Generative Model based Training of Deep Neural Networks for Event Detection in Microscopy Data

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Artur Speiser
aus Shanatas / Kasachstan

Tübingen
2022

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	09.12.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Jakob H. Macke
2. Berichterstatter/-in:	Prof. Dr. Philipp Berens

Abstract

Several imaging techniques employed in the life sciences heavily rely on machine learning methods to make sense of the data that they produce. These include calcium imaging and multi-electrode recordings of neural activity, single molecule localization microscopy, spatially-resolved transcriptomics and particle tracking, among others. All of them only produce indirect readouts of the spatiotemporal events they aim to record. The objective when analysing data from these methods is the identification of patterns that indicate the location of the sought-after events, e.g. spikes in neural recordings or fluorescent particles in microscopy data.

Existing approaches for this task invert a forward model, i.e. a mathematical description of the process that generates the observed patterns for a given set of underlying events, using established methods like MCMC or variational inference. Perhaps surprisingly, for a long time deep learning saw little use in this domain, even though it became the dominant approach in the field of pattern recognition over the previous decade. The principal reason is that in the absence of labeled data needed for supervised optimization it remains unclear how neural networks can be trained to solve these tasks. To unlock the potential of deep learning, this thesis proposes different methods for training neural networks using forward models and without relying on labeled data. The thesis rests on two publications:

In the first publication we introduce an algorithm for spike extraction from calcium imaging time traces. Building on the variational autoencoder framework, we simultaneously train a neural network that performs spike inference and optimize the parameters of the forward model. This approach combines several advantages that were previously incongruous: it is fast at test-time, can be applied to different non-linear forward models and produces samples from the posterior distribution over spike trains.

The second publication deals with the localization of fluorescent particles in single molecule localization microscopy. We show that an accurate forward model can be used to generate simulations that act as a surrogate for labeled training data. Careful design of the output representation and loss function result in a method with outstanding precision across experimental designs and imaging conditions.

Overall this thesis highlights how neural networks can be applied for precise, fast and flexible model inversion on this class of problems and how this opens up new avenues to achieve performance beyond what was previously possible.

Zusammenfassung

Eine Reihe von bildgebenden Verfahren in den Biowissenschaften ist auf Methoden des maschinellen Lernens angewiesen um die Daten, die sie produzieren, auszuwerten. Dazu gehören, unter anderen, Aufnahmen neuronaler Aktivität mittels Kalzium Imaging und Mikroelektrodenarrays, Ortsauflösende Transkriptomik und Partikelverfolgung. All diese Verfahren deuten nur indirekt auf die raum-zeitlichen Ereignisse, die sie versuchen aufzunehmen, hin. Deshalb ist das Ziel bei der Auswertung der entstehenden Daten die Identifikation von Mustern die auf die Position der gesuchten Ereignisse hinweisen, z.B. Spikes in neuronalen Aufnahmen oder fluoreszierende Partikel in Mikroskopie Daten. Vorhandene Ansätze für diese Aufgabe invertieren ein generatives Modell, also eine mathematische Beschreibung des Prozesses, der die beobachteten Muster für eine gegebene Zusammenstellung von Ereignissen erzeugt. Dies geschieht mit bewährten Methoden wie MCMC oder Variational Inference. Methoden des Deep Learnings haben lange Zeit nur sehr begrenzte Anwendung in diesem Feld gefunden, und das obwohl sie in der vergangenen Dekade der dominante Ansatz in der Mustererkennung geworden sind. Der Hauptgrund dafür ist, dass es ohne verfügbare Ziel-Variablen die für das überwachte Lernen nötig sind, unklar ist wie neuronale Netzwerke trainiert werden können um diese Aufgaben zu lösen. Um das Potential des Deep Learnings zu erschließen stellt diese Arbeit verschiedene Methoden vor, mit denen neuronale Netzwerke mithilfe von generativen Modellen trainiert werden können, ohne dabei auf bereits ausgewertete Daten angewiesen zu sein. Die Arbeit beruht auf zwei Publikationen:

In der ersten Publikation beschreiben wir einen Algorithmus zur Identifikation von Spikes in Kalzium Imaging Zeitreihen. Auf dem Konzept des Variational Autoencoders aufbauend, trainieren wir ein Netzwerk das Spikes identifiziert und optimieren gleichzeitig die Parameter des generativen Modells. Dieser Ansatz vereinigt mehrere Vorteile die bisher unvereinbar waren: er ist schnell bei der Auswertung, kann einfach mit verschiedenen nicht-linearen generativen Modellen verwendet werden und produziert Stichproben aus der A-posteriori-Verteilung über Abfolgen von Spikes.

Die zweite Publikation dreht sich um die Lokalisation von fluoreszenten Partikeln in der Einzelmolekül-Fluoreszenzmikroskopie. Wir zeigen, dass ein akkurates generatives Modell uns erlaubt Simulationen zu generieren, die als Surrogat für echte Aufnahmen als Trainingsdaten verwendet werden können. Die sorgfältige Ausarbeitung der Ausgaberepräsentation des Netzwerks und der Verlustfunktion ergeben eine Methode, die hervorragende Leistung unter diversen experimentellen Bedingungen erreicht.

Zusammenfassend stellt diese Arbeit heraus wie neuronale Netzwerke für die präzise, schnelle und flexible Invertierung von generativen Modellen in dieser Klasse von Problemen als Lösungsstrategie verwendet werden können.

Acknowledgements

Even more than in our natural state of being, as a PhD student one often feels subject to circumstances beyond one's control. When working on research projects that might go nowhere for years on end to finally send them to review processes that approach total randomness, we depend on the people around us to keep the experience wholesome.

First and foremost my gratitude goes to my supervisor Jakob Macke for assembling such a group of people and letting me be part of it. Beyond that I'm thankful that I could rely on his continuous support, his trust and the freedom he afforded me to work in my own idiosyncratic ways. Similarly I want to thank Srinivas Turaga who I worked with continuously throughout my PhD and who gave me the great opportunity to repeatedly travel to Janelia Research Campus. He has been an inexhaustible source of ideas and was willing to discuss science at any time of the day.

I am especially grateful for the people that made up our group in the past and present. They made me feel at home wherever our lab happened to be at any specific time. Special thanks go to Marcel Nonnenmacher with whom I shared my office and my opinions throughout this time. I want to thank Auguste Schulz for her dedication and resourcefulness in bringing this group even closer together. My special gratitude goes to Poornima Ramesh and her continued support right till the end when she provided valuable feedback on this thesis.

I want to thank Jonas Ries and Lucas-Raphael Müller for their help in bringing the DECODE project to a successful end and making it something I'm proud of.

I want to thank everyone who made it possible to participate in the various conferences, workshops and summer schools I was lucky enough to attend. More specifically, I am grateful for the outstanding support I got from Ezgi Bulca, Bettina Bosse and Franziska Weiler in all administrative matters. I would like to extend my gratitude to Philipp Berens for agreeing to referee this thesis.

I want to thank my family for their support, their care and attention.

My final thanks go to my partner Zoe, who became an expert in the business of blob detection just so she could partake in every small success and mishap, for her encouragement, her help and love.

Contents

Chapter 1

Introduction

In recent years researchers claim that we have entered a ’golden age’ of neuroscience.[29,75] What they refer to is the proliferation in the availability of data which can be observed across the life sciences. This growth progresses on two axis simultaneously. On the one hand we can observe biological processes in ever greater detail, on the other hand the sheer amount of available data is growing rapidly. To name just some of the most visible examples, the connectome for a large area of the fly brain is now mapped and available;[72] spatial transcriptomics can be used to analyze the organization of cell types in the brain,[13] and the growth in the number of neurons for which we can simultaneously record activity has recently further accelerated due to the development of optical calcium imaging techniques[82] (see Fig. 1.1). This is obviously an exciting development for scientists from many disciplines, who are challenged to generate insights from this data.

Before that can happen though, the data often has to go through multiple processing steps to extract the variables of interest from the raw recordings. The optical imaging of neural activity is a good example: the raw data consists of videos showing a number of shapes that vary in brightness over time. Each of them indicates the activity-induced changes in calcium concentration in a single neuron.[22] The desired outputs are the exact timings of action potentials for all the depicted neurons. Going from the videos to the desired spike times is a non-trivial task that has spawned a large amount of research[61,80] dealing with multiple issues common to statistical analysis of imaging data: How do we quantify uncertainty? How can we incorporate our prior knowledge of the underlying processes to increase performance? How do we deal with background?

In parallel to the advances in imaging techniques the last decade saw a paradigm shift in machine learning. Since its resurgence in 2012,[36,40] deep learning has begun to dominate methodological development in many disparate fields. Unsurprisingly, this is also the case for the analysis of microscopy data: deep learning has unique potential to achieve performance beyond what was hitherto possible.

In this thesis I present two deep learning algorithms: DeepSpike for the analysis of calcium imaging (CI) data, and DECODE for single molecule localization microscopy (SMLM) recordings. While apparently very distinct applications, they, and several other imaging methods, share some crucial properties that should guide the development of any analysis algorithm. In both cases, we want to recover a sparse signal consisting of discrete spatiotemporal events from noisy observations. Labeled data, i.e. pairs of images and ground truth information of the desired outputs, is rare or not available at all. Instead, we have extensive knowledge about the underlying biophysical processes that constitute the image-formation. We can express this knowledge via equations which encapsulate the process that goes from a set of latent events to the recorded images. Such a set of equations is often called the forward or generative model. The task at at hand can therefore be framed as one of inverting the generative model to go from recorded images to a (preferably probabilistic) estimate of the latents. Many of the available methods for such problems follow this logic, using well established techniques like Markov chain Monte Carlo (MCMC),[1] maximum likelihood estimation (MLE),[39] deconvolution[18] and variational inference[8] to carry out the inversion, each with their unique advantages and drawbacks.

This thesis explores how we can use neural networks to perform *amortized* inference under such circumstances. The generative model is used to optimize the parameters of the neural network. Once this is done, the network performs model inversion by means of a single forward pass. The result is an algorithmic framework that is precise and fast at test time. Furthermore, it provides a large amount of flexibility with regard to the design of the generative model and the inputs that are used to carry out the inference.

Following this introduction, in chapter 2 I will briefly introduce the two microscopy methods, calcium imaging of neural activity and SMLM, and describe why they pose challenging inference tasks. In chapter 3 the problem settings are formalized and I discuss various approaches that have been previously implemented to solve these two tasks. In chapter 4 I show how deep neural networks can be trained on such data and introduce the methods that are employed in the two publications discussed in this thesis. Chapter 5 contains brief summaries and discussions of the publications. They are attached in full in the appendix. Finally, in chapter 6 I offer some concluding remarks and speculate on future developments.

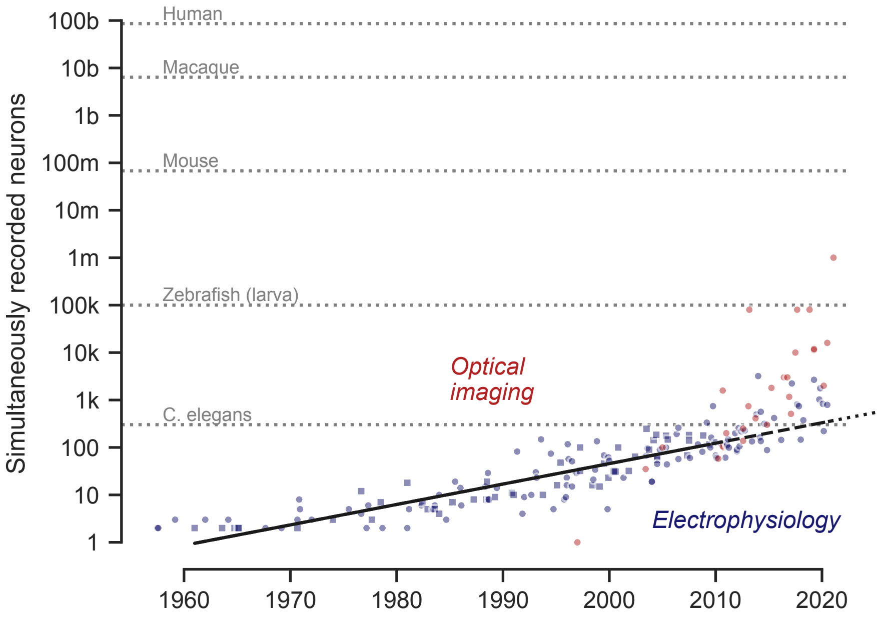


Figure 1.1: Scaling of neural recordings over time. Each data point indicates the number of simultaneously recorded neurons in an experiment using either electrophysiology (blue) or optical calcium imaging (red). Black line shows an exponential fit from the year 2011 that predicted a doubling time of 7.4 years.[79] For comparison the approximate number of neurons in the brains of several species is given. Figure from Urai et alia.[82]

Chapter 2

Background

In this chapter I briefly introduce calcium imaging (CI) and single molecule fluorescence microscopy (SMLM). To this end I characterize the observations that they produce and show examples. Finally, I state the goal of the inference task in each case.

CI as well as SMLM are variants of fluorescence microscopy. Many materials have physical properties that allow them to fluoresce. This means that their molecules are able to absorb photons which leads to their excitation to a higher energetic state. This state usually persists for less than a microsecond, after which the molecule drops into a lower state and emits a photon with a different energy than that of the excitation light.[27,44] This difference in energy, and therefore wavelength, makes it possible to effectively separate the excitation light from the fluorescence signal. Due to its unique capabilities for visualizing cells and sub-microscopic cellular components the technique has become ubiquitous throughout the biological sciences. Nowadays a wide range of fluorophores (molecules capable of fluorescing) are available and can be used to illuminate different structures and molecules in a targeted fashion.[16]

2.1 Two-photon calcium imaging

Across animal species action potentials, or spikes, play a dominant role in the encoding and transmission of information in the central nervous system. Recordings of spiking activity are therefore one of the most valuable means to learn about the inner workings of the brain. Such recordings were first acquired in 1939 by Hodgkin and Huxley, when they measured the membrane potential in the giant squid axon using glass electrodes.[24] The methodology to perform electrophysiology, i.e. of using electrodes to directly measure the membrane potential of neurons, has undergone considerable development since then. Using multi-electrode arrays, it is today possible to measure the activity of small populations of cells simultaneously (see Fig. 1.1). However, the total number of simultaneously recorded neurons remains limited and the method is highly invasive. An altogether different approach is the use of fluorescence microscopy to monitor the intra-cellular calcium concentration. Within neurons, voltage-gated calcium channels open whenever an action potential occurs. This causes a sudden influx of calcium into the cell. These transient changes in can be measured using specifically designed calcium indicators, which change their fluorescent properties when they bind to calcium ions. The development of calcium indicators began as early as 1962[74] and their sensitivity and applicability have steadily improved since then. Compared to classical approaches for the recording of neural activity based on electrodes, calcium imaging allows for a much higher number of neurons to be observed simultaneously (Fig. 1.1). It is therefore extensively used for to analyze the activity of whole populations of neurons of animals in vivo. Another important innovation that drastically improved the performance of calcium imaging was two-photon excitation microscopy (2PE). In 2PE, two coincident photons are needed for an electron to bridge the energy gap and spawn a fluorescent photon. This increases the resolution of the images and allows for imaging in deeper tissue layers while avoiding tissue damage.[60] The data recorded in calcium imaging experiments consists of videos showing the cells bodies that "light up" when an action potential occurs (see Fig. 2.1). The analysis of such videos is usually a two-step process. First, each neuron is identified and segmented. The brightness of each pixel assigned to a neuron is then added up for each frame, resulting in an indirect measurement of the calcium concentration in the cell across time. In the second step, the time traces are analyzed to extract a more immediate readout of the neuronal activity, e.g. a continuous firing rate estimate or discrete spike times. While both steps are interesting problems, I focused on the second task. This means that we assume the spatial separation was successfully executed and we are provided with one-dimensional time traces for each neuron.

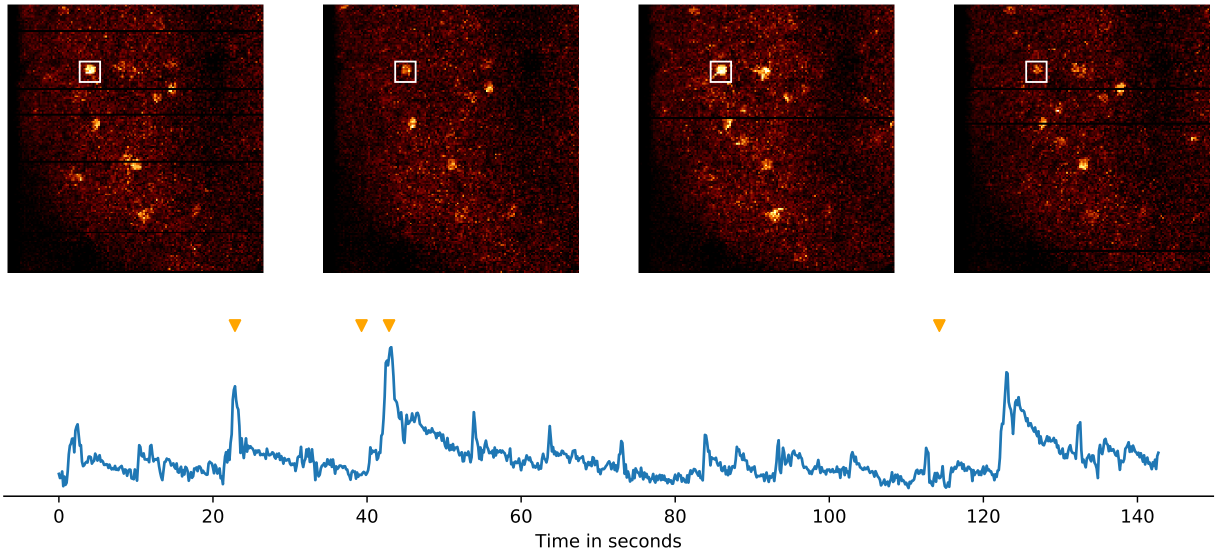


Figure 2.1: Calcium imaging data. Top: Four frames from a dataset of two-photon recordings showing a neuronal population expressing GCaMP6s. The white box indicates a manual segmentation of a neuron. Bottom: Time trace of the summed fluorescence from the area indicated by the white box. Orange arrows indicate the time points corresponding to the four frames in the top row.

A part of such a time trace is shown in Fig. 2.2 . It was recorded in vivo in the mouse visual cortex using the genetically encoded calcium indicator GCaMP6s.[12,81] This data was specifically recorded to measure the exact relationship between observed fluorescence signal and spiking activity. In parallel to the calcium imaging experiment, electrophysiological measurements of the membrane potential using loose-seal cell-attached electrodes were carried out. Such data is very valuable as it provides ground-truth information on the spike timings. The example illustrates the task of spike inference from calcium imaging and some of the properties which make it a challenging problem: An isolated spike causes a sharp increase in fluorescence, followed by a slower decline back to the baseline. For multiple spikes the dynamics are highly nonlinear. A quick succession of spiking events causes a much larger response than the linear addition of single-spike responses, an effect called facilitation. We can also observe saturation, which means that there is a maximum amount of fluorescence which can not be surpassed regardless of the spike rate. Oftentimes, background activity in the form of transients in fluorescence on long timescales further complicates the analysis. Besides the nonlinearity of the observed behaviour and background activity, another factor which makes spike inference so difficult is the heterogeneity of the data, both across and within datasets. The amplitude of the transient caused by a spike, its rise time and the degree of nonlinearity in the dynamics are properties of the specific calcium dye used. They determine our ability to identify spikes to a critical degree and novel dyes are constantly being developed to optimize these features.[45] However, these parameters, as well as the average firing rates, also vary considerably between individual cells. Finally, the experimental conditions, such as whether the experiment is carried out in vitro or in vivo, how deep in the tissue the measurement takes place, or how large the field of view is (which in turn determines the imaging rate) also have considerable impact on the collected data.

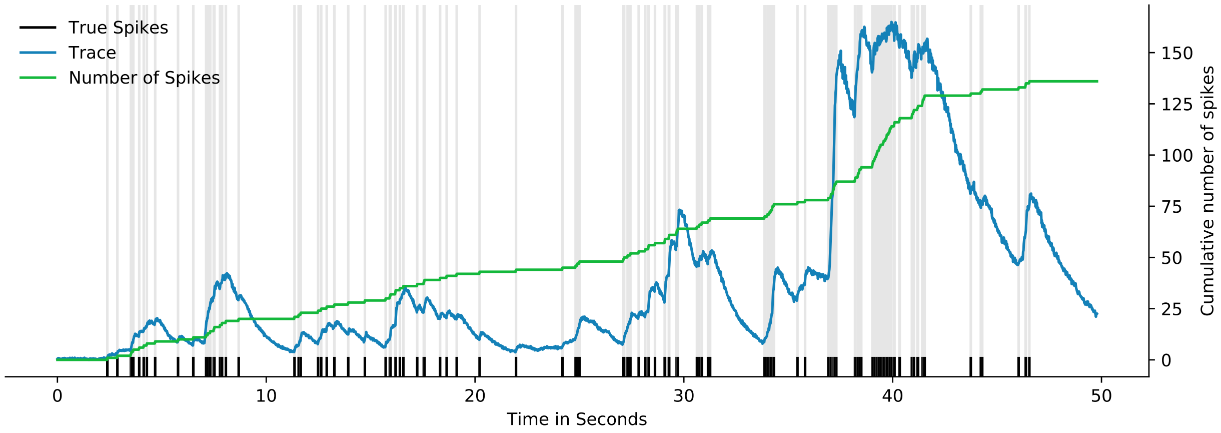


Figure 2.2: Segment of a fluorescence trace recorded in the mouse visual cortex using GCaMP6s. Electrophysiologically measured spike timings are shown in black. The cumulative number of spikes is drawn in green.

2.2 Single molecule localization microscopy

Like with other forms of light microscopy, the resolution that can be achieved in fluorescence microscopy is limited by the physical properties of light itself. When looking at an arbitrarily small light source through a microscope, a fraction of that light is collected by the objective and focused in the imaging plane. However, due to inference of the light waves at the focal point what we observe is not an infinitely small point but a point spread function (PSF) with a size on the order of roughly half the wavelength of the fluorescent light. When two such light sources are to close to each other, their images merge into each other and they can no longer be distinguished. This diffraction limit prevents us from resolving subcellular structures using conventional microscopy. Crucially though, a single PSF can be located very precisely. To illustrate this core principle we can look at simulations. Using an intensity profile of a typical PSF and a noise model that accounts for the random number of emitted photons and the camera properties we create synthetic images of a single fluorescent spot. If we run a well calibrated localization algorithm on these images, we will obtain localizations which are distributed around the ground truth position. Figure 2.3a shows one such image, the ground truth position of the simulated fluorophore and 130 localizations. It is apparent that the localizations are centered in an area which is much smaller than the corresponding PSF. As subpixel shifts of the position are reflected in the asymmetry of recorded intensity values, the precision is not limited by the pixel size nor the wavelength. Single-molecule localization microscopy makes use of this fact by activating a random small subset of all present fluorophores at a time and then recording a long series of images (usually in the range of one thousand to one hundred thousand) showing isolated spots. These spots can then be localized with high precision and recombined to form an image of the specimen with a resolution that is much higher than what the classical limit would allow for (see Figure 2.3b). There are different methods to achieve the switching of fluorophores between dark OFF and bright ON states. For example, fluorescence photoactivated localization microscopy (PALM)[7] uses proteins that can be activated with UV light, while stochastic optical reconstruction microscopy (STORM)[70] relies on suitable buffers to control the switching behavior. Point accumulation in nanoscale topography (PAINT)[73] is not based on photoswitching at all: instead fluorophores are imaged when they temporarily bind to a target. Additionally, various optical systems are able to modify the shape of the PSF depending on the position of the emitter relative to the focal plane of the microscopy. This allows for the localization of emitters in three dimensions. For an extensive review of different SMLM techniques see Lelek et al..[41]

Independent of the exact technical choices, the central task when analyzing SMLM data is the exact detection and localization of PSFs.

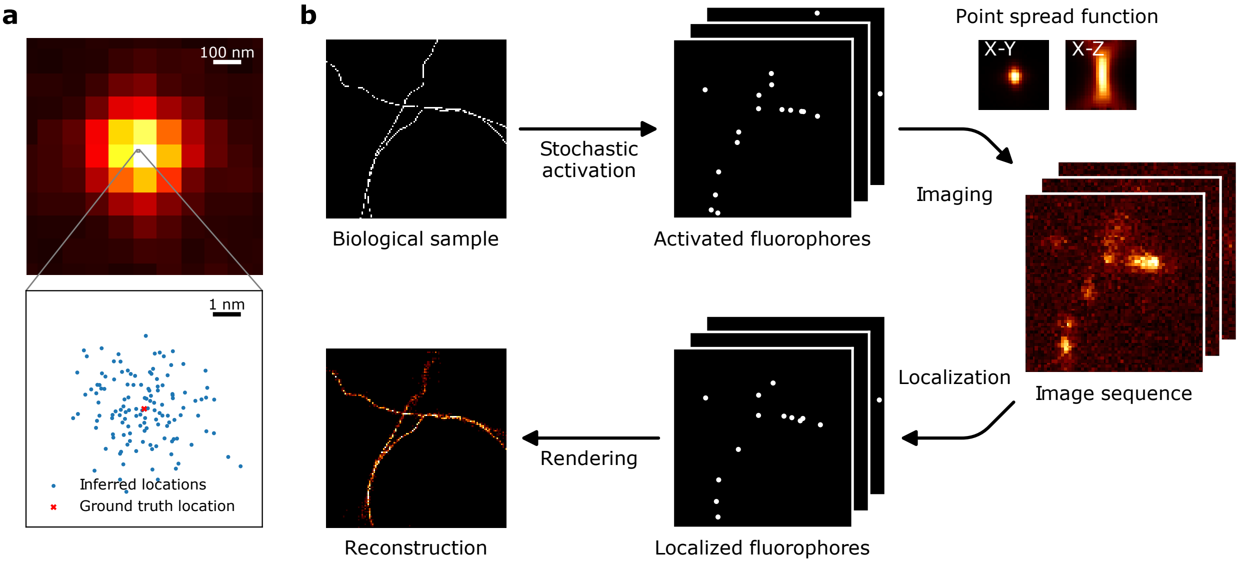


Figure 2.3: SMLM a) Localization precision of isolated PSFs. Using an empirical PSF model we generated a typical spatial intensity distribution of an activated emitter. We then sampled 130 images from the noise model which includes the statistics of the number of photons hitting each pixel and the camera properties. The upper image shows one such sample. Running a localization algorithm on the 130 images results in localizations that are tightly clustered around the true location. **b)** Principle of SMLM. Fluorophores are stochastically activated and recorded using fluorescence microscopy. A localization algorithm infers the underlying sources from noisy and blurred imaging measurements. Rendering methods turn inferred sources into an estimate of the underlying structure.

The difficulty of this task is mainly driven by the signal-to-noise ratio (SNR) and the density of the emitters. While the SNR is mostly determined by the microscopy setup and the SMLM technique used for the experiment, the emitter density can be directly controlled over a large range by adjusting the intensity of the activation light. Fig. 2.4 shows example frames and reconstructions from an experiment where the same structure was imaged multiple times at different emitter densities. It makes sense that a simple procedure that first detects emitters using a peak finding approach and then fits a model of the PSF to each spot would give good results in the low density regime, while it might struggle for higher densities where PSFs regularly overlap. Because a certain overall number of localizations is needed to obtain a contiguous image, low densities require a higher number of recorded images, and therefore lead to longer experiments. When setting the activation density the experimenter therefore has to trade off the achievable localization precision against the imaging time. Long imaging times are at best inconvenient and at worst prohibiting for certain applications, for example when imaging moving specimens. Algorithms that can correctly identify emitters in dense configurations where their PSFs regularly overlap are therefore highly sought after.

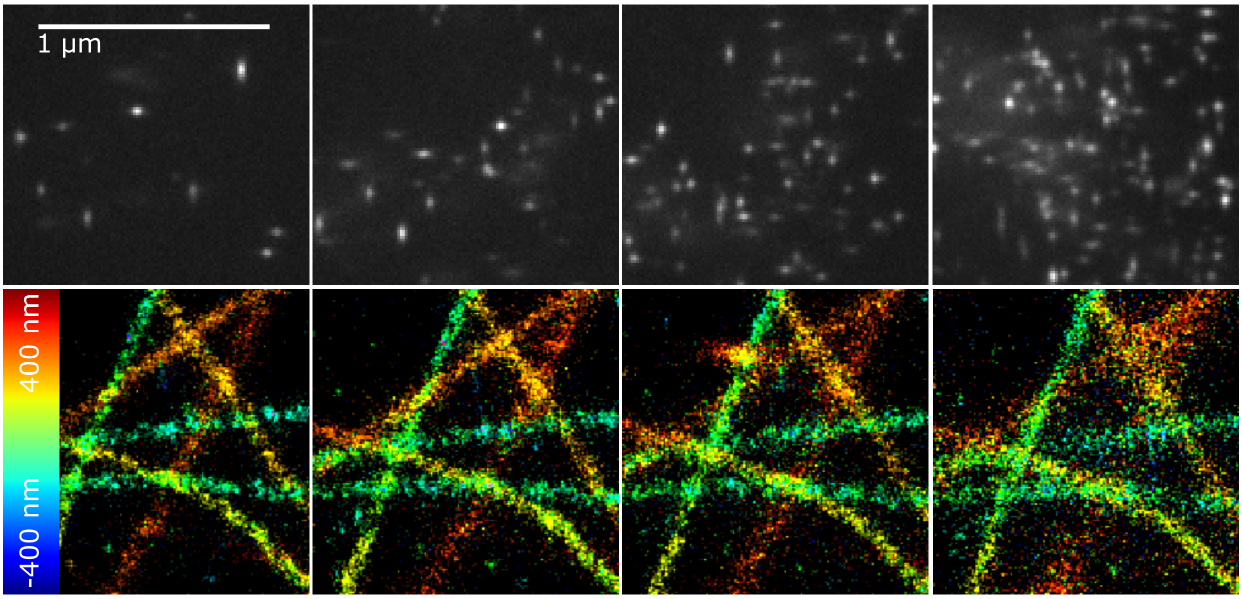


Figure 2.4: Effect of emitter density on reconstruction quality Microtubules were repeatedly imaged at four different emitter densities using STORM microscopy. Upper row shows representative images from each dataset. Lower row shows reconstructions obtained with an iterative localization method (CSpline[4]). Colorcoding indicates inferred axial location.

.

Chapter 3

Inference methods

In this chapter the problem setting for spike inference (SI) and single molecule localization (SML) is formalized and relevant notation introduced. Afterwards, different approaches that have been employed for both problems are introduced and discussed. The problem is formulated as an attempt to perform statistical inference using probabilistic modeling.[6] It should be noted that not all algorithmic approaches work within this framework. For example, a peak-finding method could serve as a heuristic approach for identifying spikes or emitters, without explicitly using a probabilistic model.

3.1 Problem setting

Both CI and SML, as well as many other problems in imaging, neuroscience and other disciplines share some crucial properties: We have observations $\mathbf{X}^* = \mathbf{x}_1^*, ..., \mathbf{x}_N^*$, which are samples from a probability distribution which spans the space of all possible observations $\mathcal{X}$. They are referred to as observations, recordings, or real data. This probability distribution p^* represents the true underlying data generating process. If we use CI as an example, where the observations are N time traces, p^* includes many factors: the underlying biology of neural activity, the properties of the calcium dye, the imaging properties of the microscope, as well as several steps of data-processing, e.g. neuron segmentation and signal normalization. In this scenario, and basically any other case of a real-world process, the true generative process is unknown. Instead we work with a parametrized observational model which tries to mimic p^*. The observational model is a probability density function $p(x|\phi)$ over the observational space i.e., $x \in \mathcal{X}$, where ϕ includes all the parameters for all components of the observational model (e.g. dye dynamics, microscope settings).

The two problems in this thesis share a more specific structure which should be reflected in the observational model. In both cases the the observations are directly linked to binary latents $z^* \in 0, 1$ which are the primary items of interest. The remaining parameters play an important role but only in so far as they help us to correctly infer z^*.

This structure can be easily expressed in our observational model: $p(x|\phi) = p_\theta(x|z)$ where we make the dependence on the latent variables explicit and introduce a new set of parameters θ that does not include z.

