Incorporating structural knowledge into unsupervised deep learning for two-photon imaging data

Florian Eichin[1,2*] Maren Hackenberg[1,2*†] Caroline Broichhagen[1,2]
Antje Kilias[3] Jan Schmoranzer[4] Marlene Bartos[3] Harald Binder[1,2]

[1] Institute of Medical Biometry and Statistics, Faculty of Medicine and Medical Center, University of Freiburg, Germany
[2] Freiburg Center for Data Analysis and Modelling, Faculty of Mathematics and Physics, University of Freiburg, Germany
[3] Institute for Physiology I, Systemic and Cellular Neurophysiology, Faculty of Medicine, University of Freiburg, Germany

[4] Core Facility Advanced Medical Bioimaging, Charité - Universitätsmedizin, Humboldt University Berlin, Germany
August 31, 2021

Abstract

Live imaging techniques, such as two-photon imaging, promise novel insights into cellular activity patterns at a high spatio-temporal resolution. While current deep learning approaches typically focus on specific supervised tasks in the analysis of such data, we investigate how structural knowledge can be incorporated into an unsupervised generative deep learning model directly at the level of the video frames. We exemplify the proposed approach with two-photon imaging data from hippocampal CA1 neurons in mice, where we account for spatial structure with convolutional neural network components, disentangle the neural activity of interest from the neuropil background signal with separate foreground and background encoders and model gradual temporal changes by imposing smoothness constraints. Taken together, our results illustrate how such architecture choices facilitate a modeling approach that combines the flexibility of deep learning with the benefits of domain knowledge, providing an interpretable, purely image-based model of activity signals from live imaging data.
Key words: deep generative models, network architecture, live cell imaging, domain knowledge
Teaser sentence: Using a neural network architecture that reflects domain knowledge provides an interpretable model of live cell imaging data.

1 Introduction

In the last decade, deep learning-based approaches have been successfully adapted for the analysis of image data, rivaling and surpassing human performance in identifying structure from data. This encompasses both supervised tasks with ground truth information, such as image segmentation with the U-Net [1], and unsupervised tasks where structure can be uncovered without prior human labeling, e.g., by generative models such as variational autoencoders (VAEs) [2]. Recently, in particular such unsupervised approaches have been extended to incorporate structural knowledge on temporal patterns in the form of explicit models [3, 4].

*The authors wish to be regarded as joint first authors.
†Corresponding author: maren@imbi.uni-freiburg.de

In line with such ideas of a gradual transition from black-box deep learning towards more explicit models we want to illustrate approaches for incorporating structural knowledge into deep generative modeling of live cell imaging data. Besides convolutional neural networks, which take into account spatial structure, we propose an architecture for distinguishing between the image foreground, containing the biological signal of interest, and a background. In addition, knowledge about gradual temporal changes in the biological processes being observed is incorporated in the architecture.

Specifically, we explore how a generative deep learning model based on VAEs can be adapted to such live imaging data. Exemplarily, we consider data from an *in-vivo* two-photon imaging experiment and show how gradually incorporating constraints and structural assumptions that reflect the properties of the data can provide an interpretable general model of neural activity. Such a model is then not specifically adapted to a particular prediction task but can be flexibly used for various downstream analyses.

Two-photon calcium imaging, as an exemplary application to illustrate our approach, represents an invasive technique in the context of 'neural decoding' for recording activities of individual neurons over time. Here, two-photon microscopy is used to capture movies of a neuronal population that expresses a fluorescent calcium indicator and allows to visualize the increase in intracellular Ca2+-concentration that accompanies neuronal spiking activity [5]. More generally, 'neural decoding' refers to techniques that use brain signals to make predictions about behavior, perception, or cognitive state and are becoming increasingly important for neuroscientific research [6–9]. For example, calcium imaging techniques enable optical measurements of large neural populations at a high spatio-temporal resolution. The resulting calcium imaging movies thus facilitate insights into neural activity [5, 10, 11]. To realize their full potential, computational methods are needed to derive a model of neural activity from the indirect optical measurement of the Ca2+ indicator. Here, to analyze the resulting video data, the typical two-step analysis framework of cell population imaging is employed, where cells are identified through segmentation in a first step, and the signals of interest, e.g., temporal fluorescence traces or firing rates, are identified subsequently [9, 12–14]. Current approaches to derive an explicit generative model typically rely on simplifying assumptions of, e.g., a linear combination of additive signals, that allow to directly incorporate known structure, but limit the flexibility and versatility of the obtained data model [15, 16]. On the other hand, deep learning techniques provide a more flexible class of models and have recently been successfully applied to infer spike activity from calcium imaging data (e.g., [17–19]). Yet, as more opaque black-box approaches, they lack the interpretability of an explicit data model and have so far been mainly used on two-photon imaging data for supervised prediction tasks, requiring large amounts of labeled training data that is typically not available.

Additionally, these approaches often only provide one of several components in the entire data analysis workflow [14], where prior steps, such as motion correction, can considerably affect performance [20]. Yet, results showing that temporal information can improve segmentation [17] indicate that it might be beneficial to consider both steps simultaneously and perform modeling directly based on the temporal sequence of images without explicitly extracting traces [9]. Such an integrated modeling strategy can help to build a flexible and versatile model that is not limited to a specific task.

We hypothesize that unsupervised generative deep learning approaches, which provide a model of the underlying data generating distribution, can be useful to build such integrated models. Specifically, we consider VAEs for this task, as they can be trained in an unsupervised way and infer a low-dimensional latent representation of the central factors of variation underlying the data. This facilitates interpretability and has been shown to be useful for capturing and extracting patterns in the data in an explainable artificial intelligence (AI) approach [21]. While VAEs have been adapted to two-photon imaging data for the specific task of inferring neural spike rates from fluorescence traces [19], we aim to exemplify how a deep learning approach based on a VAE architecture can be adapted to provide a flexible and versatile model that is not restricted to a specific task, but tailored to the properties of two-photon imaging data.

This paper is structured as follows. We first give an overview of our proposed modeling approach

of gradually incorporating increasing levels of structural information into a VAE architecture. Next, we present results of the models trained on a two-photon imaging dataset of hippocampal CA1 neurons and discuss our findings. We additionally provide an overview of the typical image processing workflow as well as an introduction to generative deep learning in general and VAEs in particular, and the convolutional architecture integrated in our models. Finally, we give details on the two-photon dataset and the proposed modeling approaches and their implementation.

The source code of the implementation and an accompanying Jupyter notebook illustrating an exemplary application of the approach is available at `https://gitlab.imbi.uni-freiburg.de/maren/incorporating-structural-knowledge-into-unsupervised-deep-learning-for-two-photon-imaging-data`.

2 Results

2.1 Adapting a VAE-based model to the properties of two-photon imaging data by incorporating different amounts of structural knowledge

We employ a VAE as an unsupervised generative deep learning approach to model neuronal activity based on two-photon imaging movies. Specifically, we consider a sequence of individual frames from such a video as input data and propose an approach to tackle the challenging modeling task of simultaneously capturing the foreground signal with the distinct cellular activity and the coarse-grained, blurry background signal, while accounting for the temporal correlation between neighboring frames. Starting with a simple model architecture, we gradually increase the level of additional structural information integrated into the model. Thus, we explore how different encoded structural properties affect the activity patterns and the identified latent structure.

