Biomedical Optics EXPRESS

Fast light-field 3D microscopy with out-of-distribution detection and adaptation through conditional normalizing flows

JOSUÉ PAGE VIZCAÍNO,[1,2,*] PANAGIOTIS SYMVOULIDIS,[3] ZEGUAN WANG,[3] JONAS JELTEN,[1,2] PAOLO FAVARO,[4] EDWARD S. BOYDEN,[3] AND TOBIAS LASSER[1,2]

[1] *Computational Imaging and Inverse Problems, Department of Computer Science, School of Computation, Information and Technology, Technical University of Munich, Germany*
[2] *Munich Institute of Biomedical Engineering, Technical University of Munich, Germany*
[3] *Synthetic Neurobiology Group, Massachusetts Institute of Technology, USA*
[4] *Computer Vision Group, University of Bern, Switzerland*
[*] *pv.josue@gmail.com*

Abstract: Real-time 3D fluorescence microscopy is crucial for the spatiotemporal analysis of live organisms, such as neural activity monitoring. The eXtended field-of-view light field microscope (XLFM), also known as Fourier light field microscope, is a straightforward, single snapshot solution to achieve this. The XLFM acquires spatial-angular information in a single camera exposure. In a subsequent step, a 3D volume can be algorithmically reconstructed, making it exceptionally well-suited for real-time 3D acquisition and potential analysis. Unfortunately, traditional reconstruction methods (like deconvolution) require lengthy processing times (0.0220 Hz), hampering the speed advantages of the XLFM. Neural network architectures can overcome the speed constraints but do not automatically provide a way to certify the realism of their reconstructions, which is essential in the biomedical realm. To address these shortcomings, this work proposes a novel architecture to perform fast 3D reconstructions of live immobilized zebrafish neural activity based on a conditional normalizing flow. It reconstructs volumes at 8 Hz spanning 512x512x96 voxels, and it can be trained in under two hours due to the small dataset requirements (50 image-volume pairs). Furthermore, normalizing flows provides a way to compute the exact likelihood of a sample. This allows us to certify whether the predicted output is in- or ood, and retrain the system when a novel sample is detected. We evaluate the proposed method on a cross-validation approach involving multiple in-distribution samples (genetically identical zebrafish) and various out-of-distribution ones.

1. Introduction

Analysis of fast biological processes on live specimens is a crucial step in biomedical research, where fluorescence 3D microscopy plays an essential role due to its ability to visualize specific structures and processes, either through intrinsic contrast or an ever-grown collection of labeling techniques, including for example genetically encoded indicators of neuronal activity.

The xlfm [1], or flfmic [2–5], offers scan-less captures of transparent samples, affordability, and simplicity over scanning microscopes like spinning disk confocal and light sheet (capturing at 10Hz [6]). But it comes at the expense of requiring a 3D reconstruction in a post-acquisition step.

Traditionally, reconstructions are done using iterative methods, like the Richardson-Lucy deconvolution [7], where the reconstruction quality is sufficient to discern neural activity. However, large computation hardware allocations and long waiting times are required (1 second

#504039 https://doi.org/10.1364/BOE.504039
Journal © 2024 Received 24 Aug 2023; revised 9 Nov 2023; accepted 20 Nov 2023; published 30 Jan 2024

per iteration in an iterative algorithm), making it impractical for real datasets where thousands of images must be processed.

Deep learning approaches arose as an alternative, where fast reconstructions are possible (up to 50Hz). These networks are trained on pairs of either raw XLFM or LFM [8] images and 3D volumes. For example XLFMNet [9], VCD [10], LFMNet [11], HyLFM-Net [12], and others.

However, within these methods, only the HyLFM-Net can detect network deviations or incapability of handling new sample types, but at the expense of a complex imaging system mixing light-sheet illumination with lf imaging for continuous validation. An algorithm's lack of certainty metrics renders it unsafe to be integrated into established experimental imaging workflows, as the network might introduce artifacts or hallucinations that cannot be detected.

These motivations bring our attention to more statistically informed methods, such as nfs [13,14], a type of invertible neural network recently used for biomedical imaging [15], inverse problems [16–19] and other applications like image generation [20,21]. nfs learn a mapping between an arbitrary statistical distribution and a normal distribution through a set of invertible and differentiable functions. Also, a tractable exact likelihood allows for probing the quality of this mapping for individual samples, which, in turn, allows for deciding what to do with the new sample, perhaps retraining the network with the new data, if desired.

One disadvantage of nfs is that due to the required invertible mapping, no data bottlenecks are possible (like in an encoder-decoder approach) as this would lead to information loss, making it necessary to store all the tensors and gradients in memory during training. This limits the data size that can be used due to the limited gpu memory. The conventional solution to this issue is to split the processed tensor after each invertible function [22], feed only one part to the following function, and concatenate the other part to the output tensor of the nf. However, when working with large tensors, the computational cost of processing the gradients during training still overwhelms conventional gpus like the ones found on image acquisition workstations.

Hence, wf [20] was introduced, where an invertible down-sampling operation is used (such as the Haar transform [23]) to serially down-sample the input image to the desired size (could be down to a single pixel), where a nf is used to learn the Haar detail coefficients required to perform up-sampling when performing reconstruction. The last down-sampling step comprises an nf that directly learns the probability distribution of the lowest-resolution image. This allows independent training of each down-sampling nf with commercial gpus, allowing their usage for high data throughput for the first time. Generating a new image involves running the wf backward: the lowest resolution image is sampled from an NF, then up-sampled through all the wfs until reaching the original resolution.

The original wf approach is not designed for inverse problems, as it ignores the image formation model prior knowledge, such as the raw XLFM image or the point spread function. Furthermore, when training each nfs, the Haar transform operator generates a down-sample of the gt volume down to the lowest resolution. And when performing a 3D reconstruction, the low-resolution volume received by the nfs might slightly deviate from the gt. These deviations accumulate as the volumes are propagated upwards through the network, affecting the output 3D volume heavily. The quality of the lowest-resolution volume dramatically impacts the final reconstruction, making the lowest-resolution initial guess fundamental to the quality of the full-resolution reconstruction.

Our work proposes the cwfa, shown in Fig. 1, suited for the 3D reconstruction and OODD of live immobilized fluorescent zebrafish, imaged through an xlfm. We used a hierarchical multi-scale approach based on wfs, conditioned by the XLFM-measured image and a 3D volume prior (the mean of the training volumes). It could be easily modified to work on freely behaving animals by removing the dependency on a 3D volume prior. The cwfs reconstruction and ood capabilities enable a fast, accurate, and robust method for 3D fluorescence microscopy. Fast enough to be applied within closed-loop or human-in-the-loop experiments, where on the fly analysis of the activity could trigger further steps in an experiment.