It is important to note that a parametric observational model will never perfectly match the true generative process, as we will not be able to capture the full complexity of the real world. This difference is called the model mismatch. The amount of model mismatch depends on the fidelity of our generative model and the parameters ϕ. We will now introduce the two observational models that are most commonly used to describe CI and SMLM data and are at the core of many of the algorithms introduced later.

Linear calcium dynamics model Observations from CI take the form of time traces, where each value f_t represents the spatially summed measured fluorescence at a certain time-point for a previously identified neuron. The calcium dynamics are described as a sequence of equally sized instant rises in calcium concentration at the spike times followed by an exponential decay

$$c_t = \gamma c_{t-1} + \delta s_t \tag{3.1}$$

with a decay constant $\gamma < 1$ and spike-amplitude δ. s_t is the number of spikes that the neuron fired at timestep t and corresponds to the latent variable of interest z, in this case. The fluorescence f_t is then simply c_t re-scaled by a factor α, with an added baseline β and additive measurement noise $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$:

$$f_t = \alpha c_t + \beta + \epsilon_t \tag{3.2}$$

This model was first described by Vogelstein et al[85] specifically as a observational model for calcium induced fluorescence. However, the idea of using a series of decaying exponentials to fit the signal was around even earlier.[30] We will refer to this model as SCF, since the generative process is described by the *s*pikes, the *c*alcium concentration and the *f*luorescence intensity.

Image formation model SMLM The observational model of SMLM data describes the purely physical processes of light diffraction and the image registration of a camera. These have been studied extensively and are very well understood. Observations are images I of a set of N activated fluorophores $S_n, n = 1, 2, ...N$. The fluorescent signal of these images can be modeled as the sum of their diffracted images, or PSFs, scaled by individual intensities.

$$I \sim \sum_{n}^{i} A_n S_n \tag{3.3}$$

where $A_n = \alpha_n \cdot \mathbf{PSF}(Z_n)$ is a matrix implementing convolution with a normalized PSF, down-sampling to the camera resolution and scaling by the number α_n of photons emitted by a fluorophore during the time the frame was recorded. Importantly, the exact shape of the PSF usually depends on the distance Z_n of the fluorophore to the focal plane of the microscope. This property can be used to perform localization in three dimensions. The recorded signal then depends on the exact noise model used to model a given camera and the assumed background. The observational models used for various SMLM inference methods mostly vary in the detail of their noise modeling and in the way the PSF is parametrized, In the simplest and most common case, the PSF is modeled as a 2D Gaussian with a single σ-parameter.[58] More elaborate approaches use Zernike polynomials[2] or splines.[4,43]

Both of these models are linear with respect to z. For example, in the SCF model the response to two concurrent spikes equals twice the response from a single spike. This property makes these models amenable for many different inference algorithms.

3.2 Inference

We want to find configurations of latent variables z that underlie our observations $\mathbf{x}^*$. Statistical inference relies on the likelihood function which gives us a relative measure of consistency for different settings of z. We obtain the likelihood of a given observation $\mathbf{x}_0^*$ by evaluating $\mathcal{L}_{\mathbf{x}_0^*}(z,\theta)$: $p_\theta(x = \mathbf{x}_0^*|z)$. While the absolute probability value is usually of little use, when comparing the likelihood values for different z, a higher likelihood value indicates a set of latents that is more consistent with the observation.

Following this line of reasoning, the maximum likelihood (ML) approach tries to identify the single point in latent parameter space that has the highest likelihood value for a given observation.

$$\hat{z}_{\theta,ML} = \arg\max_z p_\theta(x = \mathbf{x}_0^*|z) \tag{3.4}$$

As discussed below, ML is a widely used approach for the kind of problems we deal with in this thesis. In contrast, Bayesian inference targets the full distribution of latents $p(z|x)$. Therefore, when working as intended, it not only identifies parameter points that best explain our observation under the model, but also the uncertainty of the inferred values.

The principle of Bayesian inference is expressed in Bayes' formula:

$$p_\theta(z|x) = \frac{p_\theta(x|z)p(z)}{p(x)} \tag{3.5}$$

The posterior $p_\theta(z|x)$ is our inference target. It is proportional to the product of the likelihood $p(x|z)$ and the prior $p(z)$, and normalized by the marginal. The prior distribution encapsulates our prior belief about the parameter distribution. As an example, this could be the firing rates we expect to observe in the recorded neurons when performing spike inference. Intuitively, Bayes' formula states how to correctly update our prior beliefs after collecting evidence. The joint distribution $p(x,z) = p(x|z)p(z)$ is the full generative model as it allows us to draw samples $\mathbf{X} = \mathbf{x}_1, ..., \mathbf{x}_N$ (referred to as simulated data) [1].

While appealing in principle, a posterior distribution can only be analytically calculated using Bayes formula in rare cases where all terms involved belong to specific probability distributions. Oftentimes, the normalizing factor, which is the integral over all possible parameter settings $p(x) = \int_\theta p(x|z,)p(z)\,dz$, is intractable. In SML, for example, this would require integrating over all possible emitter arrangements which is a very large combinatorial space. We discuss two methods that can overcome this difficulty: Markov chain Monte Carlo and variational inference, below.

Somewhat in between ML and Bayesian inference lies the maximum a posteriori (MAP) estimation. Similar to ML we are looking for a point estimate, however instead of maximizing the likelihood, we target the mode of the posterior by maximizing

$$\hat{z}_{\theta,MAP} = \arg\max_z p_\theta(x = \mathbf{x}_0^*|z)p(z) \tag{3.6}$$

3.3 Variational inference

Variational inference (VI) is a method for Bayesian inference which lets us approximate the intractable posterior $p(z|x)$ introduced in 3.5. As our DeepSpike method is based on VI we discuss

[1] In the literature, oftentimes the likelihood is referred to as the generative model. In our definition, a generative model also includes a prior distribution.

it in more detail. The core idea is to approximate the true posterior with a parametrized density of our choosing i.e., $p(z|x) \approx q_\psi(z)$ (the proposal distribution) and optimize the parameters ψ to minimize the distance between the approximate and the true posterior. This requires a measurement of similarity between the two distributions that is amenable to optimization. Most commonly, the Kullback–Leibler divergence (KL-divergence) is used, which is defined as:

$$D_{KL}(q_\phi(z)||p_\theta(z|x)) = \int q_\phi(z) \log \frac{q_\phi(z)}{p_\theta(z|x)} dz \tag{3.7}$$

Rewriting $p_\theta(z|x) = p_\theta(x,z)/p_\theta(x)$ and reshuffling the terms we obtain:

$$D_{KL}(q_\phi(z)||p_\theta(z|x)) = \underbrace{\int q_\phi(z) \log \frac{q_\phi(z)}{p_\theta(x,z)} dz}_{-\mathcal{L}(\theta,\phi;x)} + \log p_\theta(x) \tag{3.8}$$

This leaves us with the compact relationship between the log marginal, also called log evidence $p_\theta(x)$, the KL-divergence and the evidence lower bound (ELBO) $\mathcal{L}(\theta, \phi; x)$:

$$\log p_\theta(x) = D_{KL}(q_\phi(z)||p_\theta(z|x)) + \mathcal{L}(\theta, \phi; x) \tag{3.9}$$

The KL-divergence is always positive and only vanishes if the two distributions are identical. Furthermore, $p_\theta(x)$ is independent of $q_\phi(z)$ and the ELBO is tractable if we chose $q_\phi(z)$ appropriately. Therefore, we have transformed the intractable problem of Bayesian inference into an optimization problem that lets us minimize the dissimilarity between $q_\phi(z)$ and $p_\theta(x|z)$ by maximizing the ELBO. Nevertheless, coming up with a parametric model $q_\phi(z)$ that is suitable for a given problem and for which we can obtain closed form gradients for optimization is laborious and often not possible. A much less restrictive approach is Black Box Variational Inference[65] where the expectation in the ELBO is approximated using samples from the proposal distribution. This allows us to obtain the necessary gradients as long as we can effectively evaluate $p_\theta(x, z)$ and draw samples from $q_\phi(z)$ which is a very mild constraint.

Sun et al. use VI to perform SML. Specifically they use outputs of another algorithm as initial estimates for the emitter number and positions and then refine them with VI. A unique characteristic of the algorithm is that when performing inference for a given frame, an estimate of the structure obtained from all other frames (global context) is used as a sparsifying prior. This improves performance in difficult conditions. One drawback is that the predictions lie on a super-resolved grid, which limits resolution. Gabitto et al.[20] apply VI on the output of any SML algorithm with the goal to infer the identities of emitters, grouping them across frames by precisely modeling the temporal dynamics of fluorophores. To my knowledge, DeepSpike was the first attempt to apply VI to the problem of SI.

3.4 Markov chain Monte Carlo (MCMC)

MCMC methods are a well established class of algorithms that allow sampling from a desired probability distribution. In our case this distribution is a constellation of discrete events in space and/or time. A Markov chain can be constructed by defining operations that add, remove or shift these events around. In the case of SI such an operation could be the addition of a spike event in a specific time bin. This new state would then be accepted or rejected with a probability that reflects how well the new setting (together with the generative model) explains the observation compared to the previous state. The parameters of the generative model can be treated in the same way, or instead be optimized in a more efficient manner between sampling steps (for example using MAP). MCMC is a widely used method that comes with a theoretical guarantee of eventually generating samples from the correct posterior. It is in principle also very flexible in regards to the generative model used. The biggest issue of the approach is the computational cost and slow sampling time, which can easily render it impractical depending on how it is implemented.

Pnevmatikakis et al. developed an MCMC algorithm for spike inference[62] which uses a linear generative model and is still slow. Greenberg et al.[21] used a sequential Monte Carlo (SMC) algorithm with an elaborate nonlinear generative model. However, even with a heavily optimized GPU implementation, the computation time is on the order of seconds for a single trace. Cox et al.[14] adopted MCMC for the problem of localization, modeling all fluorophores of an image sequence simultaneously with a generative model that includes blinking dynamics and bleaching. Again, this is computationally extremely expensive, requiring multiple hours to process a small dataset.

3.5 Nonnegative deconvolution

A much faster, but also more approximate approach is nonnegative deconvolution. The generative model is centered on a convolution of the latent variables z with a kernel function with an added noise variable ϵ.

$$z \circledast g + \epsilon = f \tag{3.10}$$

In the case of single molecule data, the kernel g would correspond to the PSF. For calcium imaging data, usually an instant rise in calcium concentration at spike time, followed by an exponential decay is assumed. Vogelstein et al.[84] derive the deconvolution method starting from the MAP objective. They note that optimizing this objective is intractable because "it requires a nonlinear search over an infinite number of possible spike trains". Therefore, they relax the constraint $z_t \in \mathbb{N}_0$ to $z_t >= 0$. That means the latents are assumed to be continuous instead of discrete. Such a model can be effectively optimized on recorded data using available optimization methods like the Newton-Raphson algorithm or Richardson-Lucy deconvolution.[68] However, the inferred latents cannot be directly interpreted as spikes or detected emitters anymore. The practitioner therefore either has to adapt the downstream analysis or threshold the outputs to obtain binary values. Another limiting factor is that the generative model is linear by definition. This is a serious drawback when analyzing calcium imaging data because, as described in section 2.1, many calcium dyes exhibit nonlinear properties.

Nevertheless, algorithms based on non-negative deconvolution[19,59,84] are extremely popular for spike inference due to their ease of application and speed. Deconvolution is also extensively used in the context of SMLM.[3,90] In this case the localization is usually performed on a sub-pixel grid, followed by a center-of-mass algorithm to obtain a final position estimate. The FALCON algorithm[50] uses these estimates to initialize a final step which optimizes the locations on a continuum.

3.6 Maximum-Likelihood Estimation

Maximum-likelihood estimation (MLE) is one of the most prominent approaches in SMLM. An initial heuristic detection step finds spots in the images which are treated as candidates for emitters. The exact continuous position coordinates in 2 or 3 dimensions as well as the intensity of the spot are then optimized to maximize the likelihood of the data under the generative model. The parameters describing the PSF are not optimized during inference, but instead determined beforehand using calibration datasets. For an isolated spot that was correctly detected, this approach achieves optimal performance, i.e. it reaches the Cramér–Rao lower bound (CRLB).[53] Different methods are used to carry out the initial detection step. Single-emitter algorithms focus on identifying well separated emitters in low density data. Emitter candidates are identified by extracting local maxima or by calculating the correlation between the image and a model PSF and thresholding the correlation image. A region of interest (ROI) around each candidate is then cut out from the original image and the MLE optimization is carried out under the assumption that the ROI includes a single emitter. Samples where this is not the case often result in lower likelihoods, and can be discarded. On the other hand, multi-emitter approaches try to detect all emitters in dense configurations with overlapping PSFs. To this end, steps of adding or removing localizations are alternated with MLE optimization of individual localizations. This procedure is repeated until a maximum number of iterations, or until some some criterion is reached that indicates a stable configuration.

For SI, the MLspike algorithm[15] uses MLE to find the spike train that provides the best explanation for a given fluorescence trace. It uses a variation of the Viterbi algorithm:[83] the likelihood for all possible calcium trajectories is evaluated going backwards in time, and the most likely trajectory chosen. Jumps in the inferred calcium trace are then read out as spike times. This approach works even for nonlinear generative models and the authors propose their own ’MLphys’ model which includes effects like facilitation and saturation. The parameters of the generative model are not part of the MLE procedure and are instead estimated beforehand using several heuristics.

Chapter 4

Deep Learning

Even though deep learning started to be used for more and more tasks since the seminal ImageNet paper in 2012,[36] it took some time until it was applied to the class of problems addressed in this thesis. This might be surprising given that these problems are typical instances of pattern recognition tasks at which deep learning usually excels. One reason is that deep learning usually requires large amounts of labeled data, i.e. pairs of recorded data and ground truth information, for network training. In the case of calcium imaging, such labeled data is hard to obtain and therefore scarce. In SMLM such ground truth labels don't exist altogether. Current deep learning approaches therefore either try to optimally use the limited amount of available labeled data (for calcium imaging), generate training data using a generative model or use unsupervised methods like variational autoencoders. I will discuss these approaches in turn.

4.1 Supervised learning

Supervised learning refers to the optimization of network parameters by back-propagating errors between network outputs and ground truth labels. When enough labeled data is available it can be regarded as the go-to approach for deep learning. As mentioned before obtaining labeled data for CI is very laborious though. Berens et al.[5] assembled a dataset of labeled CI data and designed a benchmarking challenge to compare different available SI methods. To that end, the dataset was split into a training set including electrophysiologically measured spike times and a test set. Several submissions [1] trained different deep neural network (DNN) architectures on the training data and achieved high overall scores competitive with the MLE-based state-of-the-art algorithm MLspike.[15] However, it should be noted that the quality of the dataset puts certain limitations on the analysis: Some of the time-traces contained large recording artifacts and the SNRs were often very low. This resulted in many algorithms achieving similar (low) performance numbers, and makes it difficult to draw conclusions about their relative strength. It was also unclear how the deep learning approaches could be adopted to a completely different dataset for which no labeled data is available. Rupprecht et al.[69] tackled this problem very recently by collecting a large database of labeled CI data specifically with the goal of training DNNs for SI. To ensure that their method also performed on data from experiments for which no ground truth information was available, they developed a procedure for re-sampling their labeled dataset with the sampling rate and noise level of the target data, so that they could retrain a network on data with similar statistics.

[1] Including a version of our DeepSpike network which we trained in a supervised fashion

4.2 Simulator learning

Especially for SML, the data generating process is well understood.[71,78] This allows us to design realistic generative models $p_\theta(x, z)$. Data sampled from such a model can act as a surrogate for labeled observations for the purpose of network training. To this end we take samples $z \sim p(z), x \sim p_\theta(x|z)$ and optimize the log-likelihood $\log q_\psi(z|x)$. This procedure minimizes the *forward* KL-divergence between the posterior of the generative model and the recognition network $D_{KL}(p_\theta(z|x)||q_\phi(z))$ where we take the expectation over the simulated data distribution.[46]

Multiple groups developed deep learning based SML approaches[10,54] concurrently to our work on DECODE. We will discuss the major differences in chapter 5. More specialized methods use deep learning to extract parameters describing single isolated emitters such as color, emitter orientation, z coordinate, background or aberrations.[31,52,86,88]

4.3 Autoencoder learning

The variational autoencoder (VAE)[32,67] adopts VI for the task of neural network training. Instead of optimizing $q_\phi(z)$ for each observation $\mathbf{x}^*$ separately, a network is trained which takes observations x as input and infers latents z conditional on x. This network, usually called the recognition network, parametrizes a distribution of the form $q_\phi(z|x)$.

We can obtain a useful identity by rewriting 3.9 and replacing $q_\phi(z)$ with $q_\phi(z|x)$:

$$\mathcal{L}(\theta, \phi; x) = \log p_\theta(x|z) - D_{KL}(q_\phi(z|x)||p(z)) \tag{4.1}$$

The name "variational autoencoder" highlights the similarity to deterministic autoencoders,[35] which becomes apparent in equation above. Both methods attempt to find encodings of unlabeled data that allow to reconstruct the input from the encoding. Optimizing the ELBO maximizes the likelihood $p_\theta(x|z)$, which is similar to minimizing a reconstruction error. The main difference between deterministic and variational autoencoders is that in the latter distributions over latents are inferred (instead of point estimates) and that the loss function not only rewards good reconstructions but also tries to minimize the dissimilarity between the posterior estimate $q_\phi(z|x)$ and the prior $p(z)$.

VAEs allow for amortized inference: once the network is trained, inference on an observation amounts to a single forward pass through the network, which is cheap. Usually, for example when VAEs are trained on natural images, in parallel to the recognition network, a second network which parameterizes the likelihood $p_\theta(x|z)$ is trained. In my methods for SI and SML the observation model instead takes the form of an explicit parametric model as introduced in 3.1 with much fewer parameters.

It should be noted that the gradients of the ELBO with respect to ϕ can not be evaluated in a straightforward manner when the latents are discrete, as it is the case in SI and SML. Various methods to train such models on multi-sample variations of the ELBO are discussed in Le et al..[38] We here include any such algorithm as performing autoencoder learning. Fig. 4.1 graphically represents the training loops for simulator learning and autoencoder learning using SML as an example.

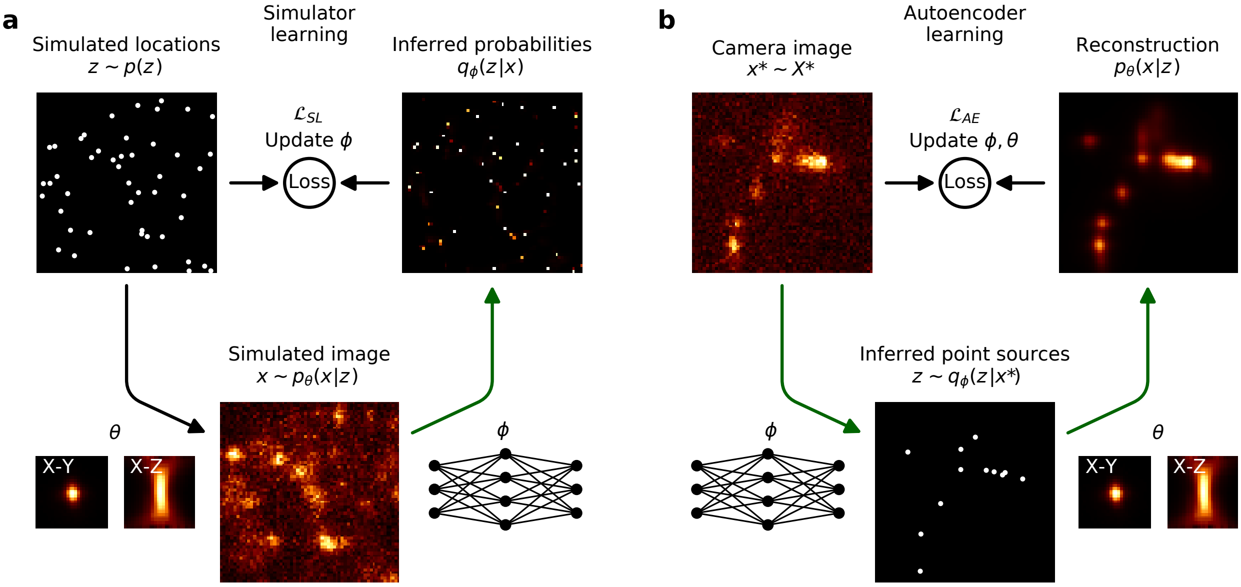


Figure 4.1: Network training approaches on SMLM data. a) Simulator learning. Synthetic images are constructed by the simulated imaging of randomly located fluorophore point sources using a generative model, and a network is trained to detect and localize the fluorophores using supervised learning. **b)** Autoencoder learning. A neural network is used to infer putative locations from a measured camera image, and subsequently the generative model is used to reconstruct the original camera image. Both the parameters of the generative model and of the DECODE network are optimized. The loss is computed between measured and reconstructed images.

4.4 Influence of model mismatch on NN training

During my work on the DECODE algorithm, I extensively tested and compared different versions of autoencoder learning, simulator learning as well as combinations of both. In the final version of the algorithm, simulator learning is used to train the network and for the sake of clarity, the publication contains no reference to VAE learning. I therefore want to use the opportunity to summarize the findings of this comparative work here as it might be of value for researchers who have to make similar choices in different problem settings.

As a first step, it is important to note that if we have access to the true generative model, simulator learning sets an upper bound to the achievable performance for a given network architecture and optimization procedure. In this case, simulator learning corresponds to supervised learning with an amount of labeled data that is only limited by the available computational resources.

In such a scenario, autoencoder learning is unnecessary and even counterproductive for multiple reasons: Training is much slower because it requires evaluations of the generative model for each sample from the recognition network (up to 50 in my experiments); this process is also memory intensive and can quickly exhaust the available resources of even modern GPUs. Furthermore, the final performance achieved is usually lower, since the gradients on the network weights obtained by autoencoder learning are generally noisier than those from simulator learning. This effect is especially severe because we are dealing with discrete latent variables and cannot make use of the reparametrization trick,[33] which facilitates learning in VAEs with continuous latents.

Many methods to reduce gradient variance and bias when training VAEs with discrete variables have been proposed. However, my own experiments with different VAE variants, specifically with reweighted wake sleep (using only the wake updates),[9] VIMCO[51] and the thermodynamical variational objective[48] have yielded very similar results. Nevertheless, it is possible that future developments will alleviate this issue.[11]

Noisy gradients not only lead to reduced performance after convergence, but also make the training process less stable (which can lead to failed runs) as well as sensitive to hyperparameters and random seeding. Finally, since real data (even unlabeled) is harder to obtain than simulations, VAEs can overfit to a specific dataset.

After these deliberations we return to the real world where, as we have noted before, we will never have access to the true data generating process. As our generative models will always be a simplifying approximation, a varying amount of model mismatch is always present. It is therefore important to analyze how autoencoder and simulator learning behave in the presence of model mismatch. To this end, I devised an experiment using simulated SMLM data, where the amount of model mismatch could be directly controlled. Specifically, we would train networks using a generative model with a circular Gaussian PSF and evaluate their performance on datasets generated with an elliptical PSF and different degrees of ellipticity. All other parts of the generative model were the same as for the model used to generate the test data.

Let us consider how simulator learning is expected to behave in such a scenario: It is not obvious how a network that during the training process has only 'seen' circular PSFs, responds to data that is made up of elliptical PSFs. In principle, given that neural networks are generally seen as universal function approximators, one could conceive a network that performs optimally on circular PSFs and completely fails to recognize elliptical ones. In practice, what we observe instead is a reduction of performance that scales with the amount of mismatch between training and test data. It is likely that similar principles apply in this case as in the more common scenario where networks are trained on limited amounts of labeled observations and tested on data that was not part of the training set. There is a vast literature on the generalization of neural networks,[55] but questions about why and how well networks generalize have not yet been conclusively answered.[87]

VAE learning could have several advantages in this scenario. The network would be trained on data from the true data distribution and we would expect that it would learn to localize elliptical PSFs even if the PSF shape is mismatched. The reason is that minimizing the reconstruction error would still require placing the emitters at the center of the PSF. Furthermore, autoencoder learning also lets us optimize the model parameters θ. Therefore, as long as our generative model is flexible enough this would enable reducing the model mismatch.

It is also possible to combine simulator and autoencoder learning, for example by alternating between both objectives during training. We call this combined learning, of which reweighted wake sleep[9,38] is one variant. Such an approach could combine the advantages from both learning methods, giving us the low variance gradients in the simulator phase, and the ability to train the generative model in the autoencoder phase.

As shown in Fig. 4.2 our experiments corroborate these intuitions. Here the lateral efficiency, a performance metric (higher is better) that is calculated from the detection accuracy and the localization precision [2], is plotted over the ellipticity (or aspect ratio) of the simulated PSF. For an ellipticity of 1.0($\implies$ circular PSFs) the two models used for model training and to generate observations are identical. In this case, simulator learning clearly outperforms autoencoder learning. However, simulator learning is also more sensitive to model mismatch, with the performance dropping sharply as ellipticity increases. We also observe that combined learning unites the advantage of both approaches, reaching the high performance of simulator learning for small model

[2] This measure was devised to compare performance with a single number, see [71] for details

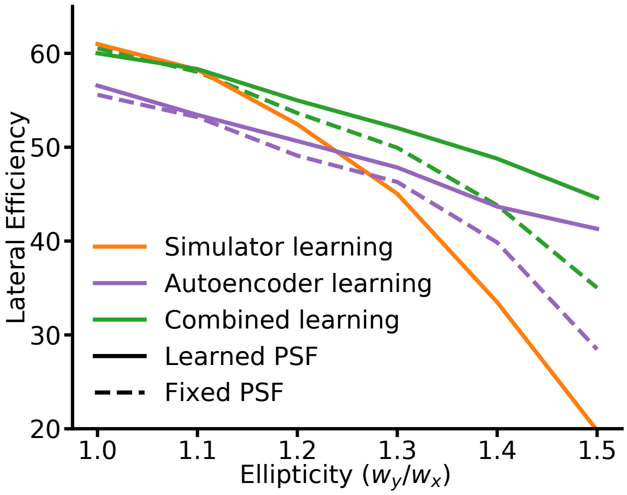


Figure 4.2: Performance of different training methods for different degrees of model mismatch. Models using a circular PSF are fit to datasets simulated from PSFs with varying ellipticity. PSF parameters for autoencoder and combined learning could be either fixed (solid line) or learned (dashed line).

mismatch, while also being more robust to increasing model mismatch. Furthermore, it is evident from the autoencoder and combined learning performance that learning the model parameters can further improve performance at high mismatch values.

While these results suggest that combined learning with learned generative model parameters is the method of choice, the optimal strategy will likely vary for different problem settings. For example, when testing our SI model, we did not observe the same performance gap between autoencoder and simulator learning when comparing them on simulated data generated with the model used for training. This suggest that the inference task is easier and less sensitive to gradient noise. The fact that the observations are one dimensional might also play a role as this allows us to employ higher batch and sample sizes. Another consideration is that while attractive in theory, learning the generative model parameters often introduces considerable difficulties in practice: mainly, the reconstruction loss can generally be minimized using different constellations of latents z and generative model parameters θ. For example, let us assume we use the linear observation model introduced in 3.1 and observe a transient that was caused by a single spike with a spike amplitude of $\delta = 10$. The same transient could just as well be explained using 10 spikes with an amplitude of 1. The model might even try to fit the noise using a large number spikes with tiny amplitudes. While such degeneracy should in theory be prevented by setting appropriate priors on the spike rate and the model parameters, in practice this is often insufficient and extensive tuning of hyperparameters might be necessary to stabilize training.

In conclusion, choosing the optimal method for network training will always depend on the specifics of the task at hand. Critical properties to consider are the expected amount of model mismatch, whether it is desirable and possible to learn the parameters of the forward model, how sensitive the network performance is to gradient noise and what the resource constraints are especially with respect to GPU memory demand.

Chapter 5

Publications

The two publications that are the basis for this dissertation describe how deep neural networks can be applied to perform inference of discrete variables to analyze data from two widespread microscopy methods. In both publications, we develop a training approach that uses a biophysical generative model to train the network without labeled data. However, the exact optimization strategy and network design vary considerably. For each publication, I will briefly summarize the specific adaptations made for the corresponding application, discuss their impact and outline my personal contribution. The full publications are included in the appendix 6.

5.1 Fast amortized inference of neural activity from calcium imaging data with variational autoencoders

This paper was presented at the 31st Conference on Neural Information Processing Systems and published in the conference proceedings.[77] It was one of the 697 out of 3240 total submissions to be accepted. Additionally, it was selected as one of 112 papers for a spotlight presentation.

Spike inference is an important step in the analysis of CI data. It also an extremely difficult problem, and fundamental questions regarding the right methodology, and how to interpret the results are hotly debated.[17,25] Our goal was to develop a deep learning approach for the analysis of CI data that could overcome some of the shortcomings of existing methods. Specifically, we wanted an algorithm that would be fast at test time and produce posteriors over spike trains. To not be constrained by the availability of labeled data, we optimized the network in an unsupervised manner using a generative model. At the same time, we wanted the training method to be flexible with respect to the generative model, i.e. not tuned to a specific generative model. With these goals in mind, we developed DeepSpike, a deep neural network trained on CI data using autoencoder learning. When trained on recordings from multiple cells, one common inference network is trained for all cells, while the generative model parameters are optimized individually for each cell.

We trained and evaluated DeepSpike using three different generative models: the widely used linear exponential model introduced in 3.1, a simple non-linear model which includes facilitation and saturation, and a more detailed model from the literature.[15] On data simulated with the linear model, we attained the same performance as deconvolution[63] and MCMC[62] algorithms. As expected, when working with real data recorded with the genetically encoded GCaMP6 dyes, DeepSpike trained with a nonlinear generative model performed much better and achieved state of the art results when evaluated with common performance metrics. Using a recurrent neural network approach, we obtained a correlated posterior that enabled us to sample realistic spike trains. While network training took several hours, inference at test time was very fast. When run on a GPU, time traces for hundreds of cells could be analyzed in seconds, which allows for the analysis of large populations of cells and for real-time applications.

We also emphasized that spike inference has many similarities with other analysis problems in biological imaging, where a sparse signal needs to be inferred from imaging data, and knowledge about the image-formation process can be used to perform statistical inference. Therefore, we predicted that the approach could find application in other inference tasks. As a concrete example of generalization, we proposed an extension of the algorithm to multi-dimensional inference of inputs from dendritic imaging data, and illustrated it on simulated data.

Given that more than four years have passed since the publication of DeepSpike, we can assess its impact and contributions. To the best of my knowledge, this work was the first instance of training a VAE with a parametric biophyisical generative model, with the purpose to infer latent variables that correspond to actual underlying events (though in a similar vein, Jiminez et al.[28] replace the generative neural network with a 3D renderer). This is a considerable step from the usual application of VAEs as a means to find arbitrary and condensed latent representations of data with the primary goal to produce realistic samples. This approach was afterwards applied to multiple other problems: Kirschbaum et. al[34] used it to infer neuronal motifs from CI videos; Hurwitz et al.[26] used it to localize spike sources in extracellular recordings of action potentials; Prince et al.[64] attempted to directly infer latent dynamics from CI videos.

On the downside, our method was not adopted as a tool for the analysis of CI data by the research community. The main reason is likely a lack of usability. Without a background in machine learning, the method is hard to use, especially as the training procedure is sensitive to the initialization of the generative model and hyperparameters. Due to the heterogeneity of the data, this is a problem that plagues any inference algorithm. However, it is especially acute for DNN approaches because each attempt to alter the settings requires retraining of the network. Possible improvements would therefore include heuristic procedures to automatically determine good initial parameter values.

One relevant and recent development in the application of deep learning to the task of spike inference is the CASCADE algorithm.[69] Instead of relying on unsupervised training methods to circumvent the lack of labeled data, the authors systematically collected a large data base of simultaneous calcium and electrophysiological recordings to cover a wide range of experimental settings and cell types. For unseen data, the ground truth data is resampled to match the respective sampling rate and noise level. This is a promising approach as it significantly facilitates network training for the end user. However, it seems that continuous integration of novel datasets would be necessary to maintain the applicability into the future when novel calcium dyes become available. Regrettably, the authors of CASCADE and other deep learning approaches[89] failed to build on our method for training end-to-end networks that produce posteriors over spike trains. Instead their networks output spike-rate estimates that can be transformed into discrete spikes using various heuristic post-processing methods.