Figure 1 provides an overview of this framework, representing steps from a black-box model towards a more explicit model tailored to the data: In a first step, we train a fully-connected vanilla VAE on the dataset, as described in Section 4.3, where each image frame is treated as a separate observational unit. In a second step, we employ a convolutional architecture for the VAE encoder and decoder to account for the characteristics of the image data (see Section 4.4). Next, we tackle the mixed-source activity by building separate encoders and decoders for the foreground and background, respectively, that share a joint latent space. In a final model extension, we consider latent kernel smoothing to account for the temporal correlation across frames, thus incorporating a temporal smoothness constraint in the latent space.

The recorded two-photon video frame images are characterized by a clear distinction (at least for the human observer) between foreground signal, showing the somatic calcium increase during firing of the neuron over time that we aim to capture with our modeling approach, and background, consisting primarily of more coarse-grained, noisy neuropil signal (for details on the data, see Section 4.7). Hence, we explicitly model the dataset of images as a combination of foreground and background signal that we seek to disentangle in the latent space of our VAE model.

To achieve this, we introduce separate VAE models for the foreground and background with a distinct latent representation of the foreground and background respectively. We realize these models as convolutional neural networks (CNNs) to account for the spatial structure of the image: In a CNN layer, a filter weight matrix is convolved (i.e., multiplied element-wise and summed up) with different patches of the image, like a window sliding over the image with its size given by the size of the filter matrix (for details, see Section 4.4). This accommodates central features of image data, such as distinctive local motifs, e.g., groups of active neurons, that can appear in any part of the image. We employ CNNS with different filter sizes for the foreground and background model to accommodate both sharp and local foreground signal and blurry, large-spanning background signal. To obtain a joint model of the entire image including both foreground and background, we integrate the latent representations of both VAE models into a joint latent space. The resulting *dual-target VAE* model

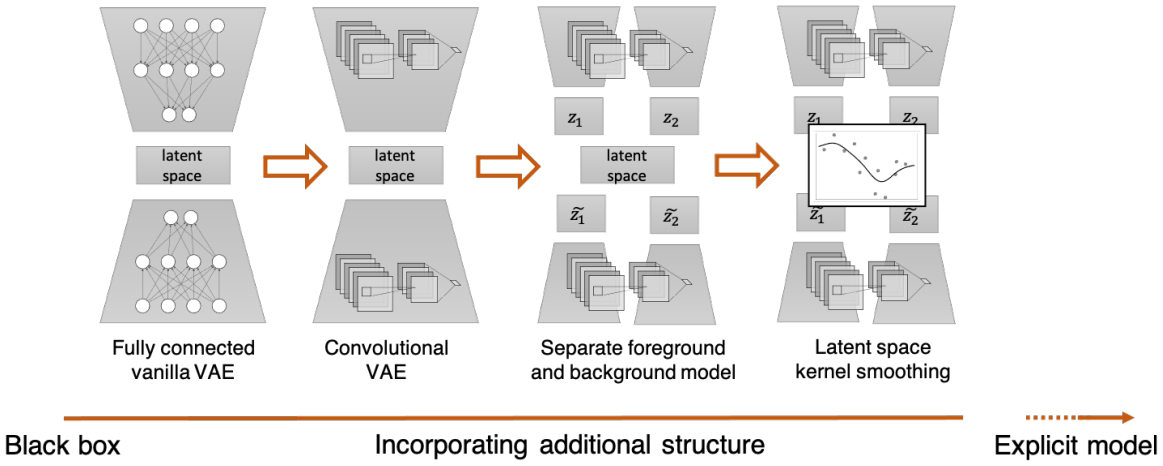

Figure 1: **Overview of deep generative model architectures with increasing amounts of domain knowledge incorporated.** Layers of structural information can be gradually combined with a fully connected vanilla VAE model (1) to enable a step-wise transition from a black-box deep learning model towards a more explicit modeling approach that takes into account problem structure by adding a convolutional architecture (2), separate foreground and background encodings (3), and latent kernel smoothing (4).

is trained by optimizing an adapted version of the evidence lower bound, a lower bound on the data likelihood (for details, see Section 4.5 and the accompanying Jupyter notebook).

Since the images correspond to frames in a video, subsequent images exhibit similar cellular activity patterns and are thus strongly temporally correlated. We explicitly incorporate this spatio-temporal information as a structural property of the data into our model for constraining the latent representation. This will allow to model smoother developments and thus more accurately reflect underlying activity patterns in the data. Specifically, we propose kernel smoothing in the latent space and train the model on neighborhoods of adjacent images, i.e., subsequent frames in the video sequence. For each frame, we consider a neighborhood of a fixed size around the frame and obtain the latent representation of this frame as a weighted sum of the latent representations of all frames in the neighborhood, where the weights are given by a kernel that takes into account distance between frames (see Section 4.6). In this way, we impose a smoothness constraint on the latent space in that each frame's representation is also influenced by the representations of neighboring frames. This facilitates to capture gradual temporal changes in activity patterns. We additionally compare this approach to a similar smoothing step performed at the level of the loss function, i.e., for each frame, we take the weighted average of all loss values of the frames in its neighborhood as the loss.

In the following, we compare (1) a VAE with a CNN architecture, (2) a dual-target VAE, and (3) a dual-target VAE with kernel smoothing.

2.2 Convolutional VAE

First, we evaluate the performance of the baseline CNN approach against a standard VAE with a fully-connected ANN encoder and decoder. With this fully-connected approach, we were not able to achieve dynamic reconstruction of the foreground and thus omit the results. In contrast, Figure 2 exemplarily shows that incorporating a CNN architecture in the VAE encoder and decoder allows for good reconstruction results of prominent foreground cells. This implies that the advantages of CNNs over a standard fully-connected neural network structure also apply to frames from two-photon imaging videos. Modeling of theses images hence benefits from taking into account local motives and patterns as well as their spatial invariance across the entire image, the main characteristics motivating the CNN