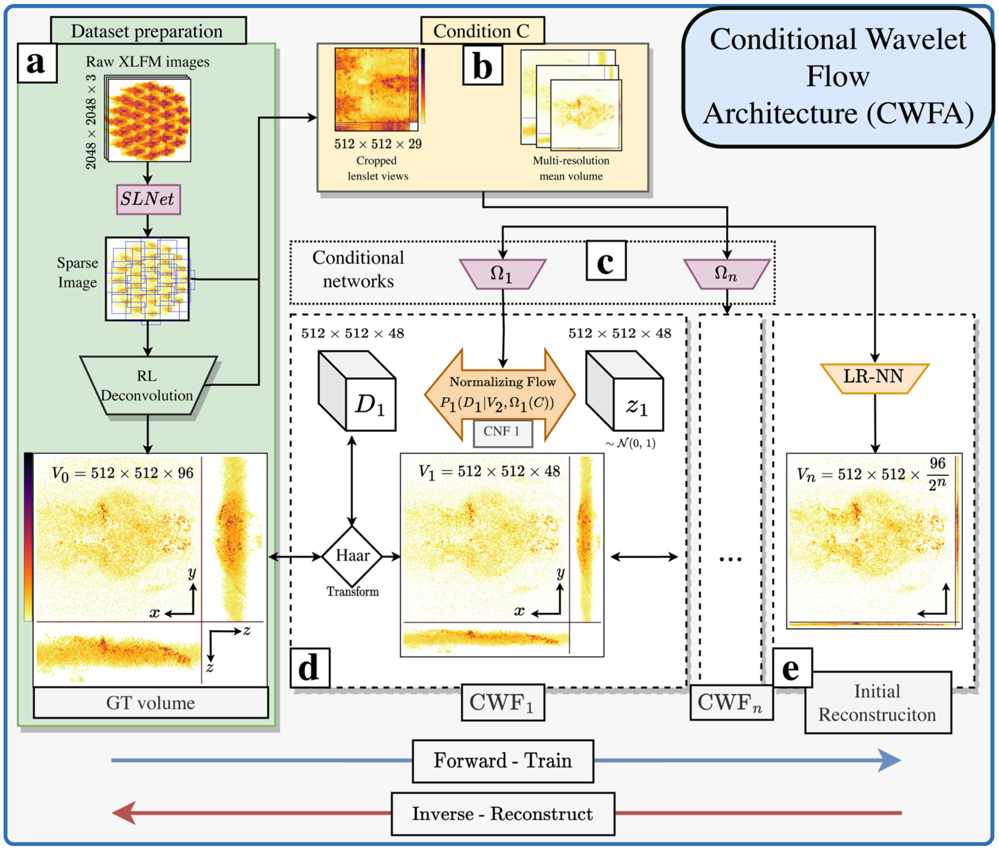

Fig. 1. The Conditional Wavelet Flow Architecture and workflow: (a) Data preparation, extracting the sparse spatiotemporal signal from the raw XLFM acquisitions with the SLNet [9] and performing 3D reconstructions using the RL algorithm. In (b), the conditions are prepared by cropping and stacking the images and computing the mean of the training volumes. In (d), the full-resolution GT volume V_0 and conditions (b) are used to train the cwfs. Training is performed in each cwf individually and consists in feeding V_0 and the processed condition $\Omega_1(C)$ to the cwf$_1$. This generates as outputs V_1 and z_1. The latter is used in loss function (Eq. (3)). V_1 is fed to the next cwf, which is trained similarly. This is repeated until it reaches the lowest resolution output. The low-resolution neural network (LR-NN) is trained deterministically with V_n and C pairs.

OODD is possible given the access to an exact likelihood computation. This allows for evaluating how likely a sample is to belong to the training distribution. Specifically, in a testing step, the likelihood of a novel sample can be computed by processing it with the cwf, as described in sec. 3.2 and Fig. 2. Even though the literature mentions that nfs are not reliable for detecting ood samples [24], we find that the proposed cwfa and sample type enables this capability.

We demonstrate the reliability of the proposed cwfa on XLFM acquisitions of live immobilized zebrafish larvae full-brain neural activity, processed with the SLNet [9], which separates the neural activity from the background of the acquisitions. As a gold standard or gt, we used a state-of-the-art 3D reconstruction method (100 iterations with the RL algorithm) of the SLNet output.

In short, our work is motivated by the need for fast and reliable 3D reconstructions. The proposed CWFA archives this by combining the best of traditional methods and cnns; integrating

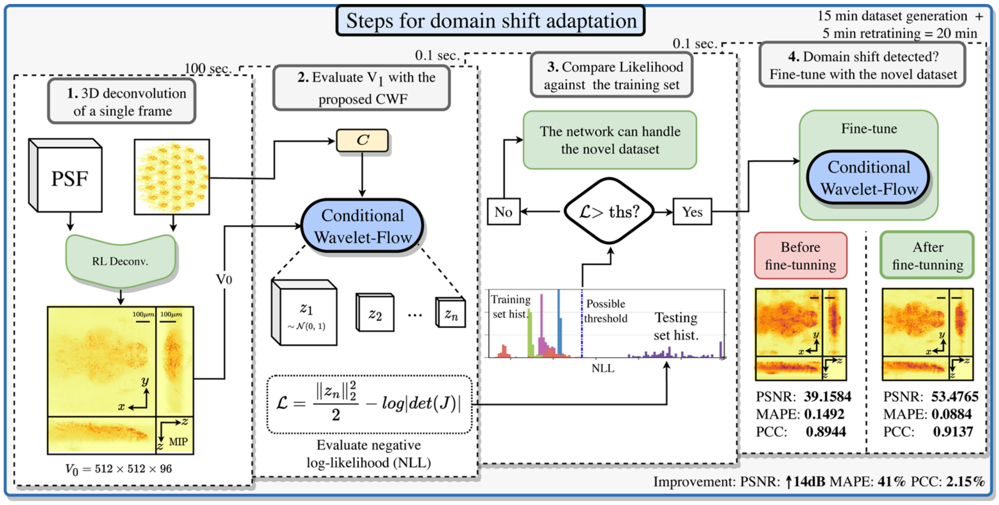
Steps for domain shift adaptation
100 sec.
1. 3D deconvolution of a single frame
PSF
RL Deconv.
100μm
100μm
y
x
z
z
MIP
$V_0 = 512 \times 512 \times 96$
V0
0.1 sec.
2. Evaluate V_1 with the proposed CWF
C
Conditional Wavelet-Flow
z_1
$\sim \mathcal{N}(0,1)$
z_2
...
z_n
$\mathcal{L} = \frac{\|z_n\|_2^2}{2} - log|det(J)|$
Evaluate negative log-likelihood (NLL)
0.1 sec.
3. Compare Likelihood against the training set
The network can handle the novel dataset
No
$\mathcal{L}>$ ths?
Yes
Training set hist.
Possible threshold
Testing set hist.
NLL
15 min dataset generation + 5 min retratining = 20 min
4. Domain shift detected? Fine-tune with the novel dataset
Fine-tune
Conditional Wavelet-Flow
Before fine-tunning
After fine-tunning
PSNR: 39.1584
MAPE: 0.1492
PCC: 0.8944
PSNR: 53.4765
MAPE: 0.0884
PCC: 0.9137
Improvement: PSNR: ↑14dB MAPE: 41% PCC: 2.15%

