Self-Supervised Deep Learning Two-Photon Microscopy: supplemental document

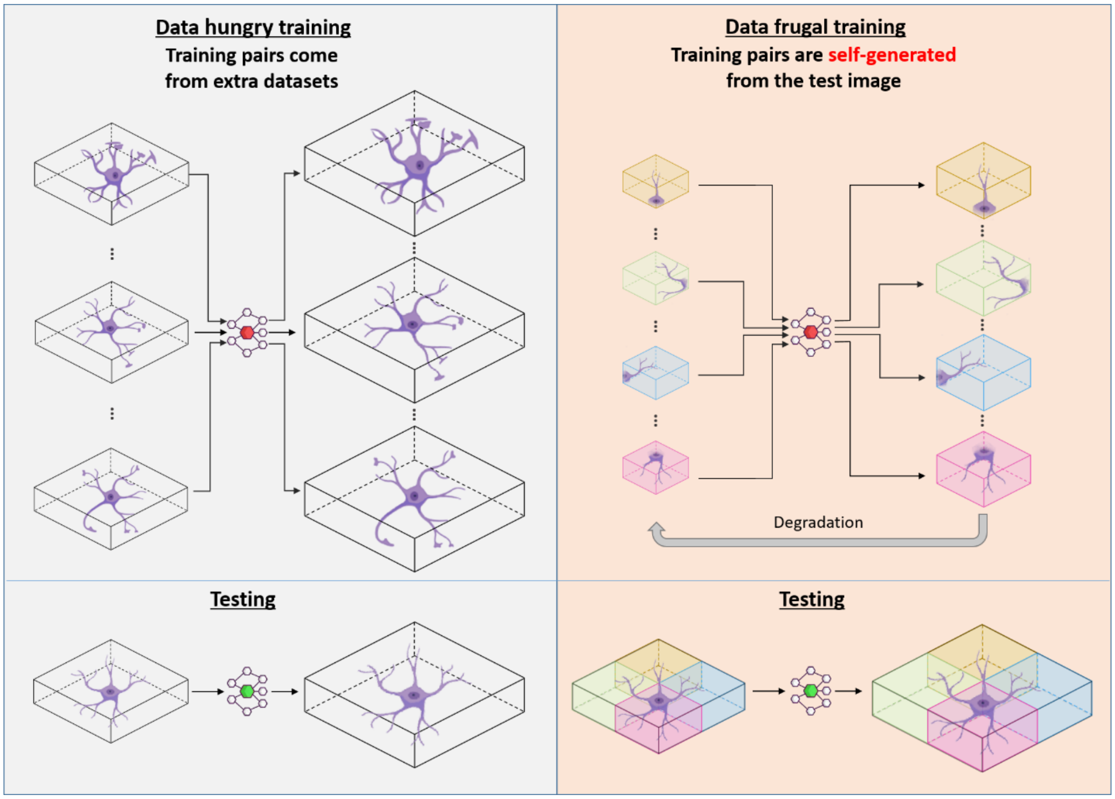
Data hungry training
Training pairs come from extra datasets
Data frugal training
Training pairs are self-generated from the test image
Degradation
Testing
Testing

Fig. S1 A comparison between two training strategies.

Evaluation of simulated beads images. A volume of size $512 \times 512 \times 60$ px containing 500 beads is first generated as the original image (shown in **Supplementary Fig. 2(a)** and **2(e)**). To simulate the image acquisition process, a low-resolution image is produced by downsampling the original image by a factor of four in all dimensions, resulting in an input of size $128 \times 128 \times 15$. The low-resolution image is then fed to the network. After training using augmented patches, the network generates the high-resolution image shown in **Supplementary Fig. 2(d)** (lateral) & **2(h)** (axial). The results from interpolation are shown in **Supplementary Fig. 2(c)** and **2(g)** for comparison. From both the lateral and axial images, we can see that the network output has improved contrast against the background, and the beads display a smoother profile. To quantify the improvements in terms of resolution, 30 beads from a lateral plane were used to calculate the full width at half maximum (FWHM); the average line profile is shown in **Supplementary Fig. 3**. It should be noted that the horizontal axis of **Supplementary Fig. 3** is the pixel (px) number of the simulated beads, not the actual size. Laterally, the input FWHM is 8.43 px. The mean of the inferred FWHM is 6.79 which matches the ground truth (mean FWHM = 6.70). Similar improvements can be seen axially, where the average FWHM is reduced from 14.51 to 13.32 px.

