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] Core Facility Advanced Medical Bioimaging, Charité - Universitätsmedizin, Humboldt University Berlin, Germany
[4] Institute for Physiology I, Systemic and Cellular Neurophysiology, Faculty of Medicine, University of Freiburg, Germany

May 18, 2021

Abstract

Live imaging techniques, such as two-photon imaging, promise novel insights into cellular activity patterns at a high spatial and temporal resolution. While current deep learning approaches typically focus on specific supervised tasks in the analysis of such data, e.g., learning a segmentation mask as a basis for subsequent signal extraction steps, we investigate how unsupervised generative deep learning can be adapted to obtain interpretable models directly at the level of the video frames. Specifically, we consider variational autoencoders for models that infer a compressed representation of the data in a low-dimensional latent space, allowing for insight into what has been learned. Based on this approach, we illustrate how structural knowledge can be incorporated into the model architecture to improve model fitting and interpretability. Besides standard convolutional neural network components, we propose an architecture for separately encoding the foreground and background of live imaging data. We exemplify the proposed approach with two-photon imaging data from hippocampal CA1 neurons in mice, where we can disentangle the neural activity of interest from the neuropil background signal. Subsequently, we illustrate how to impose smoothness constraints onto the latent space for leveraging knowledge about gradual temporal changes. As a starting point for adaptation to similar live imaging applications, we provide a Jupyter notebook with code for exploration. Taken together, our results illustrate how architecture choices for deep generative models, such as for spatial structure, foreground vs. background, and gradual temporal changes, facilitate a modeling approach that combines the flexibility of deep learning with the benefits of incorporating domain knowledge. Such a strategy is seen to enable interpretable, purely image-based models of activity signals from live imaging, such as for two-photon 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 authors wish to be regarded as joint first authors.
†Corresponding author: maren@imbi.uni-freiburg.de

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].

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 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, and also for incorporating knowledge about gradual temporal changes in the processes observed by live cell imaging.

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 then is 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 images of a neuronal population that expresses a fluorescent calcium indicator and allows to visualize the increase in intracellular Ca2+-concentration accompanying neurons' spiking activity [5]. More generally, 'neural decoding' refers to techniques that use brain signals to make predictions about behaviour, perception, or cognitive state and are becoming increasingly important for neuroscientific research [6–9]. For example, calcium imaging techniques enable optical measurement of large neural populations at a high spatio-temporal resolution and 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 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, which 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 not restricted to a specific task, yet be tailored to the properties of two-photon imaging data.

This paper is structured as follows. We first give an overview of the typical steps of image processing, before introducing generative deep learning in general and VAEs in particular, as well as the convolutional architecture integrated into our models. Next, we detail the proposed approaches for distinguishing between foreground and background, and for incorporating smoothness assumptions corresponding to gradual temporal changes. We then present results of the models trained on a two-photon imaging dataset and finally discuss our findings, pointing out limitations and directions for future research. The implementation and an exemplary application of the approach is illustrated in an accompanying Jupyter notebook available at `https://gitlab.imbi.uni-freiburg.de/maren/incorporating-structural-knowledge-into-unsupervised-deep-learning-for-two-photon-imaging-data`.

2 Background

2.1 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.

Since the advent of bioimage analysis, a plethora of successful methods and tools has been developed to perform conventional computational bioimage analysis. The main advantage of such non-deep-learning analysis 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., intra-vital time-lapse imaging of the brain activity of behaving animals), the produced bioimage data has enormously increased in size and complexity within the last decades. Clearly, the conventional bioimage 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 using the conventional approaches. 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, since these flexible algorithms can be successfully employed to efficiently analyze large, complex data sets.

2.2 Generative deep learning

Our proposed analysis strategy is based on *variational autoencoders* (VAEs), a generative deep learning approach first presented in [2]. Here, the term 'generative' refers to the fact that during training, the model learns a joint distribution over all input variables that should ideally approximate the true underlying data distribution. Before presenting the VAE model in more detail, we give a brief introduction to neural networks and generative deep learning.

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.

2.3 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:

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

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

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

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].