Fig. 2. Steps for OOD detection and domain shift adaptation. One image is deconvolved and used for OOD detection, and 10 images for fine-tuning the networks. The total data-set generation time is 15 minutes (deconvolving 10 samples x 1.5 min/sample), and training was performed in 5 minutes, adding a total fine-tuning time of 20 minutes.

the reliability given by the out-of-distribution detection and adaptation of normalizing flows and the speed of convolutional neural networks (as in the internal blocks of the CWFA).

This manuscript starts with the materials and methods in Sec. 2, from the basics of normalizing flows until deep into the proposed conditional wavelet flow architecture. Later, in the experiments section, we characterize the architecture in terms of speed, image reconstruction quality, ood detection capabilities, and estimate the effort needed to re-train and adapt a network with ood samples. Finally, in the discussion at Sec. 4, we comment on the results and ideas for future work.

2. Materials and methods

2.1. *Traditional conditional normalizing flows*

A normalizing flow [14] (NF), through a sequence of invertible and differentiable functions, transforms an arbitrary distribution $p_X(\boldsymbol{x})$ into the desired distribution $p_Z(z)$ (usually a normal distribution, hence the name Normalizing Flows). This is possible through the change of variables formula from probability theory. The density function of the random variable $\boldsymbol{X}$ is given by:

$$p_X(\boldsymbol{x}) = p_Z(z)|\det(J)|, \tag{1}$$

where z is normally distributed (mean 0 and variance 1), with probability density function $p_Z(z)$. An NF can be trained by setting $z = f_\Theta(\boldsymbol{x})$, where f_Θ is an invertible differentiable function parameterized by Θ, and $J = J_\Theta = \frac{\partial f_\Theta(\boldsymbol{x})}{\partial x}$ is the Jacobian of f_Θ with respect to $\boldsymbol{x}$, also known as the volume correction term. As seen in Eq. (1), a tractable and easily computable Jacobian determinant of $f_\Theta(\boldsymbol{x})$ is preferred (for example, where the Jacobian is block-triangular or diagonal). Hence, choosing the functions $f_\Theta(\boldsymbol{x})$ is a crucial and well-studied step [22,25,26], out of the scope of this work.

A nf can be modified into a cnf [15] and represent a conditional distribution $p_X(\boldsymbol{X}|\boldsymbol{C})$, for a set of observations $\boldsymbol{X} = \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}$ and conditions $\boldsymbol{C} = \boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \dots, \boldsymbol{c}^{(N)}$. The likelihood from Eq. (1) for a single sample (i) becomes:

$$p_{\boldsymbol{X}}(\boldsymbol{x}^{(i)}|\boldsymbol{c}^{(i)},\Theta) = p_{\boldsymbol{Z}}(z^{(i)} = f_\Theta(\boldsymbol{x}^{(i)},\boldsymbol{c}^{(i)})) \cdot |\det(J_\Theta^{(i)})|. \tag{2}$$

To train a cnf, we need to find the optimal values of the parameters Θ that maximize the likelihood or minimize the negative log-likelihood of Eq. (2), defined as follows:

$$\Theta^* = \arg\min_\Theta \sum_{i=1}^{N} \left[\frac{||f_\Theta(x^{(i)},c^{(i)})||_2^2}{2} - \log|\det(J_\Theta^{(i)})| + \rho||\Theta||_2^2 \right], \tag{3}$$

where $\det(J_\Theta^{(i)})$ is the determinant of the Jacobian matrix with respect to $x^{(i)}$, evaluated at $f_\Theta(x^{(i)},c^{(i)})$. And $\rho||\Theta||_2^2$ is the likelihood of the posterior over the model's parameters, assuming a Gaussian distribution weighted by ρ.

During training, we minimize the negative log-likelihood function with respect to the parameters Θ using the Lion optimizer [27] using weight decay for the parameters posterior. The networks were trained for 500 epochs (100 epochs for each CWF step and 100 for the LR-NN), with a learning rate of 2.21e-5 for the CWFs, 8.84e-5 for the conditional networks, and 0.8e-5 for the LR-NN.

After training the cnf, we can perform inference on a new image by sampling from the base distribution $p_{\boldsymbol{Z}}(z)$ (in our case, a normal distribution $\mathcal{N}(\mu,\sigma^2)$) and obtaining $\boldsymbol{x}$ by applying the inverse transformation of the flow: $\boldsymbol{x} = f_\Theta^{-1}(z)$. The resulting $\boldsymbol{x}$ will be a sample from the conditional distribution $p_{\boldsymbol{X}}(\boldsymbol{X}|\boldsymbol{C})$. Fig. S1 shows a graphical representation of a cnf used for inference. Even though this method is mathematically sound, its lossless nature limits its usability in practice for large 3D volumes due to high memory requirements.

2.2. Proposed conditional wavelet flow architecture

The WF architecture [20], uses a multi-scale hierarchical approach that allows training each up/down-scale independently, allowing flexibility during memory management. Each down-sampled operation is a Haar transform chosen due to its orthonormality and invertibility. Conversely, to regular wfs, we used the Haar transform to scale the volumes in the axial dimension, preserving lateral resolution across all steps. The likelihood function of a WF model is given by:

$$p(V_0) = p(V_n)\prod_{i=1}^{n-1} p(D_i|V_i) \tag{4}$$

where V_0 is the final high-resolution volume, D_i the Haar transform detail coefficients and V_i the volume down-sampled i times. $p(V_n)$ and all $p(D_i|V_i)$ are normalizing flows.

