Physics-based Deep Learning for Imaging Neuronal Activity via Two-photon and Light Field Microscopy

Herman Verinaz-Jadan, *Student Member, IEEE,* Carmel L. Howe, *Member, IEEE,* Pingfan Song, *Member, IEEE,* Flavie Lesept, *Member, IEEE,* Josef Kittler, *Member, IEEE,* Amanda J. Foust, *Member, IEEE,* and Pier Luigi Dragotti, *Fellow, IEEE*

***Abstract*—Light Field Microscopy (LFM) is an imaging technique that offers the opportunity to study fast dynamics in biological systems due to its 3D imaging speed and is particularly attractive for functional neuroimaging. Traditional model-based approaches employed in microscopy for reconstructing 3D images from light-field data are affected by reconstruction artifacts and are computationally demanding. This work introduces a deep neural network for LFM to image neuronal activity under adverse conditions: limited training data, background noise, and scattering mammalian brain tissue. The architecture of the network is obtained by unfolding the ISTA algorithm and is based on the observation that neurons in the tissue are sparse. Our approach is also based on a novel modelling of the imaging system that uses a linear convolutional neural network to fit the physics of the acquisition process. We train the network in a semi-supervised manner based on an adversarial training framework. The small labelled dataset required for training is acquired from a single sample via two-photon microscopy, a point-scanning 3D imaging technique that achieves high spatial resolution and deep tissue penetration but at a lower speed than LFM. We introduce physics knowledge of the system in the design of the network architecture and during training to complete our semi-supervised approach. We experimentally show that in the proposed scenario, our method performs better than typical deep learning and model-based reconstruction strategies for imaging neuronal activity in mammalian brain tissue via LFM, considering reconstruction quality, generalization to functional imaging, and reconstruction speed.**

***Index Terms*—Light Field Microscopy, deep learning, model-based learning, deconvolution.**

I. Introduction

STUDYING the rapid dynamics of hundreds of neurons in brain tissue poses a challenge for conventional microscopy imaging methods. Typical optical techniques struggle to achieve simultaneous 3D imaging of multiple neurons since they focus on a single plane or point in space. Furthermore, brain tissue scatters light, which increases the difficulty of capturing high-quality images of neurons.

Two-photon (2P) microscopy is an appealing modality for performing 3D imaging in brain tissue. Two-photon microscopy uses near-infrared illumination, which provides deeper tissue penetration and reduced scattering. Furthermore, it restricts excitation to a small volume, mitigating photobleaching and providing optical sectioning [2]. As explained in [2], 2P and onwards microscopy has already enabled imaging of neurons in scattering brain tissue. However, the point-scanning acquisition used in 2P microscopy limits the imaging speed and has largely restricted its use to planar recordings

On the other hand, light field microscopy (LFM) is a method to image 3D volumes with a 2D camera sensor in a single snapshot. LFM is a fast, non-scanning imaging modality with a volumetric acquisition rate limited only by the camera frame rate, indicator brightness, and required signal-to-noise ratio. This performance is achieved by placing a microlens array (MLA) between the tube lens and the camera sensor of a standard microscope, allowing for the acquisition of both angular and spatial information from the light simultaneously [3].

In LFM, a 3D image is reconstructed from a 2D light field (LF) image using computational reconstruction methods. However, using a single sensor array to capture 3D information limits the reconstruction quality due to the inherent trade-off between spatial and angular resolution [4], [5]. Therefore, recovering a high-quality 3D volume from a single 2D LF image is usually challenging. Furthermore, conventional model-based reconstruction methods for LFM require high computational time. The computation of the forward model and back projection (transpose) usually involves a large amount of computation due to the high number of views (hundreds) in the LF data, the lateral upscaling factor of the 3D volume, and the lack of shift-invariance in the system. Although new model-based approaches have recently emerged to address these issues [6], [7], [5], [8], the computational time required for these approaches still conflicts with the primary goal of LFM, which is to reconstruct 3D volume time series.

Reconstruction methods based on deep learning are potential candidates to solve typical problems in LFM [9], [10], [11]. However, current learning-based approaches are tested in controlled environments that are difficult to achieve in many realistic situations. For instance, when studying neuronal activity in mammalian brain tissue, the sample is highly scattering, non-transparent and contains high background noise, which makes training artificial neural networks (ANN) challenging. On the other hand, model-based optimization approaches have shown to be more robust in these adverse experimental

This work is in part supported by BBSRC BB/R009007/1 and in part by the Royal Academy of Engineering under the RAEng Research Fellowship scheme (RF1415/14/26). Flavie Lesept is supported by Marie Skłodowska-Curie Fellowship (707478). Herman Verinaz-Jadan was in part supported by SENESCYT. (Corresponding author: Herman Verinaz-Jadan.)

The work in this paper was in part presented at [1].

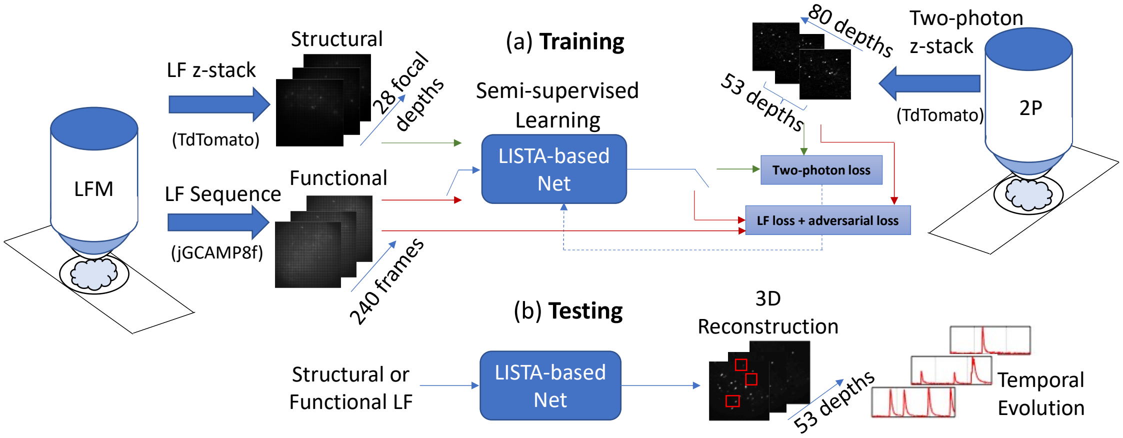

Fig. 1: Overview of our approach.In part (a), we show the experimental setup used for training. We acquire a two-photon 3D image (80 depths), along with the focal stack of LF images for a single brain sample, tagged with the TdTomato fluorophore. This data shows the anatomy of the network of neurons without any temporal activity, called structural imaging. We reconstruct 53 depths from each LF, spanning a range of 104 μm. Since the two-photon stack contains 80 depths over a depth range of 158 μm, we acquire $(28 = 80 - 53 + 1)$ LF at different depths, producing a labelled dataset of size 28. In addition, we collect unlabelled LF images from brain samples tagged with the jGCaMP8f protein that encodes calcium responses, an indirect measurement of electrical brain activity known as functional imaging. For training, we use three jGCaMP8f sequences of 80 frames from three different samples, and input structural and functional LF into the network at different times. The network is designed based on the unfolding of the ISTA algorithm. We use a training loss that exploits the forward model regularized with an adversarial loss. Testing is performed on structural LF stacks or functional LF sequences, and we produce one volume per frame. Neuronal activity is extracted from these volumes, as illustrated in part (b) of our approach

conditions [12], [13].

This work proposes a novel multimodal imaging approach leveraging the respective strengths of 2P microscopy and LFM. We label neurons in the mouse brain tissue using two types of fluorescent proteins: tdTomato and jGCaMP8f. TdTomato captures the static neuron distribution in space, disregarding its activity. On the other hand, the jGCaMP8f is an indicator of calcium concentration which indirectly measures electrical, and therefore functional, activity in the brain. In our setting, we capture the distribution of the neurons in a 400-μm-thick brain slice labelled with the tdTomato protein at high resolution using 2P microscopy. Similarly, the LF microscope captures the corresponding LF images for different focal depths. This approach gives us a labelled dataset. In addition, multiple LF temporal sequences are captured from different brain samples using jGCaMP8f protein. In this case, the ground truth data is unavailable since scanning-based techniques are not fast enough to capture the temporal activity of multiple neurons revealed by the jGCaMP8f in 3D space. Thus, the small labelled dataset and a fraction of the LF temporal sequences are used to train our network in a semi-supervised manner, as shown in Figure 1. After training, our network can reconstruct volume time series from LF sequences with high accuracy and speed, despite the fact that the temporal LF sequences are obtained with the jGCaMP8f protein, for which we do not have a labelled dataset for training.

We achieve these results by introducing a deep neural network whose architecture is driven by a proper modelling of the forward model and the fact that labelled neurons in tissue are sparse. We leverage the sparsity assumption to design the architecture by unfolding the ISTA algorithm. Deep unfolding has been successfully applied to other modalities such as compressive sensing, image superresolution [14], [15], [16], and microscopy modalities such as ultrasound localization microscopy [17], super-resolution microscopy [18] or 3D deconvolution for microscopy [19]. Our reconstruction method introduces a deep unfolding approach for LFM that exploits the physics of the system in the design of the architecture and in the computation of a LF loss during training, which is regularized with an adversarial loss. Overall, this approach allows us to exploit the best of two optical techniques to achieve the accurate and fast reconstruction of volume time series of neuronal activity in mammalian brain tissue.

II. Previous Work

LF imaging has found many interesting applications due to its unique capability of recording the direction of arrival and the location of the light rays. It has been explored for post-capture refocus, change of point of view, change of focal length, and for robot navigation [20]. Levoy et al. proposed extending LF imaging to microscopy by adapting the Plenoptic 1.0 configuration used for cameras to microscopes [3]. This application is particularly interesting because it allows for capturing volume time series at the camera frame rate, enabling the study of microscopic biological systems, such as networks of neurons in brain tissue.

LFM is a technique that relies on computational approaches to perform the reconstruction of 3D images. Reconstruction techniques in LFM differ from those in photography, in that they must consider wave-optics for an accurate system description, whereas the ray-optics model is enough in photography. The first computational approach proposed for reconstruction was presented by Levoy et al. in their pioneering work on LFM [3]. Levoy et al. algorithm performs digital refocusing to obtain a raw focal stack that is then sharpened by a 3D deconvolution process. Later, Broxton et al. [4] proposed an improvement in the reconstruction by computing the measurement matrix of the system to state a linear inverse problem that is solved using the Richardson-Lucy (RL) algorithm. Currently, many conventional model-based reconstruction strategies rely on similar approaches [21], [22][6][7].

Novel model-based methods have also emerged recently. One method that alleviates reconstruction artifacts and is faster than conventional RL strategies is proposed in [5]. This approach exploits the low-rank nature of the measurement matrix and uses the Alternating Direction Method of Multipliers (ADMM) to impose additional priors for reconstruction. Moreover, alternative model-based approaches exist that only focus on recovering point-like sources, as proposed in [8].

Apart from model-based methods, various approaches that exploit deep learning for reconstruction have been proposed recently. In [9], Wang, et al. describe the first approach that uses an end-to-end convolutional neural network (CNN) for reconstruction. The VCD-Net network is a 2D U-Net trained using synthetic LF data and 3D images obtained with confocal microscopy as labels. The VCD-Net is tested on real LF data by imaging neuron activity in C. elegans and blood flow in the heart of zebrafish larvae. Later, a technique that uses a mixed reconstruction approach was proposed by Li et al. in [23]. The network named deepLFM is designed to enhance the reconstruction obtained after a few RL iterations on LF images. DeepLFM is a 3D U-Net trained and tested using labels obtained by 3D imaging K562 cells with confocal fluorescence microscopy. Then, Page et al. proposed 3D reconstruction from LF images using a network based on a 2D U-net named LFMNet [10]. LFMNet is trained on real LF data and 3D stacks obtained via confocal microscopy. The training data is obtained after imaging brain slices with fluorescently labelled blood vessels. Finally, a convolutional neural network (CNN) named HyLFM that can be retrained to refine the 3D reconstruction with the aid of an additional selective-plane illumination microscopy (SPIM) image has been proposed in [11]. HyLFM is trained on real LF images and SPIM stacks as labels. HyLFM is specifically tested to image medaka heart dynamics and zebrafish neuronal activity.

Works exploring deep-learning methods for LFM show that these techniques are faster and perform better than classic iterative approaches if they are evaluated in controlled scenarios. For instance, they need huge training datasets, low background noise, non-scattering media, or transparent samples. In contrast, model-based approaches are more robust than learning methods and helpful in adverse conditions, as shown in works studying mammalian brain tissue [12], [13]. Thus, we propose fusing appealing features from model-based and learning-based methods. We design an artificial neural network following the physics of the system. The network is trained in a semi-supervised manner by using an adversarial regularizer and exploiting the knowledge of the forward model. Our method achieves high-speed reconstruction after training, and it is robust when imaging scattering samples since it also relies on the knowledge of the forward model as in model-based approaches.