Finally, I want to mention two advances that could directly be used to improve DeepSpike. Greenberg et al.[21] developed an elaborate biophysical model of calcium binding kinetics for the GCaMP6s indicator, and determined ranges or fixed values for all its parameters. A more precise generative would obviously benefit DeepSpike. Furthermore, it would be worthwhile to investigate if this model could be used for simulator learning.

Another exciting deep learning method is DeepCAD.[42] Based on the principle of self-supervised learning, it trains networks that can significantly increase the SNR of arbitrary calcium imaging recordings. Such preprocessing would benefit any SI method and could be especially advantageous for DeepSpike as it would facilitate the training process.

Author contributions

The publication is co-authored with Jinyao Yan, Evan Archer, Lars Buesing, Srinivas C.Turaga and Jakob H. Macke. The initial idea to apply VAEs to the problem of spike inference came from Lars Buesing and was further elaborated by Evan Archer, Srinivas Turaga and Jakob Macke. Jinyao Yan worked on the inference of somatic spikes and synaptic input spikes from dendritic imaging data. I had the idea for the correlated posterior network, implemented the code for the algorithm, and carried out all remaining experiments and prepared all figures in the manuscript except figure 4. Jakob Macke, Srinivas Turaga and I wrote and revised the manuscript. Special thanks goes to David Greenberg who proposed one of the nonlinear generative models we used.

5.2 Deep learning enables fast and dense single-molecule localization with high accuracy

The paper was published in the October 2021 issue of Nature Methods, which is the premier journal in the category of "Biochemical Research Methods". Two earlier versions of this work were made available as preprints.[76] These describe previous iterations of the algorithm that still made use of autoencoder learning and and also include the experiments detailed in section 4.4

The starting point for this project was the idea that the methods we developed for spike inference could also be applied to the problem of SMLM. We realized quickly that deep neural network (DNN)s are uniquely capable of resolving dense data because, unlike traditional algorithms, they are able to perform detection and localization simultaneously.

As discussed in section 4.4, we extensively compared different training strategies for this task. Our final algorithm relies solely on simulator learning because our generative model is able to simulate images that are very similar to recorded data. That means there is very little model mismatch and the possible benefits of autoencoder learning are not worth the additional overhead. Furthermore, 3D SMLM relies on microscopes that distort the PSF as a function of the axial offset from the focal plane. The exact relationship between the PSF shape and the axial position can not be learned from unlabeled data. We instead rely on so-called bead stacks: recordings of isolated, very bright emitters across the valid axial range at fixed and known intervals. Such bead stacks allow us to fit our PSF models with high fidelity, and therefor eliminate the need for further optimization during training.

Two other groups developed deep learning algorithms based on simulator learning for SMLM, concurrently to our work.[10,54] However, DECODE has several distinct features that set it apart from these methods. The major differences lie in the output representation, the loss function and the generative model.

Developing a network architecture that takes 2D images a input and produces a set of 3D coordinates for a previously unknown number of emitters is a nontrivial task. We solved this problem by designing an output representation that uses information from multiple channels to construct each localization. Specifically, a detection channel indicates the probability that an emitter is present within a pixel, three offset channels point to the exact continuous position within the pixel and three additional channels output the individual uncertainties in each dimension. Unlike DeepLoco[10] we maintain the local correspondence between the PSFs in the input image and the outputs. This facilitates training, since the network does not have to decide for an arbitrary ordering.

Furthermore, our network is fully convolutional which allows for the straightforward evaluation of datasets with different image sizes. DeepSTORM3D[54] also uses a fully convolutional network, but their networks predict the emitter density on a super-resolved grid. This fundamentally limits the achievable resolution. We showed that our architecture allows us to achieve the theoretical optimal performance on isolated emitters. This is a remarkable result for a network trained on dense data.

Our loss function is based on fitting a Gaussian mixture model to the ground truth locations and allows for the simultaneous optimization of detection, localization, and uncertainty estimation. Several experiments show that our uncertainty estimates are well calibrated and can be used to remove bad localizations or to optimize the rendering procedure.

Activated fluorophores are often visible over multiple adjacent frames. While the potential payoff for utilizing temporal information was well known,[17] most methods did not exploit it. Our generative model accounts for the temporal dynamics of fluorophore activation. Together with a network architecture that takes multiple adjacent images as input, this allows us to achieve better detection accuracy and to reduce localization error by up to a factor of two.

Extensive evaluation on simulated data and a diverse set of experimental data shows that our approach outperforms previous algorithms substantially and enables much faster imaging. For example, on a public benchmark challenge that evaluated the performance on sophisticated simulated datasets,[71] DECODE outperforms 39 other algorithms on 12 out of 12 datasets that covered different microscope setups and emitter densities.

A lot of effort was taken to make the method easily usable by the entire research community. To this end we created code that is easy to install, and provided detailed documentation and multiple tutorials. This effort seems to have paid off, as multiple groups have successfully applied DECODE to their data already.

Author contributions

The publication is co-authored with Lucas-Raphael Müller (who contributed equally), Philipp Hoess, Ulf Matti, Christopher J. Obara, Wesley R. Legant, Anna Kreshuk, Jakob H. Macke, Jonas Ries and Srinivas C. Turaga. The idea to adopt our method for spike inference to the problem of SMLM came from Srinivas Turaga. Philip Hoess and Ulf Matti recorded the data shown in Figure 4. Wesley Legant and Christopher Obara helped with the analysis of the lattice light-sheet dataset shown in Figure 5 of the paper. Lucas-Raphael Müller, under the supervision of Jonas Ries and Anna Kreshuk, implemented the public software package and performed the analysis of the datasets shown in Figure 4d-h and part of the analysis shown in Figure 2. Jakob Macke and Srinivas Turaga and I developed the algorithm. I implemented an initial version of the algorithm and carried out the analysis of the challenge dataset (Figure 3),the lattice light sheet data (Figure 6), the comparison in Figure 4a and the comparisons in Figure 2. Jakob Macke, Srinivas Turaga, Jonas Ries, Lucas-Raphael Müller and I wrote and revised the manuscript. Jonas Ries and I prepared all main figures.

Chapter 6

Conclusion

Modern imaging methods generate datasets that give us unprecedented insight into the inner workings of biological systems. While it is obvious how these advancements are driven by new developments in microscopy, the impact of novel computational analysis algorithms should not be underestimated. Such algorithms can not only qualitatively improve the image fidelity and the inference of latent variables, but facilitate the imaging process itself and open up new avenues. Deep learning will likely play a dominant role in the field of bio-image analyses. This prediction is not only supported by the success deep learning had across other disciplines, but also by the abundance of recent research in the field and the growing attention it receives.[37,49] This thesis showed how DNNs can be trained to infer discrete latent events from imaging data without relying on labeled data by making use of prior knowledge about the data generating process. We were able to show that such an approach can have significant advantages over traditional inference methods. We hope that this will spur further research, since many fundamental issues remain open. Specifically, a principled analysis of how different training methods behave in various conditions and how autoencoder learning, simulator learning and supervised learning can be optimally combined remains an avenue open to exploration.

Following the empirical evidence from extensive experimentation, we arrived at two different training approaches for DeepSpike and DECODE. DeepSpike uses autoencoder learning, simultaneously optimizing the parameters of the recognition network and the generative model, whereas DECODE uses simulator learning with a fixed set of predefined model parameters. While autoencoder learning is in principle very promising, it currently suffers from multiple problems in practice, especially when working with discrete latents. Theoretical and practical advances are needed to reduce gradient variance and memory demand. Another common issue is the propensity of the model to degrade towards pathological local optima while ignoring sparsity inducing priors. New methods are also needed to stabilize training. This will also be critical if we want to ensure that our algorithms not only perform well on benchmark problems, but are actually used by the scientific community and specifically non-machine learning experts. The main obstacle for widespread use is the network training step. While classical methods like deconvolution can be quickly applied to single observations to get an idea of how well they work and to test different parameter settings, this is usually not possible with DNN algorithms as they require extended training. It is therefore critical to develop methods that do not rely on extensive hyperparameter tuning and that allow for the straightforward tracking of training progress. Applicability could be further increased by providing pre-trained models whenever possible. This is an especially promising approach for integrated commercial microscope systems,[56,57] as these provide fixed and stable configurations. Such machines could be distributed together with fully optimized networks, therefore providing all the advantages of deep learning approaches without any of the drawbacks in usability.

Finally, it should be noted that improvements on this class of problems will not only advance CI and SI but other bio-imaging fields where similar problems abound: For example, particle tracking and the analysis of spatially resolved FISH data in its myriad variants.[47] In physics, equivalent problems are the detection and classification of celestial objects in astronomical images[66] and the analysis of data from collider experiments.[23]

I am therefore hopeful that the methods described in this thesis will be further improved and more widely applied to different problem settings, ultimately facilitating exciting new discoveries in the future.

Bibliography

[1] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. “An introduction to MCMC for machine learning”. In: *Machine learning* 50.1 (2003), pp. 5–43 (cit. on p. 2).

[2] Andrey Aristov, Benoit Lelandais, Elena Rensen, and Christophe Zimmer. “ZOLA-3D allows flexible 3D localization microscopy over an adjustable axial range”. In: *Nature communications* 9.1 (2018), pp. 1–8 (cit. on p. 10).

[3] Hazen P Babcock, Jeffrey R Moffitt, Yunlong Cao, and Xiaowei Zhuang. “Fast compressed sensing analysis for super-resolution imaging using L1-homotopy”. In: *Optics express* 21.23 (2013), pp. 28583–28596 (cit. on p. 13).

[4] Hazen P Babcock and Xiaowei Zhuang. “Analyzing single molecule localization microscopy data using cubic splines”. In: *Scientific reports* 7.1 (2017), p. 552 (cit. on pp. 7, 10).

[5] Philipp Berens, Jeremy Freeman, Thomas Deneux, Nikolay Chenkov, Thomas McColgan, Artur Speiser, Jakob H Macke, Srinivas C Turaga, Patrick Mineault, Peter Rupprecht, et al. “Community-based benchmarking improves spike rate inference from two-photon calcium imaging data”. In: *PLoS computational biology* 14.5 (2018), e1006157 (cit. on p. 15).

[6] M Betancourt. “Probabilistic modeling and statistical inference”. In: *GitHub repository* (2019) (cit. on p. 9).

[7] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. “Imaging intracellular fluorescent proteins at nanometer resolution”. In: *Science* 313.5793 (2006), pp. 1642–1645 (cit. on p. 6).

[8] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877 (cit. on p. 2).

[9] Jörg Bornschein and Yoshua Bengio. “Reweighted Wake-Sleep”. In: *CoRR* abs/1406.2751 (2014). arXiv: 1406.2751 (cit. on p. 18).

[10] Nicholas Boyd, Eric Jonas, Hazen P Babcock, and Benjamin Recht. “DeepLoco: Fast 3D Localization Microscopy Using Neural Networks”. In: *bioRxiv* (2018). DOI: 10.1101/267096 (cit. on pp. 16, 23).

[11] Junya Chen, Danni Lu, Zidi Xiu, Ke Bai, Lawrence Carin, and Chenyang Tao. “Variational Inference with Holder Bounds”. In: *arXiv preprint arXiv:2111.02947* (2021) (cit. on p. 18).

[12] 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, et al. “Ultrasensitive fluorescent proteins for imaging neuronal activity”. In: *Nature* 499.7458 (2013), pp. 295–300 (cit. on p. 4).

[13] Jennie L Close, Brian R Long, and Hongkui Zeng. “Spatially resolved transcriptomics in neuroscience”. In: *Nature Methods* 18.1 (2021), pp. 23–25 (cit. on p. 1).

[14] Susan Cox, Edward Rosten, James Monypenny, Tijana Jovanovic-Talisman, Dylan T Burnette, Jennifer Lippincott-Schwartz, Gareth E Jones, and Rainer Heintzmann. “Bayesian localization microscopy reveals nanoscale podosome dynamics”. In: *Nature methods* 9.2 (2012), p. 195 (cit. on p. 13).

[15] Thomas Deneux, Attila Kaszas, Gergely Szalay, Gergely Katona, Tamás Lakner, Amiram Grinvald, Balázs Rózsa, and Ivo Vanzetta. “Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo”. In: *Nature communications* 7.1 (2016), pp. 1–17 (cit. on pp. 14, 15, 21).

[16] Gregor PC Drummen. “Fluorescent probes and fluorescence (microscopy) techniques—illuminating biological and biomedical research”. In: *Molecules* 17.12 (2012), pp. 14067–14090 (cit. on p. 3).

[17] Mathew H Evans, Rasmus S Petersen, and Mark D Humphries. “On the use of calcium deconvolution algorithms in practical contexts”. In: *bioRxiv* (2019), p. 871137 (cit. on pp. 21, 24).

[18] DA Fish, AM Brinicombe, ER Pike, and JG Walker. “Blind deconvolution by means of the Richardson–Lucy algorithm”. In: *JOSA A* 12.1 (1995), pp. 58–65 (cit. on p. 2).

[19] Johannes Friedrich, Pengcheng Zhou, and Liam Paninski. “Fast online deconvolution of calcium imaging data”. In: *PLoS computational biology* 13.3 (2017), e1005423 (cit. on p. 13).

[20] Mariano I Gabitto, Herve Marie-Nelly, Ari Pakman, Andras Pataki, Xavier Darzacq, and Michael I Jordan. “A Bayesian nonparametric approach to super-resolution single-molecule localization”. In: *bioRxiv* (2020) (cit. on p. 12).

[21] David S Greenberg, Damian J Wallace, Kay-Michael Voit, Silvia Wuertenberger, Uwe Czubayko, Arne Monsees, Takashi Handa, Joshua T Vogelstein, Reinhard Seifert, Yvonne Groemping, et al. “Accurate action potential inference from a calcium sensor protein through biophysical modeling”. In: *BioRxiv* (2018), p. 479055 (cit. on pp. 13, 22).

[22] Christine Grienberger and Arthur Konnerth. “Imaging Calcium in Neurons”. In: *Neuron* 73.5 (2012), pp. 862–885. DOI: 10.1016/j.neuron.2012.02.011 (cit. on p. 1).

[23] Dan Guest, Kyle Cranmer, and Daniel Whiteson. “Deep learning and its application to LHC physics”. In: *Annual Review of Nuclear and Particle Science* 68 (2018), pp. 161–181 (cit. on p. 26).

[24] Alan L Hodgkin and Andrew F Huxley. “Action potentials recorded from inside a nerve fibre”. In: *Nature* 144.3651 (1939), pp. 710–711 (cit. on p. 3).

[25] Lawrence Huang, Peter Ledochowitsch, Ulf Knoblich, Jérôme Lecoq, Gabe J Murphy, R Clay Reid, Saskia EJ de Vries, Christof Koch, Hongkui Zeng, Michael A Buice, et al. “Relationship between simultaneously recorded spiking activity and fluorescence signal in GCaMP6 transgenic mice”. In: *Elife* 10 (2021), e51675 (cit. on p. 21).

[26] Cole Hurwitz, Kai Xu, Akash Srivastava, Alessio Buccino, and Matthias Hennig. "Scalable spike source localization in extracellular recordings using amortized variational inference". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 4724–4736 (cit. on p. 22).

[27] Aleksander Jablonski. "Efficiency of anti-Stokes fluorescence in dyes". In: *Nature* 131.3319 (1933), pp. 839–840 (cit. on p. 3).

[28] Danilo Jimenez Rezende, SM Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. "Unsupervised learning of 3d structure from images". In: *Advances in neural information processing systems* 29 (2016), pp. 4996–5004 (cit. on p. 22).

[29] Michio Kaku. *https://www.wsj.com/articles/michio-kaku-the-golden-age-of-neuroscience-has-arrived-1408577023*. `https://www.wsj.com/articles/michio-kaku-the-golden-age-of-neuroscience-has-arrived-1408577023`. Accessed: 2021-11-15 (cit. on p. 1).

[30] Jason ND Kerr, David Greenberg, and Fritjof Helmchen. "Imaging input and output of neocortical networks in vivo". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.39 (2005), pp. 14063–14068 (cit. on p. 10).

[31] Taehwan Kim, Seonah Moon, and Ke Xu. "Information-rich localization microscopy through machine learning". In: *Nature communications* 10.1 (2019), pp. 1–8 (cit. on p. 16).

[32] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013) (cit. on p. 16).

[33] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013) (cit. on p. 17).

[34] Elke Kirschbaum, Manuel Haußmann, Steffen Wolf, Hannah Jakobi, Justus Schneider, Shehabeldin Elzoheiry, Oliver Kann, Daniel Durstewitz, and Fred A Hamprecht. "LeMoNADe: Learned Motif and Neuronal Assembly Detection in calcium imaging videos". In: *arXiv preprint arXiv:1806.09963* (2018) (cit. on p. 22).

[35] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243 (cit. on p. 16).

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. on pp. 1, 15).

[37] Romain F Laine, Ignacio Arganda-Carreras, Ricardo Henriques, and Guillaume Jacquemet. "Avoiding a replication crisis in deep-learning-based bioimage analysis". In: *Nature methods* 18.10 (2021), pp. 1136–1144 (cit. on p. 25).

[38] Tuan Anh Le, Adam R Kosiorek, N Siddharth, Yee Whye Teh, and Frank Wood. "Revisiting reweighted wake-sleep for models with stochastic control flow". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1039–1049 (cit. on pp. 16, 18).

[39] Lucien Le Cam. "Maximum likelihood: an introduction". In: *International Statistical Review/Revue Internationale de Statistique* (1990), pp. 153–171 (cit. on p. 2).

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444 (cit. on p. 1).

[41] Mickaël Lelek, Melina T Gyparaki, Gerti Beliu, Florian Schueder, Juliette Griffié, Suliana Manley, Ralf Jungmann, Markus Sauer, Melike Lakadamyali, and Christophe Zimmer. “Single-molecule localization microscopy”. In: *Nature Reviews Methods Primers* 1.1 (2021), pp. 1–27 (cit. on p. 6).

[42] Xinyang Li, Guoxun Zhang, Jiamin Wu, Yuanlong Zhang, Zhifeng Zhao, Xing Lin, Hui Qiao, Hao Xie, Haoqian Wang, Lu Fang, et al. “Reinforcing neuron extraction and spike inference in calcium imaging using deep self-supervised denoising”. In: *Nature Methods* 18.11 (2021), pp. 1395–1400 (cit. on p. 22).

[43] Yiming Li, Markus Mund, Philipp Hoess, Joran Deschamps, Ulf Matti, Bianca Nijmeijer, Vilma Jimenez Sabinina, Jan Ellenberg, Ingmar Schoen, and Jonas Ries. “Real-time 3D single-molecule localization using experimental point spread functions”. In: *Nature methods* 15.5 (2018), p. 367 (cit. on p. 10).

[44] Jeff W Lichtman and José-Angel Conchello. “Fluorescence microscopy”. In: *Nature methods* 2.12 (2005), pp. 910–919 (cit. on p. 3).

[45] L Looger, Y Zhang, M Rózsa, Y Liang, D Bushey, Z Wei, J Zheng, D Reep, GJ Broussard, A Tsang, et al. “Fast and sensitive GCaMP calcium indicators for imaging neural populations”. In: (2021) (cit. on p. 5).

[46] James Martens. “New insights and perspectives on the natural gradient method”. In: *arXiv preprint arXiv:1412.1193* (2014) (cit. on p. 16).

[47] Vivien Marx. “Method of the Year: spatially resolved transcriptomics”. In: *Nature Methods* 18.1 (2021), pp. 9–14 (cit. on p. 26).

[48] Vaden Masrani, Tuan Anh Le, and Frank Wood. “The Thermodynamic Variational Objective”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 11521–11530 (cit. on p. 18).

[49] Erik Meijering. “A bird’s-eye view of deep learning in bioimage analysis”. In: *Computational and Structural Biotechnology Journal* 18 (2020), p. 2312 (cit. on p. 25).

[50] Junhong Min, Cédric Vonesch, Hagai Kirshner, Lina Carlini, Nicolas Olivier, Seamus Holden, Suliana Manley, Jong Chul Ye, and Michael Unser. “FALCON: fast and unbiased reconstruction of high-density super-resolution microscopy data”. In: *Scientific reports* 4.1 (2014), pp. 1–9 (cit. on p. 13).

[51] Andriy Mnih and Danilo Jimenez Rezende. “Variational inference for Monte Carlo objectives”. In: *Proceedings of the 33st International Conference on Machine Learning*. 2016 (cit. on p. 18).

[52] Leonhard Möckl, Anish R Roy, Petar N Petrov, and WE Moerner. “Accurate and rapid background estimation in single-molecule localization microscopy using the deep neural network BGnet”. In: *Proceedings of the National Academy of Sciences* 117.1 (2020), pp. 60–67 (cit. on p. 16).

[53] Kim I Mortensen, L Stirling Churchman, James A Spudich, and Henrik Flyvbjerg. “Optimized localization analysis for single-molecule tracking and super-resolution microscopy”. In: *Nature methods* 7.5 (2010), pp. 377–381 (cit. on p. 14).

[54] Elias Nehme, Daniel Freedman, Racheli Gordon, Boris Ferdman, Lucien E Weiss, Onit Alalouf, Tal Naor, Reut Orange, Tomer Michaeli, and Yoav Shechtman. “DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning”. In: *Nature Methods* (2020), pp. 1–7 (cit. on pp. 16, 23, 24).

[55] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. "Exploring generalization in deep learning". In: *arXiv preprint arXiv:1706.08947* (2017) (cit. on p. 18).

[56] *Nikon N-STORM*. `https://www.microscope.healthcare.nikon.com/products/super-resolution-microscopes/n-storm-super-resolution`. Accessed: 2021-1-24 (cit. on p. 25).

[57] *ONI Nanoimager*. `https://oni.bio/nanoimager/`. Accessed: 2021-1-24 (cit. on p. 25).

[58] Martin Ovesn, Pavel Křıžek, Josef Borkovec, Zdeněk Švindrych, and Guy M Hagen. "ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging". In: *Bioinformatics* 30.16 (2014), pp. 2389–2390 (cit. on p. 10).

[59] Marius Pachitariu, Carsen Stringer, and Kenneth D Harris. "Robustness of spike deconvolution for neuronal calcium imaging". In: *Journal of Neuroscience* 38.37 (2018), pp. 7976–7985 (cit. on p. 13).

[60] David W Piston. "Imaging living cells and tissues by two-photon excitation microscopy". In: *Trends in cell biology* 9.2 (1999), pp. 66–69 (cit. on p. 4).

[61] Eftychios A Pnevmatikakis. "Analysis pipelines for calcium imaging data". In: *Current opinion in neurobiology* 55 (2019), pp. 15–21 (cit. on p. 1).

[62] Eftychios A Pnevmatikakis, Josh Merel, Ari Pakman, and Liam Paninski. "Bayesian spike inference from calcium imaging data". In: *Signals, Systems and Computers, 2013 Asilomar Conference on*. IEEE. 2013, pp. 349–353 (cit. on pp. 13, 21).

[63] Eftychios A Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, et al. "Simultaneous denoising, deconvolution, and demixing of calcium imaging data". In: *Neuron* 89.2 (2016), pp. 285–299 (cit. on p. 21).

[64] Luke Yuri Prince, Shahab Bakhtiari, Colleen J Gillon, and Blake A Richards. "Parallel inference of hierarchical latent dynamics in two-photon calcium imaging of neuronal populations". In: *bioRxiv* (2021) (cit. on p. 22).

[65] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference". In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 814–822 (cit. on p. 12).

[66] Jeffrey Regier, Andrew Miller, Jon McAuliffe, Ryan Adams, Matt Hoffman, Dustin Lang, David Schlegel, and Mr Prabhat. "Celeste: Variational inference for a generative model of astronomical images". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2095–2103 (cit. on p. 26).

[67] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *arXiv preprint arXiv:1401.4082* (2014) (cit. on p. 16).

[68] William Hadley Richardson. "Bayesian-based iterative method of image restoration". In: *JoSA* 62.1 (1972), pp. 55–59 (cit. on p. 13).

[69] Peter Rupprecht, Stefano Carta, Adrian Hoffmann, Mayumi Echizen, Antonin Blot, Alex C Kwan, Yang Dan, Sonja B Hofer, Kazuo Kitamura, Fritjof Helmchen, et al. "A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging". In: *Nature Neuroscience* 24.9 (2021), pp. 1324–1337 (cit. on pp. 15, 22).

[70] Michael J Rust, Mark Bates, and Xiaowei Zhuang. “Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)”. In: *Nature methods* 3.10 (2006), p. 793 (cit. on p. 6).

[71] Daniel Sage, Thanh-An Pham, Hazen Babcock, Tomas Lukes, Thomas Pengo, Jerry Chao, Ramraj Velmurugan, Alex Herbert, Anurag Agrawal, Silvia Colabrese, et al. “Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software”. In: *Nature methods* 16.5 (2019), pp. 387–395 (cit. on pp. 16, 18, 24).

[72] Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, et al. “A connectome and analysis of the adult Drosophila central brain”. In: *Elife* 9 (2020), e57443 (cit. on p. 1).

[73] Joerg Schnitzbauer, Maximilian T Strauss, Thomas Schlichthaerle, Florian Schueder, and Ralf Jungmann. “Super-resolution microscopy with DNA-PAINT”. In: *Nature protocols* 12.6 (2017), p. 1198 (cit. on p. 6).

[74] Osamu Shimomura, Frank H. Johnson, and Yo Saiga. “Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan, Aequorea”. In: *Journal of Cellular and Comparative Physiology* 59.3 (1962), pp. 223–239. DOI: 10.1002/jcp.1030590302 (cit. on p. 3).

[75] Emily Singer. *Why Neuroscience Needs Data Scientists*. https://www.simonsfoundation.org/2018/11/19/why-neuroscience-needs-data-scientists/. Accessed: 2021-11-15 (cit. on p. 1).

[76] Artur Speiser, Lucas-Raphael Müller, Ulf Matti, Christopher J Obara, Wesley R Legant, Jonas Ries, Jakob H Macke, and Srinivas C Turaga. “Teaching deep neural networks to localize single molecules for super-resolution microscopy”. In: *arXiv preprint arXiv:1907.00770* (2019) (cit. on p. 23).

[77] 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: *Advances in Neural Information Processing Systems 30* (2017). Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp. 4024–4034 (cit. on p. 21).

[78] Sjoerd Stallinga and Bernd Rieger. “Accuracy of the Gaussian point spread function model in 2D localization microscopy”. In: *Optics express* 18.24 (2010), pp. 24461–24476 (cit. on p. 16).

[79] Ian H Stevenson and Konrad P Kording. “How advances in neural recording affect data analysis”. In: *Nature neuroscience* 14.2 (2011), pp. 139–142 (cit. on p. 2).

[80] Carsen Stringer and Marius Pachitariu. “Computational processing of neural recordings from calcium imaging data”. In: *Current opinion in neurobiology* 55 (2019), pp. 22–31 (cit. on p. 1).

[81] Karel Svoboda. “GENIE project, Janelia Farm Campus, HHMI; Karel Svoboda (contact). (2015). Simultaneous imaging and loose-seal cell-attached electrical recordings from neurons expressing a variety of genetically encoded calcium indicators”. In: (2015). DOI: http://dx.doi.org/10.6080/K02R3PMN (cit. on p. 4).

[82] Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. “Large-scale neural recordings call for new insights to link brain and behavior”. In: *Nature neuroscience* (2022), pp. 1–9 (cit. on pp. 1, 2).

[83] Andrew Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. In: *IEEE transactions on Information Theory* 13.2 (1967), pp. 260–269 (cit. on p. 14).

[84] Joshua T Vogelstein, Adam M Packer, Timothy A Machado, Tanya Sippy, Baktash Babadi, Rafael Yuste, and Liam Paninski. “Fast nonnegative deconvolution for spike train inference from population calcium imaging”. In: *Journal of neurophysiology* 104.6 (2010), pp. 3691–3704 (cit. on p. 13).

[85] Joshua T Vogelstein, Brendon O Watson, Adam M Packer, Rafael Yuste, Bruno Jedynak, and Liam Paninski. “Spike inference from calcium imaging using sequential Monte Carlo methods”. In: *Biophysical journal* 97.2 (2009), pp. 636–655 (cit. on p. 10).

[86] P Zelger, K Kaser, B Rossboth, L Velas, GJ Schütz, and A Jesacher. “Three-dimensional localization microscopy using deep learning”. In: *Optics express* 26.25 (2018), pp. 33166–33179 (cit. on p. 16).

[87] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115 (cit. on p. 18).

[88] Peiyi Zhang, Sheng Liu, Abhishek Chaurasia, Donghan Ma, Michael J Mlodzianoski, Eugenio Culurciello, and Fang Huang. “Analyzing complex single-molecule emission patterns with deep learning”. In: *Nature methods* 15.11 (2018), pp. 913–916 (cit. on p. 16).

[89] Zhanhong Zhou and Chung Tin. “Effective and Efficient Neural Networks for Spike Inference from In Vivo Calcium Imaging”. In: *bioRxiv* (2021) (cit. on p. 22).

[90] Lei Zhu, Wei Zhang, Daniel Elnatan, and Bo Huang. “Faster STORM using compressed sensing”. In: *Nature methods* 9.7 (2012), pp. 721–723 (cit. on p. 13).

Appendices

Fast amortized inference of neural activity from calcium imaging data with variational autoencoders

Artur Speiser[12], **Jinyao Yan**[3], **Evan Archer**[4][*], **Lars Buesing**[4][†],
Srinivas C. Turaga[3][‡] **and Jakob H. Macke**[1][‡][§]
[1]research center caesar, an associate of the Max Planck Society, Bonn, Germany
[2]IMPRS Brain and Behavior Bonn/Florida
[3]HHMI Janelia Research Campus
[4]Columbia University
artur.speiser@caesar.de, turagas@janelia.hhmi.org, jakob.macke@caesar.de

Abstract

Calcium imaging permits optical measurement of neural activity. Since intracellular calcium concentration is an indirect measurement of neural activity, computational tools are necessary to infer the true underlying spiking activity from fluorescence measurements. Bayesian model inversion can be used to solve this problem, but typically requires either computationally expensive MCMC sampling, or faster but approximate maximum-a-posteriori optimization. Here, we introduce a flexible algorithmic framework for fast, efficient and accurate extraction of neural spikes from imaging data. Using the framework of variational autoencoders, we propose to amortize inference by training a deep neural network to perform model inversion efficiently. The recognition network is trained to produce samples from the posterior distribution over spike trains. Once trained, performing inference amounts to a fast single forward pass through the network, without the need for iterative optimization or sampling. We show that amortization can be applied flexibly to a wide range of nonlinear generative models and significantly improves upon the state of the art in computation time, while achieving competitive accuracy. Our framework is also able to represent posterior distributions over spike-trains. We demonstrate the generality of our method by proposing the first probabilistic approach for separating backpropagating action potentials from putative synaptic inputs in calcium imaging of dendritic spines.

1 Introduction

Spiking activity in neurons leads to changes in intra-cellular calcium concentration which can be measured by fluorescence microscopy of synthetic calcium indicators such as Oregon Green BAPTA-1 [1] or genetically encoded calcium indictors such as GCaMP6 [2]. Such calcium imaging has become important since it enables the parallel measurement of large neural populations in a spatially resolved and minimally invasive manner [3, 4]. Calcium imaging can also be used to study neural activity at subcellular resolution, e.g. for measuring the tuning of dendritic spines [5, 6]. However, due to the indirect nature of calcium imaging, spike inference algorithms must be used to infer the underlying neural spiking activity leading to measured fluorescence dynamics.

[*] current affiliation: Cogitai.Inc

[†] current affiliation: DeepMind

[‡] equal contribution

[§] current primary affiliation: Centre for Cognitive Science, Technical University Darmstadt

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Most commonly-used approaches to spike inference [7, 8, 9, 10, 11, 12, 13, 14] are based on carefully designed generative models that describe the process by which spiking activity leads to fluorescence measurements. Spikes are treated as latent variables, and spike-prediction is performed by inferring both the parameters of the model and the spike latent variables from fluorescence time series, or "traces" [7, 8, 9, 10]. The advantage of this approach is that it does not require extensive ground truth data for training, since simultaneous electrophysiological and fluorescence recordings of neural activity are difficult to acquire, and that prior knowledge can be incorporated in the specification of the generative model. The accuracy of the predictions depends on the faithfulness of the generative model of the transformation of spike trains into fluorescence measurements [14, 12]. The disadvantage of this approach is that spike-inference requires either Markov-Chain Monte Carlo (MCMC) or Sequential Monte-Carlo techniques to sample from the posterior distribution over spike-trains or alternatively, iterative optimization to obtain an approximate maximum a-posteriori (MAP) prediction. Currently used approaches rely on bespoke, model-specific inference algorithms, which can limit the flexibility in designing suitable generative models. Most commonly used methods are based on simple phenomenological (and often linear) models [7, 8, 9, 10, 13].