In the proposed cwfa, we substituted the lowest resolution $p(V_n)$ with a deterministic cnn. This is necessary because the hierarchical reconstruction approach used, depends on a low-resolution initial reconstruction. We found quality advantages when using a cnn over a nf as in the traditional wf as shown in the comparisons presented in the results section. Additionally, we conditioned all the up-sampling nfs on a set of external conditions C processed by the cnn Ω_i, as shown in Fig. 1. Modifying the likelihood to:

$$p(V_0) = p(V_n)\prod_{i=1}^{n-1} p(D_i|\Omega_i(C)), \tag{5}$$

with the ll,

$$\log p(V_0) = \log p(V_n) + \sum_{i=1}^{n-1} \log p(V_i|\Omega_i(C)), \tag{6}$$

The final loss function can be optimized independently, as $\log p(V_0)$ comprises a sum of independent probabilities. This is an advantage of the Wavelet-Flow architecture when using conventional computing hardware.

2.2.1. Network implementation

The proposed architecture uses four cwfs, each comprised of a Haar transform down-sampling operation and an cnf, as seen in Fig. 1. The internal cwf learns the mapping between the Haar coefficients and a normal distribution. It is built from 6 cat blocks (see Supplement 1). The number of blocks and their parameters, such as the type of invertible blocks (like GLOW [26], RNVP [22], HINT [28], NICE [13], cat [25] etc.), the number of parameters in each internal convolution, were optimized for Pearson-correlation-coefficient in a grid-like fashion shown in Fig. S2. The architecture was implemented in PyTorch aided by the Freia framework for easily invertible architectures [29]. We invite the reader to explore this project's source code for further implementation details under https://github.com/pvjosue/CWFA. The network's total number of parameters is 73 million, where 3 million comprise the cwfs, $82K$ the conditional networks Ω_{0-n} and 63 million used by the low-resolution neural network (LR-NN). The large size of the latter indicates that the low-resolution reconstruction is a particularly hard problem, mainly due to the high dimensionality of the volumes and the lack of statistical priors. We refer the reader to the Supplement 1 for additional information.

2.3. *3D reconstruction: sampling from the conditional wavelet flow architecture*

Figure 1 depicts how to train the cwfa (forward pass) and reconstruct 3D volumes from XLFM images (inverse pass). Reconstruction with a trained cwfa involves, first, reconstructing a low-resolution volume ($\tilde{V}_n$ = LR-NN(C)), where LR-NN is a deterministic cnn and C the conditions, and using it as input to the cwf$_{n-1}$. To up-sample on CWF$_{n-1}$: first, sample z_{n-1} from a normal distribution, and input the pre-processed XLFM image and 3D prior conditions $\Omega_{n-1}(C)$ as a condition to the NF. Then, generate the Haar coefficients (D_{n-1}) used for up-sampling $\tilde{V}_{n-1}$ by a factor of 2 on the axial dimension with the Haar transform and generate $\tilde{V}_{n-2}$. We repeat this process until reaching $\tilde{V}_0$ at full resolution. Our architecture comprises 4 cwf and LR-NN. Each cwf uses 6 cnfs internally, with 14 channels per convolution and cat invertible blocks, as illustrated in Fig. S1. A key parameter during sampling is the temperature parameter, which determines if the z should be sampled from a truncated distribution and to what degree, which is discussed in sec. 3.1.2.

2.4. *Input for training*

As an input for training the network, the gt volume at full resolution V_0 and the conditions C are required. We pre-processed the raw data with the SLNet [9] for both, extracting the sparse activity from the image sequences. This approach was chosen due to the fluorescent labeling used (GCaMP), which has only a slight intensity increase (<10%) when intracellular calcium concentration increases as a result of neuronal firing. In other words, the neural activity is practically invisible on the raw sequences that suffer from substantial auto-fluorescence.

2.5. *Conditioning the wavelet flows*

In the original WF [20], each NF uses only the next level low-resolution volume as a condition, intending to generate human faces from a learned distribution, where the low-resolution Haar transformed image is used as a condition on each up-sampling step. In such a case, face distribution is the only known information.

However, when dealing with inverse problems (for example, fluorescence microscopy), prior information about the system and volume to reconstruct are known, such as the forward process in the form of a point spread function (PSF), the captured microscope image and in our case the structural information of a fish, as these are immobilized with agarose and do not move during the acquisition.

After ablating different configurations (see Fig. S2), we chose to use cat blocks due to their excellent performance and simplicity. These split the input condition in two, comprised of a translation and a scaling factor (as seen in Fig. S1) that is applied to the cwf's input in the forward or backward direction.

The following conditions are fed to each CWF$_i$ after being pre-processed by $\Omega_i(C)$.

2.5.1. Condition 1: views cropped from the XLFM image

This condition acts as the scaling factor of the block and informs the network about neural potential changes. Due to the nature of the XLFM microscope, each microlens acts as an individual camera, and the 2D image can be interpreted as a multi-view camera problem. The raw XLFM input image is prepared first by detecting the center of each micro-lens from the central depth of the measured PSF (using the Python library *findpeaks* [30]), then cropping a 512 × 512 area around the 29 centers, and stacking the images in the channel dimension, as seen in Fig. 1 panel (b).

2.5.2. Condition 2: a 3D volume structural prior

When working with immobilized animals, there is the advantage that only the neural activity changes within consecutive frames. Hence, we included a 3D volume as a condition; when training, this volume is the mean of the GT volumes per fish, which are either way required to train the network. These mean volumes are computed once and stored for use during testing, meaning that no additional deconvolution is required. For novel samples, we deconvolve the first volume of the series and use that as a condition.

Providing a volumetric prior simplifies the reconstruction problem and allows the network to focus only on updating the neural activity instead of reconstructing a complete 3D volume. Furthermore, as the Haar coefficients along the channel dimension are a discrete derivative of the volume, we found that a processed version of the volume is a very initial approximation, which will be fine-tuned by the cwf.