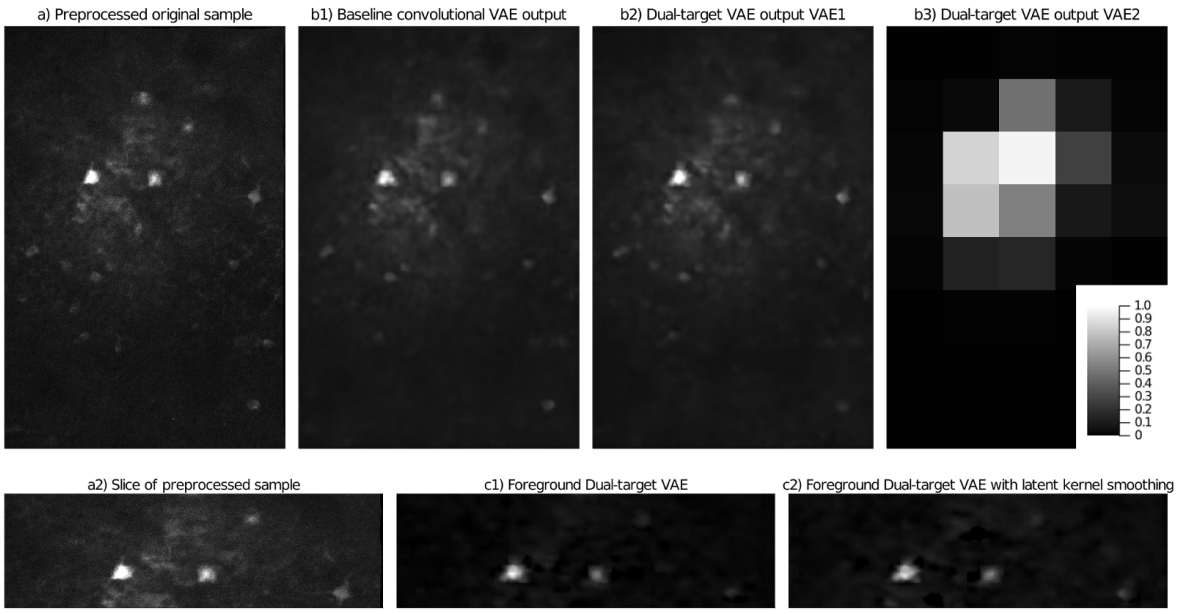

Figure 2: **Comparison of reconstructions from a convolutional VAE, a dual-target VAE and a dual-target VAE with kernel smoothing.** Top row: Preprocessed sample frame from the data (a) and its reconstructions of the convolutional baseline VAE approach (b1) and of dual-target VAE (b2) respectively. Bottom row: Slice of the same sample (a2) with corresponding reconstructions of only the foreground of dual-target VAE (c1) and dual-target VAE with latent kernel smoothing (c2).

architecture.

Yet, the signal of smaller cells with Calcium-transients of shorter duration and smaller amplitudes is poorly reconstructed by the model in most of these smaller cells with lower firing intensity in Figure 2. This can be more generally observed over most other frames in the video, where the activity of the less prominent cells is reconstructed with low accuracy. Also, when observed over the course of contiguous frames, rapid alternations in cell activity, which are not present within the original data, are visible in the output underlining the problems of the model with this kind of signal. The noise in the original sample is not retained in the output, thus creating a slightly more blurred but denoised version of the input, which might be useful, e.g., for visual presentation.

2.3 Dual-target VAE

Next, we investigated the separation of foreground and background activity within the latent space of the convolutional baseline approach. To this end, we decoded one-hot configurations of all dimensions of the latent space (i.e., vectors with a value of 0 in all but one dimension) and observed the amount of foreground and background present in the output. In Figure 3, we display the outputs of the configurations with the most and least foreground activity. Even in dimensions with low foreground activity, at least some foreground cell activity is present and thus, this approach does not allow to fully disentangle fore- and background signal.

On the other hand, the dual-target VAE showed no foreground activity within the two latent dimensions of the background VAE model and thus achieved better separation. For any given image, we can thus set $\mathbf{z}_1 = 0$ and decode the given configuration to an image of only the background activity. Subsequently, this background can be subtracted element-wise from the reconstructed image and we obtain an image of the foreground activity. Figure 2 exemplarily shows the reconstructed foreground of the given frame. The depicted frame is representative of the rest of the data, where static and dynamic parts of the background are eliminated with high accuracy and the dominant foreground cell activity

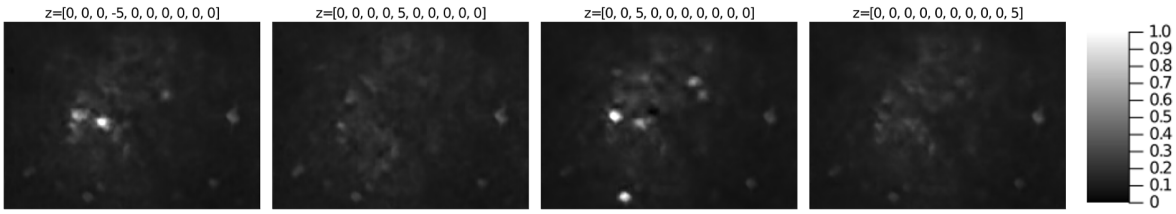

Figure 3: **Exploration of the latent space dimensions with most and least foreground activity for a convolutional VAE and a dual-target VAE.** Exemplary exploration plots (sliced) of foreground and background model of the baseline convolutional VAE model (left) and the dual-target VAE (right). The configuration of the latent space that was used to create the plot is annotated on top. Dimensions of the baseline model are chosen according to visual inspection for most and least foreground activity respectively.

is retained in the output. Yet there is still a tendency of the model to weaken or omit the activity of smaller, less active cells in the foreground.

2.4 Dual-target VAE with latent kernel smoothing

Finally, we evaluate the impact of kernel smoothing on the model performance by comparing the dual-target VAE with kernel smoothing on the latent representation and to the dual-target VAE without temporal smoothing (Figure 2): Comparing the foreground activity (bottom row) show that especially low-activity cells are more prominent in the reconstruction of the approach with latent kernel smoothing. Effectively, calculating the kernel weighted average of latent representations across a neighborhood of subsequent frames corresponds to imposing a smoothness constraint on the latent space. To further investigate whether this smoothness constraint can help to capture the temporal correlations between subsequent frames, we considered sequences of frames with a distance of two, i.e., used every second frame in the original video sequence for building the neighborhood. This means that the first, second and third frame in the neighborhood used for training the model correspond to the first, third and fifth frame in the respective part of the original video (see Figure 4). Specifically, we compared the differences between two adjacent frames in the sequence (i.e., between every second frame in the original video) when reconstructed with the dual-target VAE without kernel smoothing (Figure 4, b)) versus the dual-target VAE with latent kernel smoothing (Figure 4, c)). Here, we can observe that the differences between frames reconstructed based on a kernel smoothed latent representation are less pronounced than without the smoothing step.

We more generally observed a tendency towards higher intensity and precision of the reconstructed foreground activity with fewer flickering artifacts, indicating that our approach can indeed provide a smoother reconstruction of frames across time that reflects the underlying time-dependent biological process.

Additionally, we compare the kernel smoothing in the latent space with a dual-target VAE trained on neighborhoods with a smoothing step performed at the level of the loss values, as in 4, Section 4.6, i.e., using for each frame its individual latent representation, but employing a kernel weighted average across the loss values of all frames in a neighborhood as training objective. Here, the improvements described above are observed to a lesser extent (results not shown), implying that constrains should be imposed directly on the latent representation, which encodes the central structure underlying the data and the reconstruction.