2.4 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, 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), before 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.,

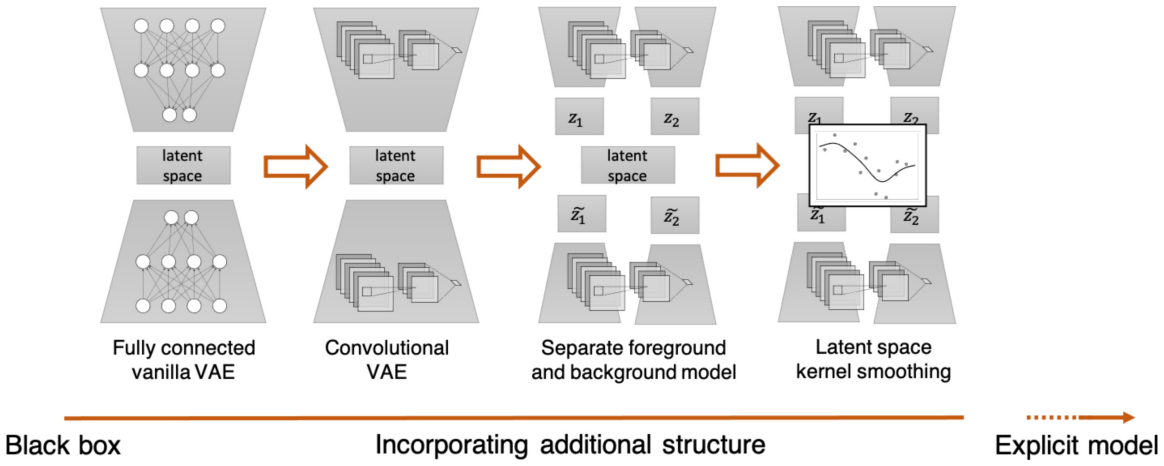

Figure 1: Layers of structural information can be gradually combined with a fully connected vanilla VAE model (1) to enable a step-wise transition from a black-box deep learning model towards a more explicit modeling approach that takes into account problem structure by adding a convolutional architecture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

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 merge similar features, in order 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 filter matrices of its layers and can be estimated with backpropagation analogously to ANNs.

3 Materials and methods

In the following sections, we describe our proposal for adapting a VAE-based model to the properties of the two-photon imaging data. To tackle the challenging modeling task of simultaneously capturing the foreground signal with the distinct cell activity and the more coarse-grained, blurry background signal, while accounting for the temporal correlation between neighboring frames, we gradually increase the level of additional structural information integrated into the model and 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 one that is tailored to the data: In a first step, we train a fully-connected vanilla VAE on the dataset, as described in Section 2.3, 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 2.4). 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.

3.1 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 $10\mathrm{Hz}$.

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

3.2 Separating foreground and background: the dual-target VAE

Formally, we can model the dataset $\mathbf{x} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ 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 fore- and background signal in 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.

The foreground signal $\mathbf{x}_1$ cannot be approximated as simply. Hence, as a pragmatic solution, we can optimize the foreground likelihood for $\mathbf{x}$ instead of $\mathbf{x}_1$, ignoring the bias introduced by the background signal. Alternatively, we approximate $\mathbf{x}_1$ by $\mathbf{x}_{>t} = \mathbf{x} \cdot \mathbb{I}_{\{\mathbf{x}>t\}}(\mathbf{x})$, i.e., we introduce a threshold k that serves as a cut-off, such that only pixel values larger than k corresponding to pixels with high activity are retained.

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.

In order 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 2.3). To simplify the notation for the practical discussion in the following, the functions representing the mapping of an input $\mathbf{x}$ to the outputs of the foreground and background VAE will be referred to as $\text{VAE}_1(\mathbf{x})$ and $\text{VAE}_2(\mathbf{x})$ respectively. Note that these are not deterministic functions, as each includes drawing a sample from the latent variable $\mathbf{z}$ via a reparameterization (see Section 2.3). 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.

3.3 Smoothing the latent representation with a kernel

Since the images correspond to frames in a video, subsequent frames exhibit similar cell activity patterns, which implies a strong temporal correlation between a frame and its neighboring images. Hence, we want to explicitly incorporate this 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. We specifically propose kernel smoothing in the latent space. Instead of training the model on randomly selected batches comprised of arbitrarily temporally distant frames, we consider a neighborhood of adjacent frames for updating parameters via the gradient of the loss functions, to make information about temporal proximity accessible to the model. We define a neighborhood by its size n and position $i \in \mathbb{N}$. 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.

We can now replace batch learning by *neighborhood learning*. Instead of randomly partitioning the data set in disjoint batches $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_J$, where $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} = \boldsymbol{B}_1 \dot\cup \boldsymbol{B}_2 \cdots \dot\cup \boldsymbol{B}_J$, 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.,

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

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 Results

4.1 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.

We compare (1) a VAE with a CNN architecture, (2) a dual-target VAE, and (3) a dual-target VAE with kernel smoothing. To ensure comparability, all three scenarios use the same architecture for their encoders and decoders (e.g., the encoder of VAE_1 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.