2.5.3. Structure of the conditional networks Ω_{0-n}

Previous methods using cnf used feature extractors from pre-trained networks. We don't rely on pre-trained networks. Instead, we trained simultaneously the conditional networks Ω with the cwfs. This is achieved by adding a second data term to the loss function from Eq. (3), resulting in the final loss function:

$$\Theta_i = \Theta_i^* + \alpha * \arg\min_{\Theta} \sum_{i=1}^{N} \left[||V_i - \tilde{V}_i)||_2^2 \right], \tag{7}$$

where α is a weighting factor (0.48 found through a grid search approach), $\tilde{V}_i$ is the reconstructed volume at the cwf i and V_i is the GT volume down-sampled i times. With this modified loss function, we ensure that the volumes generated by the cwfa have not just a statistical constraint but a spatial constraint penalizing images deviating from fish-like structures in our case.

2.6. Out-of-distribution detection

The OODD workflow is visually depicted in Fig. 2. Once we trained a cwfa on a set of fish image-volume pairs, we can evaluate if the network can adequately handle a novel sample (I_{novel}). First, by deconvolving I_{novel} into V_{novel} (with 100 iterations of rl deconvolution), using this same volume and the raw image as the condition C_{novel}, and processing these through the network in the forward direction. Finally, we can evaluate the Likelihood of the generated z_{0-n} as in Fig. 2(2).

If the nll obtained are above a pre-defined threshold, we have an ood sample in hand. Once detected, there are different solutions, for example we can fine-tune the cwfa on the test sample or add the test sample to the training set. We explore these two options in sec. 3.3.

The OODD threshold is picked by selecting a small number of samples from all cross-validation training and testing sets and evaluating their likelihood. Then, we define 1000 thresholds uniformly distributed spanning the full range of the data, and pick the one achieving the most significant AUC.

2.7. Dataset acquisition and pre-processing

Two Pan-neuronal nuclear localized GCaMP6s Tg(HuC:H2B:GCaMP6s) and two pan-neuronal soma localized GCaMP7f Tg(HuC:somaGCaMP7f) [31] zebrafish larvae were imaged at 4–6 days post fertilization. Additionally, two NLS GCaMP6s fish of unknown age. The transgenic larvae were kept at 28°C and paralyzed in standard fish water containing 0.25 mg/ml of pancuronium bromide (Sigma-Aldrich) for 2 min before imaging to reduce motion. The paralyzed larvae were then embedded in agar with 0.5% agarose (SeaKem GTG) and 1% low-melting point agarose (Sigma-Aldrich) in Petri dishes.

Additionally, a beads dataset was created by imaging 1-μm-diameter green fluorescent beads (ThermoFisher) randomly distributed in 1% low-melting-point agarose (Sigma-Aldrich). The stock beads were serially diluted using melted agarose to 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} of the original concentration.

Each fish was imaged for 1000 frames at 10Hz. Neural activity images were extracted using the SLNet [32], as seen in Fig. 1(a). Later, the resulting images were 3D reconstructed with the RL algorithm for 100 iterations, which takes roughly 1.5 minutes per frame.

2.8. Data splitting for evaluation

A cross-validation approach was used to evaluate the system, using the six different fish image sequences (described in sec. 2.7 and in Table S1). Additionally, we have the raw data (without pre-processing) and fluorescent bead images.

The rationale behind this data splitting is the following. As the distribution of 3D volume fish pixel intensities is likely unknown, assuming that fish with the same labeling come from the same distribution might pose a risk of unwanted bias. This, as the distribution might depend on many different variables that could describe a fish, such as: the molecule used for fluorescence, expression pattern (pan-neuronal, brain-wide, targeting specific neurons), background fish line (wild type or with modifications to increase/decrease melanophores and/or iridophores), age, fixing methodology, etc.

For testing OODD we used the testing fish on each fold, non-sparse images (deconvolutions of raw data, without pre-processing of the SLNet), and fluorescent beads images.

Ten XLFM pre-processed images and volumes per dataset were used to train the cwfa, as each cross-validation set has five training datasets; in total, 50 images were used for training, 250 for testing belonging to the training fish, and 50 for testing. We tested different amounts of data used for training (5, 10, 20, and 50 pairs), where using ten images dealt the best performance and a training time of 1:43h; see Supplement 1 for details.

3. Experiments

3.1. 3D reconstruction of sparse images with the CWFA

We compared the proposed method against the XLFMNet [9] aiming to match the number of parameters (103M) and a modified version of the Wavelet Flow [20] (WF). The XLFMNet is a U-net-based architecture [33] that achieves high reconstruction speeds but lacks certainty metrics (most conventional deep learning techniques).