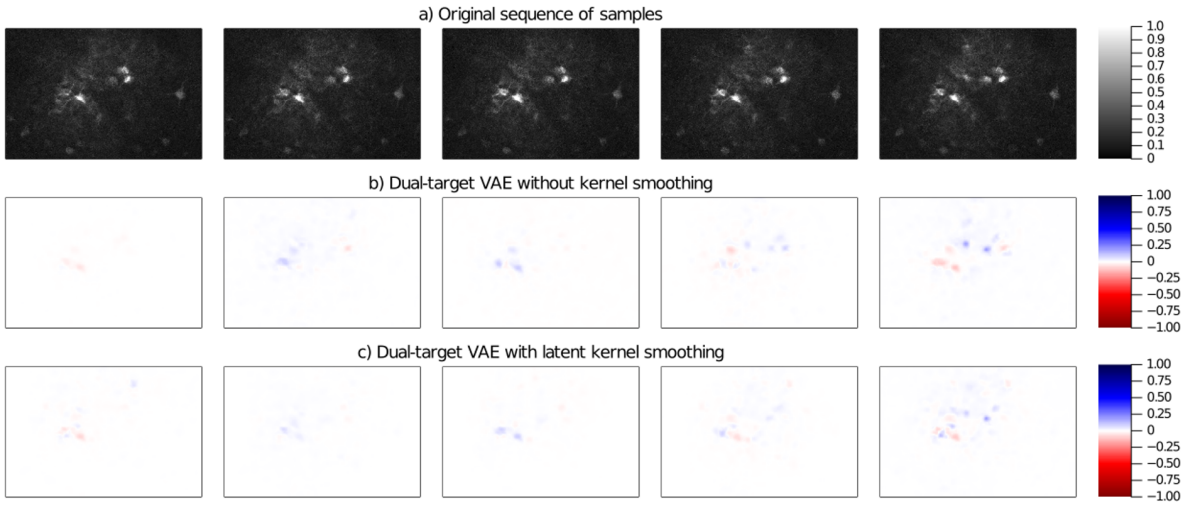

Figure 4: **Comparison of differences between model reconstructions of adjacent frames with and without latent kernel smoothing.** Exemplary sequence of frames and the differences between the model reconstructions of adjacent frames with and without latent kernel smoothing. a) Original exemplary sequence of samples with a distance of two, i.e., adjacent frames in the plot correspond to every second frame in the original video. b) Differences between the reconstructions of the frames shown in a) obtained from training a dual-target VAE without kernel smoothing. c) Differences between the reconstructions of the frames shown in a) obtained from training a dual-target VAE with kernel smoothing. The color coding describes the sign of the difference (blue: positive, red: negative), while the color intensity encodes the absolute magnitude.

3 Discussion and conclusions

Deep learning has been shown to be a useful tool for analyzing biomedical imaging data across a wide range of tasks. Specifically, such approaches have been developed for neural decoding [9] and are increasingly becoming a part of analysis workflows for live cell imaging data, such as for two-photon imaging [14, 17, 18]. Here, automatic extraction and modeling of neural activity from individual cells over time and in large cell populations is a crucial step towards a better understanding of cellular activity. This can ultimately be linked to phenotype data to facilitate biological insight. While current approaches are dominated by supervised models that require large amounts of labeled training data to learn a specific classification of prediction task, generative models infer a model of the underlying data distribution. Explicit data models that describe cellular activity as, e.g., linear combination of additive signals, are less opaque than deep learning approaches, but limited in their flexibility due to simplifying modeling assumptions. Ideally, a modeling approach should provide a combination of both, i.e., an interpretable model that allows to encode known structural properties of the data while being as flexible and versatile as possible. We have thus investigated how structural knowledge can be incorporated into an unsupervised generative deep learning approach. Specifically, we have employed VAEs, directly applied to the video frames from two-photon imaging data, for modeling neural activity in a latent representation. Here, one focus was on exploring how gradually incorporating structural components, which reflect key properties of two-photon imaging data, affect the latent representation, and thus facilitate insight into what the model has learned. Further, we exemplified to what extent a generative deep learning approach can be tailored to model signals activity in live cell imaging data.

Specifically, we have compared VAEs with different amounts of explicit structure incorporated into the model. While a standard fully-connected VAE did not permit to accurately reconstruct the cell images, employing convolutional layers, the *de facto* standard architecture for deep learning on image data, allowed to infer a latent representation based on which the input images could be recon-

structed. Next, we have taken into account the typical mixed-source activity of two-photon imaging videos, where the fluorescent traces of active neurons in the background should ideally be separated and deconvolved from the noisy neuropil background signal. To model this structure, we have proposed separate VAE encoders and decoders for the foreground and background of each image, while still learning a joint latent representation of the image. With this, we have been able to obtain a latent representation that can disentangle foreground and background signals, which was only possible by explicitly encouraging it with the distinct encodings. Finally, we have illustrated a straightforward approach for imposing a smoothness constraint on the latent space, to take into account the temporal correlation and smooth structure of activity traces in subsequent frames. Specifically, we have formed neighborhoods of adjacent images around each frame and obtained the latent representation of each frame as a smoothed average of the activity signal over the entire neighborhood. This approach allowed to retain weaker signals from cells in the foreground more frequently and more clearly, thus capturing the overall activity more accurately by exploiting the similarity of subsequent frames for modeling.

Still, it is difficult to assess the model performance objectively beyond visual inspection of reconstructed videos and images, and more rigorous quantitative criteria are needed to evaluate further model refinement. Another drawback of the model in its current form is the complexity, due to many parameters in the convolutional layers, resulting in a computationally rather extensive training procedure. Additionally, more sophisticated approaches for smoothness constraints could be considered to facilitate dynamic modeling of the latent representation over time. For example, differential equations could be incorporated in the latent space. A further extension would be to consider stochasticity, and to more explicitly model the spatial structure. Here, recent works suggest underlying low-dimensional spatio-temporal dynamics in two-photon-imaging data [32], which could potentially be captured in the latent space of a VAE model, while the spatio-temporal patterns could be modeled with an approach as in [33]. Another interesting direction for future research are methods for linking the inferred low-dimensional activity patterns, e.g., to phenotype data regarding the behavior of the animal. This could also benefit from incorporating structural assumptions on the spatio-temporal dynamics, as in [34].

In the present work, our focus was on illustrating how such a deep learning-based approach can be adapted to the specific properties of the data using two-photon imaging as an exemplary application, rather than a detailed comparative study of existing deep learning approaches for cell imaging data. We specifically highlighted how modeling can benefit from incorporating elements of more explicit data models, and investigated how such structural assumptions affect the patterns learned by the model.

More generally, our approach illustrates how generative deep learning approaches can be combined with various amounts of structural assumptions for greater interpretability, and how the learned representations can be inspected to assess the influence of these assumptions. It can be seen as an example of the currently ongoing shift from purely algorithmic black-box deep learning models towards the more explicit data modeling of classical statistics. Rather than viewing the modeling cultures as a dichotomy [35], our approach thus represents a step towards uniting the two worlds and combining their respective advantages [36].

Overall, our proposed strategy offers an integrated one-step modeling approach to extract activity signals from two-photon imaging data, operating directly on the video frames. Further, the approach exemplifies how tailoring a model to the specific properties of the data can provide an interpretable representation of the central patterns in the data, accounting for the distinct foreground and background activity as well as temporal correlations. It thus suggests that combining generative deep learning with structural information can also more generally be a promising approach for uncovering activity patterns in cell imaging data.

4 Materials and Methods

4.1 Typical steps during conventional bioimage analysis

The ultimate goal of bioimage analysis is to gain knowledge of biological processes by extracting relevant information from microscopy images, including time-lapse sequences that record the dynamics of the processes. In non-machine learning-based ('conventional') bioimage analysis, a sequence of operations is employed to extract the information [22, 23]. Depending on the specific application, this analysis workflow typically involves initial image processing (e.g., enhancement, noise reduction, filtering), followed by a sequence of operations that may include image segmentation and object detection (e.g., cell outline, nuclei), tracking of object movements, and intensity-based quantification and classification of objects (e.g., brightness, distance, size, co-localization). Finally, downstream data visualization and analytics as well as mathematical or statistical modeling are applied to allow a meaningful interpretation of the biological results, especially by comparing different experimental conditions. Depending on the complexity and size of the raw data set, it is necessary to design automated custom workflows that subsequently apply these operations to the data.