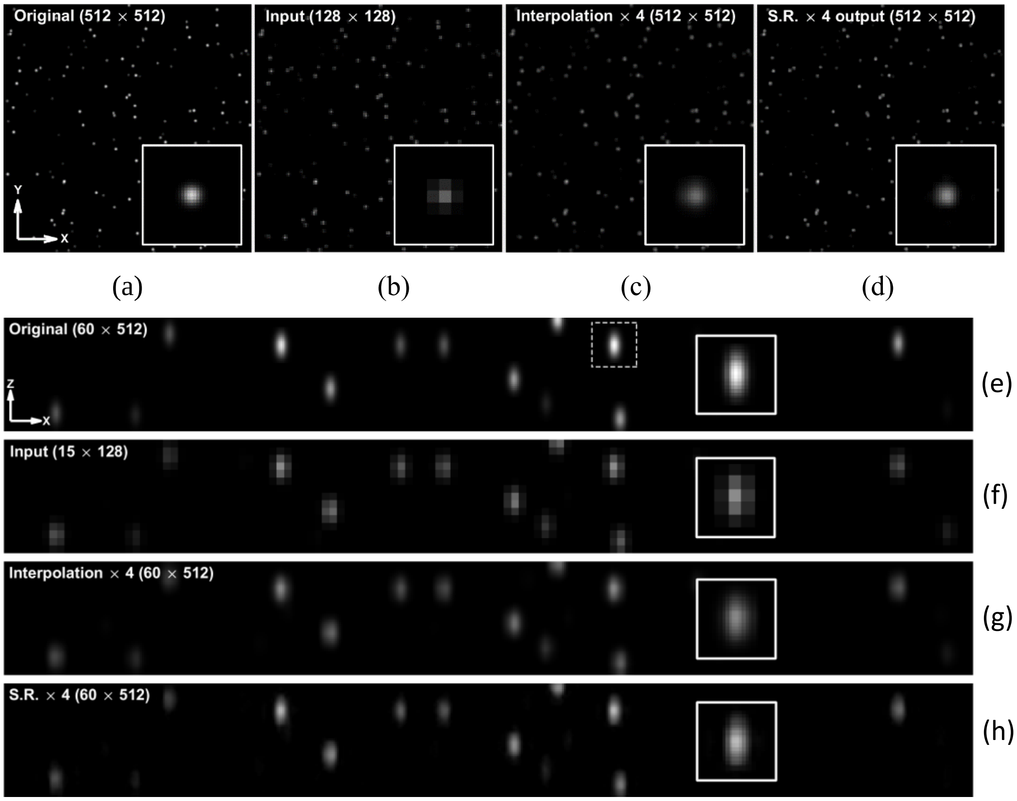

Fig. S2 Network output (lateral) and network output (axial) for simulated bead images. Lateral (a)–(d) and axial (e)–(h) bead images. Comparison between the input (b) & (f) and the network output (d) & (h). The insets show a representative profile from a single bead.

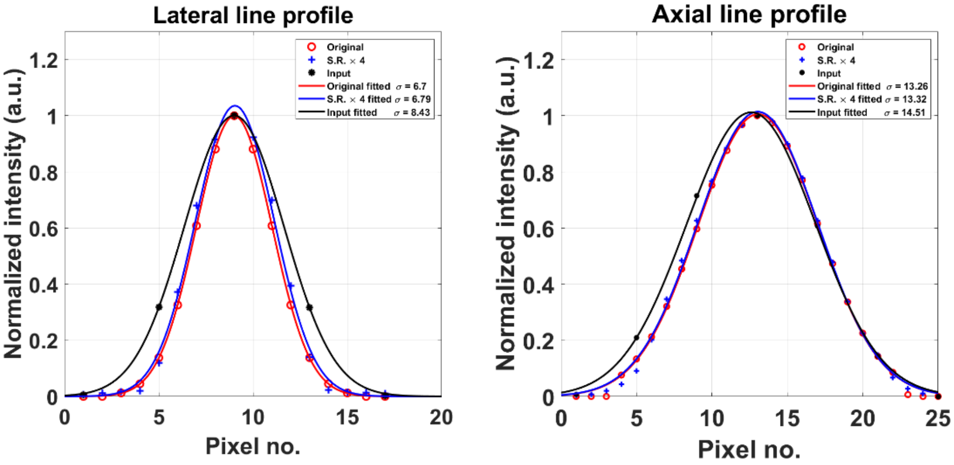

Fig. S3 Lateral and axial FWHM of the beads. Lateral line profiles shown on the left and axial line profiles shown on the right.