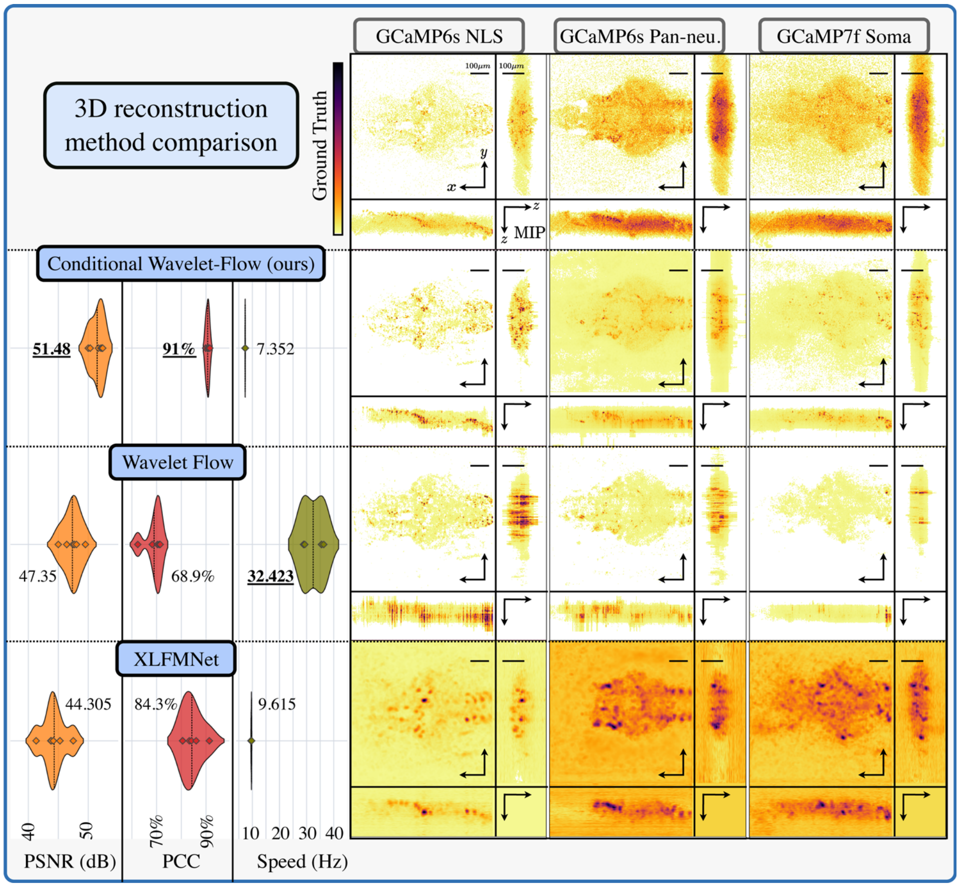

Fig. 3. 3D reconstruction comparison of zebrafish images with different methods. On the top row, gt volumes were generated through 100 iterations of the RL algorithm using a measured PSF. A fish line is presented in each column, coming from 3 of the six cross-validation folds in-distribution testing data (see sec. 3.2.1) for details on the data splitting methodology). Each row shows reconstructions performed with different methods, and the left-most column shows the performance metrics used for comparison: psnr as a structural metric, followed by the mean pcc as a temporal consistency metric for neural activity, followed by reconstruction speed for a single 3D volume. Details of these metrics can be found on sec. 3.1.1.

In the original WF, the lowest resolution prediction was reconstructed from an unconditioned NF based purely on the training data distribution, which does not comply with inverse problems modus-operandi, where a measurement is required to reconstruct the variable in question. The modified version follows the original design in which only the low-resolution image is used as a condition in each flow; however, for a fairer comparison, we substituted the lowest-resolution NF by a cnn Ω_n with access to the raw data, informing the system about the measurement. We used similar settings optimized for Pearson-correlation-coefficient as the cwfa (Fig. S2).

The wf reconstructions seen in Fig. 3 suffer heavily from axial artifacts, which is an expected result. The network starts with a low axial resolution volume and doubles after each up-sampling step until reaching the final axial resolution. However, as only the low-resolution step has access to a data prior, the error accumulates on the consecutive steps propagating. This is a strong motivation for adding data priors on each step of the cwfa.

3.1.1. Comparison metrics

We identified three relevant aspects for the quality evaluation of a 3D reconstruction:

- General image quality: We used peak signal-to-noise ratio (psnr) as it compares the pixel-wise quality of reconstruction against the gt.
- Sparse image quality: As the volumetric data used is highly sparse (mostly zeros), we created a mask of the non-zero values on both the gt and reconstructions, and, used the mean absolute percentage error (mape) for single frame quality assessment.
- Temporal consistency: An important aspect is that the neural activity is correctly reconstructed across frames. Hence, we used single neurons' Pearson-correlation-coefficient (pcc) across time. The position of the top 50 most active neurons per fish was determined by processing the gt volumes with the suite2p framework [34]. The result of this analysis can be found in Fig. S5 from the Supplement 1, where six neurons are shown in detail.

As seen in Fig. 3, the proposed method outperforms the other two approaches regarding these metrics. However, XLFMNet provides faster inference capabilities but lacks any certainty metrics.

3.1.2. Sampling temperature

In our experiments, we found that a temperature of zero produced the highest quality reconstruction. This might be the optimal parameter as the gt used is the result of the RL algorithm, which converges towards the maximum-likelihood estimate. A zero temperature means the network reconstructs the most likely sample, which makes sense for an inverse problem. Furthermore, zero temperature means no sampling is required, increasing the network performance. A comparison of different sample temperatures can be found in the Supplement 1.

3.2. Domain shift or out-of-distribution-detection (OODD)

We evaluated our method by training the cwfa on six cross-validation folds of zebrafish fluorescent activity datasets pre-processed with the SLNet extracting the neural activity. Then, we presented the pipeline with different sample types, such as previously unseen fish sparse images (processed with the SLNet), raw XLFM fish images (not pre-processed), and fluorescent bead images with different concentrations, as shown in Fig. 4.

Our algorithm achieves across all cross-validation folds a mean AUC of 0.9964 and F1-score of 0.9916 on CWF_1, with a nll threshold of -1.33. The achieved AUC and F1-scores for all the down-sampling cwfs are presented in Table S3.

The ood nll threshold was established by computing the ROC curve on all cross-validation sets simultaneously and choosing the threshold with the highest F1-score and AUC, in our case -1.33.

3.2.1. Alternative data splitting methodologies

Even though we chose a cross-validation approach for generality (as described in sec. 3.2.1), different data-splitting approaches can be considered. For example, splitting the training/testing data by fish line is an experiment that could provide information on how the network clusters the data. However, there is no guarantee that the network will cluster the data human-intuitively.

In Fig. S6, an OODD experiment was performed, where we trained a network on one fish line, and tested on the other two. In these results, a couple of points can be observed: The proposed CWFA shows to be well-calibrated as it has a predictable relationship between image quality (PSNR) and log-likelihood, which helps decide reliably whether a novel sample can be processed reliably by the network without the need to deconvolve the whole dataset.

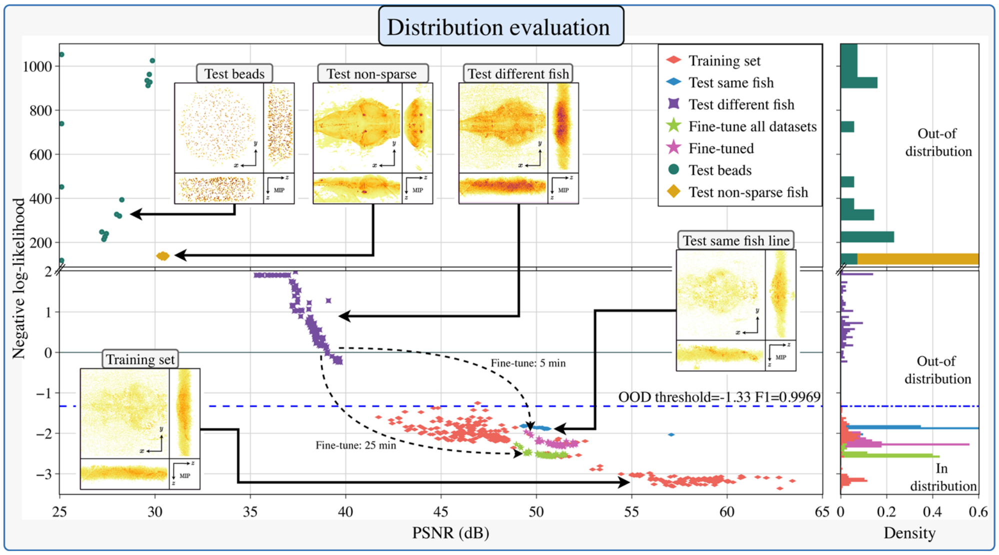

Fig. 4. Distribution analysis of different sample types. The negative log-likelihood vs. PSNR of the first cwf block is shown on the left panel. Different sample types are evaluated, and a ood threshold is shown, used to determine if the network can handle a sample reliably. For the case of the 'Test different fish', we present two re-training approaches, as mentioned in sec. 3.3. On the right panel, the nll density is shown, where a threshold can separate *in* vs. *out* of distribution data.

