a rigid transformation network, including a 9-layer convolution for feature extraction and a 2-layer convolution for rotation and translation matrix computation. Similar to the architecture of rigid networks, the affine transformation network is trained to learn the deformation and translation matrix. The rigid and affine transformations in the minimum resolution inputs are also applied to higher resolution, including the 160×228×264 voxels and the 320×456×528 voxels sized inputs. Subsequently, we extended the VoxelMorph network[18] in a multi-scale format, for the training of non-rigid deformation from the source brain to the reference brain. The VoxelMorph is first trained on the minimum resolution, with 10-layers of convolutional encoders for feature extraction of inputs, and 10-layers of convolutional decoders for the computation of deformation fields. The corresponding layers are connected by the concatenated skip connections. After training on the minimum resolution, the network parameters are fixed and used for the training of inputs with higher resolution. There are 12-layers of both convolution encoders and decoders in the second resolution network (i.e., 160×228×264 voxels), and 14-layers of both convolutional encoders and decoders in the third resolution network (i.e., 320×456×528 voxels). The loss function in the rigid network is the mean square error (MSE) loss, while the affine network includes the MSE loss and the regularization loss, for the similarity measuring between the source and reference brains. The loss function in the VoxelMorph network includes an unsupervised loss and an auxiliary data loss, following the settings from the original VoxelMorph network. The network is trained by integrating the above loss functions, where the overall loss function $\mathcal{L}_{WBR}$ for whole-brain registration can be formatted as:

$$\mathcal{L}_{WBR} = \mathcal{L}_r + \mathcal{L}_a + \mathcal{L}_v \quad (13)$$

$$\mathcal{L}_r = \mathcal{L}_{sim}$$

$$= \begin{cases} \mathcal{L}_{MI}(X,Y) = -\sum_{x\in V_X}\sum_{y\in V_Y} p(x,y)log\left(\frac{p(x,y)}{\sum_{x\in V_X} p(x,y)\sum_{y\in V_Y} p(x,y)}\right), \text{for the original brain} \\ \mathcal{L}_{CC}(X,Y) = \frac{1}{N}\sum_{j=1}^{N} \frac{\sum_{i\in\delta_j}(X_i-\bar{X}_i)(Y_i-\bar{Y}_i)}{\sqrt{\sum_{i\in\delta_j}(X_i-\bar{X}_i)^2\sum_{i\in\delta_j}(Y_i-\bar{Y}_i)^2}}, \text{for style transferred brain} \\ \mathcal{L}_{MSE}(X,Y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i-Y_i)^2}, \quad \text{for brain regions} \end{cases} \tag{14}$$

$$\mathcal{L}_a = \mathcal{L}_{sim} + \mathcal{L}_{ind} + \mathcal{L}_{rank} \tag{15}$$

$$\mathcal{L}_{ind} = \left|\left|A-I\right|\right|_F^2 + \left|\left|b\right|\right|_2^2 \tag{16}$$

$$\mathcal{L}_{rank} = |rank(A) - 1| \tag{17}$$

$$L_v = L_{sim} + L_{reg} + L_{inv} \tag{18}$$

$$L_{reg} = \left|\left|\nabla\Phi\right|\right|_1 \tag{19}$$

$$L_{inv} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\Phi_i - \Phi_i^{-1}\right)^2} \tag{20}$$