Recently, a small number of cell-attached electrophysiological recordings of neural activity have become available, with simultaneous fluorescence calcium measurements in the same neurons. This has made it possible to train powerful and fast classifiers to perform spike-inference in a discriminative manner, precluding the need for accurate generative models of calcium dynamics [15]. The disadvantage of this approach is that it can require large labeled data-sets for every new combination of calcium indicator, cell-type and microscopy method, which can be expensive or impossible to acquire. Further, these discriminative methods do not easily allow the incorporation of prior knowledge about the generative process. Finally, current classification approaches yield only pointwise predictions of spike probability (i.e. firing rates), independent across time, and ignore temporal correlations in the posterior distribution of spikes.

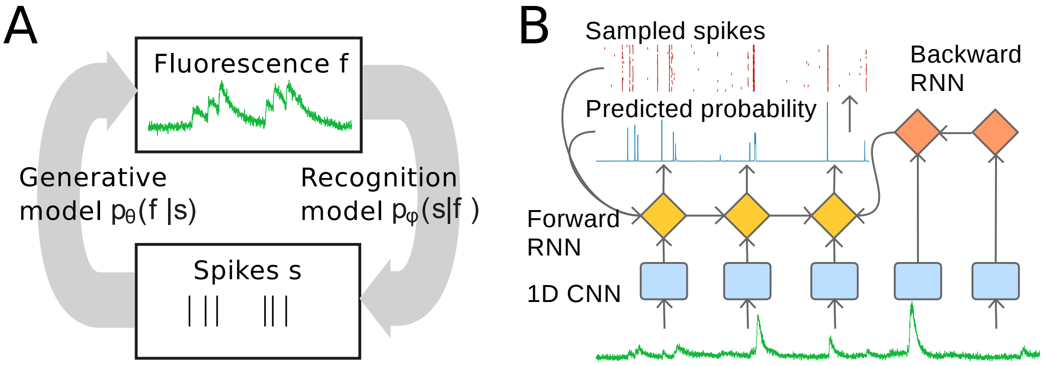


Figure 1: **Amortized inference for predicting spikes from imaging data. A)** Our goal is to infer a spike train s from an observed time-series of fluorescence-measurements f. We assume that we have a generative model of fluorescence given spikes with (unknown) parameters θ, and we simultaneously learn θ as well as a 'recognition model' which approximates the posterior over spikes s given f and which can be used for decoding a spike train from imaging data. **B)** We parameterize the recognition-model by a multi-layer network architecture: Fluorescence-data is first filtered by a deep 1D convolutional network (CNN), providing input to a stochastic forward running recurrent neural network (RNN) which predicts spike-probabilities and takes previously sampled spikes as additional input. An additional deterministic RNN runs backward in time and provides further context.

Here, we develop a new spike inference framework called DeepSpike (DS) based on the variational autoencoder technique which uses stochastic variational inference (SVI) to teach a classifier to predict spikes in an unsupervised manner using a generative model. This new strategy allows us to combine the advantages of generative [7] and discriminative approaches [15] into a single fast classifier-based method for spike inference. In the variational autoencoder framework, the classifier is called a *recognition model* and represents an approximate posterior distribution over spike trains from which samples can be drawn in an efficient manner. Once trained to perform spike inference on one dataset, the recognition model can be applied to perform inference on statistically similar datasets without any retraining: The computational cost of variational spike inference is *amortized*, dramatically speeding up inference at test-time by exploiting fast, classifier based recognition models.

We introduce two recognition models: The first is a temporal convolutional network which produces a posterior distribution which is factorized in time, similar to standard classifier-based methods [15]. The second is a recurrent neural network-based recognition model, similar to [16, 17] which can represent any correlated posterior distribution in the non-parametric limit. Once trained, both models perform spike inference with state-of-the-art accuracy, and enable simultaneous spike inference for populations as large as 10^4 in real time on a single GPU.

We show the generality of this black-box amortized inference method by demonstrating its accuracy for inference with a classic linear generative model [7, 8], as well as two nonlinear generative models [12]. Finally, we show an extension of the spike inference method to simultaneous inference and demixing of synaptic inputs from backpropagating somatic action potentials from simultaneous somatic and dendritic calcium imaging.

2 Amortized inference using variational autoencoders

2.1 Approach and training procedure

We observe fluorescence traces $f_t^i, t = 1 \dots T^i$ representing noisy measurements of the dynamics of somatic calcium concentration in neurons $i = 1 \dots N$. We assume a parametrised, probabilistic, differentiable generative model $p_{\theta^i}(f|s)$ with (unknown) parameters θ^i. The generative model predicts a fluorescence trace given an underlying binary spike train s^i, where $s_t^i = 1$ indicates that the neuron i produced an action potential in the interval indexed by t. Our goal is to infer a latent spike-train s given only fluorescence observations f. We will solve this problem by training a deep neural network as a "recognition model" [18, 19, 20] parametrized by weights ϕ. Use of a recognition model enables fast computation of an approximate posterior distribution over spike trains from a fluorescence trace $q_\phi(s|f)$. We will share one recognition model across multiple cells, i.e. that $q_\phi(s^i|f^i) \approx p_{\theta^i}(s^i|f^i)$ for each i. We describe an unsupervised training procedure which jointly optimizes parameters of the generative model θ and the recognition network ϕ in order to maximize a lower bound on the log likelihood of the observed data, $\log p(f)$ [19, 18, 20].

We learn the parameters ϕ and θ simultaneously by jointly maximizing $\mathcal{L}^K(\theta, \phi)$, a multi-sample importance-weighting lower bound on the log likelihood $\log p(f)$ given by [21]

$$\mathcal{L}^K(\theta, \phi) = \mathbb{E}_{s^1, \dots, s^K \sim q_\phi(s|f)} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(s^k, f)}{q_\phi(s^k|f)} \right] \leq \log p(f), \tag{1}$$

where s^k are spike trains sampled from the recognition model $q_\phi(s|f)$. This stochastic objective involves drawing K samples from the recognition model, and evaluating their likelihood by passing them through the generative model. When $K = 1$, the bound reduces to the *evidence lower bound* (ELBO). Increasing K yields a tighter lower bound (than the ELBO) on the marginal log likelihood, at the cost of additional training time. We found that increasing the number of samples leads to better fits of the generative model; in our experiments, we used $K = 64$.

To train θ and ϕ by stochastic gradient ascent, we must estimate the gradient $\nabla_{\phi,\theta}\mathcal{L}(\theta, \phi)$. As our recognition model produces an approximate posterior over binary spike trains, the gradients have to be estimated based on samples. Obtaining functional estimates of the gradients $\nabla_\phi \mathcal{L}(\theta, \phi)$ with respect to parameters of the recognition model is challenging and relies on constructing effective control variates to reduce variance [22]. We use the *variational inference for monte carlo objectives* (VIMCO) approach of [23] to produce low-variance unbiased estimates of the gradients $\nabla_{\phi,\theta}\mathcal{L}^K(\theta, \phi)$. The generative training procedure could be augmented with a supervised cost term [24, 25], resulting in a semi-supervised training method.

Gradient optimization: We use ADAM [26], an adaptive gradient update scheme, to perform online stochastic gradient ascent. The training data is cut into short chunks of several hundred time-steps and arranged in batches containing samples from a single cell. As we train only one recognition model but multiple generative models in parallel, we load the respective generative model and ADAM parameters at each iteration. Finally, we use norm-clipping to scale the gradients acting on the recognition model: the norm of all gradients is calculated, and if it exceeds a fixed threshold the gradients are rescaled. While norm-clipping was introduced to prevent exploding gradients in RNNs

[27], we found it to be critical to achieve high performance both for RNN and CNN architectures in our learning problem. Very small threshold values (0.02) empirically yielded best results.

2.2 Generative models $p_\theta(f|s)$

To demonstrate that our computational strategy can be applied to a wide range of differentiable models in a black-box manner, we consider four generative models: a simple, but commonly used linear model of calcium dynamics [7, 8, 9, 10], two more sophisticated nonlinear models which additionally incorporate saturation and facilitation resulting from the dynamics of calcium binding to the calcium sensor, and finally a multi-dimensional model for dendritic imaging data.

Linear auto-regressive generative model (SCF): We use the name *SCF* for the classic linear convolutional generative model used in [7, 8, 9, 10], since this generative process is described by the Spikes s_t, which linearly impact Calcium concentration c_t, which in turn determines the observed Fluorescence intensity f_t,

$$c_t = \sum_{t'=1}^{p} \gamma_{t'} c_{t-t'} + \delta s_t, \qquad f_t = \alpha c_t + \beta + e_t, \tag{2}$$

with linear auto-regressive dynamics of order p for the calcium concentration with parameters γ, spike-amplitude δ, gain α, constant fluorescence baseline β, and additive measurement noise $e_t \sim \mathcal{N}(0, \sigma^2)$.

Nonlinear auto-regressive and sensor dynamics generative models (SCDF & MLphys): As examples of nonlinear generative models [28], we consider two simple models of the discrete-time dynamics of the calcium sensor or dye. In the first (SCDF), the concentration of fluorescent dye molecules d_t is a function of the somatic Calcium concentration c_t, and has Dynamics

$$d_t - d_{t-1} = \kappa_{\text{on}} c_t^{\eta}([D] - d_{t-1}) - \kappa_{\text{off}} d_{t-1}, \qquad f_t = \alpha d_t + \beta + e_t, \tag{3}$$

where κ_{on} and κ_{off} are the rates at which the calcium sensor binds and unbinds calcium ions, and η is a Hill coefficient. We constrained these parameters to be non-negative. $[D]$ is the total concentration of the dye molecule in the soma, which sets the maximum possible value of d_t. The richer dynamics of the SCDF model allow for facilitation of fluorescence at low firing rates, and saturation at high rates. The parameters of the SCDF model are $\theta = \{\alpha, \beta, \gamma, \kappa_{\text{on}}, \kappa_{\text{off}}, \eta, [D], \sigma^2\}$.

The second nonlinear model (MLphys) is a discrete-time version of the MLspike generative model [12], simplified by not including a model of the time-varying baseline. The dynamics for f_t and c_t are as above, with $\delta = 1$. We replace the dynamics for d_t by

$$d_t - d_{t-1} = \frac{1}{\tau_{on}}(1 + \omega((c_0 + c_t)^{\eta} - c_0^{\eta}))(\frac{((c_0 + c_t)^{\eta} - c_0^{\eta})}{(1 + \omega((c_0 + c_t)^{\eta} - c_0^{\eta}))} - d_{t-1}). \tag{4}$$

Multi-dimensional soma + dendrite generative model (DS-F-DEN): The dendritic generative model is a multi-dimensional SCDF model that incorporates back-propagating action potentials (bAPs). The calcium concentration at the cell body (superscript c) is generated as for SCDF, whereas for the spine (superscript s), there are two components: synaptic inputs and bAPs from the soma,

$$c_t^c = \sum_{t'=1}^{p} \gamma_{t'}^c c_{t-t'}^c + \delta^c s_t^c, \qquad c_t^s = \sum_{t'=1}^{p} \gamma_{t'}^s c_{t-t'}^s + \delta^s s_t^s + \delta^{b_s} s_t^c, \tag{5}$$

where δ^{b_s} are the amplitude coefficients of bAPs for different spine locations, and $c \in \{1, ..., N_c\}$, $s \in \{1, ..., N_s\}$. The spines and soma share the same dye dynamics as in (3). The parameters of the dendritic integration model are $\theta = \{\alpha_{s,c}, \beta_{s,c}, \gamma_{s,c}, \kappa_{\text{on}}, \kappa_{\text{off}}, \eta, [D], \sigma_{s,c}^2\}$. We note that this simple generative model does not attempt to capture the full complexity of nonlinear processing in dendrites (e.g. it does not incorporate nonlinear phenomena such as dendritic plateau potentials). Its goal is to separate local influences (synaptic inputs) from global events (bAPs, or potentially regenerative dendritic events).

2.3 Recognition models: parametrization of the approximate posterior $q_\phi(s|f)$

The goal of the recognition model is to provide a fast and efficient approximation $q_\phi(s|f)$ to the true posterior $p(s|f)$ over discrete latent spike trains s. We will use both a factorized, localized approximation (parameterized as a convolutional neural network), and a more flexible, non-factorized and non-localized approximation (parameterized using additional recurrent neural networks).

Convolutional neural network: Factorized posterior approximation (DS-F) In [15], it was reported that good spike-prediction performance can be achieved by making the spike probability $q_\phi(s_t|f_{t-\tau...t+\tau})$ depend on a local window of the fluorescence trace of length $2\tau+1$ centered at t when training such a model fully supervised. We implement a scaled up version of this idea, using a deep neural network which is convolutional in time as the recognition model. We use architectures with up to five hidden layers and ≈ 20 filters per layer with Leaky ReLUs units [29]. The output layer uses a sigmoid nonlinearity to compute the Bernoulli spike probabilities $q_\phi(s_t|f)$.

Recurrent neural network: Capturing temporal correlations in the posterior (DS-NF) The fully-factorized posterior approximation (DS-F) above ignores temporal correlations in the posterior over spike trains. Such correlations can be useful in modeling uncertainty in the precise timing of a spike, which induces negative correlations between nearby time bins. To model temporal correlations, we developed a RNN-based non-factorizing distribution which can approach the true posterior in the non-parametric limit (see figure 1B). Similar to [16], we use the temporal ordering over spikes and factorize the joint distribution over spikes as $q_\phi(s|f) = \prod_t q_\phi(s_t|f, s_0, ..., s_{t-1})$, by conditioning spikes at t on all previously sampled spikes. Our RNN uses a CNN as described above to extract features from the input trace. Additional input is provided by a a backwards RNN which also receives input from the CNN features. The outputs of the forward RNN and CNN are transformed into Bernoulli spike probabilities $q_\phi(s_t|f)$ through a dense sigmoid layer. This probability and the sample drawn from it are relayed to the forward RNN in the next time step. Forward and backward RNN have a single layer with 64 gated recurrent units each [30].

2.4 Details of synthetic and real data and evaluation methodology

We evaluated our method on simulated and experimental data. From our SCF and SCDF generative models for spike-inference, we simulated traces of length $T = 10^4$ assuming a recording frequency of 60 Hz. Initial parameters where obtained by fitting the models to real data (see below), and heterogeneity across neurons was achieved by randomly perturbing parameters. We used 50 neurons each for training and validation and 100 neurons in the test set. For each cell, we generated three traces with firing rates of 0.6, 0.9 and 1.1 Hz, assuming i.i.d. spikes.

Finally, we compared methods on two-photon imaging data from $9 + 11$ cells from [2], which is available at www.crcns.org. Layer 2/3 pyramidal neurons in mouse visual cortex were imaged at 60 Hz using the genetically encoded calcium-indicators GCaMP6s and GCaMP6f, while action-potentials were measured electrophysiologically using cell-attached recordings. Data was pre-processed by removing a slow moving baseline using the 5th percentile in a window of 6000 time steps. Furthermore we used this baseline estimate to calculate $\Delta F/F$. Cross-validated results where obtained using 4 folds, where we trained and validated on 3/4 of the cells in each dataset and tested on the remaining cells to highlight the potential for amortized inference. Early stopping was performed based on the the correlation achieved on the train/validation set, which was evaluated every 100 update steps.

We report results using the cross-correlation between true and predicted spike-rates, at the sampling discretization of $16.6\,\mathrm{ms}$ for simulated data and $40\,\mathrm{ms}$ for real data. As the predictions of our DS-NF model are not deterministic, we sample 30 times from the model and average over the resulting probability distributions to obtain an estimate of the marginal probability before we calculate cross-correlations.

We used multiple generative models to show that our inference algorithm is not tied to a particular model: SCDF for the experiments depicted in Fig. 2, SCF for a comparison with established methods based on this linear model (Table 1, column 1), and MLphys on real data as it is used by the current state-of-the-art inference algorithm (Table 1, columns 2 & 3, Fig. 3).

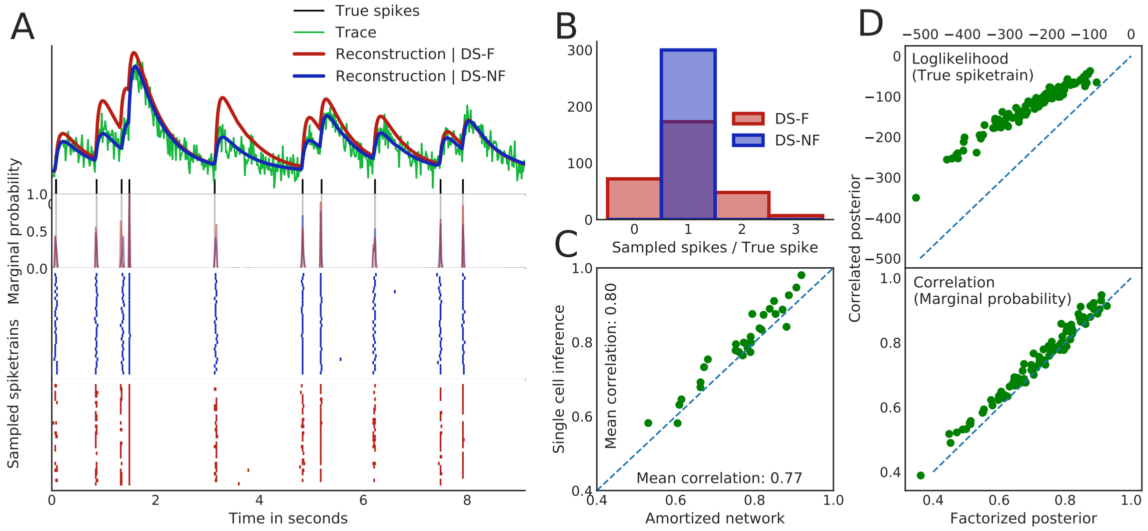


Figure 2: **Model-inversion with variational autoencoders, simulated data A)** Illustration of factorized (CNN, DS-F) and non-factorized posterior approximation (RNN, DS-NF) on simulated data (SCDF generative model). DS-NF yields more accurate reconstructions, but both methods lead to similar marginal predictions (i.e. predicted firing rates, bottom). **B)** Number of spikes sampled for every true spike for the factorized (red) and non-factorized (red) posterior. The correlated posterior consistently samples the correct number of spikes while still accounting for the uncertainty in the spike timing. **C)** Performance of amortized vs non-amortized inference on simulated data. **D)** Scatter plots of achieved log-likelihood of the true spike train under the posterior model (top) and achieved correlation coefficients between the marginalized spiking probabilities and true spike trains (bottom).

3 Results

3.1 Stochastic variational spike inference of factorized and correlated posteriors

We first illustrate our approach on synthetic data, and compare our two different architectures for recognition models. We simulated data from the SCDF nonlinear generative model and trained DeepSpike unsupervised using the same SCDF model. While only the more expressive recognition model (DS-NF) is able to achieve a close-to-perfect reconstructions of the fluorescence traces (Fig. 2 A, top row), both approaches yield similar marginal firing rate predictions (second row). However, as the factorized model does not model correlations in the posterior, it yields higher variance in the number of spikes reconstructed for each true spike (Fig. 2 B). This is because the factorized model can not capture that a fluorescence increase might be 'explained away' by a spike that has just been sampled, i.e. it can not capture the difference between uncertainty in spike-timing and uncertainty in (local) spike-counts. Therefore, while both approaches predict firing rates similarly well on simulated data (as quantified using correlation, Fig. 2 D), the DS-NF model assigns higher posterior probability to the true spike trains.

3.2 Amortizing inference leads to fast and accurate test-time inference

In principle, our unsupervised learning procedure could be re-trained on every data-set of interest. However, it also allows for amortizing inference by sharing one recognition model across multiple cells, and applying the recognition model directly on new data without additional training for fast test-time performance. Amortized inference allows for the recognition model to be used for inference in the same way as a network that was trained fully supervised. Since there is no variational optimization at test time, inference with this network is just as fast as inference with a supervised network. Similarly to supervised learning, there will be limitations on the ability of this network to generalize to different imaging conditions or indicators that where not included in the training set.

To test if our recognition model generalizes well enough for amortized inference to work across multiple cells, as well as on cells it did not see during training, we trained one DS-NF model on 50 cells (simulated data, SCDF) and evaluated its performance on a non-overlapping set of 30 cells. For comparison, we also trained 30 DS-NF models separately, on each of those cells– this amounts to standard variational inference using a neural network to parametrize the posterior approximation, but without amortizing inference. We found that amortizing inference only causes a small drop in performance (Fig. 2 C). However, this drop in performance is offset by the the large gain in computational efficiency as training a neural network takes several orders of magnitude more time then applying it at test time.

Inference using the DS-F model only requires a single forward pass through a convolutional network to predict firing rates, and DS-NF requires running a stochastic RNN for each sampled spike train. While the exact running-time of each of these applications will depend on both implementation and hardware, we give rough indications of computational speed number estimated on an Intel(R) Xeon(R) CPU E5-2697 v3. On the CPU, our DS-F approach takes $0.05\,\mathrm{s}$ to process a single trace of 10K time steps, when using a network appropriate for $60\,\mathrm{Hz}$ data. This is on the same order as the $0.07\,\mathrm{s}$ (Intel Core i5 $2.7\,\mathrm{GHz}$ CPU) reported by [31] for their OASIS algorithm, which is currently the fastest available implementation for constrained deconvolution (CDEC) of SCF, but restricted to this linear generative model. The DS-NF algorithm requires $4.6\,\mathrm{s}$ which still compares favourably to MLspike which takes $9.2\,\mathrm{s}$ (evaluated on the same CPU). As our algorithm is implemented in Theano [32] it can be easily accelerated and allows for massive parallelization on a single GPU. On a GTX Titan X, DS-F and DS-NF take $0.001\,\mathrm{s}$ and $1.5\,\mathrm{s}$, respectively. When processing 500 traces in parallel, DS-NF becomes only 2.5 times slower. Extrapolating from these results, this implies that even when using the DS-NF algorithm, we would be able to perform spike-inference on 1 hour of recordings at $60\,\mathrm{Hz}$ for 500 cells in less then $90\,\mathrm{s}$.

Table 1: Performance comparison. Values are correlations between predicted marginal probabilities and ground truth spikes.

	Dataset			Dendritic dataset	
Algorithm	SCF-Sim.	GCaMP6s	GCaMP6f	Soma	Spine
DS-F	0.88 ± 0.01	0.74 ± 0.02	0.74 ± 0.02		
DS-NF	0.89 ± 0.01	0.72 ± 0.02	0.73 ± 0.02		
CDEC [10]	0.86 ± 0.01	0.39 ± 0.03 *	0.58 ± 0.02 *		
MCMC [9]	0.87 ± 0.01	0.47 ± 0.03 *	0.53 ± 0.03 *		
MLSpike [12]		0.60 ± 0.02 *	0.67 ± 0.01 *		
DS-F-DEN				0.84 ± 0.01	0.78 ± 0.01
Foopsi-RR [2]				0.66 ± 0.02	0.60 ± 0.01

3.3 DS achieves competitive results on simulated and publicly available imaging data

The advantages of our framework (black-box inference for different generative models, fast test-time performance through amortization, correlated posteriors through RNNs) are only useful if the approach can also achieve competitive performance. To demonstrate that this is the case, we compare our approach to alternative generative-model based spike prediction methods on data sampled from the SCF model– as this is the generative model underlying commonly used methods [10, 9], it is difficult to beat their performance on this data. We find that both DS-F and DS-NF achieve competitive performance, as measured by correlation between predicted firing rates and true (simulated) spike trains (Table 1, left column. Values are means and standard error of the mean calculated over cells).

To evaluate our performance on real data we compare to the current state-of-the-art method for spike inference based on generative models[12]. For these experiments we trained separate models on each of the GCaMP variants using the MLspike generative model. We achieve competitive accuracy to the results in [12] (see Table 1, values marked with an asterisk are taken from [12], Fig. 6d) and clearly outperform methods that are based on the linear SCF model. We note that, while our method performs inference in an unsupervised fashion and is trained using an un-supervised objective, we initialized our generative model with the mean values given in [12] (Fig. S6a), which were obtained using ground truth data. An example of inference and reconstruction using the DS-NF model is shown in Fig. 3. The reconstruction based on the true spikes (purple line) was obtained using the generative model parameters which had been acquired from unsupervised learning. This explains why the reconstruction using the inferred spikes is more accurate and suggests that there is a mismatch

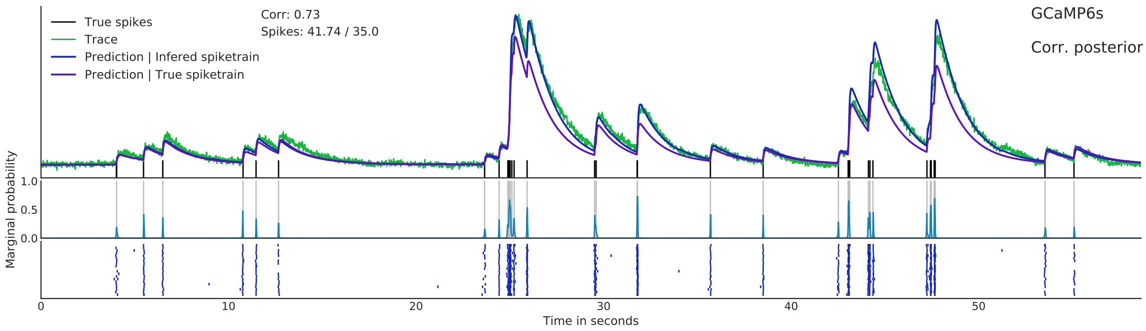


Figure 3: **Inference and reconstruction using the DS-NF algorithm on GECI data**. The reconstruction based on the inferred spike trains (blue) shows that the algorithm converges to a good joint model while the reconstruction based on the true spikes (purple) shows a mismatch of the generative model for high activity which results in an overestimate of the overall firing rate.

between the MLphys model and the true data-generating generating process. Developing more accurate generative models would therefore likely further increase the performance of the algorithm.

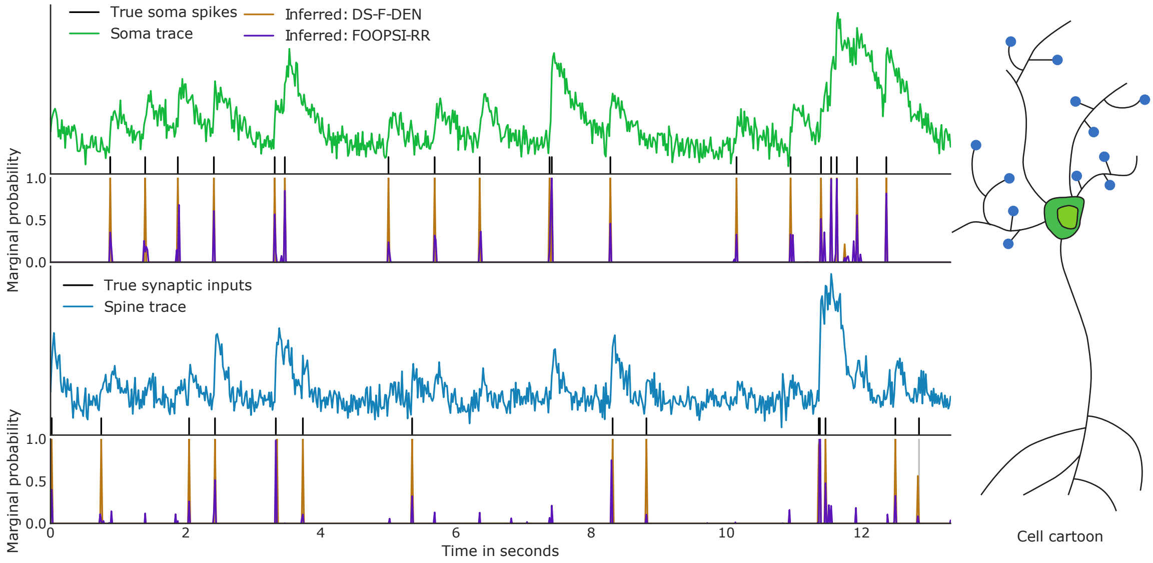


Figure 4: **Inference of somatic spikes and synaptic input spikes from simulated dendritic imaging data.** We simulated imaging data from our generative model, and compared our approach (DS-F-DEN) to an analysis inspired by [2] (Foopsi-RR), and found that our method can extract synaptic inputs more accurately. Traces at the soma and spines are used to infer somatic spikes and synaptic inputs at spines. Top: somatic trace and predictions. DS-F-DEN produces better predictions at the soma since it uses all traces to infer global events. Bottom: spine trace and predictions. DS-F-DEN performs better in terms of extracting synaptic inputs.

3.4 Extracting putative synaptic inputs from calcium imaging in dendritic spines

We generalized the DeepSpike variational-inference approach to perform simultaneous inference of backpropagating APs and synaptic inputs, imaged jointly across the entire neuronal dendritic arbor. We illustrate this idea on synthetic data based on the DS-F-DEN generative model (5). We simulated 15 cells each with 10 dendritic spines with a range of firing rates and noise levels. We then used a multi-input multi-output convolutional neural network (CNN, DS-F) in the non-amortized setting to infer a fully-factorized Bernoulli posterior distribution over global action potentials and local synaptic events.

We compared our results to an analysis technique inspired by [2] which we call Foopsi-RR. We first apply constrained deconvolution [33] to somatic and dendritic calcium traces, and then use robust linear regression to identify and subtract deconvolved components of the spine signal that correlated with global back-propagated action potential. Compared to the method suggested by [2], our model is significantly more accurate. The average correlation of our model is 0.84 for soma and 0.78 for spines, whereas for Foopsi-RR the average correlation is 0.66 for soma and 0.60 for spines (Table 1).

4 Discussion

Spike inference is an important step in the analysis of fluorescence imaging. We here propose a strategy based on variational autoencoders that combines the advantages of generative [7] and discriminative approaches [15]. The generative model makes it possible to incorporate knowledge about underlying mechanisms and thus learn from unlabeled data. A simultaneously-learned recognition network allows fast test-time performance, without the need for expensive optimization or MCMC sampling. This opens up the possibility of scaling up spike inference to very large neural populations [34], and to real-time and closed-loop applications. Furthermore, our approach is able to estimate full posteriors rather than just marginal firing rates.

It is likely that improvements in performance and interpretability will result from the design of better, biophysically accurate and possibly dye-, cell-type- and modality-specific models of the fluorescence measurement process, the dynamics of neurons [28] and indicators, as well as from taking spatial information into account. Our goal here is not to design such models or to improve accuracy *per se*, but rather to develop an inference strategy which can be applied to a large class of such potential generative models without model-specific modifications: A trained recognition model that can invert, and provide fast test-time performance, for any such model while preserving performance in spike-detection.

Our recognition model is designed to serve as the common approximate posterior for multiple, possibly heterogeneous populations of cells, requiring an expressive model. These assumptions are supported by prior work [15] and our results on simulated and publicly available data, but might be suboptimal or not appropriate in other contexts, or for other performance measures. In particular, we emphasize that our comparisons are based on a specific data-set and performance measure which is commonly used for comparing spike-inference algorithms, but which can in itself not provide conclusive evidence for performance in other settings and measures. Our approach includes rich posterior approximations [35] based on RNNs to make predictions using longer context-windows and modelling posterior correlations. Possible extensions include causal recurrent recognition models for real-time spike inference, which would require combining them with fast algorithms for detecting regions of interest from imaging-movies [10, 36]. Another promising avenue is extending our variational inference approach so it can also learn from available labeled data to obtain a semi-supervised algorithm [37].

As a statistical problem, spike inference has many similarities with other analysis problems in biological imaging– an underlying, sparse signal needs to be reconstructed from spatio-temporal imaging observations, and one has substantial prior knowledge about the image-formation process which can be encapsulated in generative models. As a concrete example of generalization, we proposed an extension to multi-dimensional inference of inputs from dendritic imaging data, and illustrated it on simulated data. We expect the approach pursued here to also be applicable in other inference tasks, such as the localization of particles from fluorescence microscopy [38].

5 Acknowledgements

We thank T. W. Chen, K. Svoboda and the GENIE project at Janelia Research Campus for sharing their published GCaMP6 data, available at http://crcns.org. We also thank T. Deneux for sharing his results for comparison and comments on the manuscript and D. Greenberg, L. Paninski and A. Mnih for discussions. This work was supported by SFB 1089 of the German Research Foundation (DFG) to J. H. Macke. A. Speiser was funded by an IMPRS for Brain & Behavior scholarship by the Max Planck Society.