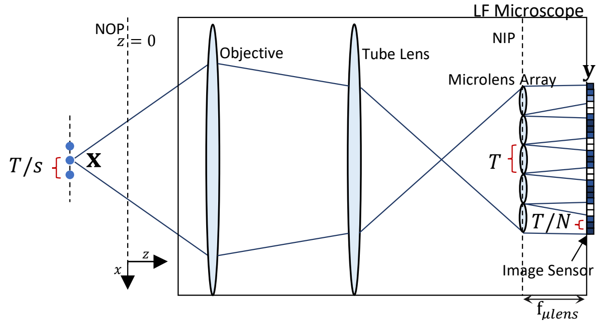

Fig. 2: Block diagram of a Plenoptic 1.0 LF system. A MLA is placed at the Native Image Plane (NIP) of a standard microscope and the sensor is placed at the focal plane of the MLA. A LF microscope maps a 3D image $\mathbf{x}$ into a 2D LF $\mathbf{y}$. The upsampling factor s and the one-dimensional number of pixels under each microlens N define the measurement matrix of the system. The optical conjugate plane of the NIP is referred to as the Native Object Plane (NOP).

III. Problem Formulation

A LF microscope can be described as a linear operator [4], [5]. For any input $x(\mathbf{p})$, the intensity $y(\mathbf{x})$ observed (before discretization) at the image sensor can be described with a superposition integral as follows:

$$y(\mathbf{x}) = \int h(\mathbf{x}, \mathbf{p}) x(\mathbf{p}) d\mathbf{p}, \tag{1}$$

where the function $h(\mathbf{x}, \mathbf{p})$ is the impulse response of the system at any location $\mathbf{x} \in \mathbb{R}^2$ for a point source located at $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}}) \in \mathbb{R}^3$. The precise computation of the impulse response is explained in the supplemental material. A LF microscope differs from standard microscopy due to the MLA placed between the tube lens and the camera sensor (see Figure 2). The MLA makes the impulse response of the system depth dependent. This property allows LFM to map a 3D input to a single 2D image. Furthermore, the MLA also makes the impulse response $h(\mathbf{x}, \mathbf{p})$ periodic in the lateral dimensions $(x_{\mathbf{p}}, y_{\mathbf{p}})$, which implies that the system is periodic-shift invariant in these dimensions [5]. After discretization, a monochromatic LF microscope can be represented in matrix form as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \tag{2}$$

where matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$ maps a vectorized volumetric input $\mathbf{x} \in \mathbb{R}^n$ into a LF image $\mathbf{y} \in \mathbb{R}^m$. The number n of voxels of the volume is usually much larger than the number m of pixels of the LF image [10] [9] [11]. In our experiments, we set $m \approx 4 \times 10^6$ and $n \approx 5 \times 10^6$. In general, the size of $\mathbf{H}$ depends on the input and output sampling intervals, which are commonly chosen to be T/s and T/N, respectively (assuming unit lens magnification for simplicity). The constant T is the microlens pitch, s is an arbitrarily chosen upsampling factor, and N is the number of pixels under each microlens (per lateral axis). See Figure 2 for clarification.

If one shifts the input of the system laterally by T, the output is shifted by N pixels, as shown in Figure 3 (a). This behaviour

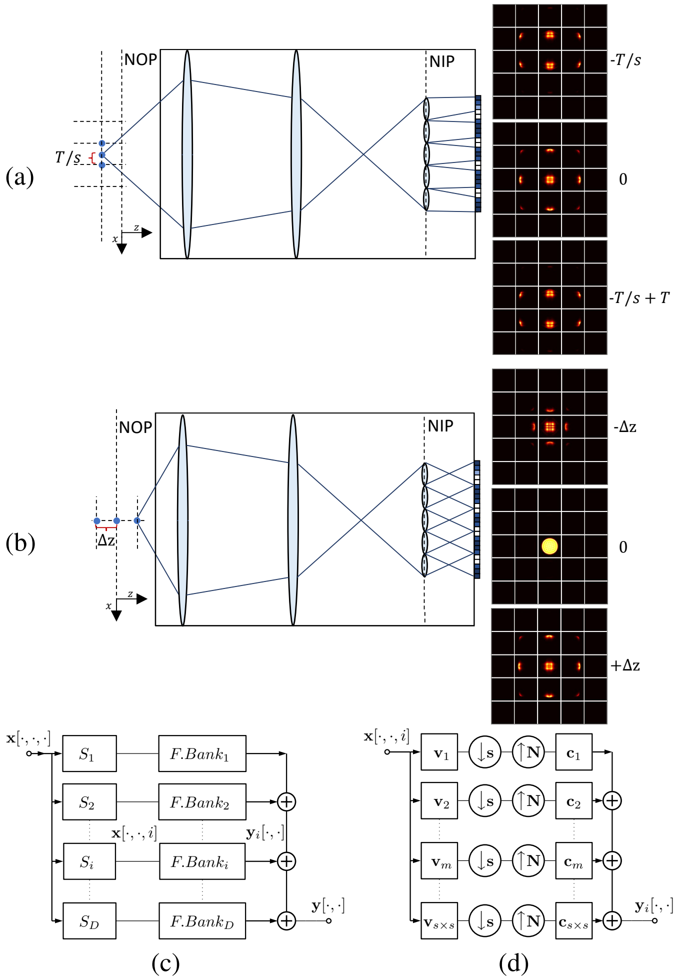

Fig. 3: Filter bank representation of the LF forward model. When the input of the system is shifted laterally by a multiple of T, the output is also shifted by T, as shown in (a). If the input is shifted along the z-axis, the output shows different patterns per depth, as shown in (b). Thus, the output of a LF microscope $\mathbf{y}$ can be described as the summation of the output of a group of filter banks. A slicing operator S_i chooses the respective i-th depth of the 3D volume, which is the input of the i-th filter bank that outputs a LF image $\mathbf{y}_i$, as in (c). As shown in (d), each filter bank has $s \times s$ branches, a downsampling factor of s and an upsampling factor of N for both lateral dimensions, where s was chosen arbitrary when computing the measurement matrix $\mathbf{H}$, and $N \times N$ is the number of pixels under each microlens.

only occurs for shifts which are multiple of T. Therefore, for a fixed depth, the forward model is periodically-shift invariant and, therefore, can be modelled by using a filter bank [5], as in Figure 3 (d). Furthermore, the impulse response is unique for each depth, which is the property that allows for localization of sources at different depths, as shown in Figure 3 (b). Thus, to describe the entire system, a different filter bank is required for each depth, as depicted in Figure 3(c). The input volume first undergoes slicing by operator S_i, which selects depth $z = i$ for $i = 1, 2, .., D$, where D is the number of depths. Then, each slice serves as input to the corresponding filter bank; i.e., for each $z = i$, the i-th filter bank outputs a LF image $\mathbf{y}_i$. The final output of the microscope $\mathbf{y}$ is the sum of all $\mathbf{y}_i$. It should be noted that the structure of each filter bank is related to the measurement matrix of the system. If there are $N \times N$ pixels under each microlens and the volume sampling interval for both lateral dimensions is T/s, then the filter bank structure has $s \times s$ branches. The downsampling factor is s, and the upsampling factor is N, as demonstrated in Figure3 (d). According to [5], the input and output filters of each branch can be obtained from the measurement matrix $\mathbf{H}$. It is important to note that the convolutions and filters are two-dimensional, and the downsampling factor s and upsampling factor N refer to both lateral dimensions.

In this work, we propose a reconstruction method to address the inverse problem derived from Equation (2) in light field microscopy (LFM). The conventional approach for reconstructing a 3D volume $\mathbf{x}$ from a single LF image $\mathbf{y}$ involves time-consuming RL-like algorithms. However, the study of the spatial and temporal behavior of neurons in brain tissue necessitates fast 3D reconstruction from LF sequences. While model-based reconstruction, as demonstrated in [7] and [5], has shown significant improvements in performance and speed, learning-based methods have the potential to achieve even better reconstruction quality and faster speed when trained appropriately in controlled scenarios.

IV. Forward model as a linear CNN

In this section, we propose a novel description of the LF system using convolutional layers. This description is fundamental for the derivation and implementation of our reconstruction method. We convert the filter-bank model to a linear Convolutional Neural Network (CNN) and propose architectures to perform the forward model computation efficiently.

A. 4D representation of the LF

The LF is conventionally represented as a 4D function in photography-related applications. Specifically, the 2D image captured with a LF camera is reordered into a 4D array that is a sampled version of the continuous LF function [24]. Two dimensions of the array represent horizontal and vertical spatial coordinates, while the other two represent horizontal and vertical spatial frequencies. The 4D LF can be interpreted as a collection of sub-aperture images or views, which are 2D images obtained when the spatial frequencies are fixed [24]. See Figure 4 (a) for clarification. In microscopy, the idea of capturing a 4D LF is still valid if the array is interpreted as a sampled version of a 4D Wigner distribution function, a generalization of the concept of LF that considers the effects of diffraction [25].

B. Linear CNN

The representation of the LF as a group of views has a convenient property. Unlike the 2D LF image, each view is not an abstract pattern. Instead, it preserves the structure of the original scene since it only carries spatial information. Furthermore, this multi-view representation is attractive for

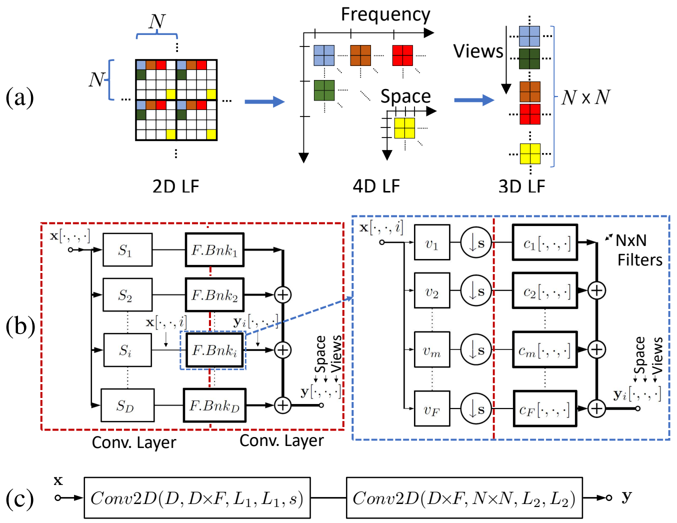

Fig. 4: Linear forward model. In (a), we show how a 2D LF can be transformed into a 4D or 3D representation. In the 4D representation, two coordinates represent space, and the other two represent spatial frequencies. The 3D representation consists of a set of $N \times N$ sub-aperture images, where $N \times N$ is the number of pixels under each microlens. In (b), we show the image formation of a 3D LF using filter banks. In this model, the upsampling blocks have been removed, and a group of $N \times N$ filters representing the view dimension has replaced the synthesis filters. In (c), we illustrate the representation of the LFM system as a forward CNN $f(\cdot)$. The architecture shown in (b) is a particular case of the CNN $f(\cdot)$. The notation $Conv2D(\cdot, \cdot, \cdot, \cdot, \cdot)$ represents a 2D convolutional layer with the following parameters ordered as follows: number of input channels, number of output channels, height and width of the filter, and stride. If the stride is omitted, it means unit stride.

3D reconstruction in LFM since it is more suitable to fit conventional CNN architectures, as also proposed in [26].

Since the 4D LF can be obtained by just rearranging pixels of the 2D LF, the filterbank description of the LF system described in Section III can be adjusted to explain the formation of a group of sub-aperture images. Specifically, reordering the synthesis filter of each branch allows simple computation of the sub-aperture images, as shown in Figure 4 (b). Note that this new representation follows the basic structure of the original filter bank in Figure 3 (a). However, there is no upsampling block since the original synthesis filter bank is replaced by a group of $N \times N$ filters that leads to multiple outputs forming a set of views or sub-aperture images.

The multiple-view filterbank description of the system can be conveniently implemented using convolutional layers. The LF system can be written as a feed forward network with 2D convolutional layers without bias terms, where the first layer has a stride given by the downsampling factor s and the second layer has a unit stride. We represent the CNN by $f(\cdot)$ and call it forward CNN (see Figure 4 (c) for clarification). Since the forward CNN is derived from the filter bank representation, the parameters of $f(\cdot)$ are related to the parameters of the microscope and the physics of the system: the number $N \times N$ of output channels is related to the number of pixels under each microlens, the number of input channels D is the number of depths, the filter size L_1 is equal to the downsampling factor s or stride [5], the filter size L_2 is instead given by the support of the PSF, specifically, if the support of the PSF related to the largest depth is M, then $L_2 = M/N$. Finally, the parameter F is related to the upsampling factor. If F is set to $s \times s$, it meets the theoretical model exactly, while if it is set to a smaller value, it performs an approximation, as explained in [5]. Note that in [5], the filter size L_1 is equal to s to allow finding the filters of the filter bank using SVD. In the model shown in Figure 4 (c), this constraint is optional since the weights can be found with any typical optimization package used in deep learning instead of SVD.