Interestingly, this experiment seems to show that the CWFA approach can differentiate the distribution of different sample types; even though the GCaMP7f Tg samples are previously unseen by the network, they still achieve a relatively high PSNR, which is well correlated with a low negative-log-likelihood. This suggests that the network might cluster the distributions by expression pattern. Further experiments and analysis would be needed to establish this, which is animal model dependent and not in the scope of this work.

3.3. Domain shift adaptation through fine-tuning

It is well known that fine-tuning a network with novel data will increase the accuracy when used on the new data type. However, evaluating how much retraining, the amount of data required, and quantification of the possible improvement are important aspects of the workflow proposed in this work.

Once a novel sample is detected as ood, the new dataset used for fine-tuning is computed by:

1. Select n images from the new dataset (the testing set of our cross-validation folds).
2. Process the images with the SLNet, to extract the neural activity, taking some seconds to process.
3. Deconvolve these images and generate image-volume pairs for training, taking around 15 minutes (10 samples x $1.5\frac{min}{sample} = 15min$).

Once the additional training samples are ready, we compared two approaches for fine-tuning the networks:

- **Fine-tune only on the new data:** We fine-tuned the network for 100 epochs (20 for each CWF block), taking roughly. Resulting in a mean PSNR increase from $39dB$ to

53dB, 40% on MAPE, and 2% on pcc, as seen in Table S2, and on Fig. 4, where we fine-tuned the 'Test different fish' (purple marker) into 'Fine-tuned' (pink stars marker). For comparison, fine-tuning the XLFMNet with the same number of images and the same amount of computation time dealt a mean PSNR increase from 43.47dB to 43.74dB, 6% on MAPE, and 1.7% on pcc. A visual comparison of both methods is shown in Fig. S7.

- **Append the new data to the original training data and fine-tune:** If we still want to use the network for the fish it already trained on, we can append the new training data to the training set (comprised of 5 different fish for each cross-validation fold) and fine-tune all the data. This approach takes roughly 25 minutes for fine-tuning. Resulting in a mean PSNR increase from 39dB to 52dB, 34% on MAPE, and 1% on pcc, as seen in Fig. 4 as 'Fine-tuned all datasets' (green stars marker).

4. Discussion

In this work, we presented a Bayesian approach to 3D reconstruction of live immobilized fluorescent zebrafish, comprised of a robust workflow for inverse problems, particularly when accurate reconstructions are a priority, as in the case of bio-medical data. The aim is to achieve spatial resolution as in the 3D reconstruction with 100 iterations of rl 3D deconvolution but with a substantial speed increase. We showcased our work on reconstructing volumes spanning a FOV of $734 \times 734 \times 225 \mu m^3$ ($512 \times 512 \times 96$ voxels), together with the XLFMNet [32] cnn and the WF [20], modified for a fair comparison by enforcing the same number of parameters and data priors. The proposed cwfa operated 735× faster than the traditional deconvolution (1 second per rl iteration, we used 100 iterations in this work), with similar quality, as indicated by a mean PSNR of 51.48, a mape [35] of 0.1011 or 10.11%.

Furthermore, when analyzing the neural activity of individual neurons in a time sequence, as seen in Fig. S5, the cwfa achieves a pcc [36] of 0.9077, where the XLFMNet and WF achieve 0.6897 and 0.8428, respectively. Highlighting the CWFA's potential to capture fine-grained temporal patterns.

Alternative methods such as the XLFMNet and wf achieved a speed increase of 3200× and 961×, PSNR of 47.35 and 44.30 and mape of 0.3594 and 0.1167 respectively. But lacking distribution information in the case of XLFMNet and an architecture inadequate for inverse problems in the case of the wf, highlighting the advantages of the proposed method against conventional cnn, traditional normalizing flows, and iterative deconvolution.

In domain shift detection, the first cwf performed best and achieved a classification F1-score [37] of 0.988 and an area under the curve (AUC) of 0.997, as seen in Fig. 4. Once an ood sample type is detected, we found that 5 minutes of fine-tuning of the proposed cwfa on a novel sample are enough to increase quality substantially, achieving a mean increase of PSNR from 39dB to 53dB, 40.71% in mape, and 2.15% pcc.

To conclude, the cwfa is 735× faster than the reconstruction gold standard (RL deconvolution [38]) and offers excellent domain shift detection that can trigger either re-training or fine-tuning on new data. Remarkably, the amount of data (50 images per cross-validation set) and training time (around 2 hours) combined with the OODD capability would allow this system to be integrated into a downstream analysis workflow. When a new fish or fluorescent sample needs analysis, 5 minutes of retraining will suffice to allow the network to reconstruct neural activity reliably.

We leave some remaining questions for future work, such as determining which element in the setup enables the oodd. The classification capability increases in higher resolution cwf steps (Table. S3), which might indicate that the hierarchical approach based on the Haar transform aids the likelihood clustering. Another relevant question is how often a user should prove the data for ood. In our experiments, deconvolving a single sample and proving the OODD from sec.

2.6 was enough to detect if the architecture could handle the new fish images. We find that nfs and Bayesian approaches are adequate for bio-medical imaging due to their potential capability of handling uncertainty, allowing for better decision-making and false positive minimization.

Funding. University of Bern (UniBE ID Grant 2018); National Institutes of Health (1R01DA045549, 1R01MH123977, R01DA029639, R01MH122971, RF1NS113287); Bundesministerium für Gesundheit (2520DAT920); National Science Foundation (1848029); Deutsche Forschungsgemeinschaft (LA 3264/2-1).

Disclosures. The authors declare no conflicts of interest.

Data Availability. The sourcecode and dataset of this project can be found under [39] and [40] respectively

Supplemental document. See Supplement 1 for supporting content.