References

[1] R. Y. Tsien, "New calcium indicators and buffers with high selectivity against magnesium and protons: design, synthesis, and properties of prototype structures," *Biochemistry*, vol. 19, no. 11, pp. 2396–2404, 1980.

[2] T.-W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. A. Kerr, M. B. Orger, V. Jayaraman, L. L. Looger, K. Svoboda, and D. S. Kim, "Ultrasensitive fluorescent proteins for imaging neuronal activity," *Nature*, vol. 499, no. 7458, pp. 295–300, 2013.

[3] J. N. D. Kerr and W. Denk, "Imaging in vivo: watching the brain in action," *Nat Rev Neurosci*, vol. 9, pp. 195–205, Mar 2008.

[4] C. Grienberger and A. Konnerth, "Imaging calcium in neurons.," *Neuron*, vol. 73, no. 5, pp. 862–885, 2012.

[5] S. L. Smith, I. T. Smith, T. Branco, and M. Häusser, "Dendritic spikes enhance stimulus selectivity in cortical neurons in vivo," *Nature*, vol. 503, no. 7474, pp. 115–120, 2013.

[6] T.-W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. A. Kerr, M. B. Orger, V. Jayaraman, *et al.*, "Ultrasensitive fluorescent proteins for imaging neuronal activity," *Nature*, vol. 499, no. 7458, pp. 295–300, 2013.

[7] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski, "Spike inference from calcium imaging using sequential monte carlo methods," *Biophysical journal*, vol. 97, no. 2, pp. 636–655, 2009.

[8] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *Journal of neurophysiology*, vol. 104, no. 6, pp. 3691–3704, 2010.

[9] E. Pnevmatikakis, J. Merel, A. Pakman, L. Paninski, *et al.*, "Bayesian spike inference from calcium imaging data," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, pp. 349–353, IEEE, 2013.

[10] E. A. Pnevmatikakis, D. Soudry, Y. Gao, T. A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, *et al.*, "Simultaneous denoising, deconvolution, and demixing of calcium imaging data," *Neuron*, 2016.

[11] E. Ganmor, M. Krumin, L. F. Rossi, M. Carandini, and E. P. Simoncelli, "Direct estimation of firing rates from calcium imaging data," *arXiv preprint arXiv:1601.00364*, 2016.

[12] T. Deneux, A. Kaszas, G. Szalay, G. Katona, T. Lakner, A. Grinvald, B. Rózsa, and I. Vanzetta, "Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo," *Nature Communications*, vol. 7, 2016.

[13] M. Pachitariu, C. Stringer, M. Dipoppa, S. Schröder, L. F. Rossi, H. Dalgleish, M. Carandini, and K. D. Harris, "Suite2p: beyond 10,000 neurons with standard two-photon microscopy," *bioRxiv*, 2017.

[14] D. Greenberg, D. Wallace, J. Vogelstein, and J. Kerr, "Spike detection with biophysical models for gcamp6 and other multivalent calcium indicator proteins," *2015 Neuroscience Meeting Planner. Washington, DC: Society for Neuroscience*, 2015.

[15] L. Theis, P. Berens, E. Froudarakis, J. Reimer, M. Román Rosón, T. Baden, T. Euler, A. S. Tolias, and M. Bethge, "Benchmarking spike rate inference in population calcium imaging," *Neuron*, vol. 90, no. 3, pp. 471–82, 2016.

[16] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[17] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator.," in *AISTATS*, vol. 1, p. 2, 2011.

[18] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[20] M. Titsias and M. Lázaro-Gredilla, "Doubly stochastic variational bayes for non-conjugate inference," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1971–1979, 2014.

[21] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015.

[22] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," *arXiv preprint arXiv:1402.0030*, 2014.

[23] A. Mnih and D. J. Rezende, "Variational inference for monte carlo objectives," in *Proceedings of the 33st International Conference on Machine Learning*, 2016.

[24] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

[25] L. Maaloe, C. K. Sonderby, S. K. Sønderby, and O. Winther, “Improving semi-supervised learning with auxiliary deep generative models,” in *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2015.

[26] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[27] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks.,” *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[28] V. Rahmati, K. Kirmse, D. Marković, K. Holthoff, and S. J. Kiebel, “Inferring neuronal dynamics from calcium imaging data using biophysical models and bayesian inference,” *PLoS Comput Biol*, vol. 12, no. 2, p. e1004736, 2016.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. ICML*, vol. 30, 2013.

[30] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[31] J. Friedrich, P. Zhou, and L. Paninski, “Fast Active Set Methods for Online Deconvolution of Calcium Imaging Data,” *arXiv.org*, Sept. 2016.

[32] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: A cpu and gpu math compiler in python,” in *Proc. 9th Python in Science Conf*, pp. 1–7, 2010.

[33] E. A. Pnevmatikakis, Y. Gao, D. Soudry, D. Pfau, C. Lacefield, K. Poskanzer, R. Bruno, R. Yuste, and L. Paninski, “A structured matrix factorization framework for large scale calcium imaging data analysis,” *arXiv preprint arXiv:1409.2903*, 2014.

[34] M. B. Ahrens, J. M. Li, M. B. Orger, D. N. Robson, A. F. Schier, F. Engert, and R. Portugues, “Brain-wide neuronal dynamics during motor adaptation in zebrafish,” *Nature*, vol. 485, pp. 471–7, May 2012.

[35] C. K. Sonderby, T. Raiko, L. Maaloe, S. K. Sonderby, and O. Winther, “How to train deep variational autoencoders and probabilistic ladder networks,” *arXiv preprint arXiv:1602.02282*, 2016.

[36] N. Apthorpe, A. Riordan, R. Aguilar, J. Homann, Y. Gu, D. Tank, and H. S. Seung, “Automatic neuron detection in calcium imaging data using convolutional networks,” in *Advances In Neural Information Processing Systems*, pp. 3270–3278, 2016.

[37] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, “Improving semi-supervised learning with auxiliary deep generative models,” in *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2015.

[38] E. Betzig, G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwartz, and H. F. Hess, “Imaging intracellular fluorescent proteins at nanometer resolution,” *Science*, vol. 313, no. 5793, pp. 1642–1645, 2006.

ARTICLES
https://doi.org/10.1038/s41592-021-01236-x

nature methods

Check for updates

Deep learning enables fast and dense single-molecule localization with high accuracy

Artur Speiser[1,2,3,4,12], Lucas-Raphael Müller[5,6,12], Philipp Hoess[5], Ulf Matti[5], Christopher J. Obara[7], Wesley R. Legant[8,9,10], Anna Kreshuk[5], Jakob H. Macke[1,2,3,11,13] ✉, Jonas Ries[5,13] ✉ and Srinivas C. Turaga[7,13] ✉

Single-molecule localization microscopy (SMLM) has had remarkable success in imaging cellular structures with nanometer resolution, but standard analysis algorithms require sparse emitters, which limits imaging speed and labeling density. Here, we overcome this major limitation using deep learning. We developed DECODE (deep context dependent), a computational tool that can localize single emitters at high density in three dimensions with highest accuracy for a large range of imaging modalities and conditions. In a public software benchmark competition, it outperformed all other fitters on 12 out of 12 datasets when comparing both detection accuracy and localization error, often by a substantial margin. DECODE allowed us to acquire fast dynamic live-cell SMLM data with reduced light exposure and to image microtubules at ultra-high labeling density. Packaged for simple installation and use, DECODE will enable many laboratories to reduce imaging times and increase localization density in SMLM.

Single-molecule localization microscopy (SMLM) (for example, PALM[1] and (d)STORM[2,3]) has become an invaluable super-resolution method for biology, as it can resolve cellular structures with nanometer precision. It is based on acquiring a large number of camera frames, in each of which only a tiny fraction of the emitters are stochastically activated into a bright 'on' state, so that their images do not overlap. This allows precise localization of the emitter coordinates by fitting a model of the point spread function (PSF). A super-resolution image is then reconstructed from these coordinates. This principle of SMLM is at the same time one of its main limitations: the need for sparse activation leads to long acquisition times. This results in low throughput, poor time resolution when imaging dynamic processes, low labeling densities and a reduced choice of fluorophores. Additionally, long acquisition times in combination with high excitation laser intensities needed for single-molecule imaging can cause strong phototoxicity in live-cell SMLM.

All of these limitations can be mitigated by activating emitters at a higher density. In this 'multi-emitter' setting, PSFs are no longer well-separated but may overlap, making both the detection of multiple nearby emitters and their accurate localization computationally challenging. This is not adequately addressed by existing algorithms: current 'multi-emitter' fitting algorithms[4–6] work reasonably well on two-dimensional (2D) samples where all emitters have the same z coordinate and thus produce identical PSFs. These algorithms, however, have had limited success for realistic three-dimensional (3D) biological structures. In a software competition that benchmarked SMLM algorithms using realistic computer-generated data, simple single-emitter fitters outperformed dedicated high-density fitters on 3D samples even in the high-density regime[7].

Deep learning is revolutionizing biological image analysis[8–10]. For SMLM, deep learning holds promise to extract emitter coordinates and additional parameters under conditions and densities too complex for traditional fitters. With enough training data, deep networks are flexible function approximators that can be trained to recognize patterns in the image and thus transform images directly into predicted emitter configurations, even for challenging high densities of emitters. While ground-truth data to train the neural network are typically not available, synthetic training data can be generated by numerically simulating the imaging process[11,12]. Convolutional neural networks (CNNs, a class of deep networks suitable for image data) have recently been used to extract parameters describing single isolated emitters such as color, emitter orientation, z coordinate, background or aberrations[13–16] and to design optimized PSFs[17]. Two recent studies (DeepSTORM3D[17] and DeepLoco[18]) used CNNs for extracting emitter coordinates, and outperformed traditional single-emitter fitting algorithms at densities higher than the single-molecule regime. These studies illustrate the potential of deep learning for SMLM, however, they have only been demonstrated either for exotic engineered PSFs or on simulated data.

Here we present the DECODE (deep context dependent) method for deep learning-based single-molecule localization that achieves high accuracy across a wide range of emitter densities and brightness levels. DECODE uses a deep network output representation, architecture, and cost function, which are optimized for simultaneous detection and subpixel localization of single emitters. Uniquely, DECODE is able to predict both the probability of detection and the uncertainty of localization for each emitter. As the timing and duration

[1]Machine Learning in Science, Excellence Cluster Machine Learning, Tübingen University, Tübingen, Germany. [2]Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany. [3]Research Center Caesar, Max Planck Society, Bonn, Germany. [4]International Max Planck Research School Brain and Behavior, Bonn, FL, USA. [5]Cell Biology and Biophysics Unit, European Molecular Biology Laboratory, Heidelberg, Germany. [6]Ruprecht Karls University of Heidelberg, Heidelberg, Germany. [7]HHMI Janelia Research Campus, Ashburn, VA, USA. [8]Joint Department of Biomedical Engineering, UNC, Chapel Hill, NC, USA. [9]NCSU Raleigh, Raleigh, NC, USA. [10]Department of Pharmacology, University of North Carolina, Chapel Hill, NC, USA. [11]Department Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen, Germany. [12]These authors contributed equally: Artur Speiser, Lucas-Raphael Müller. [13]These authors jointly directed this work: Jakob H. Macke, Jonas Ries, Srinivas C. Turaga. ✉e-mail: Jakob.Macke@uni-tuebingen.de; jonas.ries@embl.de; turagas@janelia.hhmi.org

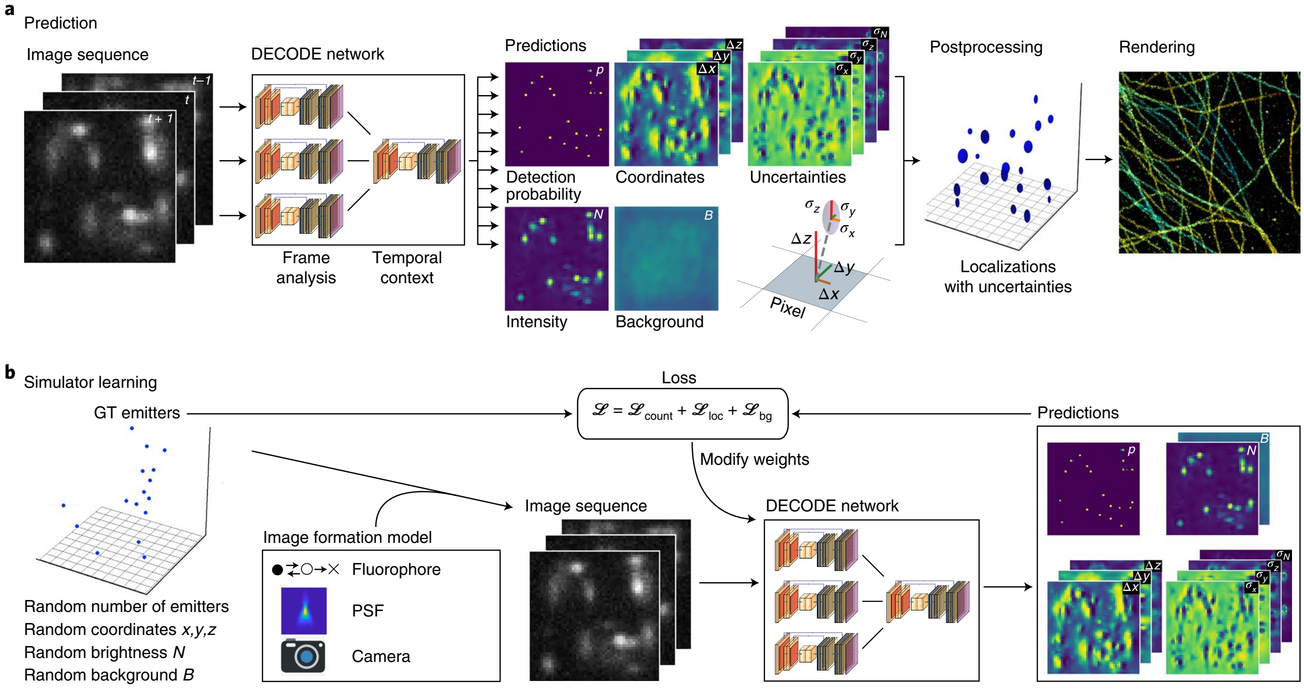


Fig. 1 | DECODE for high-density single-molecule localization. a, DECODE architecture. The DECODE network uses information from multiple frames to predict output maps representing for each pixel the probability of detecting an emitter and the emitter's subpixel spatial coordinates ($\Delta x, \Delta y, \Delta z$), brightness ($N$), the uncertainty related to those predictions ($\sigma_x, \sigma_y, \sigma_z$), and an optional background map (B). **b**, Training DECODE. The DECODE network is trained by simulator learning. Ground-truth (GT) emitter coordinates are generated randomly, and synthetic images simulated from a forward model of the image formation process are passed through the DECODE network. The loss quantifies the probability that the GT explains the output predictions and is optimized during training.

of emitter activations are stochastic, they regularly persist over several imaging frames. The DECODE architecture can integrate information across neighboring frames ('temporal context'), which improves emitter detection and localization.

In the public SMLM challenge[7], DECODE outperformed all existing methods on 12 out of 12 datasets. Compared to previous deep learning-based high-density fitters[17], DECODE is ten times faster and up to twice as accurate, and can be applied to a wide range of PSFs. We demonstrate on biological structures that DECODE allows for fivefold higher labeling densities or tenfold faster imaging compared to imaging in the single-emitter regime, and thus enables fast live-cell SMLM with reduced light exposure and visualization of dynamic processes. We show the versatility of DECODE by reanalyzing a published lattice light sheet (LLS) point accumulation for imaging of nanoscale topography labeling (PAINT) dataset[19] for which we could substantially improve fluorophore detection and localization accuracy. DECODE is packaged for simple use and can be easily trained and used by nonexpert users, without having to design new network architectures. Thus, it will enable the entire community to overcome the need of sparse activation as one of the main bottlenecks in SMLM.

Results

DECODE network. DECODE introduces a new output representation and architecture for detecting and localizing emitters. For each image frame, it predicts multiple channels with the same dimensions as the input image (Fig. 1a). The first two channels indicate the probability p that an emitter exists near that pixel, as well as its brightness N (number of photons emitted by the emitter in the frame). The next three channels describe the coordinates of the emitter with respect to the center of the pixel, $\Delta xyz = (\Delta x, \Delta y, \Delta z)$. An additional channel predicts the background intensity B in each pixel.

This architecture overcomes limitations of current deep learning[17,18] and non-deep learning–based high-density approaches in three ways. First, DECODE predictions scale only with the number of imaged pixels (not super-resolution voxels as in DeepSTORM3D), resulting in over 20-fold improvement in prediction speeds and the use of continuous subpixel coordinates eliminates a voxel size dependent limit on precision. The local output representation used by DECODE also avoids the potentially challenging nonlocal mapping of pixels to global coordinates used in DeepLoco.

Second, DECODE has four additional output channels that estimate the uncertainty of the localization along each coordinate given by $\sigma_{xyz} = (\sigma_x, \sigma_y, \sigma_z)$ and of the brightness σ_N. These predicted localization uncertainties can be used to filter out poorly localized detections to improve the rendering of super-resolution images. In addition, training the network to additionally predict the localization uncertainty corresponding to each detection also helps to improve the quality of the detection probabilities p by implicitly grouping all the detections corresponding to the same emitter. In contrast, standard output representations that only indicate the probability of detecting an emitter on a per-voxel basis make it more challenging to correctly group detection probability voxels corresponding to the same emitter in high emitter-density and high localization-uncertainty scenarios.

Third, the DECODE network integrates information across multiple frames with a two-stage design. The first stage (frame analysis module) analyses single imaging frames using a 2D multi-resolution convolutional network based on the 'U-Net' architecture[20] to compute a feature representation of the single frame (Extended Data Fig. 1). The second stage (temporal context module) integrates the feature representations of the frame with those of the previous and next imaging frame using a second 2D U-Net to produce the final predictions. As emitters persist over several frames, this improves detection and localization accuracy.

Training the DECODE network using simulator learning. We train DECODE to simultaneously detect and localize emitters in SMLM measurements. Ground-truth data for supervised learning are not easily available for SMLM. However, it is possible to simulate realistic images of activated emitters as the physics of imaging single molecules is well understood[12]. We train the DECODE network by generating a large amount of simulated data. To avoid structural bias[8], we place emitters at random coordinates, and calculate simulated images with a realistic image formation model that includes dye photophysics, a measured PSF and camera noise (Methods).

We trained the DECODE network to predict the probability of detection, along with the subpixel localization and localization uncertainty of each detected emitter. Our loss function has three terms: (1) a count loss that compares the true and detected number of emitters in the image; (2) a localization loss that trains the network to correctly localize the detected emitters and estimate the localization uncertainty and emitter brightness and (3) an optional background loss. The count and localization loss functions were derived together as an approximation to a spatial point process probability distribution. They work together to correctly train the DECODE network to predict one detection per emitter, and to correctly assign the localization uncertainty of each emitter to the corresponding detection. Together, they constitute a new loss for counting, detecting and localizing sets of discrete point-like objects.

The count loss first constructs a Gaussian approximation to the predicted number of emitters by summing the mean and the variance of the Bernoulli detection probability map, and then maximizes the probability of the true number of emitters under this distribution. Uncertain detections will lead to large predicted count variance, while confident detections will result in low variance. Thus, the count loss encourages a detection probability map with sparse but confident predictions. The localization loss models the distribution of subpixel localizations Δxyz with a coordinate-wise independent Gaussian probability distribution[21] with standard deviation σ_{xyz}. For imprecise localizations, this probability is maximized for large σ_{xyz}, for precise localizations for small σ_{xyz}. The distribution of all localizations over the entire image is approximated as a weighted average of individual localization distributions, where the weights correspond to the probability of detection. By optimizing both the probability of detection, the subpixel localization Δxyz and σ_{xyz} simultaneously, the network learns not only the best predictions for the coordinates of the emitters, but also the best estimate for their localization uncertainties. The emitter brightness predictions N and their uncertainties σ_N are optimized similarly. Finally, the optional background loss computes the mean squared error between the true and predicted background images B. While the network only uses camera images to make predictions, the network training procedure does require PSF calibration measurements.

DECODE achieves high accuracy for a wide range of simulated data. *Performance metrics.* The quality of SMLM data analysis is commonly quantified by two factors: first, the detection accuracy quantifies the fraction of emitters that are detected. The metric we use here is the Jaccard Index (JI)[7], that sets the true positives (TP) in relation to the false positives (FP) and false negatives (FN), $\mathrm{JI} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP})$. The second factor is the localization error, that is how close the measured coordinates are to the true coordinates, measured here as the root mean squared error (r.m.s.e.) averaged over the dimensions (Methods). We matched the detected emitters to the ground-truth emitters in three dimensions with a lateral threshold of 250 nm and an axial threshold of 500 nm.

There is a natural trade-off between JI and localization error: discarding all but the brightest and best separated emitters will result in a good (low) localization error but a bad (low) JI. Conversely, including also poorly localized emitters might improve JI, but deteriorates the localization error. The optimal operating point between these two extremes will depend on the experimental conditions and the scientific question. Because DECODE also provides uncertainties for each localization, it offers a straightforward way to filter localizations and thus set the desired balance between the number of detected emitters and the localization error that can be tolerated.

The Cramér–Rao lower bound (CRLB) gives the minimum achievable localization error for an optimal fitter given a known PSF, background and noise model[22]. Most commonly, it is calculated under idealized conditions (that is, nonoverlapping PSFs, homogeneous background, assuming the chosen PSF model to be the true model) and we use it here for comparison as a best-case limit for localization error.

DECODE approaches the CRLB for low densities. We simulated 100,000 frames with exactly one emitter per frame at random coordinates with a constant brightness and background, and trained DECODE without temporal context. On these data with sparse activations, DECODE approaches the single-emitter CRLB, that is the theoretical limit of precision (Fig. 2a). It thus performs as well as maximum likelihood estimation (MLE) based fitters, which have also been shown to reach the CRLB[23] in this regime.

DECODE's uncertainty estimates are well calibrated. In the high-density regime, DECODE's σ predictions correlate closely to the measured localization error (Fig. 2b), much better than the single-emitter CRLB estimate that assumes isolated emitters (correlation coefficient 0.86 for σ versus 0.07 for single-emitter CRLB). For the low-density regime, the uncertainty estimates are in line with the measured error and the single-emitter CRLB (Fig. 2a).

Temporal context improves localization error and detection. DECODE's temporal context module pools information across multiple (we used three) frames, to model the fact that emitters can persist in multiple subsequent frames. Use of this context module improves both the detection accuracy (JI) and the localization error (Fig. 2c). The increase in JI is apparent for all densities and signal to noise ratios (SNRs). In addition, the r.m.s.e. is reduced by up to 20 nm. Overall, the temporal context has a large impact across imaging conditions, and is also more powerful than 'grouping' approaches that are often applied to localizations in a postprocessing step (Extended Data Fig. 2).

DECODE architecture outperforms a voxel-based network architecture and a multi-emitter fitter. To assess how the DECODE network architecture performs against other deep learning-based and iterative methods, we directly compared to DeepSTORM3D[17] and CSpline[4], a matching pursuit style multi-emitter fitter based on MLE, using the code provided by the authors. To minimize the risk of suboptimal training, we trained DeepSTORM3D on data sampled from our generative model using the same parameters we used for the training of DECODE. For both DeepSTORM3D and CSpline we performed a parameter grid search over user-defined parameters to maximize their performance (measured as efficiency score[7]). To facilitate the comparison of localization precision, we filtered out DECODE localizations with the highest inferred uncertainties such that the remaining number match DeepSTORM3D. DECODE outperforms the other methods across all densities and SNRs (Fig. 2e and Extended Data Fig. 3) even without temporal context. When we use temporal context, DECODE reduces the localization error up to twofold compared to DeepSTORM3D. Although both methods are based on deep learning, this performance improvement is

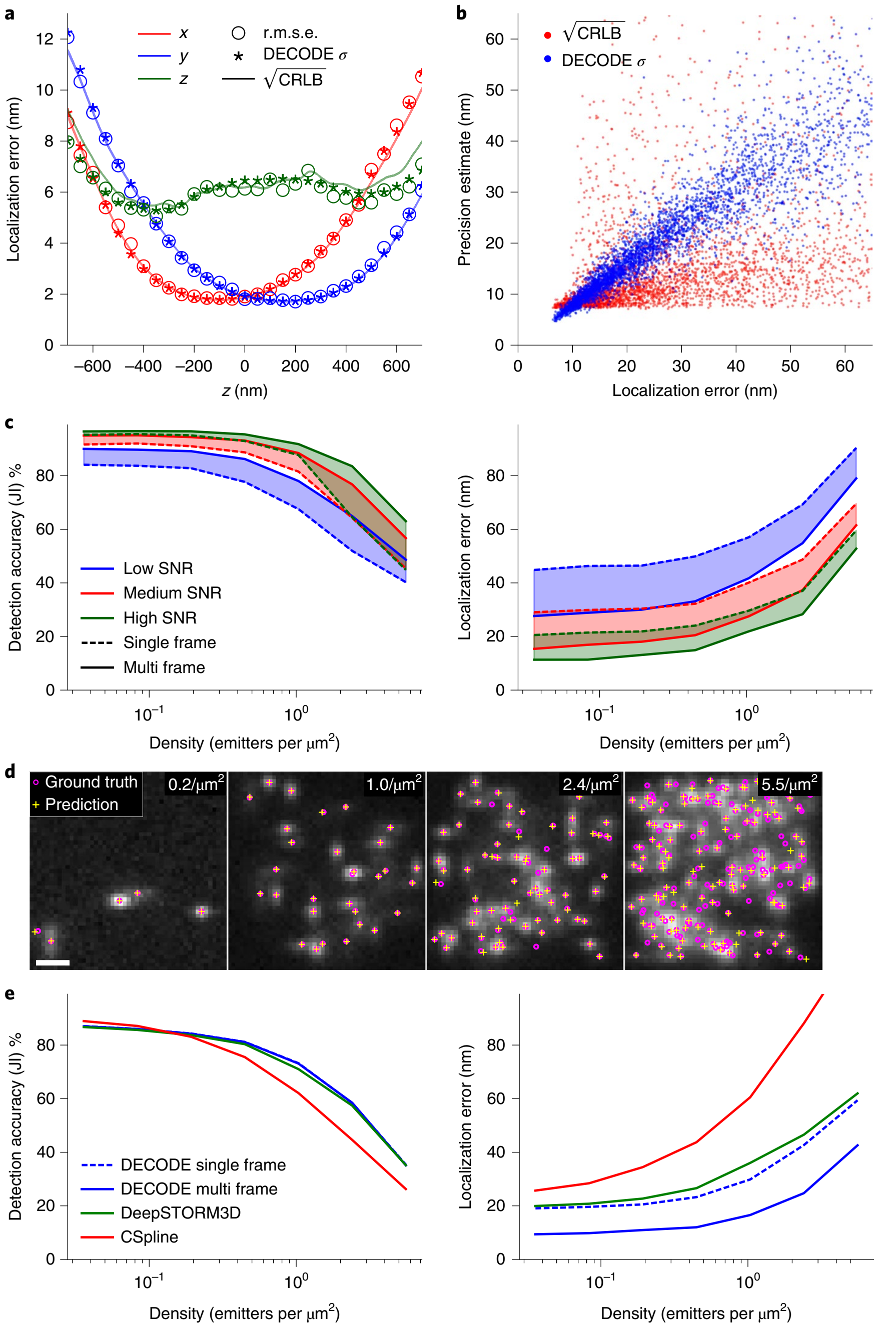


Fig. 2 | Performance of DECODE on simulated data. a, DECODE reaches the single-emitter CRLB for isolated emitters. The r.m.s.e. and DECODE σ averaged over 50 nm bins (additional comparisons in Extended Data Fig. 4). **b**, Comparison of the predicted localization uncertainty, σ, and measured localization error for densely activated emitters. We simulated the same dense emitter configuration 100 times and calculated the measured localization error as the r.m.s.e. of the predictions of the coordinates. Also shown is the (square root of the) single-emitter CRLB. See Supplementary Fig. 1 for comparisons of individual axes. **c**, Impact of temporal context on detection performance and localization error. Detection accuracy and localization error of DECODE trained with (multiframe) and without (single frame) temporal context quantified as a function of emitter density on simulations with low, medium and high SNR. **d**, Representative simulated frames with ground-truth coordinates (magenta circles) and predicted coordinates (yellow crosses) for the densities used in **c** and medium SNR. **e**, Comparison of DECODE with CSpline and DeepSTORM3D over a wide range of densities. See Extended Data Figs. 2 and 3 for additional comparisons with different conditions and metrics. The standard error of the mean (s.e.m.) on the localization error lies between 0.2 and 0.4 nm. See Methods and Supplementary Table 1 for additional details on training and evaluation.

based on the differences in output representation and loss function between DECODE and DeepSTORM3D. The localization error of DeepSTORM3D is limited by the super-resolution voxel size[17] (Extended Data Fig. 4), which prevents the method from achieving the single-emitter CRLB, unlike DECODE that has no such limitation. Because DECODE has multiple output maps it is also able to

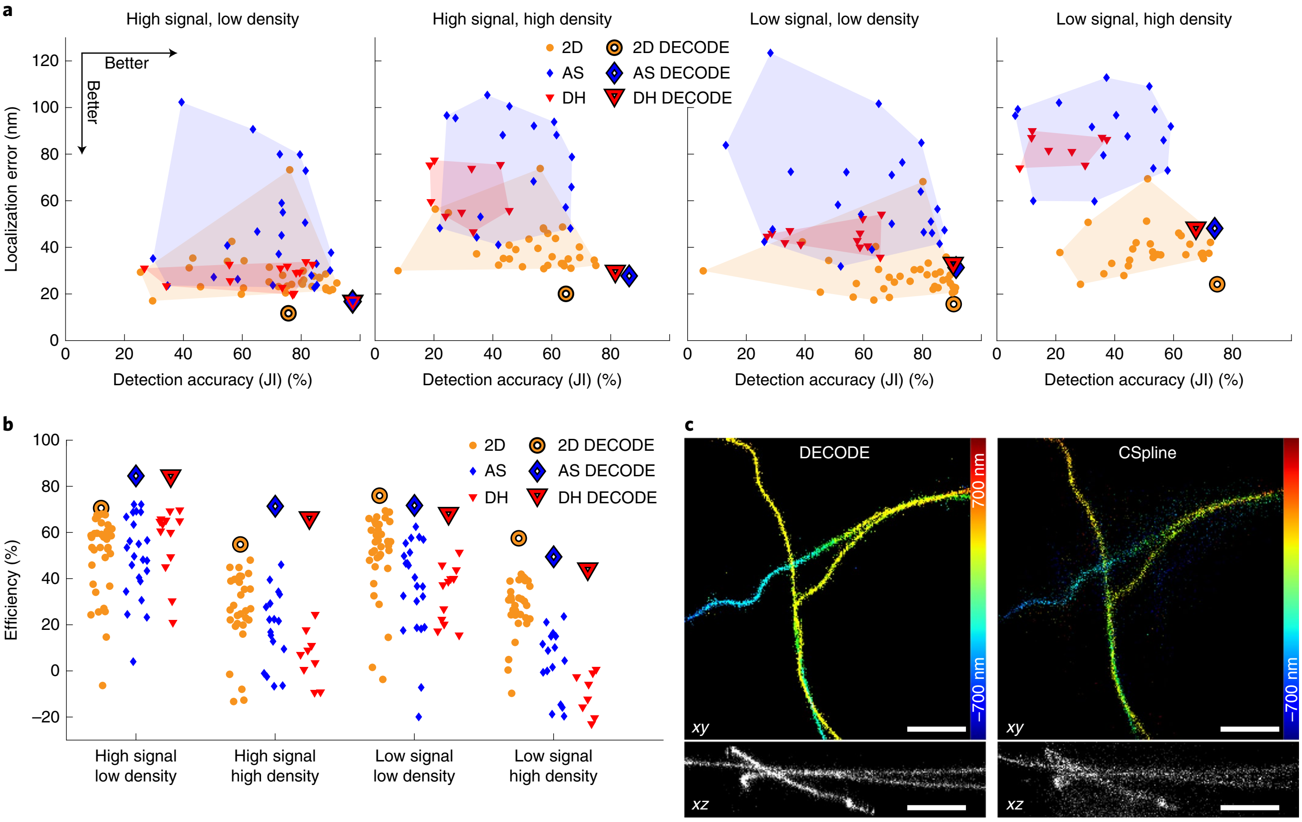


Fig. 3 | Performance comparison on the SMLM 2016 challenge. **a**, Performance evaluation on the 12 test datasets with low or high density, low or high SNR and different modalities (2D, AS, astigmatic; DH, double helix) using the detection accuracy (JI, higher is better) and localization error (lower is better) as metrics. Each marker indicates a benchmarked algorithm, large markers indicate DECODE. **b**, Efficiency scores (higher is better) where each marker indicates performance for one method. Metrics were calculated by the SMLM 2016 challenge and downloaded from the challenge website (http://bigwww.epfl.ch/smlm/challenge2016/leaderboard.html). **c**, Reconstructions by DECODE and the CSpline algorithm on the high density, low signal double-helix challenge training data. Upper panels show the *xy* view, color coded by the *z* coordinate and the lower panels show the *xz* reconstructions. Scale bars, 1 µm. See Supplementary Figs. 5 and 6 for additional comparisons with DeepSTORM3D on training datasets.

provide accurate estimates of the signal photon counts and background values (Extended Data Fig. 9).