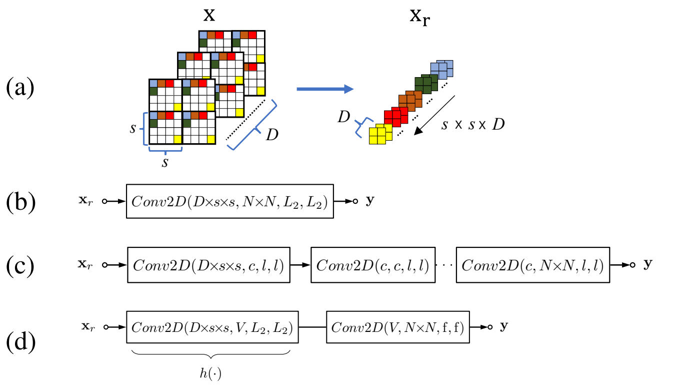
(a)
x
s
s
D
$\mathrm{x_r}$
D
s x s x D
(b)
x_r
$Conv2D(D{\times}s{\times}s, N{\times}N, L_2, L_2)$
y
(c)
x_r
$Conv2D(D{\times}s{\times}s, c, l, l)$
$Conv2D(c, c, l, l)$
$Conv2D(c, N{\times}N, l, l)$
y
(d)
x_r
$Conv2D(D{\times}s{\times}s, V, L_2, L_2)$
$Conv2D(V, N{\times}N, \mathrm{f}, \mathrm{f})$
y
$h(\cdot)$

Fig. 5: Simplified forward model. We show three different linear CNNs that transform a 3D input image $\mathbf{x}$ with D depths into a LF image y with $N \times N$ views. Here, $N \times N$ is the number of pixels under each microlens. In all cases, the input $\mathbf{x}$ is first reshaped, as shown in (a). The model in Figure 4 (c) can always be converted into the single convolutional layer shown in (b). In (c), we approximate the single convolutional layer by a sequence of convolutional layers with filters of smaller size l. Lastly, in (d), the first layer $h(\cdot)$, named compressed forward CNN, outputs V channels, and the second layer recovers the $N \times N$ views with filters of size f.

C. Dimensionality reduction

To overcome the memory and computational-complexity constraints, it is essential to develop efficient implementations of the forward CNN. In the previous section, we employed two convolutional layers to map the forward model to the filter banks. However, it is possible to explore other architectures that simplify the implementation. First, we reshape the 3D input $\mathbf{x}$ to obtain the new input $\mathbf{x}_r$, which has $D \times s \times s$ channels, as depicted in Figure 5 (a). Then, instead of using two convolutional layers as in Figure 4 (c), we can replace them with a single convolutional layer, as shown in Figure 5 (b). Moreover, we propose two different simplifications of the architecture based on two observations:

(a) Convolutional layers with large filters can usually be well-approximated by a sequence of convolutional layers with smaller filters [27], [28], [29]. Therefore, it is realistic to describe the original system with a series of convolutional layers, as shown in Figure 5 (c). We ensure this architecture

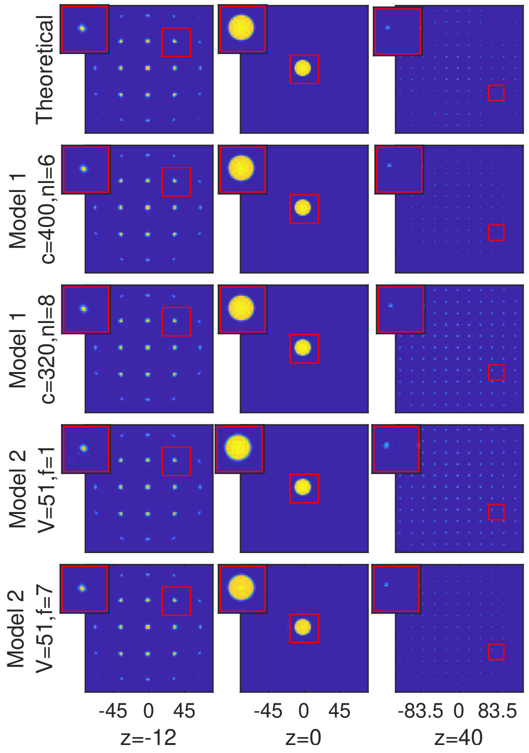

Fig. 6: System impulse response. We evaluate two linear CNN models. Model 1 and 2 correspond to the architecture shown in Figure 5 (c) and (d), respectively. We show the impulse response for a Dirac delta centered at the origin ($x = 0$, $y = 0$) and three different depths ($z = -12$, $z = 0$ and $z = 40$). All the distances are measured in μm.

uses fewer parameters than the single convolution and also ensure that the size of the equivalent filter is the same as the original one by choosing the filter size l and the number of channels c accordingly. Note that the weights of the network can be learned from the theoretical matrix $\mathbf{H}$. In our work, we use this architecture when we need to apply the forward CNN $f(\cdot)$ to a given input volume.

(b) The number of channel outputs in the system greatly impacts the number of parameters. For instance, in our setting $N \times N = 19 \times 19$. Therefore, we can reduce the number of views to a smaller value V to reduce computational complexity. Then, to restore the original number of views back to $N \times N$, we can add a linear convolutional layer, as shown in Figure 5 (d). Since the sub-aperture images are usually highly correlated, it is feasible to perform this linear approximation without significantly impairing the accuracy of the model. Note that the input convolutional layer in Figure 5 (d) is named $h(\cdot)$. This layer is named compressed forward CNN and is connected to our reconstruction approach.

In the sequel, we use $f(\cdot)$ to compute the forward model, while the compressed forward CNN $h(\cdot)$ is used in the reconstruction network. This will be clarified in the following section.

Model 1 ($l = 3$, $D = 53$, $s = 3$, $N{\times}N = 19{\times}19$)	
Parameters	MSE ($\times 10^{-6}$)
$c = 320, nl = 8$	0.19
$c = 361, nl = 8$	0.14
$c = 400, nl = 6$	1.54
$c = 400, nl = 8$	0.12
$c = 400, nl = 10$	0.13
Model 2 ($L_2 = 17$, $D = 53$, $s = 3$, $N{\times}N = 19{\times}19$)	
Parameters	MSE ($\times 10^{-6}$)
$V = 51$, f = 1	25.14
$V = 51$, f = 5	4.49
$V = 51$, f = 7	3.69
$V = 100$, f = 1	6.57
$V = 200$, f = 1	0.01

TABLE I: Numerical evaluation of forward-model approximations.

D. Experimental evaluation of the forward-model approximations.

In this section, we experimentally evaluate the performance of the CNN architectures modelling the system. Notice that the architecture proposed in Figure 4 (c) is derived from a rearrangement of the original filter bank proposed in [5]. This structure allows reproducing the forward system exactly or approximating it by adjusting the number of channels $D \times F$. Its weights can be directly found using SVD, as explained in [5]. Furthermore, the architecture mentioned in Figure 5 (b) is a rearrangement of the original forward model architecture, which does not allow approximating the system. The weights can be found directly by rearranging the original impulse response. Therefore, we only evaluate the performance of the architectures referred to as 'model 1' and 'model 2,' shown in Figure 5(c) and Figure5 (d), respectively.

For these experiments, we describe the system with an impulse response array of size $323 \times 323 \times 3 \times 3 \times 53$, computed from the theoretical model. The array is non-negative and scaled to the range of 0 to 1. The first two dimensions correspond to the lateral coordinates of the impulse response, while the following 3 dimensions represent the lateral and axial dimensions of the impulse. The 3D input has 53 depths, while the lateral size can vary due to the periodic shift-invariance of the system with a period of 3. To find the weights of models 1 and 2, we minimize the mean square error between the impulse response array of the CNN architecture and the theoretical model. We use an optimization framework with Adam optimizer, batch size=2, learning rate=10^{-5}, and 3000 epochs for training.

We experimentally study the impact of the different parameters on the approximation performance. For this experiment, we evaluated different parameters listed in Table I. In model 1, the approximation improves as the number of channels c increases, while reducing the number of layers is detrimental. This behaviour is expected since the theoretical filter size is 17 ($323/19$), which is achieved after 8 layers in 'model 1'. However, we also found that increasing the number of layers beyond 8 does not necessarily improves the performance, as shown in Table I. In 'model 2', increasing the number of channels V and filter size f improves the approximation performance, as shown in Table I. Note that for this model, f $= 1$ and $V = 360$ will reproduce exactly the forward model.

Visual inspection of the approximated impulse response shows little difference between approximation methods, as depicted in Figure 6. Note in Table I that the proposed architectures allow approximating the measurement matrix with a very low mean square error (MSE).

V. 3D Reconstruction

In this section, we design a CNN that considers the physics of the system to perform the reconstruction. The architecture of our network is constructed using the unfolding technique [14] and is obtained by unrolling sparsity-driven algorithms for reconstruction. Furthermore, our network is trained in a semi-supervised manner to alleviate the lack of data which is a typical issue for applications in neuroscience.

A. *CNN architecture*

Large distributions of labelled neurons can be modelled as compact cell bodies sparsely distributed in brain tissue [30]. Therefore, to reconstruct high-quality 3D volumes, we can consider the following optimization approach that promotes sparsity in the reconstruction:

$$\arg\min_{\mathbf{x}} \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2+\lambda\|\mathbf{x}\|_1, \tag{3}$$

where $\mathbf{y}$ is a given LF image, $\mathbf{x}$ is the reconstructed volume and the scalar λ controls the degree of regularization. This problem can be solved using the Iterative Shrinkage-Thresholding Algorithm (ISTA) [31] by computing at each iteration:

$$\mathbf{x}^{k+1}=\mathcal{T}_\lambda(\mathbf{x}^k-\mathbf{H}^\intercal\mathbf{H}\mathbf{x}^k+\mathbf{H}^\intercal\mathbf{y}), \tag{4}$$

where $\mathcal{T}_\lambda$ is the soft-thresholding operator with parameter λ. One can interpret each iteration of ISTA as a layer of a neural network with fixed weights. Therefore, it is possible to design a neural network architecture based on ISTA. LISTA [14] (the learned version of ISTA) is a neural network built such that each layer corresponds to one iteration of ISTA. Effectively, each layer of LISTA implements the following step:

$$\mathbf{x}^{k+1}=\mathcal{T}_\lambda(\mathbf{x}^k-\mathbf{H}_1^\intercal\mathbf{H}_2\mathbf{x}^k+\mathbf{H}_3{}^\intercal\mathbf{y}), \tag{5}$$

where $\mathbf{H}_1$,$\mathbf{H}_2$ and $\mathbf{H}_3$ are matrices of same size and structure as $\mathbf{H}$. These matrices are the parameters of the network that can be learned using a proper loss function. Note that, contrary to [14], we do not fuse the product $\mathbf{H}_1^\intercal\mathbf{H}_2$ into a single matrix since we want to keep the structure of each factor. This version of LISTA uses the soft-thresholding as the element-wise non-linearity due to the l_1 constraint in Equation (3). However, ISTA can be used with different types of non-linearities related to the prior imposed, as explained in [32]. For instance, replacing $\mathcal{T}_\lambda$ by a rectified linear unit (Relu) imposes non-negativity, and replacing it with a ReLU with a bias term imposes sparsity and non-negativity. This follows by computing the following proximal operator related to the $L1$ norm and the non-negativity constraint:

$$\begin{aligned}\min_{\mathbf{x}} \quad & \tfrac{1}{2}\|\mathbf{x}-\mathbf{v}\|_2^2+\lambda\|\mathbf{x}\|_1,\\ \text{s.t.} \quad & \mathbf{x}\geq 0.\end{aligned} \tag{6}$$

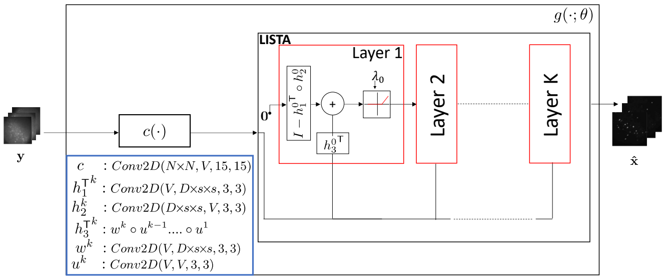
$g(\cdot;\theta)$
LISTA
Layer 1
$I-h_1^{0\intercal}\circ h_2^0$
λ_0
$h_3^{0\intercal}$
$\mathbf{0}$
Layer 2
Layer K
$c(\cdot)$
$\mathbf{y}$
$\hat{\mathbf{x}}$
c : $Conv2D(N{\times}N,V,15,15)$
${h_1^\intercal}^k$: $Conv2D(V,D{\times}s{\times}s,3,3)$
h_2^k : $Conv2D(D{\times}s{\times}s,V,3,3)$
${h_3^\intercal}^k$: $w^k\circ u^{k-1}....\circ u^1$
w^k : $Conv2D(V,D{\times}s{\times}s,3,3)$
u^k : $Conv2D(V,V,3,3)$

Fig. 7: CNN architecture. Our reconstruction network $g(\cdot)$ is composed of (1) a compression layer $c(\cdot)$, which is a linear convolutional layer and (2) a LISTA network. The architecture of LISTA is based on the assumption that reconstructed 3D data is non-negative and sparse. The LISTA network is composed of K layers. The architecture of each block is detailed in the bottom left of the figure.