In conventional computational bioimage analysis, the main advantage of non-deep-learning approaches is that the underlying algorithms are transparent with observable input-output data, so that the results are readily interpretable. However, due to recent technical advances in microscopy methods and increased experimental complexities (e.g., in-vvol time-lapse imaging of the brain activity of behaving animals), the produced bioimage data has enormously increased in size and complexity. Conventional analysis tools are limited in processing these highly complex and large data sets, because meaningful results can only be obtained after tedious optimization of the algorithms. In addition, subtle differences in biological structures and dynamics are easily overseen or masked by suboptimal parameter settings that require individual manual adjustment. For example, existing toolbox solutions [14] account for the high complexity of the data with a large set of predefined parameters that require a deep knowledge of the underlying software by the user.

Driven by these challenges, a major paradigm shift has occurred with the massive adoption of deep learning technologies that are rapidly replacing conventional bioimage analysis approaches. Especially the use of artificial neural networks, and more recently of VAEs, in bioimage analysis offers considerable advantages for a flexible, yet efficient analysis of large, complex data sets.

4.2 Generative deep learning

As our proposed analysis strategy is based on a VAE, a generative deep learning approach first presented in [2], we first give a brief introduction to neural networks and generative deep learning in general, before presenting the VAE in more detail.

An *artificial neural network* (ANN) is a function composition $f : \mathbb{R}^{k_1} \to \mathbb{R}^{k_{n+1}}, \mathbf{x} \mapsto (g_n \circ g_{n-1} \circ \cdots \circ g_2 \circ g_1)(\mathbf{x})$ for distinct continuous functions $g_i : \mathbb{R}^{k_i} \to \mathbb{R}^{k_{i+1}}$ called the *layers* of the network. Each layer is of the form $g_i(\mathbf{x}) = h_i(\mathbf{W}_i\mathbf{x}+\mathbf{b}_i)$, where $h_i : \mathbb{R} \to \mathbb{R}$ is a continuous non-linear function called the *activation function* and is applied element-wise, and $\mathbf{W}_i \in \mathbb{R}^{k_{i+1}\times k_i}$ and $\mathbf{b}_i \in \mathbb{R}^{k_{i+1}}$ are *weights* and *biases*, also called *parameters*, of the network. Intuitively, by combining many of these layers of linear combinations followed by a non-linear activation into a *deep* network, ANNs can be used to model potentially very complex, non-linear structures in the data.

The process of finding a parameter set, i.e., determining weights and biases such that an ANN approximates a specific input-output mapping is called *learning* or *training* of the ANN. Thus, the term *deep learning* refers to the process of approximating potentially complex mappings with deep ANNs. More precisely, training is performed by defining a loss function as the training objective and repeatedly applying the chain rule to obtain partial derivatives of the loss with respect to the network parameters in order to minimize the loss function by stochastic gradient descent [24]. This training strategy results in the propagation of gradients 'backwards' through the network and is hence termed

backpropagation [25]. ANNs can be trained to approximate various types of mappings from input data to outputs, e.g., mapping a dataset of images to binary classifications or segmentation masks, and have enjoyed great successes in many of these supervised tasks. They can also be employed as *generative* models that learn a joint distribution over all input variables in an unsupervised fashion to approximate the true underlying data distribution. Such a generative deep learning model then allows to draw synthetic samples from the learned distribution. Various approaches for such deep generative models have been proposed, including deep Boltzmann machines (DBMs) [26], generative adversarial networks (GANs) [27], and VAEs [2], which differ predominantly in the way the underlying data distribution is represented by the model.

4.3 Variational autoencoder

VAEs learn explicit parametrizations of the underlying probability distributions by employing two distinctly parametrized, but jointly optimized neural networks that are responsible for encoding and decoding of the data into and from a latent space that is governed by a probability distribution: The encoder maps the input data $\mathbf{x}$ to a lower-dimensional latent representation given by a random variable $\mathbf{z}$ by approximating the conditional distribution of $\mathbf{z}$ given $\mathbf{x}$, while the decoder performs the reverse transformation from the latent space back to data space, parametrizing the conditional distribution of $\mathbf{x}$ given $\mathbf{z}$.

The VAE training objective is to recover the central factors of variation underlying the data in the lower-dimensional latent space, thus obtaining a compressed representation based on which the input data distribution can be approximated. Since an ANN is used to encode the data into the latent space, the true posterior distribution $p(\mathbf{x}|\mathbf{z})$ becomes intractable. Hence, a variational approximation $q(\mathbf{z}|\mathbf{x})$ is employed – typically a Gaussian distribution with diagonal covariance matrix. A loss function for the model can then be derived based on variational inference [28]:

Minimizing the *Kullback-Leibler divergence* $D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\right]$ to the exact posterior is equivalent to maximizing a lower bound on the true data likelihood, the *evidence lower bound* (ELBO) given by $\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$. Denoting the parameters of the encoder and decoder neural networks with $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, respectively, we can define the VAE training objective as the negative ELBO:

$$\begin{aligned} &\mathcal{L}_{\mathrm{VAE}}(\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\theta}) = -\mathrm{ELBO}(\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\theta}) \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z})). \end{aligned} \tag{1}$$

Intuitively, the first term in (1) can be thought of as a reconstruction error that encourages densities placing mass on configurations of latent variables that explain the observed data, while the second term has a regularizing effect by enforcing consistency between the prior and posterior of $\mathbf{z}$. By maximizing the ELBO with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we can derive both approximate maximum likelihood estimates for $\boldsymbol{\theta}$ and an optimal variational density q_ϕ [28].

In practice, with a Gaussian prior and posterior of $\mathbf{z}$, the Kullback-Leibler divergence in (1) can be calculated analytically, while the expectation with respect to the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ has to be approximated by Monte Carlo sampling. Using S samples for this approximation and a K-dimensional latent space the ELBO of a single data point $\mathbf{x}^{(i)} \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ can be calculated as

$$\begin{aligned} \mathrm{ELBO}(\mathbf{x}^{(i)}, \boldsymbol{\phi}, \boldsymbol{\theta}) = &\frac{1}{L}\sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,s)}) \\ &+ \frac{1}{2}\sum_{k=1}^{K}(1 + 2 \cdot \log(\boldsymbol{\sigma}_k^{(i)}) - (\boldsymbol{\mu}_k^{(i)})^2 - (\boldsymbol{\sigma}_k^{(i)})^2), \end{aligned} \tag{2}$$

where $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\boldsymbol{\sigma} \in \mathbb{R}^K$ denote the parameters of the variational posterior (hence depending on its parameterization by $\boldsymbol{\phi}$) and $\boldsymbol{\mu}_k$ denotes the k-th component of the vector. The loss over all data points

$\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ is then given by

$$\mathcal{L}_{\text{VAE}}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}, \boldsymbol{\phi}, \boldsymbol{\theta}) = -\sum_{i=1}^{N} \text{ELBO}(\mathbf{x}^{(i)}, \boldsymbol{\phi}, \boldsymbol{\theta}).$$

In our implementation, we use $S = 1$ throughout and follow the common practice of adding an $\mathcal{L}^2$ regularization term $\lambda \cdot \| \sum_{\beta \in \{\boldsymbol{\phi}, \boldsymbol{\theta}\}} \beta \|_2^2$ to prevent exploding model parameters. While obtaining an estimate of the gradient with respect to $\boldsymbol{\theta}$ is straightforward, we have to employ a change of variables called the *reparameterization trick* [2] to estimate the gradient with respect to the variational parameters $\boldsymbol{\phi}$. Intuitively, instead of sampling $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we sample some $\boldsymbol{\epsilon}$ from a random variable independent of $\mathbf{x}$ and express $\mathbf{z}$ as a deterministic transformation of $\boldsymbol{\phi}$ and $\boldsymbol{\epsilon}$. Thus, we obtain unbiased estimates of the ELBO with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ that are optimized with stochastic gradient descent [24].