Notably, DECODE performs favorably in fitting time (Extended Data Fig. 6), taking less than 1.5 s to analyze 1,000 frames of 64 × 64 pixels, while DeepSTORM3D requires between 34 and 54 s and CSpline requires between 14 and 2,680 s, which is up to 1,900-fold slower than DECODE. Training the DECODE network to convergence on a NVIDIA RTX2080Ti graphical processing unit (GPU) requires around 10 h while DeepSTORM3D takes around 50 h.

DECODE outperforms all fitters on a public SMLM benchmark. The 2016 SMLM challenge is an on-going and continuously updated second generation comprehensive benchmark evaluation developed for the objective, quantitative evaluations of the plethora of available localization algorithms[7,24]. It offers synthetic datasets for training, created to emulate various experimental conditions. To avoid overfitting, evaluations are carried out on data not shared with contestants. It calculates various quality metrics, among them r.m.s.e. lateral or volume localization error, as applicable for 2D and 3D data, respectively, the JI quantifying detection accuracy and a single 'efficiency' score that combines r.m.s.e. and JI. The performance of DECODE in the SMLM 2016 challenge, including extensive evaluations and side by side comparisons, is available online (http://bigwww.epfl.ch/smlm/challenge2016/leaderboard.html). DECODE outperformed all 39 algorithms on 12 out of 12 datasets, often by a substantial margin (Fig. 3, data from challenge website, current as of 1 October, 2020). The datasets included high (N1) and low (N2, N3) SNRs, with low or high emitter densities, with 2D, astigmatism and double-helix PSF-based imaging modalities.

DECODE achieves an average efficiency score of 66.6% out of the best possible score of 100% (achievable only by a hypothetical algorithm that accurately detects 100% emitters with 0 nm localization error). This is compared to an average score of 48.3 and 45.6% for all second and third place algorithms, respectively. The difference is particularly large under difficult imaging conditions, when high emitter densities and low SNR can conspire to make detection and localization challenging, particularly so for the double-helix PSF. For example, compared to the second-best algorithm (SMAP2018) in the low-SNR/high-density/double-helix condition, DECODE improves the localization error from 75.2 to 48.4 nm and the JI from 30.0 to 67.5%.

DECODE enhances super-resolution reconstructions by improving both the detection and the localization of single molecules. An example of this can be seen in Fig. 3c, where we compare the reconstruction obtained with DECODE and CSpline[4] on a high-density 3D double-helix dataset (using settings provided by the authors, github.com/ZhuangLab/storm-analysis). Other deep learning-based approaches have not yet submitted their results. However, we performed comparisons to DeepSTORM3D on

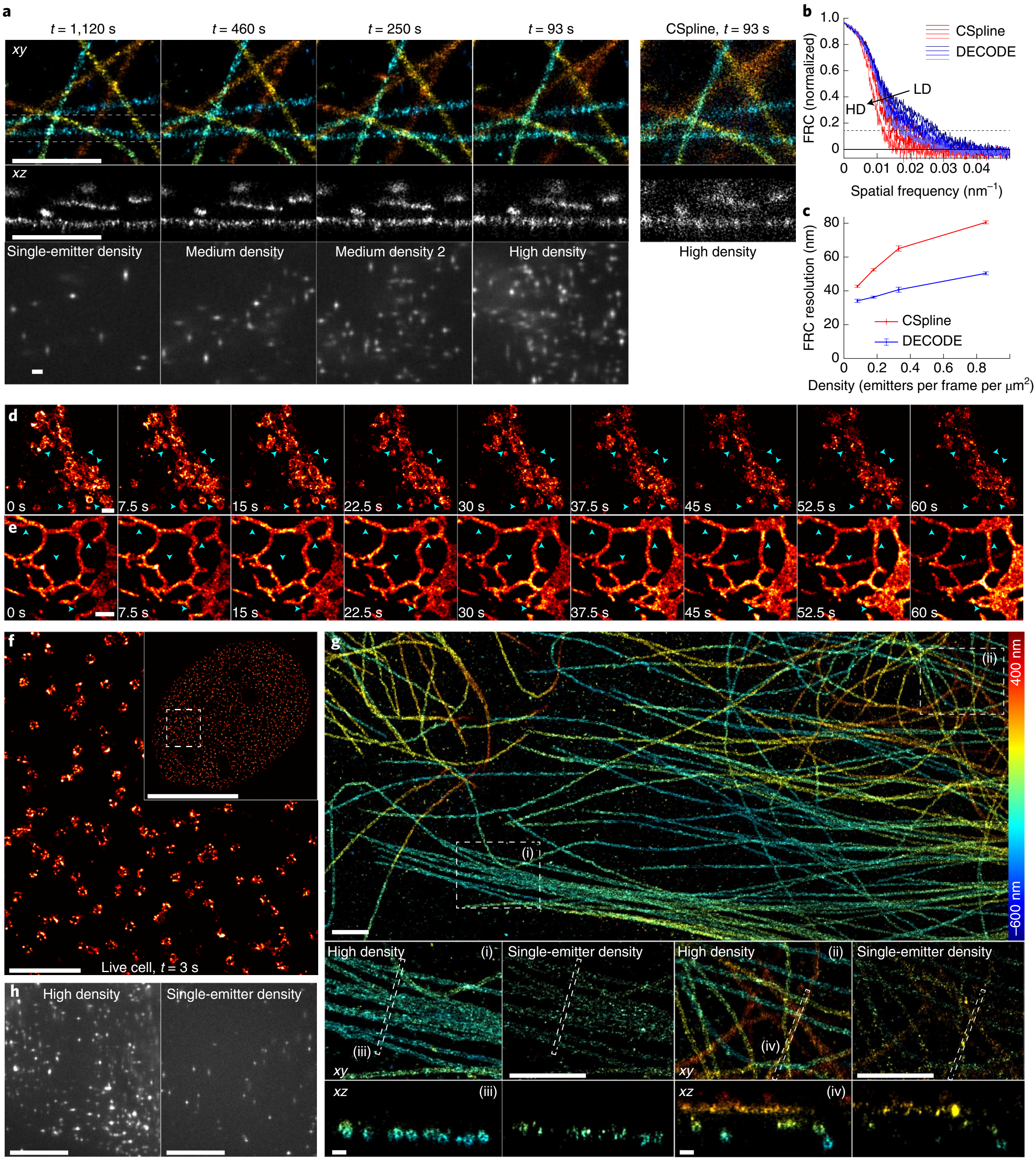


Fig. 4 | DECODE enables high-speed and live-cell SMLM and ultra-high labeling densities. a, DECODE can reduce acquisition times by one order of magnitude. The same sample of microtubules, labeled with anti-α-tubulin primary and AF647 secondary antibodies, imaged with different ultraviolet activation intensities to result in different emitter densities per frame, between 0.08 and 0.86 µm^{-2} and acquisition times between 93 and 1,120 s, while keeping the total number of localizations the same. For high-density activation, we show a comparison with CSpline. **b**, Fourier ring correlation (FRC) curves for DECODE and CSpline for different emitter densities. HD, high density; LD, low density. **c**, Resolution estimates obtained using the FRC and 0.143 criterion across densities for both methods. **d**, Fast live-cell SMLM on the Golgi apparatus labeled with α-mannosidase II-mEos3.2 (Supplementary Video 1). **e**, Fast live-cell SMLM on the endoplasmic reticulum labeled with calnexin-mEos3.2 (Supplementary Video 2 and Supplementary Fig. 3). **f**, Fast live-cell SMLM on the nuclear pore complex protein Nup96-mMaple acquired in 3 s. **g**, DECODE enables ultra-high labeling densities. Microtubules labeled with a high concentration of anti-α and anti-β-tubulin primary and Alexa Fluor 647 secondary antibodies. **g**(i),(ii), Magnified regions as indicated in **g**. Data acquired with high-density labeling show continuous structures. As a comparison, the same sample was acquired after prebleaching of the fluorophores to reach the single-molecule blinking regime. Here, single labels are resolved in the super-resolution reconstruction and lead to a sparse decoration of the microtubules. **g**(iii),(iv), Side view reconstructions of regions as indicated in **g**(i),(ii) resolving the hollow, cylinder-like structure of immunolabeled microtubules. **h**, Representative raw camera frames for the high-density and single-emitter acquisitions, respectively. Scale bars, 10 µm (**f** inset, **h**), 1 µm (**a**,**d**,**e**,**f**,**g**,**g**(i),**g**(ii)) and 100 nm (**g**(iii),**g**(iv)).

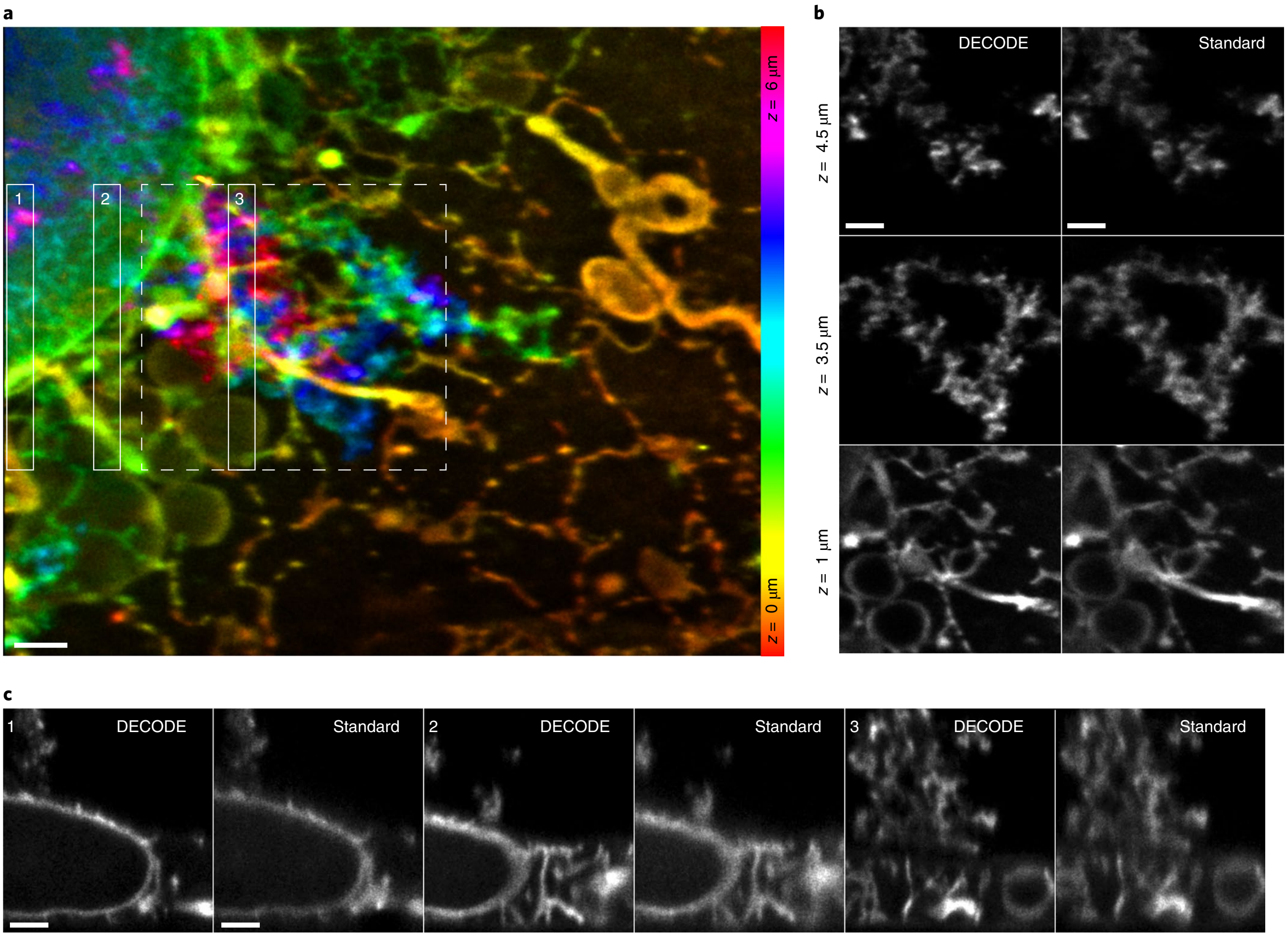


Fig. 5 | DECODE improves resolution in LLS PAINT. a, COS-7 cell imaged with LLS PAINT microscopy, overview. Data from Legant et al.[30], 70,000 volumes imaged over 2.7 days. **b**, 500-nm-thick slices of the region indicated in **a** (dashed line), comparing DECODE analysis and the original analysis using MLE fitting (standard analysis). **c**, Perpendicular (side view) reconstructions of 500-nm-thick regions as indicated in **a** comparing DECODE and standard analysis. Scale bars, 1 µm. See Extended Data Fig. 7 for additional comparisons.

low-SNR high-density training datasets and again achieved superior results (efficiency score of 51 against 32% on double helix and 45 against 31% on astigmatism data, Supplementary Figs. 5 and 6). Thus, DECODE is setting new quantitative standards for localization algorithms, across both low and high SNRs and densities.

Considerations. As with any fitter, DECODE relies on an accurate PSF model and proper parameters, otherwise artifacts will dominate the predictions. When the localization uncertainty is large, for very dim and dense localizations far from the focal plane, DECODE has a bias toward predicting localizations close to the pixel center. This effect can be overcome by filtering out localizations with large predicted uncertainty, or by rendering every localization with a Gaussian the size of the localization uncertainty, effectively dispersing these large uncertainty localizations over the pixel (Extended Data Fig. 8 and Methods).

DECODE reduces imaging times by one order of magnitude. By enabling accurate emitter localization at high densities of more than 2.5 µm^{-2} per frame (Fig. 2c), DECODE can yield high-quality super-resolution reconstructions with much shorter imaging times. We demonstrate this by imaging and reconstructing the same sample of labeled microtubules at four different activation laser powers using STORM (stochastic optical reconstruction microscopy)[2,3]. This results in different emitter densities per frame between 0.08 and 0.86 µm^{-2}. The imaging time was chosen to result in the same number of total localizations and decreased from 1,120 to 460 and 250 and 93 s for stronger activation.

We trained and applied one common DECODE model to all four datasets (Fig. 4a). Whereas CSpline reconstructions quickly degrade with high emitter densities, DECODE consistently yields reconstructions with high accuracy even for the densest sample. We quantified the lateral resolution using Fourier Ring Correlation (FRC)[25], which estimates resolution by measuring the correlation of two different reconstructions of the same image across spatial frequencies. DECODE consistently improves the *x*,*y* resolution by 20–30 nm over CSpline across all imaging densities (Fig. 4b,c) while detecting around 30% more localizations.

DECODE enables fast live-cell SMLM with reduced light exposure. Fast imaging is especially relevant for live-cell SMLM where the dynamics of the biological system under investigation dictate the necessary time resolution. At the same time, fast imaging usually requires high laser powers, deteriorates resolution[26] and leads to substantial phototoxicity[27]. As DECODE allows activating emitters to high density, it enables faster imaging with decreased light dose for a given number of localizations. We were able to image dynamic changes of the Golgi apparatus (Fig. 4d) and the endoplasmic reticulum (Fig. 4e) with 7.5 s temporal resolution. We imaged nuclear pore complexes in living cells[28] within only 3 s (Fig. 4f), seven times faster than our previous speed-optimized live-cell SMLM[26] and with a 70% reduced light dose.

DECODE enables ultra-high labeling densities. Labeling densities in SMLM are fundamentally limited by the fraction of emitters that are in the bright state. For the best performing fluorophore Alexa Fluor 647, even without ultraviolet activation about 0.05% of the emitters are in the bright state[29] due to activation by the red imaging laser and spontaneous activation. For the single-emitter blinking regime (activated emitter density $<0.1\,\mu m^{-2}$), this limits the number of total emitters to about $200\,\mu m^{-2}$. For higher labeling, prebleaching can be used to reduce the number of emitters to this regime, but the resulting low labeling limits the resolution[19] and in the super-resolution reconstructions sparse individual emitters become dominant (Fig. 4g). With DECODE, we can now image densely labeled samples that previously were inaccessible. We demonstrated this on immunolabeled microtubules that were labeled about fivefold higher than compatible with single-emitter fitting, resulting in much smoother and denser decoration of the microtubules (Fig. 4g). In 50-nm-thick orthogonal reconstructions, only the densely labeled microtubules were resolved as hollow cylinders, whereas after prebleaching to single-emitter blinking, these reconstructions only showed individual emitters (Fig. 4g(iii)(iv)). Additional comparisons with DeepSTORM3D highlight that the superior output representation and loss function of DECODE are critical to reach the optimal resolution for this dataset (Extended Data Fig. 5).

DECODE enables high fidelity reconstructions of 3D LLS PAINT. To illustrate the general applicability of DECODE, we applied it to 3D LLS microscopy combined with the PAINT technique[19]. In PAINT microscopy, the fluorophore labeling a sample stochastically binds and unbinds from the sample, providing dense labeling. In LLS microscopy, thick volumes are imaged at high resolution by scanning a thin (1.1 µm) light sheet, with axial localization within the sheet enabled by astigmatism.

Single-molecule localization in LLS PAINT is usually performed frame-wise using MLE fitting[30]. However, an emitter is visible in several adjacent *z* planes in the volumetric dataset. Thus, similar to exploiting the temporal context, we now use the same spatio-temporal context by analyzing three adjacent frames in the *z* stack at the same time to improve detection accuracy and localization error.

We reconstructed a previously reported dataset of a chemically fixed COS-7 cell with intracellular membranes labeled by azepanyl-rhodamine (AzepRh)[19,30] consisting of 70,000 3D volumes comprising more than 10 million 2D images acquired in 270-nm steps. DECODE detected 500 million emitters, compared to 200 million emitters detected by the original algorithm. Thus, for a comparable quality of the reconstruction, only half of the frames are needed, reducing imaging times by over a day from 2.7 to 1.35 days (Extended Data Fig. 7). At the same time, improved accuracy of DECODE results in sharper reconstructions (Fig. 5).

Discussion

We presented DECODE, a new deep learning-based method for single-molecule localization that performs exceptionally well on dense 3D data. DECODE differs from traditional localization algorithms by simultaneously performing detection and localization of emitters. It can be used in a flexible and general manner for a wide range of imaging parameters (including arbitrary PSFs and noise models) and imaging modalities such as 3D LLS PAINT imaging. In a publicly available benchmark challenge, it is the best performing algorithm in every condition, and often improves both localization and detection accuracy by a large margin. By making use of the temporal context, DECODE improves detection accuracy and localization error of emitters that are active across multiple imaging frames. Temporal context is also used by postprocessing steps in SMLM relying on 'merging' or 'grouping' of localizations, in which localizations occurring in consecutive images that are closer to each other than a fixed threshold are assumed to belong to the same emitter and their coordinates are averaged, weighted by the uncertainty of each localization. However, grouping does not improve detection of emitters, and it fails for dense or dim emitters whose localizations cannot be linked unambiguously across frames.

DECODE not only predicts coordinates of emitters, but also their uncertainty. This is highly useful for filtering out imprecise localizations, for reconstruction of super-resolution images in which every localization is rendered as a Gaussian with a size proportional to the coordinate uncertainty and as weights for quantitative coordinate-based analysis of SMLM data.

We demonstrated the performance of DECODE on various experimental SMLM datasets. We could show that the excellent performance on high-density data can increase the achievable localization density or decrease imaging times by one order of magnitude. This allowed us to perform live-cell measurements on nuclear pore complexes with high temporal resolution and reduced light exposure, and to achieve ultra-high labeling on microtubules. LLS PAINT data analyzed with DECODE showed markedly improved resolution due to substantial improvements in emitter detection and localization error.

Prediction of coordinates with DECODE can be as fast as GPU-based MLE fitters for sparse activation, but greatly outperforms those for high densities, as the computational complexity of DECODE depends only on the size of the image and not the number of emitters in each imaging frame. However, it requires the training of a new neural network whenever the optical properties of the microscope change. This training can currently take over 10 h on a single GPU, but after just 2 h of training time, the localization error is within 1 nm and the JI within 2% of the final value (Extended Data Fig. 10). To reduce training times further, one can likely take an existing network and fine-tune its parameters using a smaller number of simulations, rather than training it from scratch. Ultimately, it may be possible to train a single network across multiple parameter settings or even PSFs, so that the same network can 'amortize' inference across multiple experimental settings. To make DECODE easily usable by the entire community, we distribute it as a Python-based open-source software package based on the PyTorch[31] deep learning library. We provide precompiled, easily installable code, along with detailed tutorials and integration into the SMAP SMLM analysis software[32]. To enable anyone to directly use DECODE for training and prediction without relying on previous programming knowledge and dedicated local hardware, we deploy these Jupyter notebooks in Google Colab, complementing a recent initiative to make deep learning-based image analysis tools accessible to nonexperts at minimal cost[33]. Thus, DECODE will enable a large community to directly perform SMLM in a new high-density regime with greatly increased imaging speeds or localization densities and excellent localization and detection accuracy.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-021-01236-x.

Received: 23 October 2020; Accepted: 7 July 2021;
Published online: 3 September 2021

References

1. Betzig, E. et al. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**, 1642–1645 (2006).
2. Rust, M. J., Bates, M. & Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat. Methods* **3**, 793–796 (2006).
3. Van de Linde, S. et al. Direct stochastic optical reconstruction microscopy with standard fluorescent probes. *Nat. Protocols* **6**, 991–1009 (2011).
4. Babcock, H. P. & Zhuang, X. Analyzing single molecule localization microscopy data using cubic splines. *Sci. Rep.* **7**, 552 (2017).
5. Babcock, H., Sigal, Y. M. & Zhuang, X. A high-density 3d localization algorithm for stochastic optical reconstruction microscopy. *Opt. Nanoscopy* **1**, 6 (2012).
6. Ovesny, M., Krizek, P., Borkovec, J., Svindrych, Z. & Hagen, G. M. Thunderstorm: a comprehensive ImageJ plug-in for palm and storm data analysis and super-resolution imaging. *Bioinformatics* **30**, 2389–2390 (2014).
7. Sage, D. Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software. *Nat. Methods* **16**, 387–395 (2019).
8. Belthangady, C. & Royer, L. A. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods* **16**, 1215–1225 (2019).
9. Ching, T. Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 20170387 (2018).
10. Weigert, M. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090 (2018).
11. Le, T. A., Baydin, A. G., Zinkov, R., and Wood, F. Using synthetic data to train neural networks is model-based reasoning. In *Proc. International Joint Conference on Neural Networks (IJCNN)* 3514–3521 (IEEE, 2017).
12. Möckl, L., Roy, A. R. & Moerner, W. E. Deep learning in single-molecule microscopy: fundamentals, caveats, and recent developments. *Biomed. Opt. Express* **11**, 1633–1661 (2020).
13. Zhang, P. et al. Analyzing complex single-molecule emission patterns with deep learning. *Nat. Methods* **15**, 913–916 (2018).
14. Kim, T., Moon, S. & Xu, K. Information-rich localization microscopy through machine learning. *Nat. Commun.* **10**, 996 (2019).
15. Möckl, L., Roy, A. R., Petrov, P. N. & Moerner, W. E. Accurate and rapid background estimation in single-molecule localization microscopy using the deep neural network bgnet. *Proc. Natl Acad. Sci. USA* **117**, 60–67 (2020).
16. Zelger, P. et al. Three-dimensional localization microscopy using deep learning. *Opt. Express* **26**, 33166–33179 (2018).
17. Nehme, E. et al. DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning. *Nat. Methods* **17**, 734–740 (2020).
18. Boyd, N., Jonas, E., Babcock, H. P. & Recht, B. Deeploco: fast 3D localization microscopy using neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/267096 (2018).
19. Chen, B.-C. Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science* **346**, 1257998 (2014).
20. Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention* (eds Navab, N. et al.) 234–241 (Springer, 2015); https://doi.org/10.1007/978-3-319-24574-4_28
21. Rieger, B. & Stallinga, S. The lateral and axial localization uncertainty in super-resolution light microscopy. *Chem. Phys. Chem.* **15**, 664–670 (2014).
22. Chao, J., Ward, E. S. & Ober, R. J. Fisher information theory for parameter estimation in single molecule microscopy: tutorial. *JOSA A* **33**, B36–B57 (2016).
23. Li, Y. et al. Real-time 3D single-molecule localization using experimental point spread functions. *Nat. Methods* **15**, 367–369 (2018).
24. Small, A. & Stahlheber, S. Fluorophore localization algorithms for super-resolution microscopy. *Nat. Methods* **11**, 267–279 (2014).
25. P.J. Nieuwenhuizen, R. et al. Measuring image resolution in optical nanoscopy. *Nat. Methods* **10**, 557–562 (2013).
26. Diekmann, R. et al. Optimizing imaging speed and excitation intensity for single-molecule localization microscopy. *Nat. Methods* **17**, 909–912 (2020).
27. Wäldchen, S., Lehmann, J., Klein, T., Van De Linde, S. & Sauer, M. Light-induced cell damage in live-cell super-resolution microscopy. *Sci. Rep.* **5**, 15348 (2015).
28. Thevathasan, J. V. et al. Nuclear pores as versatile reference standards for quantitative superresolution microscopy. *Nat. Methods* **16**, 1045–1053 (2019).
29. Dempsey, G. T., Vaughan, J. C., Chen, K. H., Bates, M. & Zhuang, X. Evaluation of fluorophores for optimal performance in localization-based super-resolution imaging. *Nat. Methods* **8**, 1027–1036 (2011).
30. Legant, W. R. et al. High-density three-dimensional localization microscopy across large volumes. *Nat. Methods* **13**, 359–365 (2016).
31. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)* Vol. 32, 8024–8035 (2019).
32. Ries, J. SMAP: a modular super-resolution microscopy analysis platform for SMLM data. *Nat. Methods* **17**, 870–872 (2020).
33. von Chamier, L. et al. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12**, 2276 (2021).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

DECODE network architecture for probabilistic single-molecule detection and localization. Our architecture consists of two stacked U-nets[20] (Extended Data Fig. 1), each with two up- and downsampling stages and 48 filters in the first stage. Each stage consists of three fully convolutional layers with 3 × 3 filters. In each downsampling stage, the resolution is halved, and the number of filters is doubled, vice versa in each upsampling stage. Upsampling is performed using nearest neighbor interpolation to avoid checkerboard artifacts[34]. For multiframe DECODE, three consecutive frames are processed by the first frame analysis U-net (with parameters shared for every frame), and the outputs are concatenated and passed to the second temporal context U-net. The entire DECODE network is always trained end-to-end by gradient descent.

For each camera pixel k, the DECODE network predicts (1) a Bernoulli probability map p_k that an emitter was detected near that pixel, (2) the coordinates of the detected emitter $\Delta x_k, \Delta y_k, \Delta z_k$ relative to the center of the pixel x_k, y_k, z_k, (3) a nonnegative emitter brightness ('photon count') N_k and (4) the uncertainties associated with each of these predictions, $\sigma_{x,k}, \sigma_{y,k}, \sigma_{z,k}, \sigma_{N,k}$. For each of these outputs, we use two additional convolutional layers that follow the second U-net. We used the exponential linear unit activation function[35] for all hidden units, and the logistic sigmoid nonlinearity for the nonnegative detection probability p, brightness N and the uncertainty outputs $\sigma_x, \sigma_y, \sigma_z, \sigma_N$ (scaled by a prefactor of three). For the coordinate outputs $\Delta x, \Delta y, \Delta z$ we use the hyperbolic tangent nonlinearity, which limits their range to $[-1,1]$ (that is, to twice the size of a pixel). This way, even though the network can at most predict one emitter per pixel, when necessary, the neighboring pixels can each contribute to place multiple localizations within a single pixel.

New loss function for simultaneous detection, localization and uncertainty estimation. Given a set of E simulated emitters active in each imaging frame with locations for each emitter e given by x_e, y_e, z_e and brightness N_e, and a background image map B_k simulated as described below, we developed a loss function that trains the DECODE network to detect the correct number of emitters, to predict the subpixel localization and brightness for each detection (along with the uncertainty), and to predict the image background. Our loss function is a sum of three terms: a count loss $\mathcal{L}_{\text{count}}$, a localization loss $\mathcal{L}_{\text{loc}}$ and a background loss $\mathcal{L}_{\text{bg}}$,

$$\mathcal{L} = \mathcal{L}_{\text{count}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{bg}}. \tag{1}$$

The count loss $\mathcal{L}_{\text{count}}$ is a function of the detection probability map p_k with K total pixels and the total number of true emitters E. Interpreting p_k as a Bernoulli detection probability for a single-emitter, we can compute the mean and variance of the predicted total number of emitters detected, if we were to independently sample binary detections from each p_k. While the predicted count distribution $P(E|\{p_k\})$ over the number of emitters detected by this Bernoulli sampling procedure follows an intractable Poisson binomial distribution, we can approximate this predicted distribution as a Gaussian distribution,

$$P(E|\{p_k\}) \approx P(E|\mu_{\text{count}}, \sigma^2_{\text{count}}) = \frac{1}{\sqrt{2\pi}\sigma_{\text{count}}} \exp\left(-\frac{1}{2}\frac{(E-\mu_{\text{count}})^2}{\sigma^2_{\text{count}}}\right). \tag{2}$$

The mean of a sum of Bernoulli random variables is the sum of the means $\mu_{\text{count}} = \sum_{k=1}^{K} p_k$, and the variance is the sum of the variances of each independent Bernoulli random variable $\sigma^2_{\text{count}} = \sum_{k=1}^{K} p_k(1-p_k)$. This count loss maximizes the log probability of the true number of emitters E under the Gaussian approximation of the predicted count probability distribution. This loss is minimized when μ_{count} correctly matches E, sparsely predicting only one nonzero p_k per detected emitter, and when σ^2_{count} is small, which happens when p_k are confident and so nearly binary,

$$\mathcal{L}_{\text{count}} = -\log P(E|\mu_{\text{count}}, \sigma^2_{\text{count}}) = \frac{1}{2}\frac{(E-\mu_{\text{count}})^2}{\sigma^2_{\text{count}}} + \log\left(\sqrt{2\pi}\sigma_{\text{count}}\right). \tag{3}$$

The localization loss $\mathcal{L}_{\text{loc}}$ is a function of the true emitter locations, and the predicted detection probability map, and the subpixel localizations $\Delta x_k, \Delta y_k, \Delta z_k$, brightness N_k, along with the associated uncertainties $\sigma_{x,k}, \sigma_{y,k}, \sigma_{z,k}, \sigma_{N,k}$ for each detected emitter. For each pixel k, we predict a four-dimensional Gaussian distribution $P(\mathbf{u}_k|\boldsymbol{\mu}_k, \Sigma_k)$ over the absolute position and brightness of an emitter $\mathbf{u} = [x, y, z, N]$ detected in pixel k corresponding to the mean and uncertainty in the subpixel localization and brightness of the emitter detected in pixel k, with mean $\boldsymbol{\mu}_k = [x_k + \Delta x, y_k + \Delta y_k, \Delta z_k, N_k]$ and diagonal covariance matrix $\Sigma_k = \text{diag}(\sigma^2_{x,k}, \sigma^2_{y,k}, \sigma^2_{z,k}, \sigma^2_{N,k})$,

$$P(\mathbf{u}|\boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^4 \det(\Sigma_k)}} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_k - \mathbf{u})^\top \Sigma_k^{-1} (\boldsymbol{\mu}_k - \mathbf{u})\right). \tag{4}$$

Here, the x_k, y_k and z_k are the absolute coordinates for the center of pixel k, so $x_k + \Delta x_k$ corresponds to the absolute coordinates of the emitter to subpixel precision. We note that the localization loss defined below ignores the predicted localization and brightness for pixels where no emitter is detected, that is p_k is zero.