It is possible to show that the $\mathbf{x}$ that minimizes this optimization is given by:

$$x^*=\begin{cases}0 & \text{if } v\leq\lambda\\ v-\lambda & \text{if } v>\lambda\end{cases}, \tag{7}$$

which is equivalent to $x^*=ReLU(v-\lambda)$.

In our case, $\mathbf{x}$ is sparse and non-negative. Therefore, we propose a LISTA network that uses a ReLU with a bias term as non-linearity:

$$\mathbf{x}^{k+1}=ReLU(\mathbf{x}^k-{\mathbf{H}_1^\intercal}^k\mathbf{H}_2^k\mathbf{x}^k+{\mathbf{H}_3^\intercal}^k\mathbf{y}-\lambda^k), \tag{8}$$

where λ^k is a learnable parameter that can be optimized with the rest of the network weights. Furthermore, the custom $\{\mathbf{H}_i^k\}_{i=1}^3$ for each unfolded iteration k gives the network more capabilities without compromising its simplicity.

In many practical cases in LFM, the described LISTA network cannot be used directly to solve the volume reconstruction problem. The size and structure of the matrix $\mathbf{H}$ make it computationally prohibitive to perform matrix multiplications repeatedly. Therefore, we propose using the compressed forward CNN $h(\cdot)$ proposed in Section IV-B to reduce the computational complexity. The final architecture of our network is, therefore, described as follows:

$$\mathbf{x}^{k+1}=ReLU(\mathbf{x}^k-{h_1^\intercal}^k(h_2^k(\mathbf{x}^k))+{h_3^\intercal}^k(c(\mathbf{y}))-\lambda^k), \tag{9}$$

where we have replaced matrices $\mathbf{H}_i^k$ in Equation (8) with the linear mappings $\{h_i^k\}_{i=1}^3$. The computation of the mapping $\{h_i^k\}_{i=1}^3$ is given by the compressed forward CNN explained in Section IV-B. Note that the structure of the adjoint operators (transpose) $\{h_i^\intercal\}_{i=1}^3$ in Equation (9) can be computed from the permutation of the weights of $h(\cdot)$. Furthermore, the input of the network is $c(\mathbf{y})$ rather than $\mathbf{y}$. The mapping $c(\cdot)$ is defined as a single linear convolutional layer with $N\times N$ input channels and V output channels. By having V output channels, $c(\cdot)$ is compatible with the input size of the operators $\{h_i^\intercal\}_{i=1}^3$. We highlight that the coefficients of the compression layer $c(\cdot)$ are learned together with LISTA. The end-to-end network $g(\cdot;\theta)$, where θ represents the learnable parameters

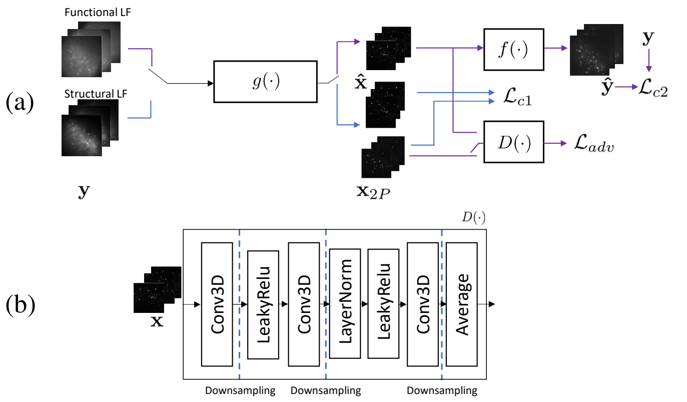

Fig. 8: Adversarial training. In (a), we show how the LISTA network $g(\cdot)$ is trained. A first content loss measured on structural data is first computed using a small labelled dataset captured with two-photon microscopy. A second content loss is computed only on LF data showing functional information (tagged with jGCaMP8f), where we exploit the knowledge of the forward model $f(\cdot)$. Finally, an adversarial loss is computed from a critic $D(\cdot)$. In (b), we show the architecture of the critic $D(\cdot)$.

of the network, is shown in Figure 7. To provide additional simplification, the layers ${h_3^\intercal}^k$ acting directly on the LF $\mathbf{y}$ are replaced by a series of convolutional layers, the filter sizes of all the layers are set to 3. Moreover, the number of layers of ${h_3^\intercal}^k$ increases with k to save extra computational complexity, as specified in Figure 7.

B. CNN training

We learn the parameters θ of our LISTA network $g(\cdot;\theta)$ using a suitable loss function and a combination of labelled and unlabelled datasets. In our scenario, a labelled dataset consists of LF images and their corresponding 2P volumes. Capturing a large labelled dataset can be impractical or prohibitively expensive for many applications in LFM. For instance, when studying the behavior of neurons in mammalian tissue, capturing a clean 3D label is challenging due to the scattering media Additionally, relying solely on synthetic data for training can be problematic if noise is not accurately modelled.

In our setting, we propose acquiring a very small labelled training dataset. We label neurons in a single brain sample using tdTomato fluorophore. Using tdTomato, we are able to capture the static distribution of the neurons in space using both 2P and LF modalities. The 2P raster scanning modality provides the ground truth volume that can be paired with the LF images acquired with the same fluorophore. Therefore, to train our network, we exploit the small labelled dataset, the large amount of unpaired LF images, and our knowledge of the forward model. The training loss is stated as follows:

$$\alpha_1 \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{c1}(\mathbf{x}^i,\hat{\mathbf{x}}^i)+\alpha_2\frac{1}{K}\sum_{j=1}^{K}\mathcal{L}_{c2}(\mathbf{y}^j,\hat{\mathbf{y}}^j)+\alpha_3\mathcal{L}_{adv}(g(\mathbf{y}^j)), \tag{10}$$

where $\mathbf{x}^i$ is the 2P 3D image, $\hat{\mathbf{x}}^i$ is the network reconstruction, $\mathbf{y}^i$ is a LF image, M is the number of 3D samples, K is the number of LF samples and $\hat{\mathbf{y}}^i = f(g(\mathbf{y}^i))$, where $f(\cdot)$ is the known forward CNN. The weight for each loss is controlled by scalars α_1, α_2, and α_3. Note that operator $f(\cdot)$ is fixed since it is already known and is based on the model [4]. The loss $\mathcal{L}c1(\cdot)$ is the 2P content loss computed on the labelled dataset. The loss $\mathcal{L}c2(\cdot)$ is the LF content loss, which ensures that the re-synthesized LF computed from the recovered volume is similar to the original LF image. The adversarial loss $\mathcal{L}adv(\cdot)$ makes the recovered volume appear realistic. This loss is computed from a trainable critic $D(\cdot)$, which functions as a regularizer. We consider this loss as a dynamic regularizer that is updated simultaneously with the reconstruction network during training. Only $\mathcal{L}c1(\cdot)$ requires a labelled dataset, while $\mathcal{L}c2(\cdot)$ and $\mathcal{L}adv(\cdot)$ use LF images and unpaired 2P data, respectively. The loss function in Equation (10) was first proposed as a supervised-learning technique for single image super-resolution [33]. Additionally, learning regularizers to solve standard inverse problems using stochastic gradient descent has been studied in [34]. However, we note that our approach is a semi-supervised technique compared to [33]. Furthermore, the critic, or regularizer, is not pre-trained, unlike in [34].

The loss $\mathcal{L}_{c1}(\cdot)$ is computed on a small labelled dataset. The LF stack showing structural information (tagged with tdTomato) has a corresponding 3D label captured with the microscope in 2P modality (we collect 28 labelled pairs in our setup). Note that the labelled dataset is small due to the demanding experimental requirements for acquisition. The second dataset is formed by only LF data, which shows functional information (tagged with jGCaMP8f). This dataset can be much larger since it does not require 2P acquisition (240 LF images in our experiments). The losses $\mathcal{L}_{c2}(\cdot)$ and $\mathcal{L}_{adv}(\cdot)$ are computed only on the unlabelled dataset, as indicated by the different summation index in Equation (10). During training, each LF data is input to the network at different times, and the corresponding loss is computed. In our work, the adversarial regularizer is learned simultaneously with the reconstruction network by using the well-known adversarial framework for least squares GANs (LSGANs) [35], as depicted in Figure 8 (a). In the next section, we explicitly define the training loss and the architecture of the critic used for the experiments.

VI. Experiments and Results

In this section, we show the performance of our approach by imaging mouse brain tissue with both LF and 2P modalities (see supplemental material). We compare the performance of our method with state-of-the-art model-based and learning-based methods for 3D reconstruction in LFM.

A. Experimental setup

The LF microscope is modelled as follows: numerical aperture = 1, refractive index = 1.33, wavelength = 514 nm for jGCaMP8f and 580 nm for tdTomato fluorophore, magnification = 25, microlens pitch =125 μm, microlens focal

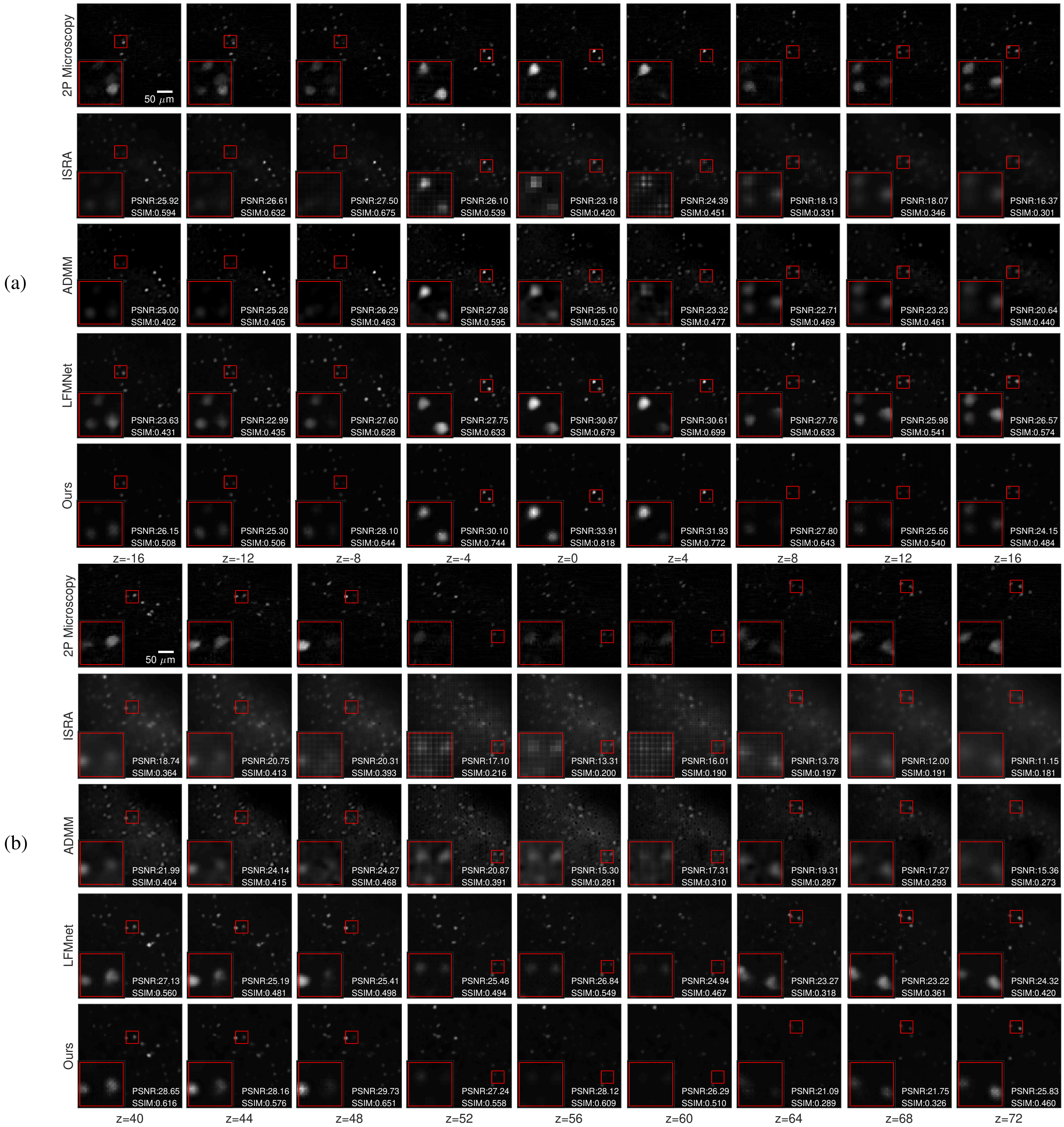

Fig. 9: Reconstruction using real LF data from acute mouse brain slices expressing tdTomato fluorophore. This fluorophore allows the study of the morphology or anatomy of the network of neurons, known as structural imaging. For this type of data, we acquired a small labelled dataset for training using 2P microscopy. In part (a), from top to bottom we show the 2P 3D image used as ground truth, the reconstruction using two model-based approaches: ISRA and ADMM, the LFMNet proposed in [10], and our approach. We show a series of 2D slices, each labelled with its corresponding depth. This reconstruction corresponds to the performance shown in the first row in Table II. In part (b), the same methods are evaluated for a LF image focused on a deeper depth, corresponding to the row 28 in Table II. The shown PSNR and SSIM are measured for the whole plane at each depth. Measures on the whole volume are shown in Table II. All the distances are measured in μm. The settings used to capture both the LF image and 2P image are specified in Section VI.