First, we evaluate the performance of the baseline CNN approach against a standard VAE with a fully-connected ANN encoder and decoder. The CNN encoder has 3 convolutional layers and one pooling layer and the decoder has 4 transposed convolutional layers. The first layer of the encoder is configured with a small filter, also called receptive field, of size 25 by 25. To keep the computational cost of the ANN approach comparable to the CNN scenario, the encoder and decoder were both realized as one-layer ANNs. Next, we add a background model to the baseline approach, which consists 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 VAE_2 is the scaled version of the original data as described above in Section 3.2. Finally, we evaluate the performance of the dual-target VAE with kernel smoothing on the latent representation as described in Section 3.3 and compare it against the dual-target VAE without temporal smoothing. Additionally, we train a dual-target VAE that employs the same smoothing procedure with a tricube kernel based on the loss values of the neighborhood of each frame.

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.

4.2 Convolutional VAE

With the 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 the CNN approach achieved good results on prominent foreground cells. This implies that the advantages of CNNs over

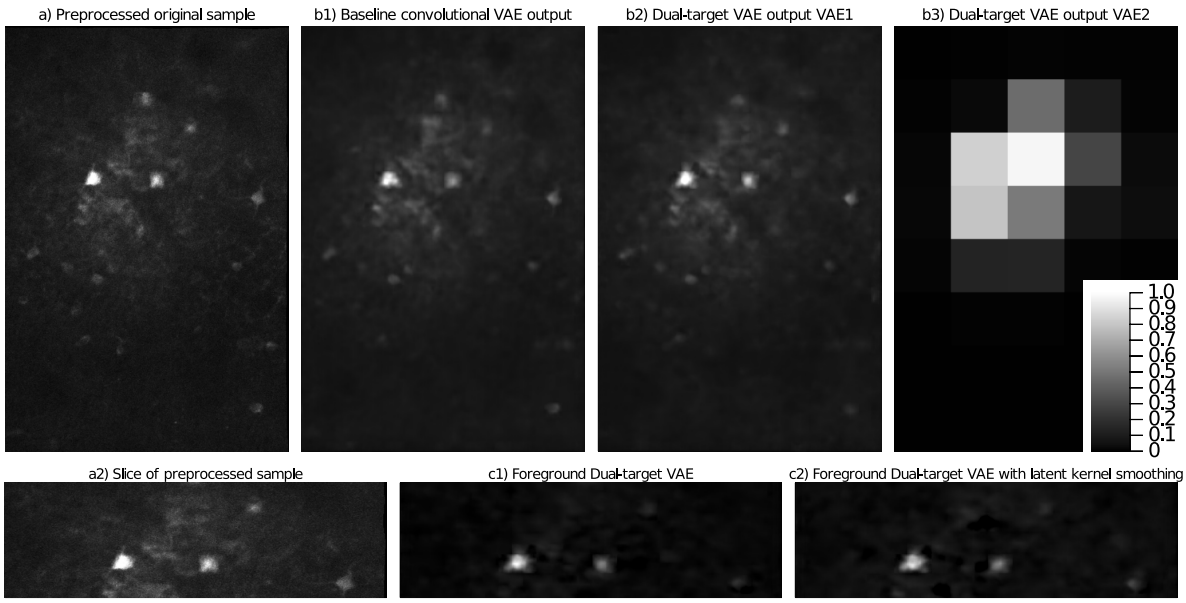

Figure 2: Top row: Preprocessed sample frame from the data and its reconstructions of the convolutional baseline VAE approach and of dual-target VAE respectively. Bottom row: Slice of the same sample with corresponding reconstructions of only the foreground of dual-target VAE and dual-target VAE with latent kernel smoothing.

a standard fully-connected neural network structure also apply to frames from two-photon imaging videos, and that 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 architecture.

Yet, the signal of smaller cells that spike with lower magnitude and more rapidly is reconstructed with lower intensity or not at all in a large proportion of these cell populations. This can be observed over most other frames. The activity of the less dominant cells is reconstructed only with low accuracy and results in a lot of false positives and false negatives. 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.