At any given point in training, the true number of emitters will not necessarily match the detected number of emitters perfectly, and we will not have a perfect correspondence between predicted emitters and true emitters. A full probabilistic loss function would sum over all possible assignments of true emitters to detected emitters to correctly evaluate $P(\mathbf{u}|\boldsymbol{\mu}_k, \Sigma_k)$. And since p_k will not necessarily be sparse, the correct cost function would include an intractably large sum over $\binom{K}{E}$ terms. We approximate this by constructing a Gaussian mixture model over the predicted per pixel distributions $P(\mathbf{u}_k|\boldsymbol{\mu}_k, \Sigma_k)$ with mixture weights equal to $p_k/\sum_{j=1}^{K} p_j$ where the denominator is a sum of the detection probability over all pixels in the image.

The resulting approximation leads to the following localization loss function, which maximizes the probability of the true absolute coordinates and brightness of each ground-truth emitter $\mathbf{u}_e^{\text{GT}}$ under the weighted mixture of per pixel probabilities,

$$\mathcal{L}_{\text{loc}} = -\frac{1}{E}\sum_{e=1}^{E} \log \sum_{k=1}^{K} \frac{p_k}{\sum_j p_j} P(\mathbf{u}_e^{\text{GT}}|\boldsymbol{\mu}_k, \Sigma_k). \tag{5}$$

The background loss $\mathcal{L}_{\text{bg}}$ computes the simple squared error between the predicted and true background maps,

$$\mathcal{L}_{\text{bg}} = \sum_k \left(B_k^{\text{GT}} - B_k^{\text{pred}}\right)^2. \tag{6}$$

Obtaining localizations and postprocessing. The DECODE network predicts the probabilities p_k of an emitter being located at a specific pixel k. To get deterministic, fast and precise final localizations we use a variant of spatial integration. A detection is considered at pixel k if one of two conditions is met. (1) $p_k > 0.6$. (2) $p_k > 0.3$ and it is a local maximum of a four-connected neighborhood. These candidates are then registered as detections if the cumulative probability of p_k and its four nearest neighbor pixels is >0.7. Therefore, if the network predicts high confidence detection probability (>0.6) in two adjacent pixels, two emitters will be considered to be detected. However, if a cluster of pixels has low predicted probability, their probabilities will be clustered toward the local maximum, if the local maximum has probability >0.3, and an emitter will be considered to have been detected if the integrated probabilities of the cluster are >0.7. The algorithm can be expressed purely in the form of pooling and convolution operations and therefore runs efficiently on a GPU.

For difficult imaging conditions when the predicted localization uncertainties are large, that is high densities, low-SNR values, and large offsets from the focal plane, the subpixel coordinates Δx, Δy and Δz can be biased toward the center of the pixels (Extended Data Fig. 8). This is because with large predicted localization uncertainty, the predicted mean location is poorly constrained. This bias toward zero (pixel center) scales with the uncertainty of the predictions and can produce artifacts in the reconstructed image depending on how the reconstruction is performed. If a reconstruction uses only the coordinates while ignoring the uncertainty, poorly localized emitters will cluster toward the pixel centers. A more expensive rendering procedure that renders a Gaussian localization distribution with variance proportional to the estimated uncertainty corresponding to each emitter will reduce the impact of this artifact since the bias is usually small relative to the localization uncertainty. Also, filtering out localizations with high uncertainty removes this artifact (Extended Data Fig. 8).

Simulating training data. Training samples are continuously generated in an asynchronous fashion and each frame is only used once as a target. For this reason, the network cannot overfit to specific frames. The performance of our approach will depend on an accurate generative model and could show reduced performance when there is a mismatch between the simulated and experimental data. Thus, we developed a realistic model for the image formation process that incorporates dye blinking behavior, a realistic PSF model and realistic camera read noise.

Structural prior. While incorporating prior structural information has shown to be beneficial[36,37], there are concerns that these priors could potentially bias the model to the training data, which could result in the presence of misleading structures after the fitting procedure. We therefore sample the coordinates of the emitters from a 3D homogeneous spatial Poisson point process distribution with density as specified in the text, limits corresponding to the size of the image and the z range for which the PSF was calibrated.

Photophysical prior. In contrast to previous work, DECODE can directly incorporate temporal context into the detection and localization of emitters, rather than as a postprocessing step. We simulate the temporal dynamics of emitters, at least over the short time scale of three imaging frames corresponding to the temporal context of the DECODE network.

For each emitter, the time of initial appearance t_0 is sampled from a continuous random distribution. The on-time of the emitter follows an exponential distribution parametrized by λ. For each emitter, we draw a photon flux from a Gaussian distribution $N(\mu_{\text{flux}}, \sigma_{\text{flux}})$. Together with the amount of time the emitter is active in each frame, this determines the total number of photons emitted in a frame. Since the input to our model is only a window of three frames, we argue that it is not necessary to model long range temporal correlations that are part of a more detailed photoactivation model[38], such as an emitter in the dark state that reappears many frames later. The aforementioned parameters are estimated by a prefit procedure as described in Estimating simulation parameters.

Point spread function. The PSF is a fundamental characteristic of a microscope, specifying the image formed by a single point emitter, and we approximate it to be spatially invariant across the field of view. Given the object $O(\mathbf{r})$ in the object plane, and PSF($\mathbf{r}$), the image $I(\mathbf{r})$ results in

$$I(\mathbf{r}) = O(\mathbf{r}) \circledast \mathrm{PSF}(\mathbf{r}), \tag{7}$$

where $\circledast$ denotes the convolution operator. While Gaussian approximations of the PSF are frequently used for both 2D and 3D[5,6] data, (cubic) spline functions have been shown to achieve more accurate results and can mimic almost arbitrary PSFs[4,23]. Following Li et al.[23] and Babcock et al.[4] a 3D PSF can be modeled as

$$f_{i,j,k}(x,y,z) = \sum_{m=0}^{3}\sum_{n=0}^{3}\sum_{p=0}^{3} a_{i,j,k,m,n,p}\left(\frac{x-x_i}{\mathrm{d}x}\right)^m\left(\frac{y-y_j}{\mathrm{d}y}\right)^n\left(\frac{z-z_k}{\mathrm{d}z}\right)^p, \tag{8}$$

where i,j,k are the voxel indices, dx, dy are the pixel sizes; dz is the step size in the axial dimension; x_i, y_j, z_k are the corner coordinates of the voxel (i,j,k) in the respective directions and $a_{i,j,k,m,n,p}$ are the respective spline coefficients, which amounts to 64 coefficients per pixel and per z slice. In a bead calibration routine, the coefficients $a_{i,j,k,m,n,p}$ are estimated and account for varying experimental conditions. Because of the simple form of equation (8), the CRLB with respect to the fitting parameters x, y, z can be calculated easily as the diagonal elements of the inverse of the Fisher information matrix[22].

Camera model. All real datasets presented in this work were recorded with an electron multiplying charge-coupled device (EMCCD) camera, with the exception of the LLS data that were recorded with an sCMOS camera. The measured camera signal is subject to various noise sources, which we will discuss in the following:

Shot noise originates from the stochastic nature of photons when interacting with the camera chip. The expected number of detected electrons is

$$\lambda_k = \lambda_{0,k}\cdot \mathrm{QE} + c_s. \tag{9}$$

Here, $\lambda_{0,k}$ is the expected number of photons that are collected in pixel k, QE is the quantum efficiency, and c_s the spurious charge, measured in electrons. The probability $p_{\mathrm{shot}}(s_k)$ of observing the signal s_k in pixel k follows a Poisson distribution,

$$p_{\mathrm{shot}}(s_k) = \frac{\lambda_k^{s_k}\mathrm{e}^{-\lambda_k}}{s_k!}. \tag{10}$$

EMCCD amplification noise stems from the amplification of photo electrons that pass through the gain register and stochastically generate additional electrons. For our EMCCD camera noise model, we follow Huang et al.[39]. EMCCD amplification noise can be described approximately by a Gamma distribution,

$$\rho_{\mathrm{EM}}(x|s_k,\theta) = \frac{1}{\Gamma(s_k)\theta^{s_k}}x^{s_k-1}\mathrm{e}^{-\frac{x}{\theta}}. \tag{11}$$

$\rho_{\mathrm{EM}}(x|s_k,\theta)$ denotes the probability that s_k input photo electrons in pixel k with an electron multiplying gain of θ create x output electrons after the gain register.

Read noise stems from the process of converting electrons into a digital signal. In this process, the signal is usually multiplied by a gain factor g and an offset o is added to avoid negative signal. In this work, we convert the input camera image to photon units before inference by subtracting o and dividing by g. In addition, when using EMCCD cameras we divide by the electron multiplying gain θ, thus the units of the read noise are photo electrons. We approximate the read noise (both for sCMOS and EMCCD cameras) by a zero mean additive Gaussian distribution with variance σ^2,

$$\rho_{\mathrm{read}}\left(x|0,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{x^2}{2\sigma^2}}. \tag{12}$$

Training details. Training was performed on 40 × 40 pixel-sized regions that are directly simulated or randomly selected from larger simulated images at each iteration. We used the AdamW optimizer[40] with a group learning rate of 6×10^{-4} for the network parameters. We reduce the learning rate by a factor of 0.9 after every 1,500 iterations with a batch size of 64. To stabilize training, we use gradient norm clipping with a maximum norm of 0.03.

Very dim emitters with less than 50 photons are excluded from the ground-truth targets (but still rendered) so that the network is discouraged to make predictions for practically invisible emitters.

Estimating simulation parameters. For training DECODE, a proper parametrization of the simulation is needed to match the real data distribution. In a prefitting step, the main parameters, that is the emitter on-time, emitter brightness and background, can be determined. The prefitting can be performed with a single-emitter MLE fitter after filtering the log-likelihood value to exclude data from overlapping PSFs. This step is incorporated in the SMAP software for the sake of ease of use[32]. We observed that the precise values of the simulation parameters of the emitters' photophysics (that is, lifetime and brightness) and density are not crucial, as the stochastic nature of the emitters' positions, brightness levels and appearance times presents the network with data that match the real experiments under different conditions and effectively covers a broad range of these parameters.

The camera parameters are usually given by the manufacturer.

The given network architecture and training parameters are effective across different real and simulated datasets and in our experience do not have to be optimized by the end user.

Evaluating localization error and reconstruction resolution. To evaluate performance on the challenge datasets, as well as our own simulations, we use two metrics.

First, instead of the Euclidean distance, we use the localization error, measured in nm, which is the r.m.s.e. averaged over the dimensions:

$$\mathrm{r.m.s.e.}_d = \left(\frac{1}{\mathrm{TP}}\sum_{i=1}^{\mathrm{TP}}\sum_{k=1}^{d}\left(x_{i,k}-x_{i,k}^{\mathrm{GT}}\right)^2/d\right)^{1/2} \tag{13}$$

TP is the number of localizations that are matched to ground-truth (GT) coordinates, d is the dimension (two for 2D data, three for 3D data), $x_k = x, y, z$ are the predicted coordinates and $x_k^{\mathrm{GT}} = x, y, z$ the ground-truth coordinates.

Second, the detection accuracy or JI, which quantifies how well an algorithm does at detecting all the emitters while avoiding false positives:

$$\mathrm{JI} = \mathrm{TP}/(\mathrm{FN}+\mathrm{FP}+\mathrm{TP}) \tag{14}$$

TP are the true positives, FN the false negatives and FP the false positives.

Localizations are matched to ground-truth coordinates when they are within a circle of 250 nm radius and the distance in z is less than 500 nm. As a single metric that evaluates the ability to reliably infer emitters with high precision, we use the efficiency metric as defined in ref.[7]:

$$E = 1 - \sqrt{(1-\mathrm{JI})^2 + \alpha^2 d\cdot \mathrm{r.m.s.e.}_d^2} \tag{15}$$

Lateral and axial efficiency are calculated based on r.m.s.e.$_2$ and r.m.s.e.$_1$ with alpha values of $\alpha = 1\times10^{-2}$ and $\alpha = 0.5\times10^{-2}\,\mathrm{nm}^{-1}$, respectively and then averaged to obtain the overall efficiency. Detection accuracy is expressed in units of 0 to 1 (or 0–100%), the efficiency ranges up to 1 (or 100%) for a perfect fitting algorithm.

The Fourier ring correlation[25,41] (FRC) in Fig. 4a was calculated by dividing the data in ten blocks of equal number of frames and constructing super-resolution images from even and odd blocks (pixel size 5 nm).

Simulating data for performance evaluation. To simulate data for performance evaluation and comparison shown in Fig. 2, we assumed an ideal camera without EMCCD or read noise and an image size of 64 × 64 pixels. We used the PSF model that was acquired for the dataset in Fig. 4a. Data used to test the effect of the SNR and density were simulated using the structural and photophysical prior previously described with an average on-time of two frames. Precise simulation parameters can be found in Supplementary Table 1.

The CRLB is evaluated as the diagonal elements of the inverse of the Fisher information matrix[22] with the simulated parameters and spline interpolated experimental PSF model and was calculated with the SMAP software[32]. A bootstrap estimate (N = 10,000) of the r.m.s.e. was used to estimate the s.e.m. on the localization error.

Sample preparation and localization microscopy. See Supplementary Note for details on sample preparation and localization microscopy.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All data can be downloaded from https://doi.org/10.25378/janelia.14674659. Raw data and bead frames are available for Figs. 2a–e and 4a–h and Extended Data Figs. 2, 3, 4a, 5 and 8. Localizations and performance metrics (for DECODE and CSpline/DeepSTORM3D when applicable) are available for Figs. 2a–e, 4a–h and 5 and Extended Data Figs. 2, 3, 4a, 7 and 8. The parametrization of the simulation for Fig. 2a–e is available and can be used to generate data. Raw data and bead frames, as well as performance metrics for Fig. 3. are publicly available at http://bigwww.epfl.ch/smlm/challenge2016/. Raw data and bead frames for Fig. 5 and Extended Data Fig. 7 are available on request from the authors of ref. [30]. All other data supporting the findings of this study are available from the corresponding authors upon reasonable request. Source data are provided with this paper.

Code availability

DECODE is available as Supplementary Software. Updated versions can be found at https://github.com/TuragaLab/DECODE.

References

34. Odena, A., Dumoulin, V. & Olah, C. Deconvolution and checkerboard artifacts. *Distill* https://distill.pub/2016/deconv-checkerboard/ (2016).
35. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). Preprint at https://arxiv.org/abs/1511.07289 (2016).
36. Ouyang, W., Aristov, A., Lelek, M., Hao, X. & Zimmer, C. Deep learning massively accelerates super-resolution localization microscopy. *Nat. Biotechnol.* **36**, 460–468 (2018).
37. Weigert, M. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090–1097 (2018).
38. Annibale, P., Vanni, S., Scarselli, M., Rothlisberger, U. & Radenovic, A. Quantitative photo activated localization microscopy: unraveling the effects of photoblinking. *PLoS ONE* **6**, e22678 (2011).
39. Huang, F. Video-rate nanoscopy using scmos camera–specific single-molecule localization algorithms. *Nat. Methods* **10**, 653–658 (2013).
40. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. Preprint at https://arxiv.org/abs/1711.05101 (2019).
41. Banterle, N., Bui, K. H., Lemke, E. A. & Beck, M. Fourier ring correlation as a resolution criterion for super-resolution microscopy. *J. Struct. Biol.* **183**, 363–367 (2013).
42. Perlin, K. An image synthesizer. *Comput. Graph. (ACM)* **19**, 287–296 (1985); https://doi.org/10.1145/325165.325247

Acknowledgements

S.C.T. is supported by the Howard Hughes Medical Institute. J.R., L.-R.M. and P.H. were supported by the European Molecular Biology Laboratory, the European Research Council (grant no. CoG-724489 to J.R.) and the National Institutes of Health Common Fund 4D Nucleome Program (grant no. U01 EB021223 to J.R.). J.H.M. and A.S. were supported by the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, project no. 390727645) and the German Federal Ministry of Education and Research (BMBF, project no. 'ADMIMEM', FKZ 01IS18052). W.R.L. acknowledges support from the Searle Scholars Program, the Beckman Young Investigator Program, an National Institutes of Health New Innovator Award (no. DP2GM136653) and the Packard Fellows Program. We thank D. Sage for useful discussions, U. Boehm, D. Greenberg and P. Ramesh for comments on the manuscript, and E. Betzig and J. Lippincott Schwartz for kindly sharing data with us. We are grateful to C. Leterrier for extensive testing and useful suggestions for improving the DECODE library, and to U. Boehm for helping create DECODE tutorials. We thank D. Olbris and D. Kutra for assistance with automation of building and deploying DECODE across multiple platforms.

Author contributions

A.S., L.-R.M., J.H.M., J.R. and S.C.T. conceived the project, analyzed the results and wrote the paper with input from all authors. A.S. and L.-R.M. wrote the software. U.M., P.H. and J.R. acquired and analyzed the biological data. A.K. provided supervision. C.J.O. and W.R.L. provided the LLS data and helped with analysis.

Competing interests

The authors declare no competing interests.

Additional information

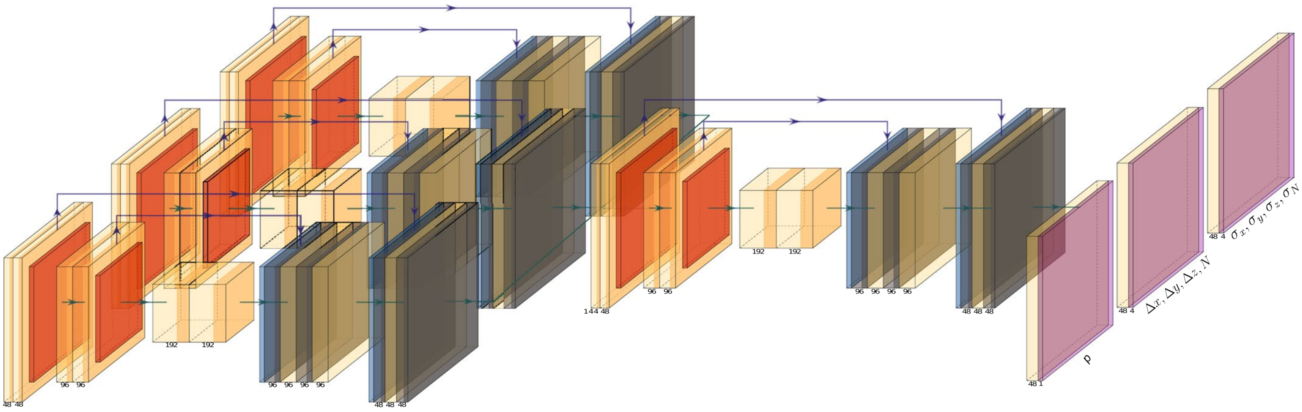
Extended data is available for this paper at https://doi.org/10.1038/s41592-021-01236-x.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01236-x.

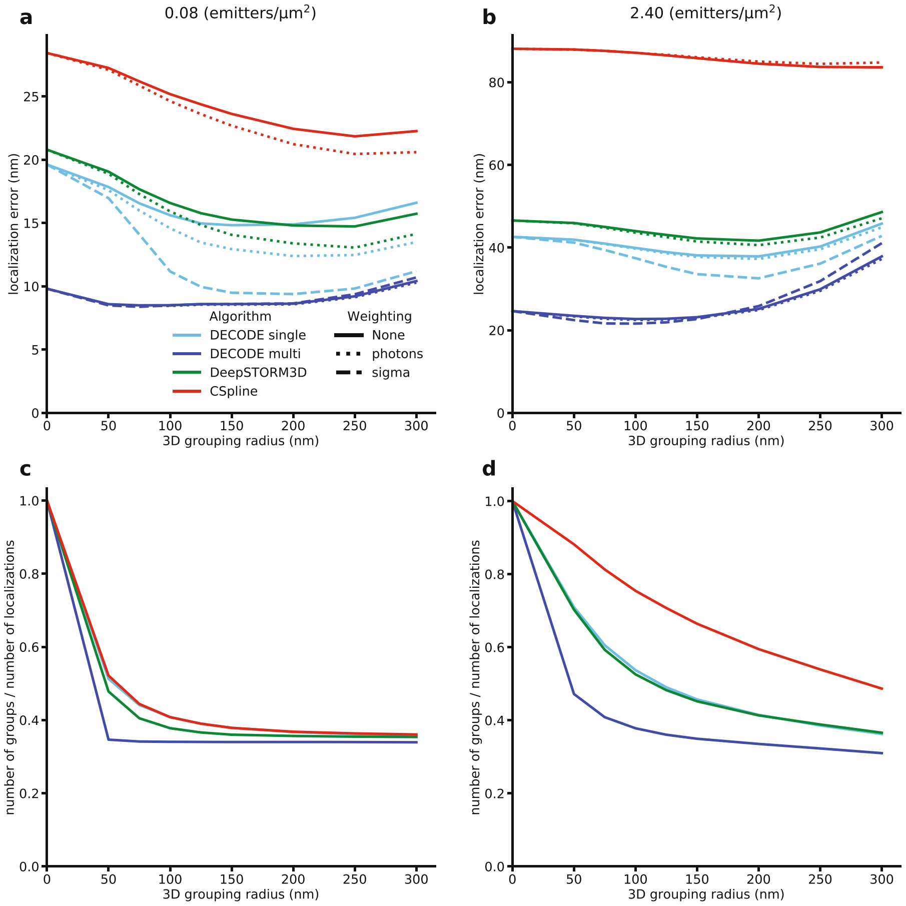
Correspondence and requests for materials should be addressed to J.H.M., J.R. or S.C.T.

Peer review information *Nature Methods* thanks Alex Herbert and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

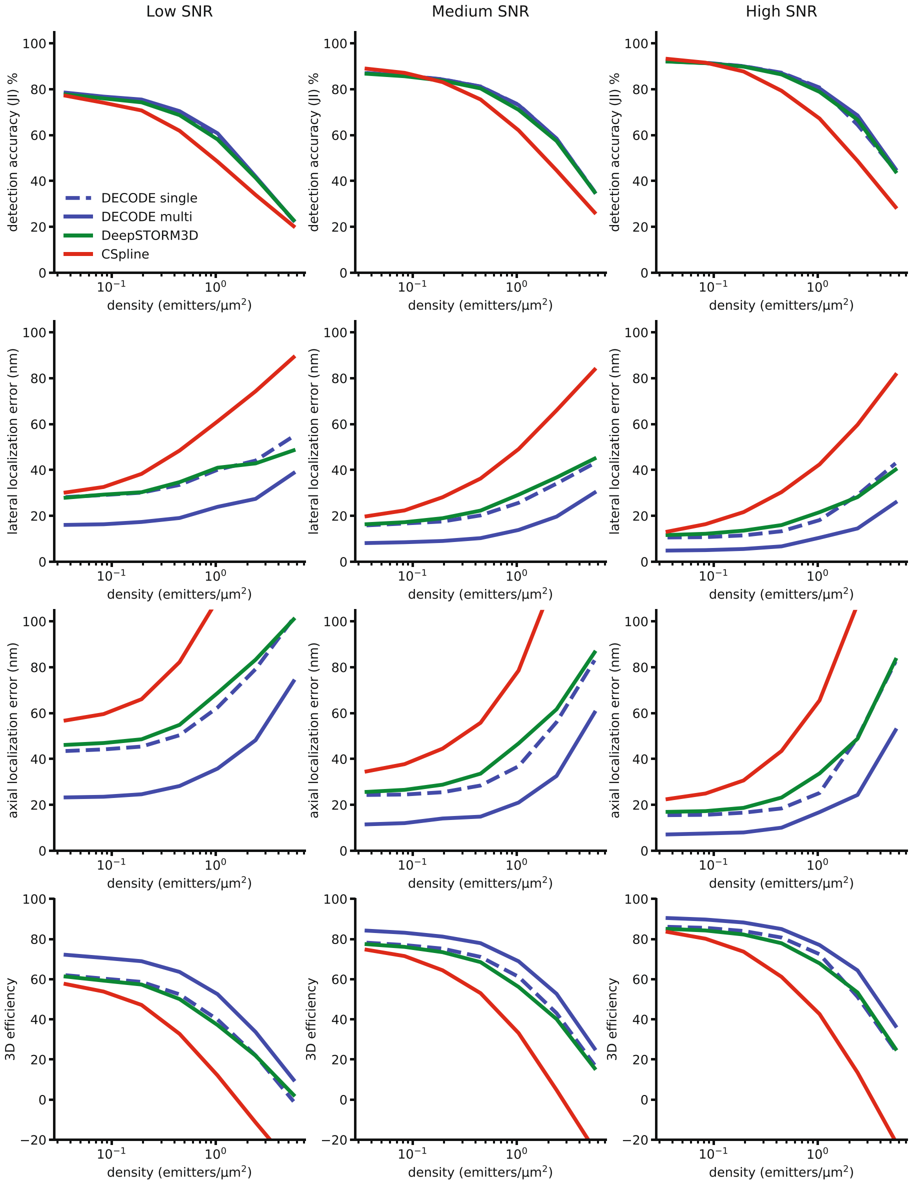
Reprints and permissions information is available at www.nature.com/reprints.



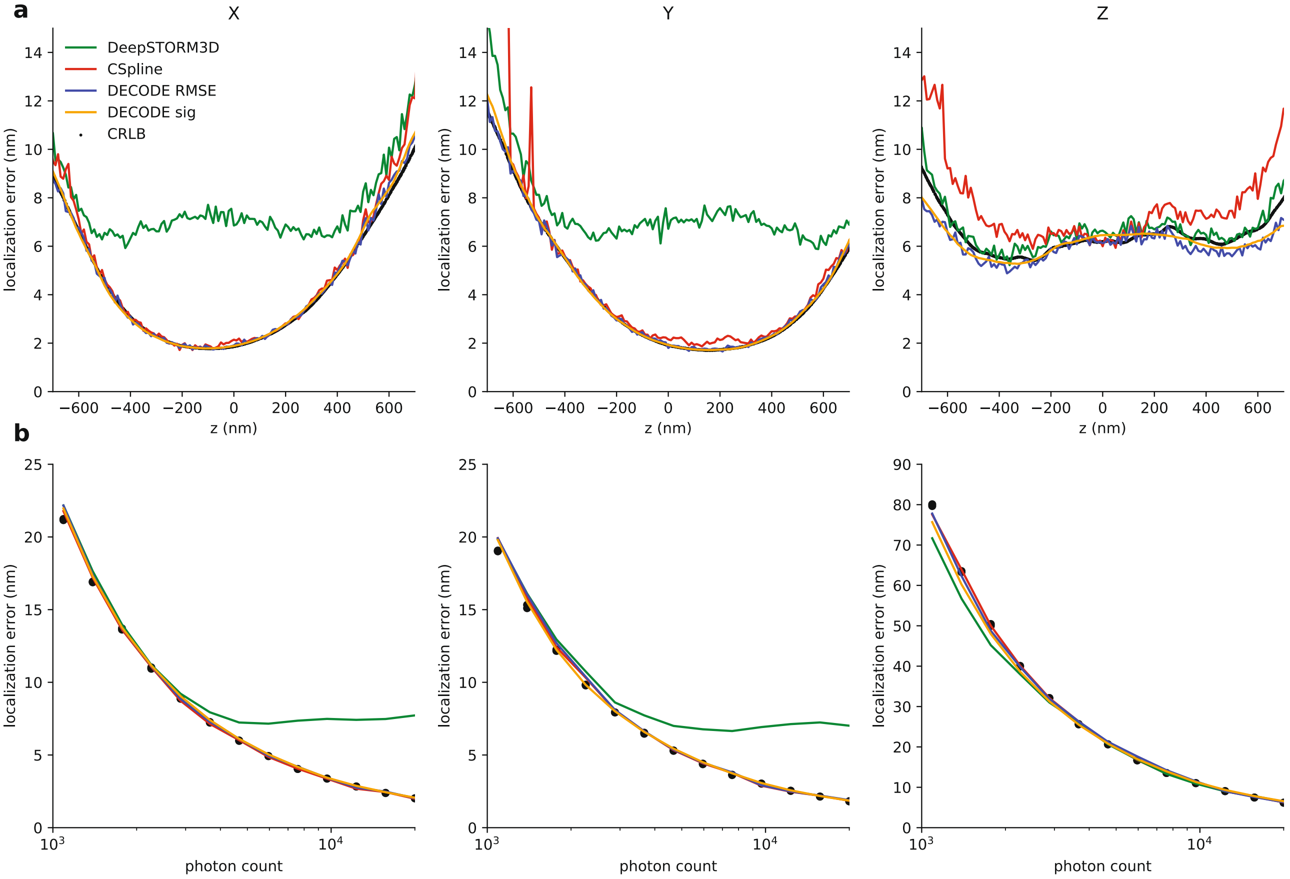
Extended Data Fig. 1 | Architecture. The DECODE network consists of two stacked U-Nets[20] with identical layouts (the three networks depicted on the left share parameters). The *frame analysis module* extracts informative features from three consecutive frames. These features are integrated by the *temporal context module*. Both U-Nets have two up- and downsampling stages and 48 filters in the first stage. Each stage consists of three fully convolutional layers with 3 × 3 filters. In each downsampling stage, the resolution is halved, and the number of filters is doubled, vice versa in each upsampling stage. Blue arrows show skip connections. Following the *temporal context module* three output heads with two convolutional layers each produce the output maps which have the same spatial dimensions as the input frames. The first head predicts the Bernoulli probability map *p*, the second head the spatial coordinates of the detected emitter Δ*x*, Δ*y*, Δ*z* and its intensity *N* and the third head the associated uncertainties σ_x, σ_y, σ_z, σ_N. An optional fourth output head can be used for background prediction.



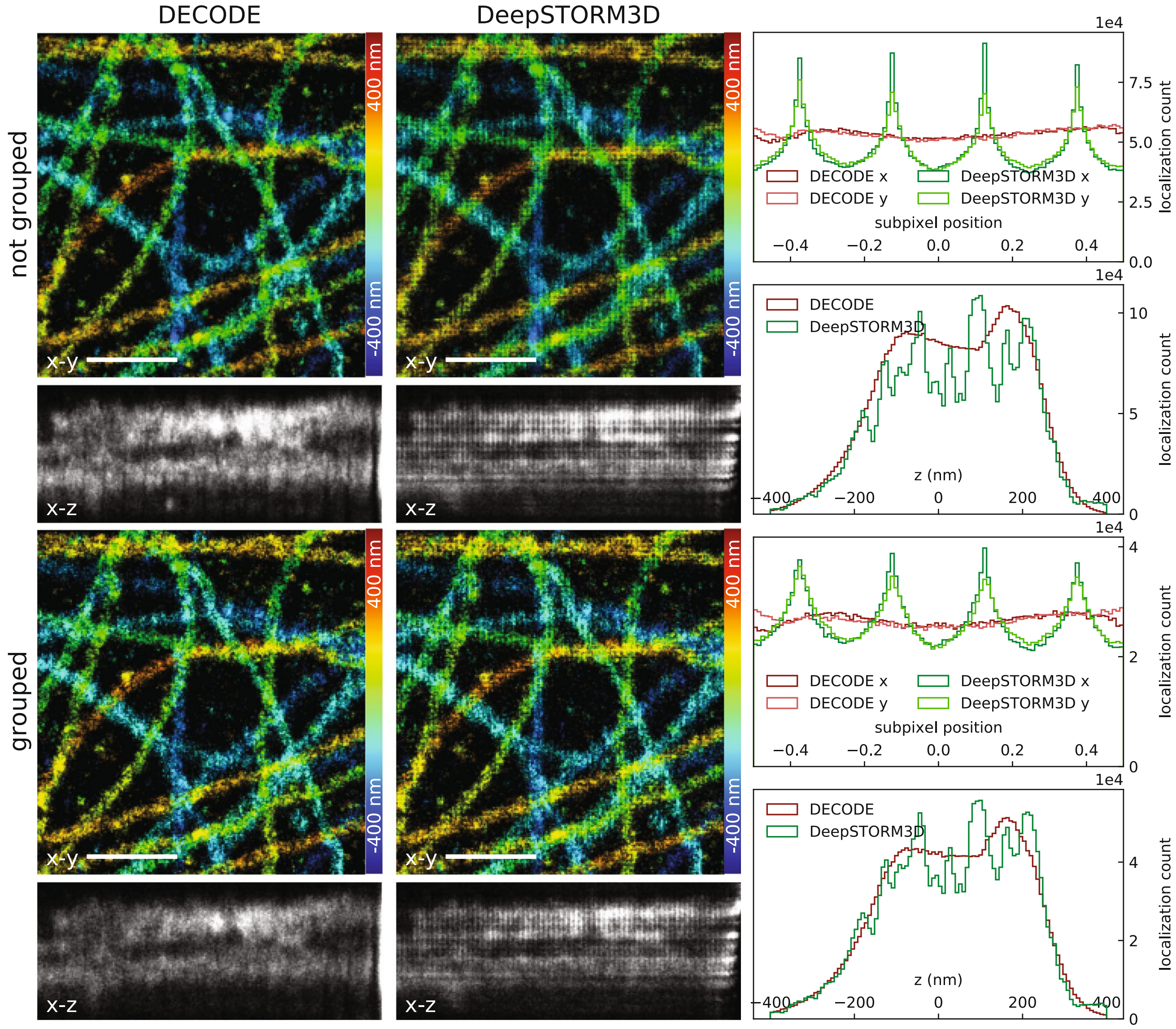
Extended Data Fig. 2 | Impact of grouping across grouping radius for different averaging weights. Predictions in consecutive frames are grouped when they are closer to each other than the given grouping radius. A grouping radius of 0 nm corresponds to not performing any grouping. Predictions within a group are assigned a common set of emitter coordinates which is calculated as weighted average of their individual coordinates. We compare three different options for the weighted average: Uniform weighting ('None', solid lines); Weighting by the inferred number of photons for CSpline and DECODE or the inferred confidence for DeepSTORM3D ('photons', dotted line); Weighting by the predicted DECODE σ values, where the x,y and z values are individually weighted by $\sigma_{x,y,z}^{-2}$. **a, b**): 3D efficiencies across grouping radii. Grouping is especially useful in the low density setting (a) where DECODE without temporal context (DECODE single) with a correctly set grouping radius can match the performance of DECODE with temporal context (DECODE multi) without grouping. This is, however, only the case when weighting by the uncertainty estimates that DECODE provides. Using grouping on top of DECODE multi offers little additional benefit. **c, d**): Number of groups divided by the number of localizations. Detecting all emitters and correctly grouping them would result in a ratio of 1:3 as on average each emitter is visible in three consecutive frames. See methods and Supplementary Table 1 for additional details on training and evaluation.