Part A

	PSNR / SSIM / MSE ($\times 10^{-2}$) / ENL / EPI			
Depth Index	VCDNet	HyLFM	LFMNet	Ours
1	30.84 / 0.65 / 0.08 / 0.97 / 0.22	31.48 / 0.67 / 0.07 / 0.99 / 0.25	31.02 / 0.64 / 0.08 / 0.98 / 0.26	**34.19 / 0.82 / 0.04 / 0.99 / 0.27**
2	32.09 / 0.72 / 0.06 / 0.97 / 0.23	32.04 / 0.70 / 0.06 / **0.98** / 0.27	30.98 / 0.64 / 0.08 / 0.98 / 0.27	**34.34 / 0.82 / 0.04** / 0.98 / **0.30**
3	31.09 / 0.66 / 0.08 / 0.97 / 0.24	31.89 / 0.68 / 0.06 / **0.99** / 0.26	30.53 / 0.62 / 0.09 / 0.98 / 0.27	**33.88 / 0.79 / 0.04** / 0.98 / **0.29**
4	29.99 / 0.62 / 0.10 / 0.97 / 0.24	32.89 / 0.74 / 0.05 / **0.98** / 0.26	29.98 / 0.60 / 0.10 / 0.98 / 0.28	**33.20 / 0.75 / 0.05** / 0.98 / **0.29**
5	29.82 / 0.61 / 0.10 / 0.97 / 0.24	32.67 / 0.73 / 0.05 / **0.98** / 0.26	29.64 / 0.60 / 0.11 / 0.98 / 0.28	**33.18 / 0.75 / 0.05** / 0.98 / **0.29**
6	28.79 / 0.57 / 0.13 / 0.97 / 0.24	**33.00 / 0.75 / 0.05** / 0.98 / 0.26	29.95 / 0.60 / 0.10 / 0.98 / 0.28	32.99 / 0.74 / 0.05 / **0.98 / 0.29**
7	28.94 / 0.57 / 0.13 / 0.97 / 0.24	31.68 / 0.68 / 0.07 / 0.98 / 0.26	31.18 / 0.65 / 0.08 / 0.98 / 0.28	**33.33 / 0.76 / 0.05 / 0.98 / 0.29**
8	29.35 / 0.59 / 0.12 / 0.97 / 0.24	29.97 / 0.61 / 0.10 / 0.98 / 0.26	31.57 / 0.67 / 0.07 / 0.98 / 0.28	**33.62 / 0.78 / 0.04 / 0.98 / 0.30**
9	29.69 / 0.61 / 0.11 / 0.97 / 0.23	30.09 / 0.62 / 0.10 / 0.98 / 0.27	32.41 / 0.71 / 0.06 / 0.98 / 0.27	**33.84 / 0.80 / 0.04 / 0.98 / 0.29**
10	29.77 / 0.61 / 0.11 / 0.97 / 0.23	30.22 / 0.62 / 0.09 / 0.98 / 0.27	31.16 / 0.65 / 0.08 / 0.98 / 0.27	**33.38 / 0.77 / 0.05 / 0.98 / 0.28**
11	31.12 / 0.68 / 0.08 / 0.96 / 0.23	30.27 / 0.62 / 0.09 / **0.98** / 0.27	30.94 / 0.65 / 0.08 / 0.98 / 0.27	**33.29 / 0.77 / 0.05** / 0.98 / **0.29**
12	30.24 / 0.63 / 0.09 / 0.97 / 0.23	29.38 / 0.58 / 0.12 / 0.98 / 0.27	29.03 / 0.56 / 0.13 / 0.98 / 0.27	**32.65 / 0.73 / 0.05 / 0.99 / 0.28**
13	30.03 / 0.63 / 0.10 / 0.97 / 0.23	29.79 / 0.60 / 0.10 / 0.98 / **0.28**	31.36 / 0.67 / 0.07 / 0.98 / 0.26	**32.25 / 0.72 / 0.06 / 0.98** / 0.27
14	29.80 / 0.61 / 0.10 / 0.97 / 0.24	29.96 / 0.60 / 0.10 / 0.99 / 0.28	30.17 / 0.61 / 0.10 / 0.98 / 0.28	**31.77 / 0.68 / 0.07 / 0.99 / 0.30**
15	29.51 / 0.60 / 0.11 / 0.97 / 0.25	29.26 / 0.57 / 0.12 / 0.99 / 0.28	29.11 / 0.57 / 0.12 / 0.98 / 0.29	**31.48 / 0.67 / 0.07 / 0.99 / 0.30**
16	29.19 / 0.58 / 0.12 / 0.97 / 0.25	28.13 / 0.52 / 0.15 / 0.99 / 0.28	28.39 / 0.53 / 0.14 / 0.98 / 0.29	**31.10 / 0.65 / 0.08 / 0.99 / 0.31**
17	28.20 / 0.53 / 0.15 / 0.98 / 0.25	27.12 / 0.48 / 0.19 / 0.99 / 0.28	28.54 / 0.54 / 0.14 / 0.98 / 0.30	**30.69 / 0.63 / 0.09 / 0.99 / 0.30**
18	29.76 / 0.60 / 0.11 / 0.98 / 0.26	26.94 / 0.48 / 0.20 / 0.99 / 0.27	28.75 / 0.55 / 0.13 / 0.98 / 0.29	**31.54 / 0.67 / 0.07 / 0.99 / 0.31**
19	25.93 / 0.44 / 0.26 / 0.98 / 0.24	26.09 / 0.44 / 0.25 / 0.99 / 0.27	27.28 / 0.49 / 0.19 / 0.98 / 0.28	**28.89 / 0.56 / 0.13 / 0.99 / 0.29**
20	27.54 / 0.50 / 0.18 / 0.98 / 0.25	27.45 / 0.49 / 0.18 / 0.99 / 0.26	27.85 / 0.51 / 0.16 / 0.98 / 0.29	**31.53 / 0.66 / 0.07 / 0.99 / 0.30**
21	25.86 / 0.43 / 0.26 / 0.98 / 0.25	27.00 / 0.47 / 0.20 / 0.99 / 0.26	27.31 / 0.49 / 0.19 / 0.98 / 0.28	**30.08 / 0.60 / 0.10 / 0.99 / 0.28**
22	26.74 / 0.47 / 0.21 / 0.98 / 0.26	26.86 / 0.47 / 0.21 / 0.99 / 0.26	27.89 / 0.51 / 0.16 / 0.98 / 0.29	**31.42 / 0.66 / 0.07 / 0.99 / 0.29**
23	25.80 / 0.43 / 0.26 / 0.98 / 0.26	26.15 / 0.44 / 0.24 / 0.99 / 0.26	28.32 / 0.52 / 0.15 / 0.98 / 0.27	**30.00 / 0.60 / 0.10 / 0.99 / 0.29**
24	25.36 / 0.42 / 0.29 / 0.98 / 0.26	26.42 / 0.45 / 0.23 / 0.99 / 0.27	27.74 / 0.50 / 0.17 / 0.98 / 0.28	**30.31 / 0.61 / 0.09 / 0.99 / 0.30**
25	26.28 / 0.45 / 0.24 / 0.98 / 0.26	26.02 / 0.43 / 0.25 / 0.99 / 0.27	27.92 / 0.51 / 0.16 / 0.98 / 0.28	**30.54 / 0.62 / 0.09 / 0.99 / 0.30**
26	27.35 / 0.49 / 0.18 / 0.98 / 0.26	25.47 / 0.41 / 0.28 / 0.99 / 0.27	27.88 / 0.50 / 0.16 / 0.98 / 0.27	**28.99 / 0.56 / 0.13 / 0.99 / 0.30**
27	27.80 / 0.50 / 0.17 / 0.98 / 0.26	24.58 / 0.38 / 0.35 / 0.99 / 0.27	26.70 / 0.45 / 0.21 / 0.99 / 0.28	**30.66 / 0.62 / 0.09 / 0.99 / 0.31**
28	26.26 / 0.44 / 0.24 / 0.99 / 0.27	25.05 / 0.39 / 0.31 / 0.99 / 0.27	27.16 / 0.47 / 0.19 / 0.99 / 0.28	**31.06 / 0.63 / 0.08 / 0.99 / 0.30**
Mean	28.68 / 0.56 / 0.15 / 0.97 / 0.24	28.85 / 0.56 / 0.15 / 0.99 / 0.27	29.31 / 0.57 / 0.12 / 0.98 / 0.28	**31.94 / 0.69 / 0.07 / 0.99 / 0.29**

Part B

	PSNR / SSIM / MSE ($\times 10^{-2}$) / ENL / EPI			
Depth Index	ISRA	ISRA TV	ISRA AF	ADMM
1	28.78 / 0.61 / 0.13 / 0.99 / 0.20	28.17 / 0.59 / 0.15 / 0.99 / 0.20	21.50 / 0.34 / 0.71 / 0.99 / 0.20	29.55 / 0.62 / 0.11 / 0.86 / 0.12
2	28.84 / 0.60 / 0.13 / 0.99 / 0.20	28.16 / 0.58 / 0.15 / 0.99 / 0.20	20.90 / 0.32 / 0.81 / 1.00 / 0.20	30.16 / 0.68 / 0.10 / 0.93 / 0.13
3	27.32 / 0.55 / 0.19 / 0.99 / 0.20	26.69 / 0.53 / 0.21 / 0.99 / 0.20	19.62 / 0.28 / 1.09 / 0.99 / 0.20	28.87 / 0.61 / 0.13 / 0.88 / 0.12
4	26.22 / 0.50 / 0.24 / 0.99 / 0.20	25.55 / 0.48 / 0.28 / 0.99 / 0.20	19.03 / 0.27 / 1.25 / 1.00 / 0.20	28.54 / 0.63 / 0.14 / 0.92 / 0.13
5	24.95 / 0.46 / 0.32 / 0.99 / 0.19	24.22 / 0.44 / 0.38 / 0.99 / 0.19	18.30 / 0.25 / 1.48 / 1.00 / 0.20	27.54 / 0.59 / 0.18 / 0.92 / 0.13
6	25.37 / 0.48 / 0.29 / 0.99 / 0.19	24.12 / 0.43 / 0.39 / 0.99 / 0.19	18.25 / 0.25 / 1.50 / 1.00 / 0.20	26.67 / 0.57 / 0.22 / 0.91 / 0.13
7	26.36 / 0.51 / 0.23 / 0.99 / 0.19	25.32 / 0.48 / 0.29 / 0.99 / 0.19	19.55 / 0.29 / 1.11 / 1.00 / 0.20	27.56 / 0.58 / 0.18 / 0.90 / 0.12
8	26.58 / 0.53 / 0.22 / 0.99 / 0.19	25.53 / 0.49 / 0.28 / 0.99 / 0.19	19.87 / 0.30 / 1.03 / 1.00 / 0.20	27.90 / 0.58 / 0.16 / 0.90 / 0.13
9	26.08 / 0.51 / 0.25 / 0.99 / 0.20	25.04 / 0.47 / 0.31 / 0.99 / 0.19	19.79 / 0.29 / 1.05 / 1.00 / 0.20	28.07 / 0.60 / 0.16 / 0.91 / 0.13
10	25.14 / 0.48 / 0.31 / 0.99 / 0.20	24.51 / 0.46 / 0.35 / 0.99 / 0.20	19.06 / 0.28 / 1.24 / 1.00 / 0.21	27.47 / 0.56 / 0.18 / 0.89 / 0.12
11	25.49 / 0.49 / 0.28 / 0.99 / 0.21	24.82 / 0.47 / 0.33 / 0.99 / 0.21	18.71 / 0.27 / 1.35 / 1.00 / 0.21	27.70 / 0.60 / 0.17 / 0.92 / 0.14
12	24.95 / 0.47 / 0.32 / 0.99 / 0.21	24.01 / 0.44 / 0.40 / 0.99 / 0.21	19.95 / 0.30 / 1.01 / 1.00 / 0.21	26.77 / 0.55 / 0.21 / 0.90 / 0.13
13	25.06 / 0.48 / 0.31 / 0.99 / 0.21	24.30 / 0.45 / 0.37 / 0.99 / 0.21	20.83 / 0.33 / 0.83 / 1.00 / 0.21	26.43 / 0.55 / 0.23 / 0.89 / 0.13
14	25.08 / 0.47 / 0.31 / 0.99 / 0.22	24.31 / 0.45 / 0.37 / 0.99 / 0.22	21.14 / 0.34 / 0.77 / 1.00 / 0.23	27.65 / 0.59 / 0.17 / 0.93 / 0.15
15	25.41 / 0.48 / 0.29 / 0.99 / 0.23	24.65 / 0.46 / 0.34 / 0.99 / 0.23	21.01 / 0.34 / 0.79 / 1.00 / 0.23	27.68 / 0.60 / 0.17 / 0.92 / 0.14
16	24.88 / 0.47 / 0.32 / 0.99 / 0.23	24.13 / 0.44 / 0.39 / 0.99 / 0.23	20.63 / 0.33 / 0.87 / 1.00 / 0.23	27.48 / 0.58 / 0.18 / 0.91 / 0.14
17	24.50 / 0.45 / 0.35 / 0.99 / 0.24	23.98 / 0.44 / 0.40 / 0.99 / 0.23	20.23 / 0.32 / 0.95 / 1.00 / 0.23	27.11 / 0.56 / 0.19 / 0.91 / 0.13
18	23.91 / 0.43 / 0.41 / 0.99 / 0.24	23.37 / 0.42 / 0.46 / 0.99 / 0.24	20.02 / 0.31 / 0.99 / 1.00 / 0.24	26.71 / 0.55 / 0.21 / 0.92 / 0.16
19	22.63 / 0.39 / 0.55 / 0.99 / 0.24	22.06 / 0.37 / 0.62 / 0.99 / 0.24	19.20 / 0.29 / 1.20 / 1.00 / 0.23	25.42 / 0.50 / 0.29 / 0.90 / 0.13
20	22.28 / 0.38 / 0.59 / 0.99 / 0.24	21.52 / 0.35 / 0.70 / 0.99 / 0.24	18.73 / 0.27 / 1.34 / 1.00 / 0.24	25.03 / 0.49 / 0.31 / 0.92 / 0.15
21	20.80 / 0.33 / 0.83 / 0.99 / 0.24	20.06 / 0.31 / 0.99 / 0.99 / 0.24	17.62 / 0.25 / 1.73 / 1.00 / 0.23	23.57 / 0.45 / 0.44 / 0.90 / 0.13
22	21.19 / 0.34 / 0.76 / 0.99 / 0.25	19.99 / 0.31 / 1.00 / 1.00 / 0.25	17.19 / 0.23 / 1.91 / 1.00 / 0.25	23.06 / 0.45 / 0.49 / 0.92 / 0.16
23	21.60 / 0.35 / 0.69 / 0.99 / 0.25	20.49 / 0.32 / 0.89 / 1.00 / 0.25	16.55 / 0.22 / 2.21 / 1.00 / 0.25	23.37 / 0.44 / 0.46 / 0.91 / 0.15
24	21.43 / 0.35 / 0.72 / 0.99 / 0.26	20.49 / 0.32 / 0.89 / 1.00 / 0.26	16.46 / 0.22 / 2.26 / 1.00 / 0.26	23.51 / 0.44 / 0.45 / 0.93 / 0.17
25	20.21 / 0.31 / 0.95 / 0.99 / 0.27	19.59 / 0.29 / 1.10 / 1.00 / 0.27	15.95 / 0.20 / 2.54 / 1.00 / 0.27	23.32 / 0.47 / 0.47 / 0.92 / 0.16
26	20.46 / 0.32 / 0.90 / 0.99 / 0.27	19.75 / 0.30 / 1.06 / 1.00 / 0.26	15.16 / 0.19 / 3.05 / 1.00 / 0.26	23.19 / 0.46 / 0.48 / 0.92 / 0.15
27	19.80 / 0.30 / 1.05 / 0.99 / 0.26	19.30 / 0.29 / 1.18 / 1.00 / 0.26	14.94 / 0.18 / 3.20 / 1.00 / 0.26	22.62 / 0.43 / 0.55 / 0.92 / 0.16
28	19.30 / 0.28 / 1.17 / 0.99 / 0.27	18.62 / 0.27 / 1.37 / 1.00 / 0.27	14.47 / 0.17 / 3.57 / 1.00 / 0.27	22.09 / 0.42 / 0.62 / 0.93 / 0.17
Mean	24.09 / 0.44 / 0.47 / 0.99 / 0.23	23.31 / 0.42 / 0.56 / 0.99 / 0.22	18.74 / 0.27 / 1.49 / 1.00 / 0.23	26.25 / 0.54 / 0.27 / 0.91 / 0.14