4.4 Convolutional neural networks

For data in the form of two-dimensional arrays such as images, typically *convolutional neural networks* (CNN) are employed, a type of ANN specifically adapted to the spatial structure of image data [29]. CNNs comprise mainly two types of layers, namely *convolutional* layers and *pooling* layers.

A convolutional layer is defined by a number of filters, also called receptive fields, where each filter is represented by a weight matrix. The filters are applied locally, i.e., the weight matrix is multiplied element-wise with local patches of the input image and all values are summed up (mathematically, this corresponds to a discrete convolution, hence the name). Finally, a non-linear activation function is applied to the resulting weighted sum. We can thus think of a filter as a window sliding over the image, convolving the filter weight matrix with different patches of the image.

Each filter is defined by the size of the filter matrix, the stride, i.e., the step size with which it slides across the image, and its corresponding activation function. The locally applied convolution operation accommodates central characteristics of array data such as images, namely the fact that they often exhibit distinctive local motifs, representing highly correlated local groups of values, and that such local motifs can appear in any part of the image, i.e., are invariant to their overall position [29]. This is reflected in the CNN architecture by the idea of using the same weights at different locations, realized by the filter matrix applied across the image.

The convolutional layer outputs a feature map, which is subsequently aggregated in a pooling layer to form higher-level features and detect motifs by building more coarse-grained representations [29]. Technically, this is achieved by aggregating a local patch of a feature map to a single value, e.g., by computing the maximum or average of all values in the patch. Thus, a pooling layer is defined by the corresponding pooling function and the size and stride of the pooling filter. The parameters of the CNN are given by the set of all layer filter matrices and can be estimated with backpropagation as in ANNs.

4.5 Separating foreground and background: the dual-target VAE

Formally, we can describe the dataset $\mathbf{x} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ of images $\mathbf{x}^{(i)} \in \mathbb{R}^{L \times M}$ for some $L, M \in \mathbb{N}$ as a combination of foreground signal $\mathbf{x}_1$ and background signal $\mathbf{x}_2$, such that $\mathbf{x} = f(\mathbf{x}_1, \mathbf{x}_2)$ for some unspecified function f that describes the merging of foreground and background to the overall image.

Here, we consider an approximation of the background $\mathbf{x}_2$ by scaling down the data in order to minimize the bias introduced by the foreground signal $\mathbf{x}_1$, that has a much higher resolution. We choose $g : \mathbb{R}^{L \times M} \to \mathbb{R}^{l \times m}$ with $L = ls$, $M = ms$ for some $l, m \in \mathbb{N}$ to be a pooling function with stride s, that samples patches of size s down to a single pixel, thus creating a smaller image that averages out the fine-grained foreground signal, retaining mostly large-scale background activity.

As the foreground signal $\mathbf{x}_1$ cannot be approximated in a similarly simple fashion, we employ a pragmatic solution by introducing a cut-off and only retain pixel values larger than this threshold, corresponding to pixels with high activity. Thus, we approximate $\mathbf{x}_1$ by $\mathbf{x}_{>t} = \mathbf{x} \cdot \mathbb{I}_{\{\mathbf{x}>t\}}(\mathbf{x})$.

To model foreground and background separately, we introduce separate VAE models with latent variables $\mathbf{z}_1$ and $\mathbf{z}_2$, that form compressed representations of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. Accordingly, we consider likelihoods $p_{\boldsymbol{\theta}_1}(\mathbf{x})$ for the foreground and $p_{\boldsymbol{\theta}_2}(\mathbf{x})$ for the background and define corresponding variational encoders, $q_{\boldsymbol{\phi}_1}(\mathbf{z}_2|\mathbf{x})$ and $q_{\boldsymbol{\phi}_2}(\mathbf{z}_2|\mathbf{x})$ realized as CNNs.

To account for the different nature of sharp and local foreground signal and blurry, large-spanning background signal, we choose a small receptive field with a size similar to the regions of interest in the foreground for $q_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{x})$, while $q_{\boldsymbol{\phi}_2}(\mathbf{z}_2|\mathbf{x})$ is parametrized by a CNN with larger receptive field. To obtain a joint model of the entire image including both foreground and background, we need to allow for interaction between the two models and thus add a fully connected ANN layer that integrates $\mathbf{z}_1$ and $\mathbf{z}_2$ into a joint latent space, before passing the output into the separate decoders. More specifically, the encoding and generative process of the resulting dual-target VAE model is given by

$$\begin{aligned}
&\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1 = \text{EncoderNeuralNet}_{\boldsymbol{\phi}_1}(\mathbf{x}) \\
&\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2 = \text{EncoderNeuralNet}_{\boldsymbol{\phi}_2}(\mathbf{x}) \\
&\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1) = q_{\boldsymbol{\phi}_1}(\mathbf{z}_1|\mathbf{x}) \\
&\mathbf{z}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2) = q_{\boldsymbol{\phi}_2}(\mathbf{z}_2|\mathbf{x}) \\
&\widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2 = \text{InteractionNeuralNet}(\mathbf{z}_1, \mathbf{z}_2) \\
&\widehat{\mathbf{x}} \sim p_{\boldsymbol{\theta}_1}(\mathbf{x}|\widetilde{\mathbf{z}}_1) \\
&\widehat{g(\mathbf{x})} \sim p_{\boldsymbol{\theta}_2}(g(\mathbf{x})|\widetilde{\mathbf{z}}_2).
\end{aligned} \tag{3}$$

Consequently, we can derive an optimization criterion for the overall model by considering both ELBOs from the foreground and background part:

$$\mathcal{L}_{\text{DualVAE}}(\mathbf{x}) = \text{ELBO}(\mathbf{x}, \boldsymbol{\phi}_1, \boldsymbol{\theta}_1) + \text{ELBO}(\mathbf{x}, \boldsymbol{\phi}_2, \boldsymbol{\theta}_2)$$

Estimators for the two ELBOs are calculated as in (2), such that the model can thus be trained by stochastic gradient descent analogously to the vanilla VAE (see Section 4.3). Note that the two coupled VAE models are not deterministic functions, as each includes drawing a sample from the latent variable $\mathbf{z}$ via a reparameterization (see Section 4.3). Also, because the latent spaces interact, each function includes the computation of both encoders.

The implementation of the approach and the training procedure of the dual-target VAE loss are illustrated and described in more detail in the accompanying Jupyter notebook.

4.6 Smoothing the latent representation with a kernel

To take into account the temporal correlation between subsequent frames, we additionally propose a smoothing step in the latent representation. Instead of training the model on randomly selected batches comprised of arbitrarily temporally distant frames, we consider neighborhoods of adjacent frames for training and thus make information about temporal proximity accessible to the model. For each neighborhood, we take a kernel-weighted average of the latent representations of all frames in the neighborhood, and use this smoothed latent representation as input for the decoder and loss function.