$$\Phi^{-1} = (-\Phi_x\odot\Phi, -\Phi_y\odot\Phi, -\Phi_z\odot\Phi) \tag{21}$$

where $\mathcal{L}_r$, $\mathcal{L}_a$, $\mathcal{L}_v$ indicate the loss function of rigid, affine, multi-scale VoxelMorph network respectively. In the rigid network, there are three types of similarity loss function ($\mathcal{L}_{sim}$) in measuring the source brain and the reference brain, i.e., the Mutual Information loss ($\mathcal{L}_{MI}$), the Cross Correlation loss ($\mathcal{L}_{CC}$), and the Mean Square Error loss ($\mathcal{L}_{MSE}$). X indicates the reference brain, where Y indicates the registered brain. For a brain location, $p(x,y)$ indicates the probability of voxel values as x in X and y in Y at the same position. V_X and V_Y indicates the two sets of all voxel values in X and Y respectively. δ_j indicates the cube with the size of $s\times s\times s$ and the centre of j. $\bar{X}_i$ indicates the average voxel value of δ_j in X. The $\mathcal{L}_{MI}$ is employed for the similarity measuring between the original source brain and the reference brain. The $\mathcal{L}_{CC}$ is employed for the similarity measuring between the style transferred source brain and the reference brain. The $\mathcal{L}_{MSE}$ is employed for the similarity measuring between each brain region in source brain and the reference brain. In the affine network, besides the similarity loss ($\mathcal{L}_{sim}$), it also includes two more items, i.e., L_{ind} for the penalization of the affine transformation from the identity, and L_{rank} for constraining brain scaling in the affine transformation matrix. A indicates the predicted affine transformation matrix, where b indicates the offset in the affine transformation. In the multi-scale VoxelMorph network, besides the similarity loss ($\mathcal{L}_{sim}$), it also includes two more items, i.e., L_{reg} for the constrain of the gradient in non-linear to keep smooth transformation, L_{inv} for the constraint of non-linear transformation space. The Φ indicates the non-linear transformation space, where Φ^{-1} indicates the approximate inverse transformation in the non-linear transformation space. $\odot$ indicates the interpolation.

For the multi-constraints, the original source brain, style transferred source brain, and the source brain regions, are set as inputs for the deep model training in different channels, which are used for the similarity computation with the corresponding reference brain and reference brain regions respectively. The original source brain is assigned with higher weights in the loss computation (i.e., 70%), while the style transferred source brain and brain regions are assigned with lower weights in the loss computation (i.e., 15% respectively). In each training batch, the above constraints' losses in different channels are integrated for model updating. Φ_x, Φ_y, Φ_z indicates the offset in the X, Y, and Z directions respectively.

During the network training, the initial learning rate in the rigid and affine network is set as 0.0001. The initial learning rate in the VoxelMorph is set as 0.01. Adam is adopted as the optimiser, with a batch size of one. In the training on the minimum resolution inputs (i.e., 80×114×132 voxels sized brains), the training epoch is set as 1000. While in the training of the higher resolution inputs (i.e., 160×228×264 voxels and 320×456×528 voxels sized brains), the training epoch is set as 300.

In the experiment of LSFM auto-fluorescence brain registration to Allen atlas, nine LSFM brains are used for multi-scale and multi-constraints deep registration model training, where the constraints of brain regions are directly obtained by the automated brain region segmentation. Then the 12 LSFM brains with manual brain region annotations are used for evaluation (Fig. 4C, Extended Data Fig. 8A). In the experiment of MRI brain registration, five MRI brains are used for model training, where the constraints of brain regions are obtained from the MRI datasets by manual annotations. Then three MRI brains are used for evaluation (Extended Data Fig. 8B, Extended Data Fig. 10A). In the experiment of LSFM stained-specific brain registration, we directly used the deep model trained on the LSFM auto-fluorescence brains for evaluation, where the inputs are the style-transferred LSFM stained-specific brains (Allen-style) (Extended Data Fig.10B).

Code and Data Availability

The source code of all D-LMBmap modules, including automated axon segmentation, brain style transfer, brain region segmentation, whole-brain 3D registration, along with the executable files of D-LMBmap software and sample data can be found at D-LMBmap's Github page (https://github.com/lmbneuron/D-LMBmap). All the automated annotated and manual annotated samples are also available at the Github page. All the LSFM brain data will be available upon request. All the MRI brain data are available at https://github.com/dmac-lab/mouse-brain-atlas.