4.3 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 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 VAE_2 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 dominant foreground cell activity is retained in the

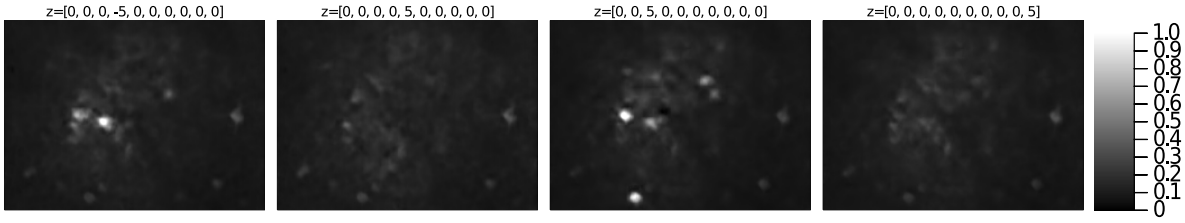

Figure 3: Exemplary exploration plots (sliced) of foreground and background model of the baseline convolutional VAE model (left) and the dual-target VAE (right). The configuration of the latent space that was used to create the plot is annotated on top. Dimensions of the baseline model are chosen according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

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

4.4 Dual-target VAE with latent kernel smoothing

As a last step, we evaluate the impact of kernel smoothing on the model performance. Comparing the foreground activity in Figure 2 (bottom row), 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., the first, second and third frame in the sequence correspond to the first, third and fifth frame in the respective part of the 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 3.3, 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.

5 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 that can ultimately be linked to phenotype data to facilitate biological insight. While current

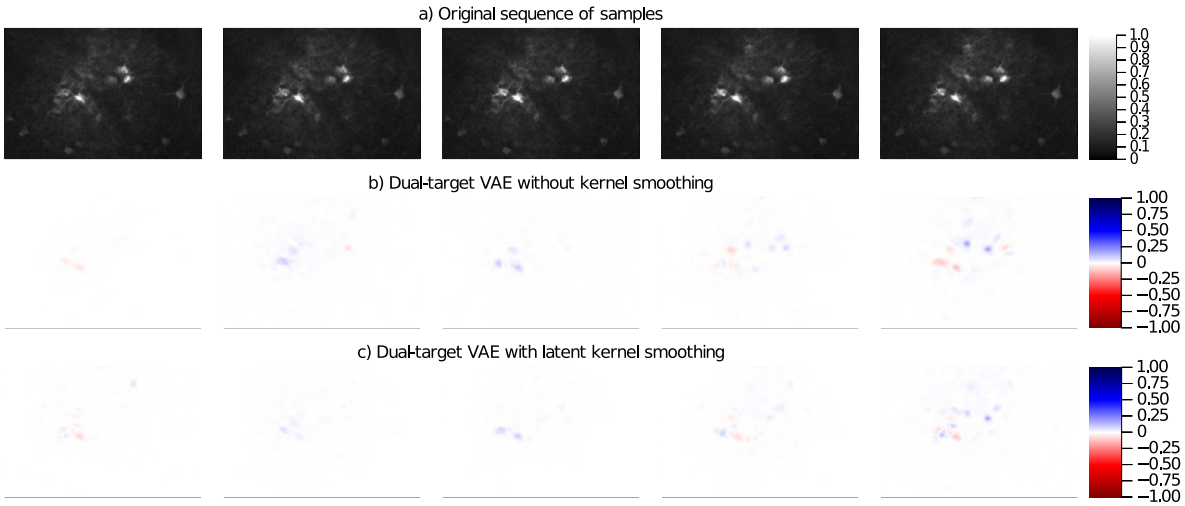

Figure 4: Exemplary sequence of frames and the differences between the model reconstructions of adjacent frames with and without latent kernel smoothing. a) Original exemplary sequence of samples with a distance of two, i.e., adjacent frames in the plot correspond to every second frame in the original 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.

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 VAEs, directly applied to the video data frames from two-photon imaging, for modeling neural activity in a latent representation. Here, the focus was on exploring how gradually incorporating structural components, which reflect key properties of two-photon imaging data, affect the latent representation, thus facilitating insight into what the model has learned and exemplifying 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, in the encoder and decoder allowed to infer a latent representation based on which the input images could be reconstructed. 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 smaller active cells in the foreground more frequently and more clearly, thus capturing the overall activity more accurately by exploiting the similarity of neighboring 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 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 respective learned representation can be inspected to assess to what extent it is influenced by 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, and, rather than viewing the modeling cultures as a dichotomy [35], represents a step towards uniting the two worlds and combine 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, and 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.

6 Competing interests

The authors have declared no competing interests.

7 Acknowledgements

The authors would like to thank Thomas Hainmueller, who generated the two-photon imaging data. This work is supported by the DFG (German Research Foundation) – 322977937/GRK2344 (MH), SFB958/Z02 (JS). Further, the authors acknowledge support by the state of Baden-Württemberg through bwHPC.

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, 12 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, 08 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.