Formally, for some $m \in \mathbb{N}$, the neighborhood of size $2m+1$ of the i-th frame $\mathbf{x}^{(i)}$ is then given by $N_m(\mathbf{x}^{(i)}) := (\mathbf{x}^{(i-m)}\ \mathbf{x}^{(i-m+1)}\ \ldots\ \mathbf{x}^{(i)}\ \ldots\ \mathbf{x}^{(i+m)})$. Denoting with L the sum of dimensions of the latent variables $z_{1,2}$, a kernel function $K : \mathbb{R}^{L\times(2m+1)} \to \mathbb{R}^L$ is applied to the latent representation mean $\mu = (\mu_1, \mu_2)$ for the respective posterior distributions of the two VAE models. As a kernel, we use the tricube function defined by $K(x) = (1 - (|x|)^3)^3$ and use the distance in frames to compute the weights with the kernel.

Instead of batch learning, we can now employ *neighborhood learning*: Instead of randomly partitioning the data set in disjoint batches $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_J$, and calculating the loss of the entire batch as $\mathcal{L}_{\text{DualVAE}}(\boldsymbol{B}_j) = \sum_{\mathbf{x}\in\boldsymbol{B}_j} \mathcal{L}_{\text{DualVAE}}(\mathbf{x})$ before applying one gradient update of the model parameters,

we form a neighborhood $\boldsymbol{N}_m(\mathbf{x})$ of size $2m+1$ around each $\mathbf{x}$ as defined above. With this, we can derive one loss value for the neighborhood by using the latent representation obtained as a weighted average of all frames in the batch with weights given by the kernel, i.e.,

$$\begin{aligned}\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1 &= K(\mathrm{EncoderNeuralNet}_{\boldsymbol{\phi}_1}(\boldsymbol{N}_m(\mathbf{x}))) \\ \boldsymbol{\mu}_2, \boldsymbol{\sigma}_2 &= K(\mathrm{EncoderNeuralNet}_{\boldsymbol{\phi}_2}(\boldsymbol{N}_m(\mathbf{x}))),\end{aligned}$$

and subsequently use $\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\sigma}_2$ as in (3) to obtain samples $\mathbf{z}_1, \mathbf{z}_2$ and reconstructions $\widehat{\mathbf{x}}, \widehat{g(\mathbf{x})}$. Thus, from each frame $\mathbf{x}$ and its respective neighborhood $\boldsymbol{N}_m(\mathbf{x})$, we calculate a single value of $\mathcal{L}_{\mathrm{DualVAE}}(\boldsymbol{N}_m(\mathbf{x}))$ before applying one gradient update. Note how in this case the complexity for computing the gradients of each frame increases, as it has to flow through $2n$ additional computations of the $\mathrm{EncoderNeuralNet}$.

To further explore our approach to obtain a smoother latent representation, we compare kernel smoothing in the latent space described above to smoothing performed at the level of the loss values of all frames in a neighborhood. This means that we calculate the loss $\mathcal{L}_{\mathrm{DualVAE}}(\mathbf{x}^{(j)})$ for a frame $\boldsymbol{N}_m(\mathbf{x})$, for all $\mathbf{x}^{(j)} \in \boldsymbol{N}_m(\mathbf{x})$, and obtain a common loss as weighted average

$$\begin{aligned}&\mathcal{L}_{\mathrm{DualVAE}}(N_m(\mathbf{x}^{(i)})) \\ &= K(\mathcal{L}_{\mathrm{DualVAE}}(\mathbf{x}^{(i-m)}, \ldots, \mathcal{L}_{\mathrm{DualVAE}}(\mathbf{x}^{(i+m)})).\end{aligned} \tag{4}$$

4.7 Data and preprocessing

We use data obtained from *in-vivo* two-photon calcium imaging of hippocampal CA1 neurons in mice. Mice were intra-hippocampally injected with adeno-associated viral constructs (AAV1.Syn.GCaMP6f. WPRE.SV4, University of Pennsylvania Vector Core) that established a panneuronal expression of the calcium-indicator GcaMP6f in CA1 neurons. Subsequently, a $3\mathrm{mm}$ wide cranial window was implanted above the hippocampal formation to provide optical access to the structure. For details of the procedure, please see [11]. During the imaging mice were head fixed and placed on an air-floating styrofoam ball that allowed them to navigated through a virtual reality which was projected to four screens placed around them. Mice were trained to run along $4\mathrm{m}$ long linear tracks to obtain goal-oriented rewards. Imaging was performed using a resonant/galvo high-speed laser scanning two-photon microscope (Neurolabware) with a frame rate of $30\mathrm{Hz}$ for bidirectional scanning. The microscope was equipped with an electrically tunable, fast z-focusing lens (optotune, Edmund optics) to switch between z-planes within less than a millisecond. GCaMP6f was excited at $930\mathrm{nm}$ with a femtosecond-pulsed two-photon laser (Mai Tai DeepSee, Spectra-Physics). To maximize the number of recorded neurons we scanned three imaging planes ($\approx 25\mu\mathrm{m}$ spacing between planes) in rapid alternation so that each plane was sampled at $10\mathrm{Hz}$.

As a preprocessing step, we performed motion correction with the NoRMCorre algorithm [30] and subsequently normalized the data.

4.8 Implementation

The experiments have been implemented in the Julia programming language (version 1.4.1) and were carried out on a Linux cluster utilizing 150GB of RAM, 16 CPU-cores, and one NVIDIA Tesla V100 GPU. Model training was realized using the Julia machine learning library Flux.jl (version 10.1.0) and CUDA.jl (version 0.1.0) for GPU support.

To ensure comparability, all considered models (VAE with a CNN architecture, dual-target VAE with and without kernel smoothing) use the same architecture for their encoders and decoders (e.g., the encoder of the foreground model in the dual-target VAE is the same as the encoder of the convolutional baseline). In the following, we briefly describe the implementation of the different models. Details on the model structure and parameters are given in the supplementary material.

For the simple CNN-based VAE, we use 3 convolutional layers and one pooling layer for the CNN encoder and 4 transposed convolutional layers for the decoder. The first layer of the encoder is configured with a small filter of size 25 by 25. To keep the computational cost of the fully-connected ANN approach comparable to the CNN scenario, both encoder and decoder were realized as one-layer ANNs. In the dual-target VAE, the foreground model has the same architecture as the CNN-based VAE described above. For the background model, we employ of a two-layer convolutional encoder with a receptive field of size 100 by 100 and a fully-connected ANN decoder. The training target of the background VAE is the scaled version of the original data as described above in Section 4.5.

The training data comprises 10000 contiguous, normalized frames of size 796 by 512 pixels. The smaller target of size 8 by 5 for the background models is obtained by padding with zeroes and pooling 100×100 patches into their sums of pixels and normalizing subsequently. For pretraining, we choose $t = 0.3$ as threshold of the foreground approximation. We use the ADAM optimizer [31] for stochastic gradient descent and train the models for 20 epochs on the cut-off data and another 20 epochs on the original data with a batch size of 5. For the approaches with kernel smoothing, we use the parameters of the dual-target VAE as initialization and train for another 20 epochs and a neighbourhood of size 5.