References

1. Q. Wen Lin Cong, Z. Wang, Y. Chaik, W. Hang, C. Shang, W. Yang, L. Bai, J. Du, and K. Wang, "Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (danio rerio)," eLife **6**, e28158 (2017).
2. X. Hua, W. Liu, and S. Jia, "High-resolution fourier light-field microscopy for volumetric multi-color live-cell imaging," Optica **8**(5), 614–620 (2021).
3. W. Liu, C. Guo, X. Hua, and S. Jia, "Fourier light-field microscopy: An integral model and experimental verification," *Biophotonics Congress: Optics in the Life Sciences Congress 2019 (BODA,BRAIN,NTM,OMA,OMP)* p. DT1B.4 (2019).
4. K. Han, X. Hua, V. Vasani, G.-A. R. Kim, W. Liu, S. Takayama, and S. Jia, "3d super-resolution live-cell imaging with radial symmetry and fourier light-field microscopy," Biomed. Opt. Express **13**(11), 5574 (2022).
5. A. Stefanoiu, G. Scrofani, G. Saavedra, M. Martinez-Corral, and T. Lasser, "What about super-resolution in fourier lightfield microscopy?" Opt. Express **28**(11), 16554 (2020).
6. E. M. C. Hillman, V. Voleti, W. Li, and H. Yu, "Light-Sheet microscopy in neuroscience," Annu. Rev. Neurosci. **42**(1), 295–313 (2019).
7. L. B. Lucy, "An iterative technique for the rectification of observed distributions," The astronomical J. **79**, 745 (1974).
8. M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," ACM Trans. Graph. **25**(3), 924–934 (2006).
9. J. Page Vizcaino, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. S. Boyden, and T. Lasser, "Real-time light field 3d microscopy via sparsity-driven learned deconvolution," in *2021 IEEE International Conference on Computational Photography (ICCP)*, (2021), pp. 1–11.
10. Z. Wang, L. Zhu, H. Zhang, G. Li, C. Yi, Y. Li, Y. Yang, Y. Ding, M. Zhen, S. Gao, T. K. Hsiai, and P. Fei, "Real-time volumetric reconstruction of biological dynamics with light-field microscopy and deep learning," Nat. Methods **18**(5), 551–556 (2021).
11. 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 Trans. on Computat. Imaging **7**, 775–788 (2021).
12. 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," Nat. Methods **18**(5), 557–563 (2021).
13. L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," arXiv, arXiv:1410.8516 (2014).
14. D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," (2015).
15. A. Denker, M. Schmidt, J. Leuschner, and P. Maass, "Conditional invertible neural networks for medical imaging," J. Imaging **7**(11), 243 (2021).
16. L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," arXiv, arXiv:1808.04730 (2018).
17. L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," *7th International Conference on Learning Representations, ICLR 2019* pp. 1–20 (2019).
18. G. Anantha Padmanabha and N. Zabaras, "Solving inverse problems using conditional invertible neural networks," J. Comput. Phys. **433**, 110194 (2021).
19. S. Kousha, A. Maleky, M. S. Brown, and M. A. Brubaker, "Modeling srgb camera noise with normalizing flows," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), pp. 17442–17450.
20. J. J. Yu, K. G. Derpanis, and M. A. Brubaker, "Wavelet flow: Fast training of high resolution normalizing flows," in *Advances in Neural Information Processing Systems*, vol. 33 H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds. (Curran Associates, Inc., 2020), pp. 6184–6196.
21. L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, "Guided image generation with conditional invertible neural networks," arXiv, arXiv:1907.02392 (2019).
22. L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," arXiv, arXiv:1605.08803 (2016).
23. A. Haar, "Zur theorie der orthogonalen funktionensysteme," Math. Ann. **69**(3), 331–371 (1910).
24. P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why normalizing flows fail to detect out-of-distribution data," Adv. neural Inform. Process. Syst. **33**, 20578–20589 (2020).

25. T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019).
26. D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” Adv. neural information processing systems **31** (2018).
27. X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, “Symbolic discovery of optimization algorithms,” (2023).
28. J. Kruse, G. Detommaso, U. Köthe, and R. Scheichl, “Hint: Hierarchical invertible neural transport for density estimation and bayesian inference,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35 (2021), pp. 8191–8199.
29. L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier, and P. Sorrenson, “Framework for Easily Invertible Architectures (FrEIA),” (2018-2022).
30. E. Taskesen, “findpeaks is for the detection of peaks and valleys in a 1D vector and 2D array (image),” (2020).
31. O. A. Shemesh, C. Linghu, K. D. Piatkevich, D. Goodwin, O. T. Celiker, H. J. Gritton, M. F. Romano, R. Gao, C.-C. J. Yu, and H.-A. Tseng, “Precision calcium imaging of dense neural populations via a cell-body-targeted calcium indicator,” Neuron **107**(3), 470–486.e11 (2020).
32. J. Page Vizcaino, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. S. Boyden, and T. Lasser, “Real-time light field 3d microscopy via sparsity-driven learned deconvolution,” in *2021 IEEE International Conference on Computational Photography (ICCP)*, (2021), pp. 1–11.
33. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” CoRR abs/1505.04597 (2015).
34. M. Pachitariu, C. Stringer, M. Dipoppa, S. Schröder, L. F. Rossi, H. Dalgleish, M. Carandini, and K. D. Harris, “Suite2p: beyond 10,000 neurons with standard two-photon microscopy,” bioRxiv (2017).
35. A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, “Mean absolute percentage error for regression models,” Neurocomputing **192**, 38–48 (2016).
36. J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise reduction in speech processing*, (Springer, 2009), pp. 1–4.
37. N. Chinchor, “Image quality metrics: Psnr vs. ssim,” in *Proc. of the Fourth Message Understanding Conference, (MUC-4)*, (1992), pp. 22–29.
38. W. Richardson, “Bayesian-based iterative method of image restoration,” J. Opt. Soc. Am. **62**(1), 55–59 (1972).
39. J. Page Vizcaíno, P. Symvoulidis, J. Jelten, *et al.*, “{F}ast light-field 3D microscopy with out-of-distribution detection and adaptation through Conditional Normalizing Flows,” Github, 2023), https://github.com/pvjosue/CWFA
40. J. Page Vizcaíno, P. Symvoulidis, J. Jelten, *et al.*, “Immobilized fluorescently stained zebrafish through the eXtended Field of view Light Field Microscope 2D-3D dataset,” Github, 2023, https://doi.org/10.5281/zenodo.8024696