TABLE II: Evaluation of deep-learning (A) and model-based (B) methods for LF imaged using tdTomato fluorophore.

length = 1250 μm, tube lens focal length = 0.18 m, pixels per microlens =19×19.

We compare our method with the following model-based approaches: ISRA, a variant of RL algorithm [21], ISRA with total variation (ISRA TV) [22], ISRA with reduced artifacts (ISRA AF) [7], and ADMM [5]. Furthermore, we evaluate other learning-based approaches by adapting the LFMNet [10], HyLFM [11], and VCDNet [9] to work with our specifications based on the respective code made available online. To train our network and to evaluate model-based approaches, we use the conventional theoretical forward model used for reconstruction proposed by Broxton et al. [4].

We used a 2P microscope to capture the spatial distribution of a network of neurons in mouse brain slices expressing tdTomato fluorescent protein. The captured stack consists of 80 planes taken at 2 μm intervals. We generated 28 volumes, each comprising 53 slices, from this stack. The first volume includes planes 1 to 53, the second volume includes planes 2 to 54, and so on. We also captured the corresponding LF stack consisting of 28 images. Therefore, we obtained a training dataset with 28 pairs of images taken from a single brain slice. To evaluate our method, we captured another dataset of the same size from a different brain sample.

In addition, we capture temporal LF sequences from samples labelled with genetically encoded calcium indicators (jGCaMP8f). We acquire 4 different temporal sequences with 500 LF images each. We took the first 80 LF images from 3 stacks for training. This additional training dataset only contains LF images without any 2P label. In our experiments, the 2P data is acquired only once and is not updated when evaluating different sample tissues.

Due to the dimensionality of the dataset, data augmentation is needed to alleviate data over-fitting. Specifically, we perform data augmentation by using random reflections on the x, y, and z-axis and axes swapping on the x, y dimension of the volume. We modify the LF data accordingly as well. Furthermore, we use patch-based training to reduce memory consumption.

B. CNN settings

To initialize the network, we pre-train it using only the labelled dataset. For this step, we use only the first content loss $\mathcal{L}_{c1}(\cdot)$ in Equation (10), which is chosen to be the normalized mean square error loss as follows:

$$\mathcal{L}_{c1}(\mathbf{x}, \hat{\mathbf{x}}) = \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} \right\|_2^2, \tag{11}$$

where $\mathbf{x}$ is the 2P 3D image and $\hat{\mathbf{x}}$ is the 3D image reconstructed by the network.

Once the network is initialized, all the losses in Equation (10), including the previous $\mathcal{L}_{c1}(\cdot)$ are considered in the minimization. The second content loss named $\mathcal{L}_{c2}(\cdot)$ is given by the following equation:

$$\mathcal{L}_{c2}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2, \tag{12}$$

where the subscript n represents mean normalization, which is performed as follows:

$$\mathbf{y}_n = \mathbf{y} - \mathbb{E}_{d\times d}[\mathbf{y}], \tag{13}$$

where the notation $\mathbb{E}_{d\times d}[\cdot]$ means that the expected value is computed for regions of size $d \times d$ pixels in the spatial dimensions. Specifically, each sub-aperture image of the LF $\mathbf{y}$ is divided into a grid of squares of size $d \times d$ pixels. The expected value is then subtracted from each square to obtain $\mathbf{y}_n$. In our experiments, the value d is experimentally chosen to be 8. Sub-aperture images preserve lateral spatial information of the input, and they are usually used to visually inspect if the acquired LF image matches the 3D scene. Therefore, it is realistic to promote reconstruction details shown in the sub-aperture images rather than background noise. Losses that focus on details or edges appear in other modalities such as learned optics [36] or dehazing [37] by exploiting high-pass filters. In wavelet theory, a more general approach to extracting details from a signal is to subtract the projection of the image over the subspace generated by a scaling function at a given level. Wavelets allow for analyzing signals in both space and frequency. In our case, we use 3 levels of the Haar scaling function. Thus, the loss only considers the horizontal, vertical, and diagonal details of levels 1, 2, and 3

The adversarial loss in Equation (10), $\mathcal{L}_{adv}(\cdot)$ is determined by the discriminator $D(\cdot)$. The discriminator tries to assign different scores to the real 3D data and the reconstruction from the network $g(\cdot)$. We name $\mathbb{P}_\theta$ the probabilistic distribution of real 3D volumes and $\mathbb{P}_r$ the distribution of 3D reconstructions from the network. Then, the adversarial loss is given by:

$$\mathcal{L}_{adv}(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\theta}[(D(\mathbf{x}) - a)^2], \tag{14}$$

where $a = 1$ and the architecture of $D(\cdot)$ is depicted in Figure 8 (b). The discriminator was designed by following standard architectures used for 3D GANs [38]. The expected value $\mathbb{E}[\cdot]$ is approximated by computing the mean on a batch of volumes generated by our network. Finally, the discriminator is trained by using the loss

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\theta}[(D(\mathbf{x}) - b)^2] + \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[(D(\mathbf{x}) - a)^2], \tag{15}$$

where $a = 1$, $b = -1$. As mentioned previously, the first expected value is computed on the batch of volumes generated with our network. Similarly, the second expected value is computed on batches of real data. Intuitively, the discriminator is trained to assign a 1 if the input has the 2P quality and a -1 if the input is generated by $g(\cdot)$ and does not have 2P quality. At the same time, $g(\cdot)$ tries to make $D(\cdot)$ to assign a 1 by improving the reconstruction quality. This training procedure is part of a standard technique proposed in [35] to train LSGANs. For pre-training, we use the Adam optimizer, batch size=2, number of epochs=400 and learning rate= 10^{-6}. We keep the same batch size and number of epochs for the adversarial training. The learned weights are used to initialize $g(\cdot)$. The learning rate is 2×10^{-7} for $g(\cdot)$ and 4×10^{-7} for the discriminator $D(\cdot)$. The scalars α_1, α_2, and α_3 in Equation (10) allow us to adjust the importance of each loss and ensure that all terms are of a similar order. We set $\alpha_1 = 2.5 \times 10^3$, $\alpha_2 = 1.5 \times 10^{-3}$ and $\alpha_3 = 10^{-3}$. For both 2P and LF images, all the pixel values are normalized to lie between 0 and 255 before training. The maximum value for normalization is found per stack or sequence to avoid altering the axial correlation between slices or frames.

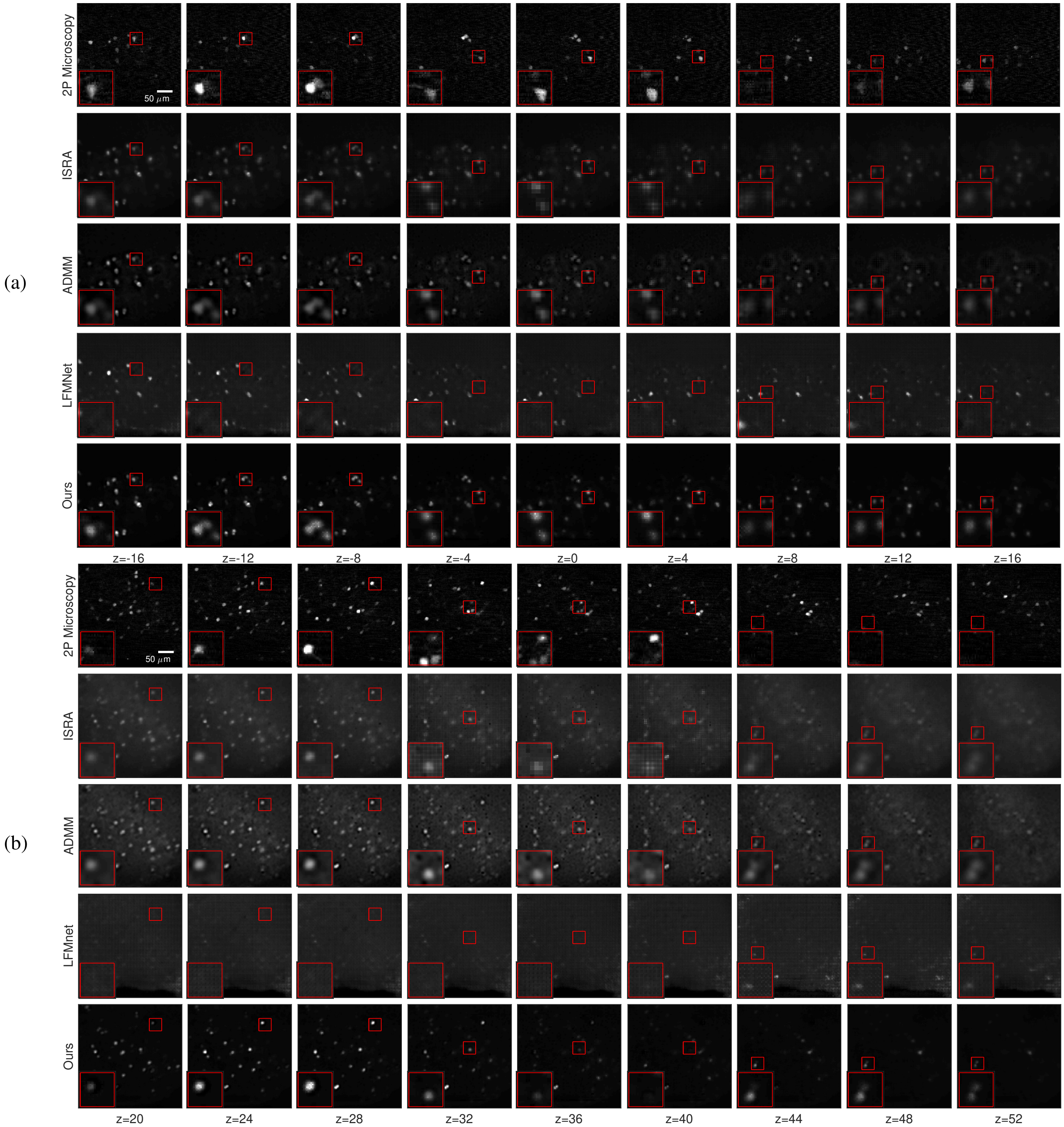

Fig. 10: Reconstruction using real LF data from acute mouse brain slices expressing the calcium indicator jGCaMP8f. The indicator jGCaMP8f is an indicator of calcium concentration which indirectly measures electrical, and therefore functional, activity in the brain. For this type of data, only LF images and the knowledge of the forward model was used for training. No ground-truth dataset is available for this type of data. The labels (a) and (b) indicate two different brain samples. From top to bottom, we show the 2P 3D image of the static tdTomato fluorophore used as a reference since the ground truth (2P jGCaMP8f volume) is unavailable. The following images show the reconstruction using ISRA, ADMM, LFMNet [10], and finally, our approach. For each row, we show a series of 2D slices, each labelled with its corresponding depth. All the distances are measured in μm. The settings used to capture both the LF image and the 2P 3D image are specified in Section VI.