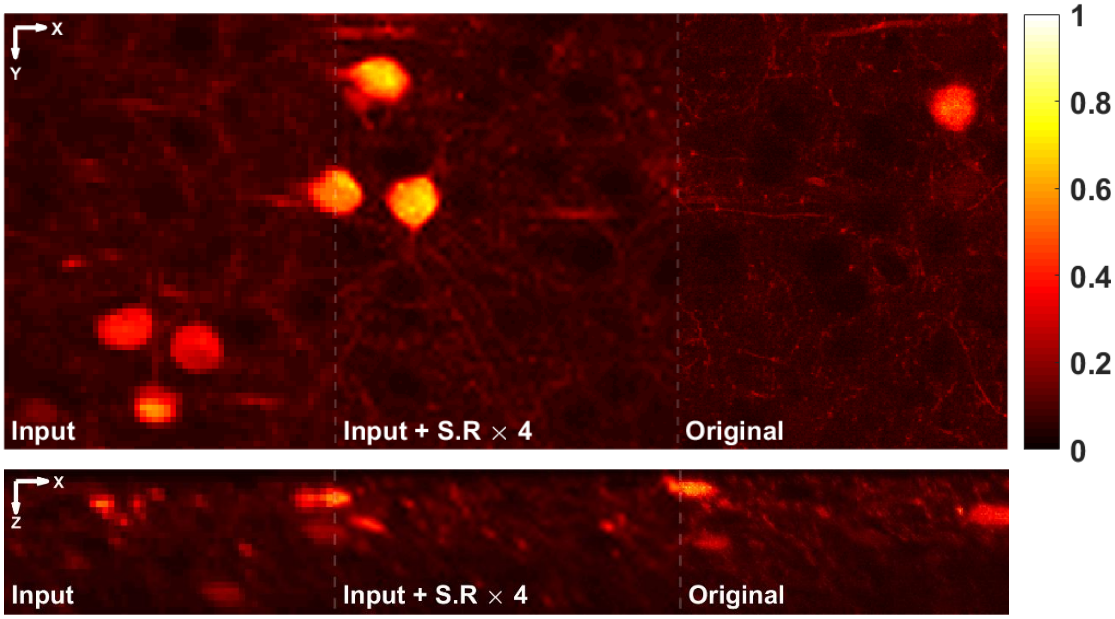

Fig. S4 Transition from low resolution (coarse input, left) to high resolution (clear output, middle) using a super-resolution network. The original image is shown on the right for reference. Lateral results are shown in the top section and axial results are shown in the bottom section.

Supplementary Media 1 The movie shows the transition from a low-resolution input image to a super-resolution network-enhanced output (see file '1-low-resolution with super-resolution.mov').

Supplementary Media 2 The movie shows the transition from a high-resolution input image to a super-resolution network-enhanced output (see the file '2-high-resolution with super-resolution.mov').

Supplementary Media 3 The movie shows the transition from a 3D (lateral) high-resolution input image to a super-resolution network-enhanced output see file '3-xy_3d_super-resolution.mov').

Supplementary Media 4 The movie shows the transition from a 3D (axial) high-resolution input image to a super-resolution network-enhanced output (see the file '4-yz_3d_super-resolution.mov).

Supplementary Note 1 Data augmentation and Self-Vision training

Data augmentation is crucial for Self-Vision training because the only data source is the low-resolution input, I_{input}. The first step of the augmentation is to crop I_{input} into patches I_p. We allowed overlapping when creating I_p so that the number of resulting patches would be greater. The resulting patches are augmented by random flipping and/or rotation by $90°/180°/270°$. To train the network, pairs of low-resolution images and their high-resolution references are generated. The low-resolution images, I_{p_lr}, are created from down sampling I_p using 'cubic' down-sampling methods with no image registration being required. Subsequently, I_p is paired with I_{p_lr} in a batch of eight and used for training. The typical epoch number for training a 3D image (output size $1024 \times 1024 \times 152$) is 1500, using a 0.0001 learning rate. For other training-related hyperparameters, refer to the configuration file in the code repository. Hyper-parameter optimisation is performed using the Ray Tune package; the search space also being provided in the automated script.

Supplementary Note 2 Training data preparation for DFCAN, PSSR, and DSP-Net

For the Thy1-GFP mouse brain specimen, raw data from five distinct regions of interest were imaged, from which four regions were used for network training and the remaining region was used for evaluation. Again, Self-Vision does not use any data from the four training regions. Each training region is composed of at least 135 2D images of size 1024×1024, resulting in 543 raw training images. The test region contains 152 images of size 1024×1024. All low-resolution images are generated by downsampling $4\times$ the original images. The implementations of the three networks are available online (we used the original implementations). The three networks have different methods for processing and augmenting training data. For instance, the 3D super-resolution network DSP-Net uses two-stage processing (denoising and up-sampling) techniques, which require two sets of training data. Consequently, we followed the instructions provided by the authors to generate training data.

Supplementary Note 3 Evaluation metrics

Two classical metrics, PSNR and SSIM, were used to quantify network performance. The PSNR is inversely proportional to the MSE. PSNR can be expressed as follows:

$$\mathrm{PSNR} = 20 \cdot \log_{10}\left(MAX_I\right) - 10 \cdot \log_{10}\left(MSE\right) \tag{S1}$$

where MAX_I is the maximum possible pixel value in the image. Our network uses an 8-bit input image; therefore, $MAX_I = 255$ in our case.

The SSIM measures the perceptual difference between two images using luminance, contrast, and structure. It follows that:

$$\mathrm{SSIM}\left(\mathrm{x}, \mathrm{y}\right) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{S2}$$

where μ_x, μ_y, σ_x, and σ_y are the mean, standard deviation, and cross-covariance for images x, y. By default, $C_1 = (0.01L)^2, C_1 = (0.03L)^2$ where L specifics the dynamic range.