The source code is available at `https://gitlab.imbi.uni-freiburg.de/maren/incorporating-structural-knowledge-into-unsupervised-deep-learning-for-two-photon-imaging-data`

Acknowledgements

The authors would like to thank Thomas Hainmueller, who generated the two-photon imaging data. Further, the authors acknowledge support by the state of Baden-Württemberg through bwHPC.

Funding

This work is supported by the DFG (German Research Foundation) – 322977937/GRK2344 (MH), SFB958/Z02 (JS).

Author contributions

Conceptualization: HB
Data curation: AK, MB
Methodology: HB, FE, MH, CB
Investigation: FE, MH, CB
Visualization: FE, MH
Writing – original draft: MH, FE, JS, AK, MB
Writing – review and editing: all authors

Competing interests

The authors have declared no competing interests.

Data and materials availability

Code to reproduce the figures in the manuscript and a Jupyter notebook tutorial illustrating the main functionalities of our method are available in a public Gitlab repository at `https://gitlab.imbi.uni-freiburg.de/maren/incorporating-structural-knowledge-into-unsupervise deep-learning-for-two-photon-imaging-data`. In the repository, we also provide a small exemplary dataset to recapitulate our analyses.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations (ICLR), Conference Track Proceedings*, 2014.

[3] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

[4] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. Universal differential equations for scientific machine learning, 2020. arXiv preprint: https://arxiv.org/abs/2001.04385.

[5] Tsai-Wen Chen, Trevor J. Wardill, Yi Sun, Stefan R. Pulver, Sabine L. Renninger, Amy Baohan, Eric R. Schreiter, Rex A. Kerr, Michael B. Orger, Vivek Jayaraman, Loren L. Looger, Karel Svoboda, and Douglas S. Kim. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499:295–300, 2013.

[6] Ricardo Augusto de Melo Reis, Hércules Rezende Freitas, and Fernando Garcia de Mello. Cell calcium imaging as a reliable method to study neuron–glial circuits. *Frontiers in Neuroscience*, 14:975, 2020.

[7] Pallavi Gupta, Nandhini Balasubramaniam, Hwan-You Chang, Fan-Gang Tseng, and Tuhin Subhra Santra. A single-neuron: Current trends and future prospects. *Cells*, 9(6), 2020.

[8] Yaesop Lee, Jing Xie, Eungjoo Lee, Srijesh Sudarsanan, Da-Ting Lin, Rong Chen, and Shuvra S. Bhattacharyya. Real-time neuron detection and neural signal extraction platform for miniature calcium imaging. *Frontiers in Computational Neuroscience*, 14:43, 2020.

[9] Jesse A Livezey and Joshua I Glaser. Deep learning approaches for neural decoding across architectures and recording modalities. *Briefings in Bioinformatics*, 22(2):1577–1591, 2020.

[10] C Grienberger and A Konnerth. Imgaing calcium in neurons. *Neuron*, 75(5):862–885, 2012.

[11] T. Hainmueller and M. Bartos. Parallel emergence of stable and dynamic memory engrams in the hippocampus. *Nature*, 558:292–296, 2018.

[12] Jiangheng Guan, Jingcheng Li, Shanshan Liang, Ruijie Li, Xingyi Li, Xiaozhe Shi, Ciyu Huang, Jianxiong Zhang, Junxia Pan, Hongbo Jia, Le Zhang, Xiaowei Chen, and Xiang Liao. Neuroseg: automated cell detection and segmentation for in vivo two-photon ca2+ imaging data. *Brain Structure and Function*, 223:519–533, 2018.

[13] Chiara Magliaro, Alejandro L Callara, Nicola Vanello, and Arti Ahluwalia. Gotta trace 'em all: A mini-review on tools and procedures for segmenting single neurons toward deciphering the structural connectome. *Frontiers in bioengineering and biotechnology*, 7:202, 2019.

[14] Eftychios A Pnevmatikakis. Analysis pipelines for calcium imaging data. *Current Opinion in Neurobiology*, 55:15 – 21, 2019. Machine Learning, Big Data, and Neuroscience.

[15] Andrea Giovannucci, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, Farzaneh Najafi, Jeffrey L Gauthier, Pengcheng Zhou, Baljit S Khakh, David W Tank, Dmitri B Chklovskii, and Eftychios A Pnevmatikakis. CaImAn an open source tool for scalable calcium imaging data analysis. *eLife*, 8:e38173, 2019.

[16] Marius Pachitariu, Carsen Stringer, Mario Dipoppa, Sylvia Schröder, L. Federico Rossi, Henry Dalgleish, Matteo Carandini, and Kenneth D. Harris. Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv*, 2017. bioRxiv preprint: https://www.biorxiv.org/content/early/2017/07/20/061507.

[17] Elke Kirschbaum, Alberto Bailoni, and Fred A. Hamprecht. DISCo: Deep learning instance segmentation, and correlations for cell segmentation in calcium imaging. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020*, pages 151–162, Cham, 2020. Springer International Publishing.

[18] Somayyeh Soltanian-Zadeh, Kaan Sahingur, Sarah Blau, Yiyang Gong, and Sina Farsiu. Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. *Proceedings of the National Academy of Sciences*, 116(17):8554–8563, 2019.

[19] Artur Speiser, Jinyao Yan, Evan W Archer, Lars Buesing, Srinivas C Turaga, and Jakob H Macke. Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4024–4034. Curran Associates, Inc., 2017.

[20] D. Soulet, J. Lamontagne-Proulix, B. Aubé, and D. Davalos. Multiphoton intravital microscopy in small animals: motion artefact challenges and technical solutions. *Journal of Microscopy*, 278 (1):3–17, 2020.

[21] Moritz Hess, Maren Hackenberg, and Harald Binder. Exploring generative deep learning for omics data using log-linear models. *Bioinformatics*, 36(20):5045–5053, 2020.

[22] Kota Miura and Nataša Sladoje. *Bioimage Data Analysis Workflows*. Learning Materials in Biosciences. Springer, Cham, 2020.

[23] Erik Meijering. A bird's-eye view of deep learning in bioimage analysis. *Research Network of Computational and Structural Biotechnology*, 18:2312–2325, 2020.

[24] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245.

[25] David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[26] Ruslan Salakhutdinov and Geoffrey Hinton. An efficient learning procedure for deep boltzmann machines. *Neural Comput.*, 24(8):1967–2006, 2012.

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[28] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[30] Eftychios A. Pnevmatikakis and Andrea Giovannucci. Normcorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods*, 291: 83–94, 2017.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations (ICLR), Conference Track Proceedings*, 2015.

[32] Camden J. MacDowell and Timothy J. Buschman. Low-dimensional spatiotemporal dynamics underlie cortex-wide neural activity. *Current Biology*, 30(14):2665–2680.e8, 2020.

[33] Nickel M Chen RTQ, Amos B. Neural spatio-temporal point processes, 2020. arXiv preprint: https://arxiv.org/abs/2011.04583.

[34] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series, 2020. arXiv preprint: https://arxiv.org/abs/2005.08926.

[35] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the

author). *Statistical Science*, 16(3):199–231, 08 2001.

[36] Bradley Efron. Prediction, estimation, and attribution. *Journal of the American Statistical Association*, 115(530):636–655, 2020.