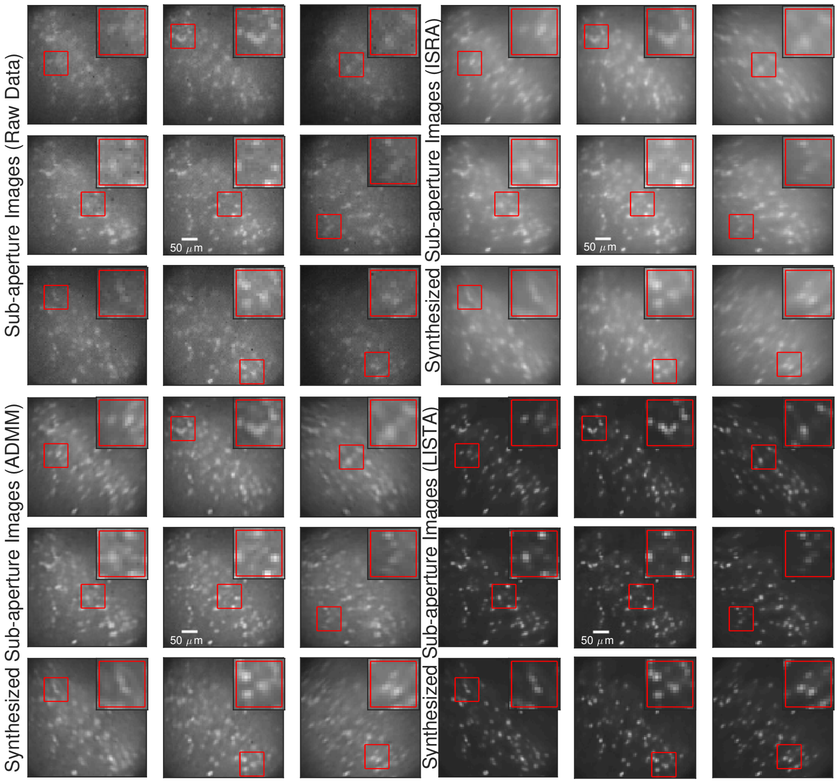
Sub-aperture Images (Raw Data)
Synthesized Sub-aperture Images (ISRA)
Synthesized Sub-aperture Images (ADMM)
Synthesized Sub-aperture Images (LISTA)
50 μm
50 μm
50 μm
50 μm

Fig. 11: Visual comparison of sub-aperture images. We show 9 different sub-aperture images (views) for 4 different cases: views from the raw LF data, LF synthesised from ISRA reconstruction, LF synthesised from ADMM method, and LF synthesised from our approach. Both model-based methods reconstruct noisy regions in the sub-aperture images that do not carry any meaningful information, which translates into a noisy 3D reconstruction. On the other hand, our approach can accurately reconstruct every neuron footprint while significantly reducing noise from scattering.

C. Reconstruction of structural 3D images from LF images

In this section, we evaluate the performance of our method for reconstructing 3D images from a single LF image. We use unseen samples to measure the reconstruction performance by measuring the Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Mean Square Error (MSE), Equivalent number of looks (ENL) [39] and Edge Preservation Index [40] (EPI). The neurons in this sample are labelled with the tdTomato fluorescent fluorophore. Since this fluorophore is bright every time it is illuminated, these LF images only give structural information and do not show neuronal activity. As mentioned previously, we captured 28 LF images for training and 28 from a different sample for testing. Each LF image corresponds to a focal depth ranging from 0 to $54\mu m$. From every LF image, we reconstruct a volume of size $321 \times 321 \times 53$ voxels covering a range $533.3 \times 533.3 \times 104$ μm^3. The size of each LF image is 2033×2033 pixels.

The number of iterations used for model-based reconstruction must be chosen properly to avoid noise amplification. As mentioned in previous works [22], [7], [41], a typical empirical number of iterations used for ISRA is between 8 and 10. We fixed this value to 8 for both ISRA and ADMM. Noise amplification is not related to the number of layers in deep-learning methods since the reconstruction mapping is learned directly from the data. Thus, to choose the number of unfolded iterations in our network, we just considered the trade-off between computational burden and network capacity. We empirically found that 6 unfolded iterations give enough capacity without affecting the computational load.

Our LISTA network achieves better average performance than other methods in a focal depth range of 54 μm in terms of PSNR, SSIM, MSE, ENL and EPI. In Table II, we show the performance for LF images taken at different focal depths, taken in steps of $2\mu m$. The depth index in Table II increases as the depth increases. Note that all methods are affected by scattering as the depth increases. Even though all deep-learning methods in Table II A outperform model-based reconstruction approaches in Table II B, our method achieves the best average performance in all the shown measures. These scores are measures on the whole volume. In our experiments, ADMM show the best average performance among model-based methods while LFMNet achieved best performance without considering ours. Since the deep learning methods are trained with a very small dataset compared to the size of the dataset used in [10],[11],[9], their performance may be affected. In contrast, our method is more robust under this adverse condition. Furthermore, deep-learning methods are much faster than model-based approaches. Table III shows the average computational time to reconstruct a volume from a single LF image. All the methods were evaluated on a GeForce GTX 1080 Ti.

As shown in Figure 9, our LISTA network achieves better qualitative reconstruction performance than other methods. In

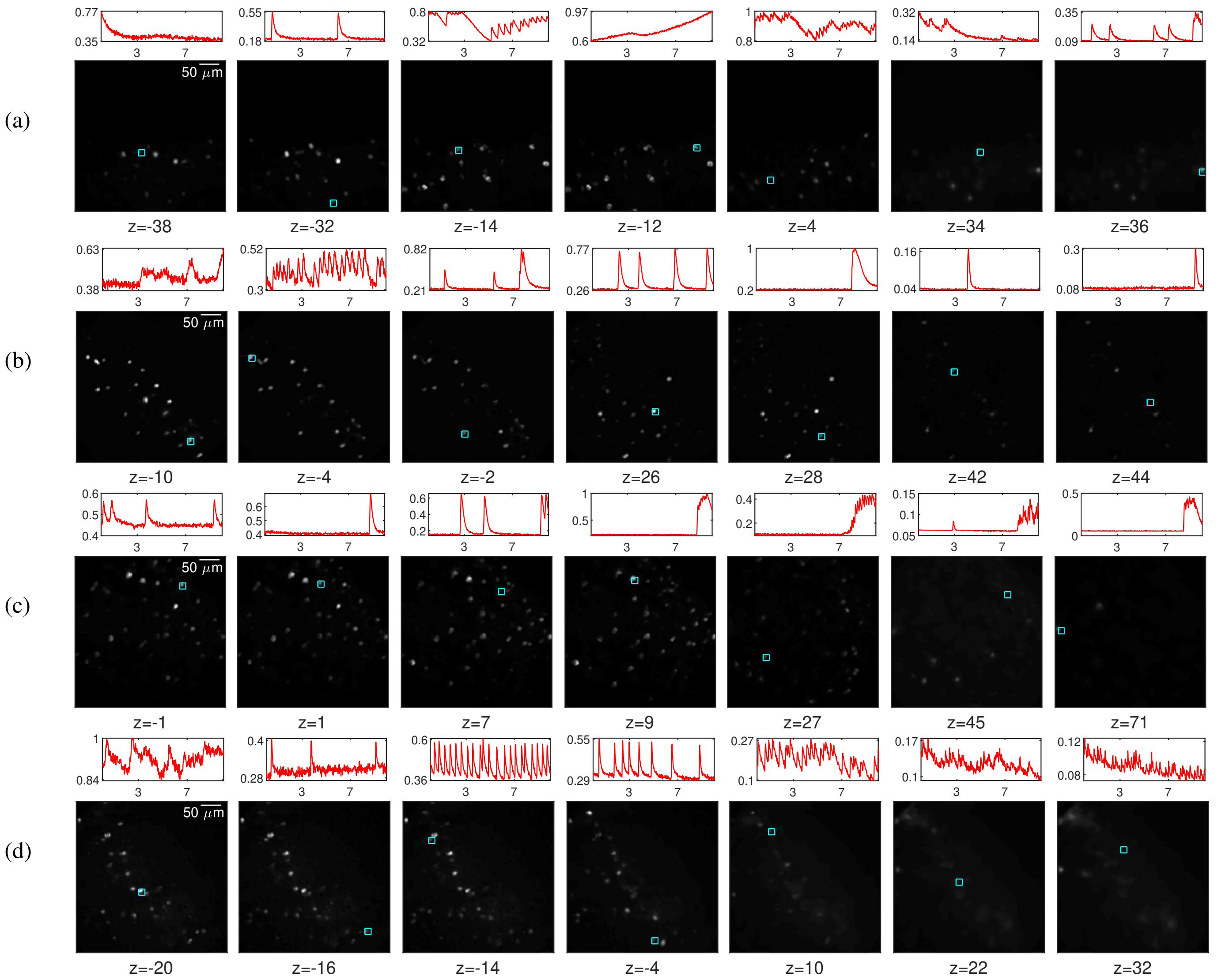

Fig. 12: Temporal evolution of neuron activity in mammalian brain tissue expressing the calcium indicator jGCaMP8f. The labels (a), (b), (c) and (d) indicate different sample tissues. We show several slices containing active neurons, the neuron of interest is marked in cyan. On top of each slice, we show the temporal evolution. The horizontal axis shows time in seconds, the vertical axis shows the normalized intensity. The normalization is performed per sample between the selected neurons. Our method reconstructs 500 frames (500 3D images) in this experiment. The LF imaging rate is 50 Hz. All the distances are measured in μm. Other additional settings are specified in Section VI.

TABLE III: Computational time.

	ISRA	ADMM	VCDNet	HyLFM	LFMNet	Ours
Time (s)	234.69	237.63	0.022	0.195	0.111	0.026

Figure 9 (a) and (b), we show visual results for two different depths corresponding to index 1 and 28 in Table II, respectively. ISRA introduces square-like artifacts strongly present near the in-focus plane, approximately from $z = -8$ μm to $z = 8$ μm in part (a) and from $z = 48$ μm to $z = 64$ μm in (b). The ADMM can effectively remove these artifacts; however, both ISRA and ADMM are affected by background noise and scattering. As one goes deeper into the tissue, the model-based methods are more affected by scattering. It is notable that learning methods achieve better performance than model-based approaches and are less affected by noise. However, our approach is visually closer to the ground truth and achieves higher PSNR and SSIM than other learning methods. For instance, see plane $z = 48$ μm. Also, note in $z = 64$ μm that the LFMNet incorrectly reconstructs neurons from neighbour depths.

D. Reconstruction of volume time series from LF images

In this section, we evaluate our method for reconstructing a temporal sequences of 3D volumes from temporal sequences of LF images. The LF sequence captures the activity of neurons labelled with the jGCaMP8f calcium indicator, which increases its fluorescence intensity when the neurons fire, at different times and focused at a fixed focal depth. Since it is impossible to capture the activity of many neurons in 3D with scanning-based techniques, ground truth data is unavailable in this case. We evaluate our approach on 4 LF sequences with 500 frames. As mentioned previously, only the first 80 LF images of three sequences were used for training, with no labels, while the rest of LF images is used for testing.