Software and Algorithms

Software	Developer	Link
PyTorch	The Linux Foundation	https://pytorch.org
PyCharm	JetBrains	https://www.jetbrains.com/
Anaconda	Anaconda Inc	https://www.anaconda.com/
Python3.8.8	Python Software Foundation	https://www.python.org/
ImageJ (Fiji)	National Institutes of Health	https://imagej.nih.gov/ij/index.html
Elastix	Image Sciences Institute	https://elastix.lumc.nl/
IMARIS	Oxford Instruments	https://imaris.oxinst.com/
Vaa3D	Allen Institute	https://alleninstitute.org/what-we-do/brain-science/research/products-tools/vaa3d/
ITK-SNAP	Paul A. Yushkevich et al. 2006[19]	www.itksnap.org
Allen Institute's Common Coordinate Framework (CCF)	Allen Institute's for Brain Science	http://atlas.brain-map.org
GraphPad Prism	GraphPad	https://www.graphpad.com/

Reference

1. Ren, J. *et al.* Single-cell transcriptomes and whole-brain projections of serotonin neurons in the mouse dorsal and median raphe nuclei. *eLife* **8**, e49424 (2019).
2. DeNardo, L. A. *et al.* Temporal evolution of cortical ensembles promoting remote memory retrieval. *Nat Neurosci* **22**, 460–469 (2019).
3. Kebschull, J. M. *et al.* Cerebellar nuclei evolved by repeatedly duplicating a conserved cell-type set. *Science* **370**, eabd5059 (2020).
4. Renier, N. *et al.* Mapping of Brain Activity by Automated Volume Analysis of Immediate Early Genes. *Cell* **165**, 1789–1802 (2016).
5. Holmes, H. E. *et al.* Comparison of In Vivo and Ex Vivo MRI for the Detection of Structural Abnormalities in a Mouse Model of Tauopathy. *Front. Neuroinform.* **11**, (2017).
6. Yun, S. *et al.* CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 6022–6031 (IEEE, 2019). doi:10.1109/ICCV.2019.00612.
7. Shapira, D., Avidan, S. & Hel-Or, Y. Multiple histogram matching. in *2013 IEEE International Conference on Image Processing* 2269–2273 (IEEE, 2013). doi:10.1109/ICIP.2013.6738468.
8. Kao, W.-C., Hsu, M.-C. & Yang, Y.-Y. Local contrast enhancement and adaptive feature extraction for illumination-invariant face recognition. *Pattern Recognition* **43**, 1736–1747 (2010).
9. Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J. & Maier-Hein, K. H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods* **18**, 203–211 (2021).
10. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (eds. Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G. & Wells, W.) vol. 9901 424–432 (Springer International Publishing, 2016).
11. Ho, J., Kalchbrenner, N., Weissenborn, D. & Salimans, T. Axial Attention in Multidimensional Transformers. Preprint at http://arxiv.org/abs/1912.12180 (2019).
12. Shit, S. *et al.* clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation. in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 16555–16564 (IEEE, 2021). doi:10.1109/CVPR46437.2021.01629.
13. Todorov, M. I. *et al.* Machine learning analysis of whole mouse brain vasculature. *Nat Methods* **17**, 442–449 (2020).
14. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. in *2017 IEEE International Conference on Computer Vision (ICCV)* 2242–2251 (IEEE, 2017). doi:10.1109/ICCV.2017.244.
15. Gu, Z. *et al.* CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Trans. Med. Imaging* **38**, 2281–2292 (2019).
16. Oktay, O. *et al.* Attention U-Net: Learning Where to Look for the Pancreas. Preprint at http://arxiv.org/abs/1804.03999 (2018).
17. Tarvainen, A. & Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. 10.
18. Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J. & Dalca, A. V. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. *IEEE Trans. Med. Imaging* **38**, 1788–1800 (2019).
19. Yushkevich, P. A. *et al.* User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage* **31**, 1116–1128 (2006).