Our LISTA network performs better than model-based approaches, while the state-of-the-art neural networks fail to reconstruct volumes for jGCaMP8f-labelled brain tissues. In Figure 10, we show the visual performance of ISRA, ADMM, and our method for reconstruction of one frame of the sequence ($300th$ frame). We show two different samples in part (a) and (b). Even though the LFMNet [10] achieved satisfactory performance in the previous section, it generalizes poorly to the reconstruction of the temporal sequence due to the small training dataset. A more specific reason is that the use of a different fluorophore, the jGCaMP8f, implies samples with different noise levels and light sources with different wavelengths than those used for training. Figure 10 suggests that model-based methods are more robust under these adverse conditions than learning-based approaches, as mentioned in the introduction. However, model-based methods are heavily affected by scattering due to the lack of a strong regularization. In addition, ISRA introduces strong artifacts near the plane $z = 0$. In our approach, we exploit the knowledge of the forward model and the few available labels to achieve improved reconstruction performance.

Our training loss is designed to avoid amplifying noise from scattering. In Figure 11, we show the LF images synthesized from the reconstructed volumes. We display 3×3 views per LF image from the total 19×19 views. The re-synthesized LF image from ADMM shows that noise is reduced compared to the ISRA approach since the ADMM method imposes additional regularizers in the objective function. However, noise in the tissue region is still significant. Our approach greatly reduces noise while reconstructing every neuron footprint in the ground truth LF image. We achieve this performance due to the adversarial regularizer and the specialized content losses used for training. We found that the content loss alone $\mathcal{L}_{c1}$ does not allow for reconstruction of functional data, even if other losses such as L_1, perceptual or Charbonnier losses are used. We also evaluate alternative architectures for reconstruction. See supplemental material for more details.

Our approach provides a new powerful tool to study the fast temporal evolution of neurons in mammalian brain tissue. In Figure 12, we show the temporal neuronal activity revealed by the jGCaMP8f. This indicator emits fluorescence due to changes in intracellular calcium concentration, which is an indirect measurement of electrical activity. We show four brain samples in parts (a), (b), (c) and (d). Note how neurons located at different positions in the 3D space show different activity. For this experiment, we reconstruct 500 3D volumes per LF sequence, each with the same size as in previous experiments. Since the imaging rate is 50 Hz, we show 10 seconds of neuronal activity. We highlight that this task is challenging to achieve with other optical modalities. For instance, multi-photon recording of genetically encoded calcium indicators is typically performed by scanning planes at sampling rates below 30 Hz [42]. Although LFM cannot penetrate as deeply into scattering tissue as multi-photon imaging, our strategy enables volume acquisition at a fast rate for imaging mammalian brain tissue with the improved quality compared to traditional reconstruction approaches for LFM. In our experiment, the equivalent plane sampling frequency is 53×50 Hz, approximately 88 times faster than the mentioned multi-photon modality.

VII. Conclusion

We have introduced a physics-driven deep learning approach to reconstruct 3D volumes from LF sequences. The network architecture is inspired by the fact that labelled neurons in tissues are sparse, which motivates an architecture based on unfolding the ISTA algorithm. Additionally, we demonstrate how the forward model of a LF microscope can be described using a deep linear convolutional network. Finally, we utilize an adversarial loss and leverage the theoretical forward model to train our network with a limited labelled dataset.

We show that our method achieves better quality than other state-of-the-art methods for reconstructing temporal LF sequences imaged with the jGCaMP8f indicator. Since other deep-learning methods require labelled data expressing the jGCaMP8f for training, they cannot perform this task in our scenario. Furthermore, our approach showed better performance than standard model-based and learning-based methods in terms of PSNR and SSIM for reconstructing structural 2P volume imaged using the tdTomato fluorophore. Although LFM cannot penetrate as deeply into scattering tissue as functional 2P imaging, our 2P-enhanced LFM strategy enables light-efficient volume acquisition at fast rates, essential to capturing neuronal dynamics transduced by fast sensors such as jGCaMP8f [43].

Our work offers a practical method to perform 3D reconstruction for LF microscopy under adverse acquisition conditions when imaging mammalian brain tissue. We believe that the proposed method could be helpful beyond this scenario in problems that require high computational cost where periodically shift-invariant property holds.

Acknowledgment

The authors would like to thank Dr. Yu Liu for preparing the plasmids and for assisting with the fluorophore transfection.

References

[1] H. Verinaz-Jadan, P. Song, C. L. Howe, P. Quicke, A. J. Foust, and P. L. Dragotti, "Deep learning for light field microscopy using physics-based models," in *ISBI 2021 - 2021 IEEE International Symposium on Biomedical Imaging (ISBI)*, 2021.

[2] S. Weisenburger and A. Vaziri, "A guide to emerging technologies for large-scale and whole brain optical imaging of neuronal activity," *Annual review of neuroscience*, vol. 41, p. 431, 2018.

[3] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 924–34, 2006. [Online]. Available: http://dx.doi.org/10.1145/1141911.1141976

[4] M. Broxton, L. Grosenick, S. Yang, A. A. N. Cohen and, K. Deisseroth, and M. Levoy, "Wave optics theory and 3-d deconvolution for the light field microscope," *Optics express*, vol. 21, no. 21, pp. 25 418–25 439, 2013.

[5] H. Verinaz-Jadan, P. Song, C. L. Howe, A. J. Foust, and P. L. Dragotti, "Shift-invariant-subspace discretization and volume reconstruction for light field microscopy," *IEEE Transactions on Computational Imaging*, vol. 8, pp. 286–301, 2022.

[6] Z. Lu, J. Wu, H. Qiao, Y. Zhou, T. Yan, Z. Zhou, X. Zhang, J. Fan, and Q. Dai, "Phase-space deconvolution for light field microscopy," *Opt.Express*, vol. 27, no. 13, pp. 18 131–18 145, Jun 2019. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-27-13-18131

[7] A. Stefanoiu, J. Page, P. Symvoulidis, G. G. Westmeyer, and T. Lasser, "Artifact-free deconvolution in light field microscopy," *Opt. Express*, vol. 27, no. 22, pp. 31 644–31 666, Oct 2019. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-27-22-31644

[8] P. Song, H. Verinaz-Jadan, C. L. Howe, P. Quicke, A. J. Foust, and P. L. Dragotti, "3D localization for light-field microscopy via convolutional sparse coding on epipolar images," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1017–1032, 2020.

[9] Z. Wang, L. Zhu, H. Zhang, G. Li, C. Yi, Y. Li, Y. Yang, Y. Ding, M. Zhen, S. Gao *et al.*, "Real-time volumetric reconstruction of biological dynamics with light-field microscopy and deep learning," *Nature Methods*, vol. 18, no. 5, pp. 551–556, 2021.

[10] J. P. Vizcaíno, F. Saltarin, Y. Belyaev, R. Lyck, T. Lasser, and P. Favaro, "Learning to reconstruct confocal microscopy stacks from single light field images," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 775–788, 2021.

[11] N. Wagner, F. Beuttenmueller, N. Norlin, J. Gierten, J. C. Boffi, J. Wittbrodt, M. Weigert, L. Hufnagel, R. Prevedel, and A. Kreshuk, "Deep learning-enhanced light-field imaging with continuous validation," *Nature Methods*, vol. 18, no. 5, pp. 557–563, 2021.

[12] P. Quicke, C. L. Howe, P. Song, H. V. Jadan, C. Song, T. Knöpfel, M. Neil, P. L. Dragotti, S. R. Schultz, and A. J. Foust, "Subcellular resolution three-dimensional light-field imaging with genetically encoded voltage indicators," *Neurophotonics*, vol. 7, no. 3, p. 035006, 2020.

[13] C. L. Howe, P. Quicke, P. Song, H. V. Verinaz-Jadan, P. L. Dragotti, and A. J. Foust, "Comparing synthetic refocusing to deconvolution for the extraction of neuronal calcium transients from light fields," *Neurophotonics*, vol. 9, no. 4, p. 041404, 2022. [Online]. Available: https://doi.org/10.1117/1.NPh.9.4.041404

[14] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on international conference on machine learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, pp. 399–406.

[15] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1828–1837.

[16] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3217–3226.

[17] R. Wang and W.-N. Lee, "A general deep learning model for ultrasound localization microscopy," in *2022 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2022, pp. 1–4.

[18] G. Dardikman-Yoffe and Y. C. Eldar, "Learned sparcom: unfolded deep super-resolution microscopy," *Optics express*, vol. 28, no. 19, pp. 27 736–27 763, 2020.

[19] Y. Li, Y. Su, M. Guo, X. Han, J. Liu, H. D. Vishwasrao, X. Li, R. Christensen, T. Sengupta, M. W. Moyle *et al.*, "Incorporating the image formation process into deep learning improves network performance," *Nature Methods*, vol. 19, no. 11, pp. 1427–1437, 2022.

[20] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging: Briefly revisiting 25 years of research," *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, 2016.

[21] M. E. Daube-Witherspoon and G. Muehllehner, "An iterative image space reconstruction algorthm suitable for volume ect," *IEEE transactions on medical imaging*, vol. 5, no. 2, pp. 61–66, 1986.

[22] T. Nöbauer, O. Skocek, A. P.-A. J., L. Weilguny, F. M. Traub, M. I. Molodtsov, and A. Vaziri, "Video rate volumetric ca2+ imaging across cortex using seeded iterative demixing (sid) microscopy," *Nature Methods*, vol. 14, p. 811, 2017. [Online]. Available: https://doi.org/10.1038/nmeth.4341

[23] X. Li, H. Qiao, J. Wu, Z. Lu, T. Yan, R. Zhang, X. Zhang, and Q. Dai, "Deeplfm: Deep learning-based 3d reconstruction for light field microscopy," in *Novel Techniques in Microscopy*. Optica Publishing Group, 2019, pp. NM3C–2.

[24] R. Ng, M. Levoy, M. B. 'edif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Tech. Rep., apr 2005. [Online]. Available: http://graphics.stanford.edu/papers/lfcamera/

[25] Z. Zhang and M. Levoy, "Wigner distributions and how they relate to the light field," pp. 1–10, 2009.

[26] X. Li, H. Qiao, J. Wu, Z. Lu, T. Yan, R. Zhang, X. Zhang, and Q. Dai, "DeepLFM: Deep learning-based 3D reconstruction for light field microscopy," pp. NM3C–2, 04/14 2019, j2: NTM; T3: The Optical Society. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=NTM-2019-NM3C.2

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[28] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] C. Zhang, S. Moeller, O. B. Demirel, K. Uğurbil, and M. Akçakaya, "Residual raki: A hybrid linear and non-linear approach for scan-specific k-space deep learning," *NeuroImage*, vol. 256, p. 119248, 2022.

[30] N. C. Pegard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller, "Compressive light-field microscopy for 3D neural activity recording," *Optica*, vol. 3, no. 5, pp. 517–524, May 2016. [Online]. Available: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-3-5-517

[31] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009. [Online]. Available: https://doi.org/10.1137/080716542

[33] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.

[34] S. Lunz, O. Öktem, and C.-B. Schönlieb, "Adversarial regularizers in inverse problems," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[35] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[36] D. Deb, Z. Jiao, R. R. Sims, A. B.-Y. Chen, M. Broxton, M. Ahrens, K. Podgorski, and S. C. Turaga, "Fouriernets enable the design of highly non-local optical encoders for computational imaging," in *Advances in Neural Information Processing Systems*.

[37] O. Susladkar, G. Deshmukh, S. Nag, A. Mantravadi, D. Makwana, S. Ravichandran, G. H. Chavhan, C. K. Mohan, S. Mittal *et al.*, "Clarifynet: A high-pass and low-pass filtering based cnn for single image dehazing," *Journal of Systems Architecture*, vol. 132, p. 102736, 2022.

[38] G. Kwon, C. Han, and D.-s. Kim, "Generation of 3D brain mri using auto-encoding generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 118–126.

[39] S. N. Anfinsen, A. P. Doulgeris, and T. Eltoft, "Estimation of the equivalent number of looks in polarimetric synthetic aperture radar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3795–3809, 2009.

[40] F. Sattar, L. Floreby, G. Salomonsson, and B. Lovstrom, "Image enhancement based on a nonlinear multiscale method," *IEEE transactions on image processing*, vol. 6, no. 6, pp. 888–895, 1997.

[41] R. Prevedel, Y. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, and A. Vaziri, "Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy," *Nature methods*, vol. 11, no. 7, p. 727, 2014.

[42] V. Villette, M. Chavarha, I. K. Dimov, J. Bradley, L. Pradhan, B. Mathieu, S. W. Evans, S. Chamberland, D. Shi, R. Yang *et al.*, "Ultrafast two-photon imaging of a high-gain voltage indicator in awake behaving mice," *Cell*, vol. 179, no. 7, pp. 1590–1608, 2019.

[43] Y. Zhang, M. Rózsa, D. Bushey, J. Zheng, D. Reep, Y. Liang, G. J. Broussard, A. Tsang, G. Tsegaye, R. Patel, S. Narayan, J.-X. Lim, R. Zhang, M. B. Ahrens, G. C. Turner, S. S.-H. Wang, K. Svoboda, W. Korff, E. R. Schreiter, J. P. Hasseman, I. Kolb, and L. L. Looger, "jGCaMP8 Fast Genetically Encoded Calcium Indicators," 12 2020. [Online]. Available: https://janelia.figshare.com/articles/online_resource/jGCaMP8_Fast_Genetically_Encoded_Calcium_Indicators/13148243