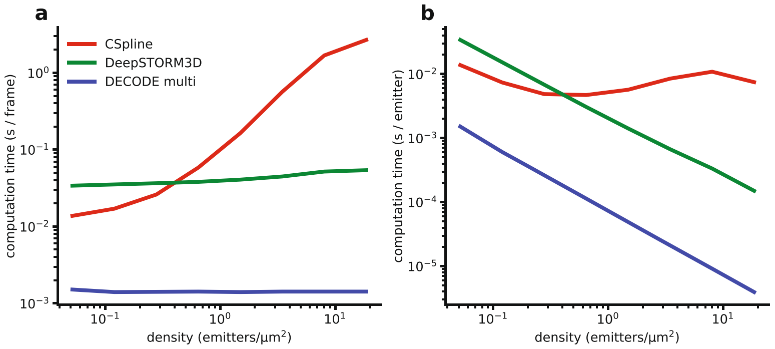
Extended Data Fig. 3 | Comparison of performance metrics across densities and SNRs. DECODE outperforms DeepSTORM3D and CSpline across densities and SNRs. See methods and Supplementary Table 1 for additional details on training and evaluation.



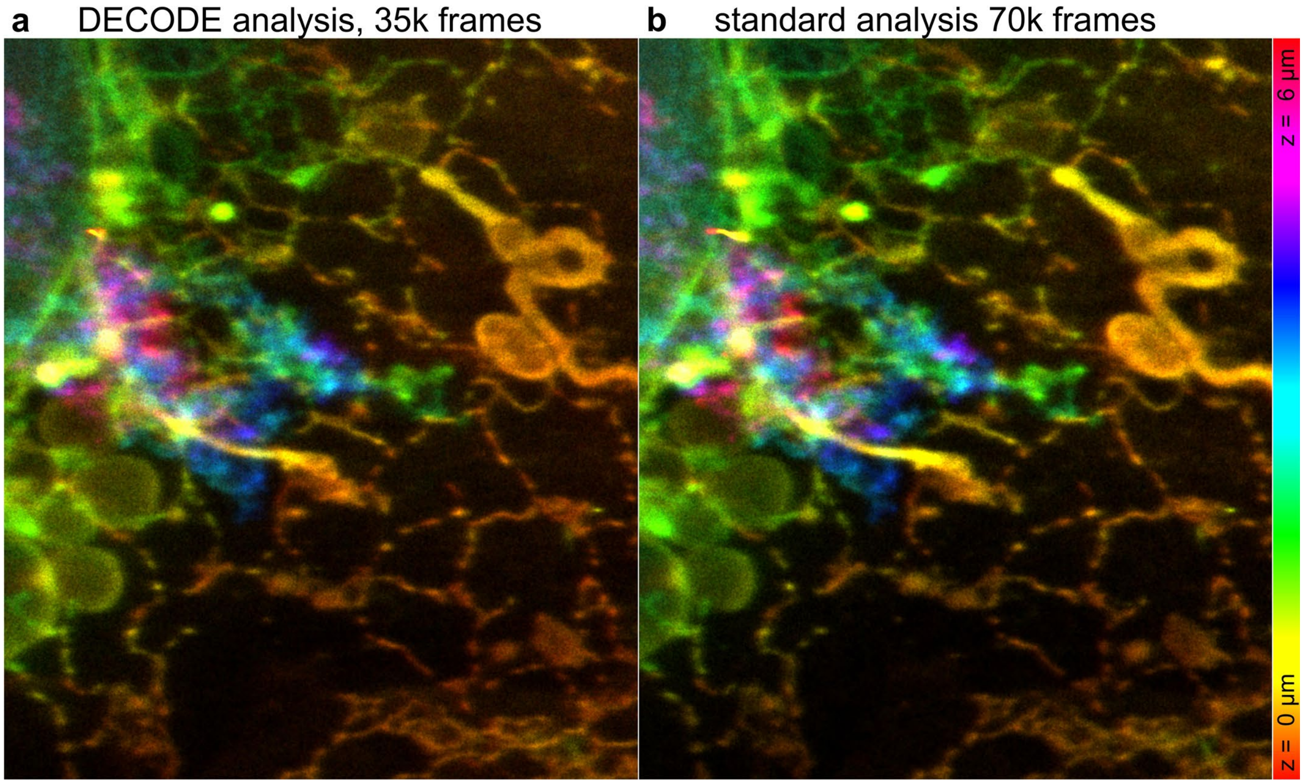
Extended Data Fig. 4 | Comparison of localization error and CRLB for single-emitter fitting. The r.m.s.e. achieved by DECODE and its predicted σ values closely match the single emitter CRLB in every dimension. CSpline is also able to achieve the CRLB, which has been shown for iterative MLE fitters before. In contrast the resolution that DeepSTORM3D can achieve is limited by its output representation and the size of the super-resolution voxels. **a**): Data simulated with high SNR (20,000 photons) and random *z*. r.m.s.e. and DECODE σ averaged over 10 nm bins. **b**): Data simulated with fixed *z* (0 nm) and varying SNR levels. See methods and Supplementary Table 1 for additional details on training and evaluation.



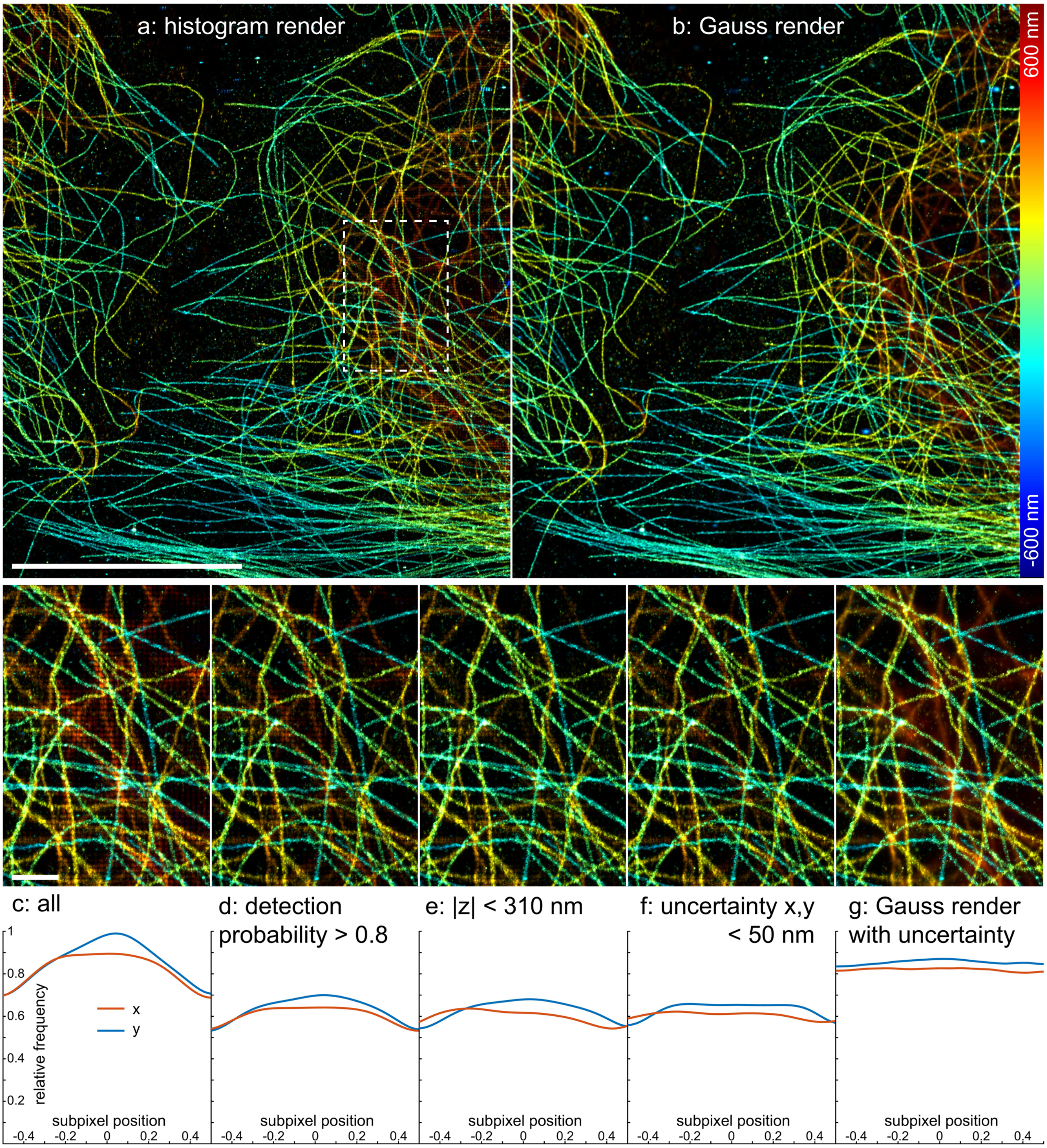
Extended Data Fig. 5 | Comparison of reconstruction quality on experimental STORM data. Reconstructions by DECODE and the DeepSTORM3D on a subset of data shown in Fig. 4g. Histograms show within pixel distribution of localizations in x and y as well as the z coordinate in nm. DeepSTORM3D has 4 significant peaks in the subpixel distribution, corresponding to the fourfold upsampling it uses for its network output. These are visible as grid artifacts in the reconstructions. In contrast the DECODE localizations are evenly distributed and no artifacts are visible. Scale bars 0.5 µm.



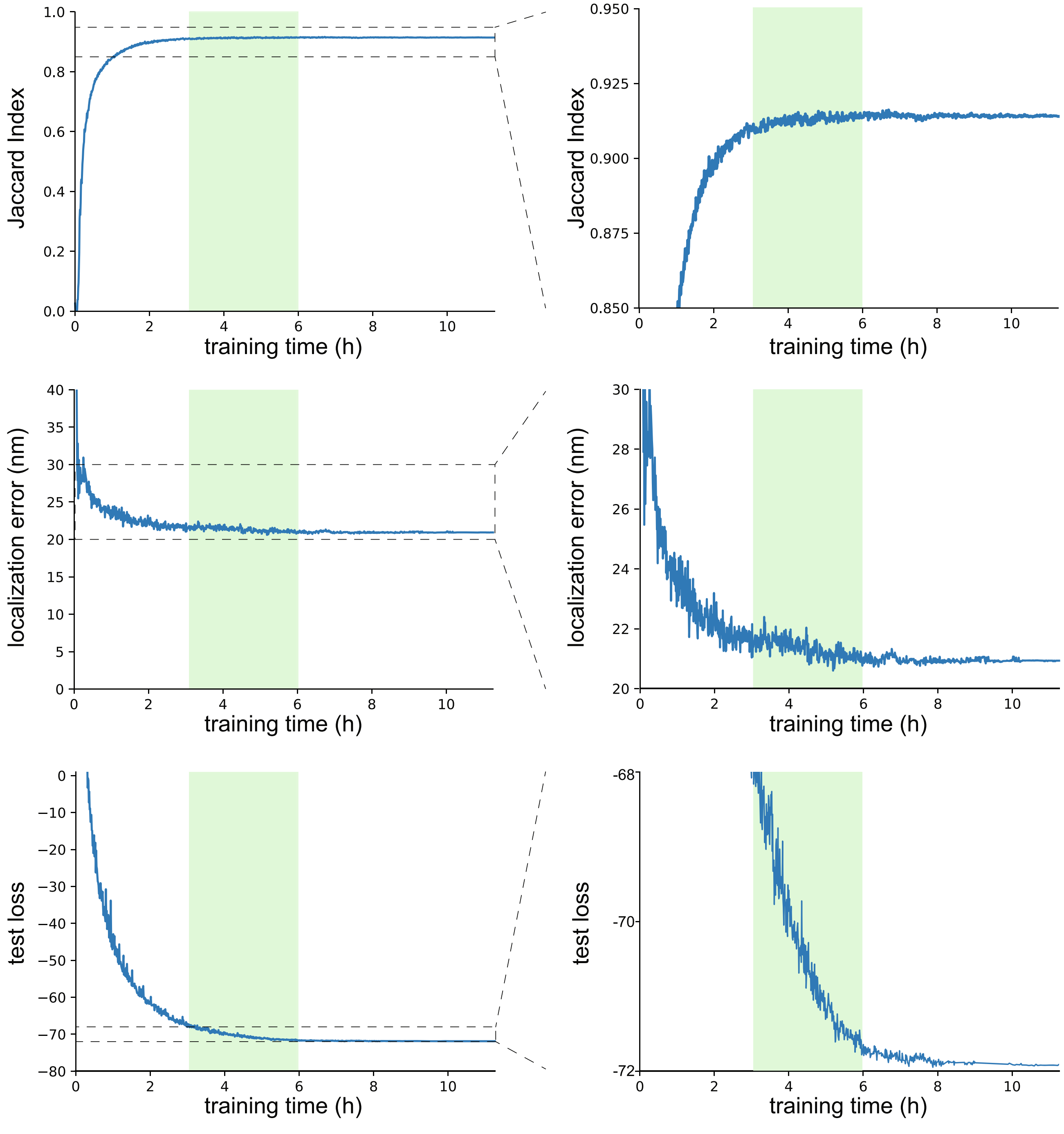
Extended Data Fig. 6 | Comparison of computation times. a) Measured as the time it takes to analyze a 64 × 64 pixel frame with varying emitter densities. Trained DECODE and DeepSTORM3D models were evaluated using a NVIDIA RTX2080Ti GPU. Computation time includes the network forward pass and postprocessing and does not include training time. CSpline was evaluated on an Intel(R)Xeon(R) CPU E5-2697 v3. **b**) Computation time per simulated emitter. The computation time of CSpline scales with the number emitters while the two deep learning based approaches scale with the number (and size) of the analyzed frames. GPU-based DECODE is about 20 times faster than GPU-based DeepSTORM3D and outperforms CPU-based CSpline even at low densities.



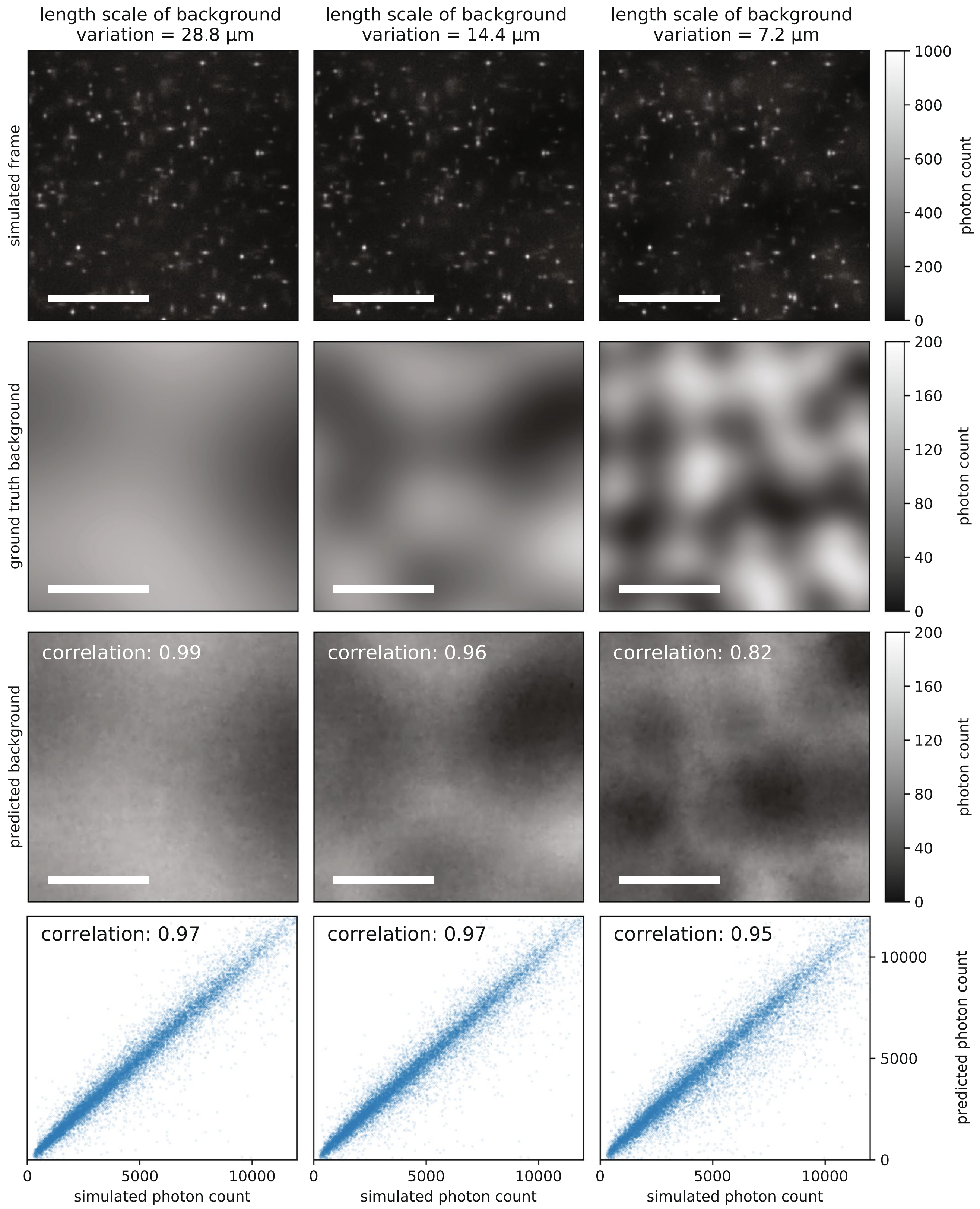
Extended Data Fig. 7 | DECODE reduces acquisition times in LLS-PAINT. DECODE reconstruction of 35,000 frames (**a**) results in the same number of localizations as the Standard reconstruction of 70,000 frames (**b**). As DECODE detects twice as many localizations as the traditional analysis, it needs only approx. half of the frames for a high-quality reconstruction.



Extended Data Fig. 8 | Removing Pixelation artifacts. Dim, dense out-of-focus localizations have a bias towards the pixel center (**a,c**). This is apparent as a non-uniform distribution of the sub-pixel positions in x and y (bottom row). This bias is not visible if every localization is rendered as a Gaussian with a standard deviation equal to the predicted uncertainty *s* (**b,g**). Filtering according to the detection probability reduces the artifact (**d**). Filtering according to the predicted uncertainty *σ* (**f**) or the fluorophore *z*-position (**e**) also removes the pixelation artifact. Scale bars 10 μm (**a,b**) and 1 μm (**c-g**). The overview images (**a,b**) are rendered with a pixel size of 10 nm, the zoom-ins (**c-g**) with a with a pixel size of 4 nm. The camera used to record the data has a pixel size of 117 × 127 nm.



Extended Data Fig. 9 | Performance as a function of deep network training time. Convergence of the accuracy of DECODE for several performance metrics. Runtimes are measured on a single nVidia RTX 2080 Ti GPU. The estimated training achievable with the maximum of 12 hours possible on the free tier of Google Colab is shown in green range (assuming that a Google Colab GPU is 2 × – 4 × slower than the nVidia RTX 2080 Ti GPU). This suggests that acceptable performance is achievable using DECODE and Google Colab at minimal cost, no GPU needed. Metrics evaluated for prediction > 0.5 detection probability estimate without sigma filtering. Training data was simulated at high SNR (as described in Fig. 2c) at an average density of 1 μm^{-2}.



Extended Data Fig. 10 | DECODE provides accurate background and signal predictions. Shown on simulated data with inhomogeneous background of various length scales. First row: sample frames. Second row: background values simulated using Perlin noise[42]. Third row: background values inferred by a DECODE network that was trained on 40 × 40 pixel sized simulations with uniform background. Fourth row: Scatter plot of inferred photon counts over simulated photon counts. Scale bars are 10 µm.

Supplementary Information
Deep learning enables fast and dense single-molecule localization with high accuracy

Artur Speiser[1,2,3,4,*], Lucas-Raphael Müller[5,6,*], Philipp Hoess[5], Ulf Matti[5], Christopher J. Obara[7], Wesley R. Legant[8,9], Anna Kreshuk[5], Jakob H. Macke[1,2,3,10,†], Jonas Ries[5,†], and Srinivas C. Turaga[7,†]

[1]Excellence Cluster Machine Learning, Tübingen University, Germany
[2]Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany
[3]research center caesar, an associate of the Max Planck Society, Bonn, Germany
[4]International Max Planck Research School ‘Brain and Behavior’, Bonn/Florida
[5]Cell Biology and Biophysics Unit, European Molecular Biology Laboratory, Heidelberg, Germany
[6]Ruprecht Karls University of Heidelberg, Heidelberg, Germany
[7]HHMI Janelia Research Campus, Ashburn, VA, USA
[8]Joint Department of Biomedical Engineering, UNC, Chapel Hill, NC, USA, and NCSU Raleigh, NC, USA
[9]Department of Pharmacology, University of North Carolina, Chapel Hill, NC, USA
[10]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[*]These authors contributed equally: Artur Speiser, Lucas-Raphael Müller
[†]For correspondence, Jakob H. Macke (Jakob.Macke@uni-tuebingen.de), Jonas Ries (jonas.ries@embl.de), Srinivas C. Turaga (turagas@janelia.hhmi.org). These three authors contributed equally.

Contents

List of Figures

1 Comparison with DeepSTORM3D and CSpline

For both methods we used the software provided by the authors. For the DeepSTORM3D comparison instead of using their PSF fitting procedure and generative model we sampled ground truth coordinates and training images using our model so that it exactly matches the simulated test data. To minimize possible effects of overfitting we generated 22,500 images with a size of 121×121 pixels (22k for training and 500 for validation). DeepSTORM3D uses a fourfold super-resolved grid in the $x - y$ dimensions and we chose discretization of 15 nm in z. As the camera we emulate in these experiments has a pixel size of 120 nm, each voxel of the output representation has a size of $30 \times 30 \times 15$ nm. For DECODE (with and without temporal context), and DeepSTORM3D we trained six networks each on training data generated with average emitter densities of 0.65 and 2.17 μm^{-2} as well as low, medium and high SNRs (1000, 5000 and 20,000 average photons). We used the low density network for the CRLB evaluation (Fig. 2a, Extended Data Fig. 4) and the simulated data with densities between 0.04 and 2.4 μm^{-2} (Fig 2c,d, Extended Data Fig. 3) and the high density networks for densities between 2.4 and 5.6 μm^{-2}. DeepSTORM3D has two hyperparameters that control the post-processing and determine the balance between recall and localization error. We performed a sweep over combinations of radius = [5,6,7,8,10] and threshold = [5,8,12,20,30,40] and picked the values that maximized the efficiency score on the validation data for each of the six networks. We discovered and fixed a bug in the DeepSTORM3D post-processing software which led to poor localizations. All DeepSTORM3D results were reported with the fixed post-processing algorithm.

For the CSpline comparison we created a bright artificial bead with 500k photons using our PSF model, which we used to generate the CSpline PSF model. The most critical settings are the find-max-radius and threshold, which we again optimized by sweeping over values find-max-radius = [2,3,4,5], threshold = [6,7,8,9,10] to maximize efficiency for each of the three SNRs on data generated with an average emitter density of 0.9 μm^{-2}.

2 DECODE for LLS-PAINT microscopy

A DECODE model for lattice light sheet point accumulation for imaging of nanoscale topography (LLS-PAINT) microscopy[1] was trained by simulating the imaging of an angled light sheet being swept through a volume. This leads to the same emitter appearing with fixed shift in the x and z coordinates relative to the imaged plane between consecutive camera frames. The offset in emitter coordinates from frame to frame are given by the microscope geometry as described in[2]. We simulated data with a high emitter density of 1 μm^{-2} to match the densities seen in LLS-PAINT.

We analyzed a large dataset corresponding to a fixed COS-7 cell with intracellular membranes labeled with azepanyl-rhodamine (AzepRh) described in Legant et al.[2]. Over a period of 2.7 days (64.8 hours), LLS-PAINT imaging yielded 70,000 3D volumes comprising more than 10 million 2D images. Significant non-uniform swelling of the sample was observed over the course of the imaging, which was approximately corrected by non-rigid registration in Legant et al.[2]. We applied the same correction transformation estimated by Legant et al.[2] to DECODE localizations.

We introduced an additional simulation-free training step and loss function to the training of the LLS-PAINT DECODE network based on the Re-weighted Wake Sleep algorithm[3] for training variational autoencoders (VAE)[4,5]. This form of auto-encoder learning allowed us to further optimize the parameters of the PSF and improve the background predictions based on the real data, as opposed to the simulation.

3 Sample preparation

Sample seeding

Before seeding of cells, high-precision 24 mm round glass coverslips (No. 1.5H, catalog no. 117640, Marienfeld) were cleaned by placing them overnight in a methanol:hydrochloric acid (50:50) mixture while stirring. After that, the coverslips were repeatedly rinsed with water until they reached a neutral pH. They were then placed overnight into a laminar flow cell culture hood to dry them before finally irradiating the coverslips by ultraviolet light for 30 min. Cells were seeded on clean glass coverslips 2 days before fixation to reach a confluency of about 50 to 70 % on the day of fixation. They were grown in growth medium (DMEM; catalog no. 11880-02, Gibco) containing 1× MEM NEAA (catalog no. 11140-035, Gibco), 1× GlutaMAX (catalog no. 35050-038, Gibco) and 10 % (v/v) fetal bovine serum (catalog no. 10270-106, Gibco) for approximately 2 days at 37 °C and 5 % CO_2.

Transfection

The plasmids encoding calnexin (Addgene plasmid #57445; http://n2t.net/addgene:57445; RRID:Addgene_57445) and α-mannosidase II (Addgene plasmid #57467; http://n2t.net/addgene:57467; RRID:Addgene_57467) tagged on their C-termini with mEos3.2 were gifts from Michael Davidson. The plasmids were isolated by midi-prep (catalog no. 12143; QIAGEN, Hilden, Germany) and transfected into U-2 OS cells using Lipofectamine™ 2000 (catalog no. 11668019; Thermo Fisher, Waltham, MA, USA) according to the manufacturer's instructions. Briefly, cells were seeded on coverslips as described in the previous section, after 2 days the medium was replaced with OptiMEM™ (catalog no. 51985026, Thermo Fisher) and the transfection solution was added dropwise. To prepare the transfection solution for 1 well (2 mL of medium), in a first step 1 µg of plasmid was added to 50 µL of OptiMEM™ medium and 3 µL of Lipofectamine™ were added to 50 µL of OptiMEM™ medium, respectively. The two solutions were mixed individually by pipetting, incubated for 3 min, and mixed together by pipetting to constitute the transfection solution after further incubation for 5 to 10 min. After 24 h, the OptiMEM™ medium was replaced by normal growth medium and the cells were grown for another 24 h before imaging.

Preparation of microtubule samples.

For microtubule staining, wild-type U-2 OS cells (ATCC HTB-96) were prefixed for 2 min with 0.3 % (v/v) glutaraldehyde in cytoskeleton buffer (CB, 10 mM MES pH 6.1, 150 mM NaCl, 5 mM EGTA, 5 mM glucose, 5 mM $MgCl_2$) + 0.25 % (v/v) Triton X-100 and fixed with 2 % (v/v) glutaraldehyde in CB for 10 min. Fluorescent background was reduced by incubation with 0.1 % (w/v) $NaBH_4$ in PBS for 7 min. After the samples had been washed three times with PBS, microtubules were stained with anti-α-tubulin (MS581; NeoMarkers, Fremont, CA, USA), and for ultra-high labeling (Fig. 4g) additionally with anti-β-tubulin (T5293; Sigma-Aldrich), each diluted 1:50 in PBS with 2 % (w/v) BSA, overnight. After being washed three times with PBS, samples were incubated with anti-mouse Alexa Fluor 647 (A21236; Invitrogen, Carlsbad, CA, USA) 1:50 in PBS + 2 % (w/v) BSA for 6 h. After being washed three times with PBS, samples were imaged in blinking buffer as described below. The holder was sealed with parafilm.

4 Localization microscopy

Microscope setup

SMLM data were acquired on a custom built widefield setup described previously[6,7]. Briefly, the free output of a commercial laser box (LightHub, Omicron-Laserage Laserprodukte) equipped with Luxx 405, 488 and 638 and Cobolt 561 lasers and an additional 640 nm booster laser (iBeam Smart, Toptica) were coupled into a square multi-mode fiber (catalog no. M103L05). The fiber was agitated as described in Ref. 8. The output of the fiber was magnified by an achromatic lens and focused into the sample to homogeneously illuminate an area of about 700 µm^2. The laser was guided through a laser cleanup filter (390/482/563/640 HC Quad, AHF) to remove fluorescence generated by the fiber. The emitted fluorescence was collected through a high numerical aperture (NA) oil immersion objective (HCX PL APO 160×/1.43 NA, Leica), filtered with a 676/37 (catalog no. FF01-676/37-25, Semrock) bandpass filter (for imaging of Alexa Fluor 647) or with a 600/60 (catalog no. NC458462, Chroma) bandpass filter (for live-cell imaging of mMaple and mEos3.2) on an EMCCD camera (Evolve 512, Photometrics). Astigmatism was introduced by a cylindrical lens (f = 1.00 m; catalog no. LJ1516L1-A, Thorlabs) to determine the z coordinates of fluorophores. The z focus was stabilized by an infrared laser that was totally internally reflected off the coverslip onto a quadrant photodiode, which was coupled into closed-loop feedback with the piezo objective positioner (Physik Instrumente). Laser control, focus stabilization and movement of filters was performed using a field-programmable gate array (Mojo, Embedded Micro). The custom microscope was controlled by Micro-Manager[9] using the EMU plugin[10]. The pulse length of the 405 nm laser could be controlled by a feedback algorithm to sustain a predefined number of localizations per frame.

Imaging conditions

Coverslips containing prepared samples were placed into a custom-built sample holder and 500 µL of blinking buffer (50 mM Tris/HCl pH 8, 10 mM NaCl, 10 % (w/v) d-glucose, 500 µg mL^{-1} glucose oxidase, 40 µg mL^{-1} catalase, 35 mM MEA) was added for imaging of Alexa Fluor 647 samples.

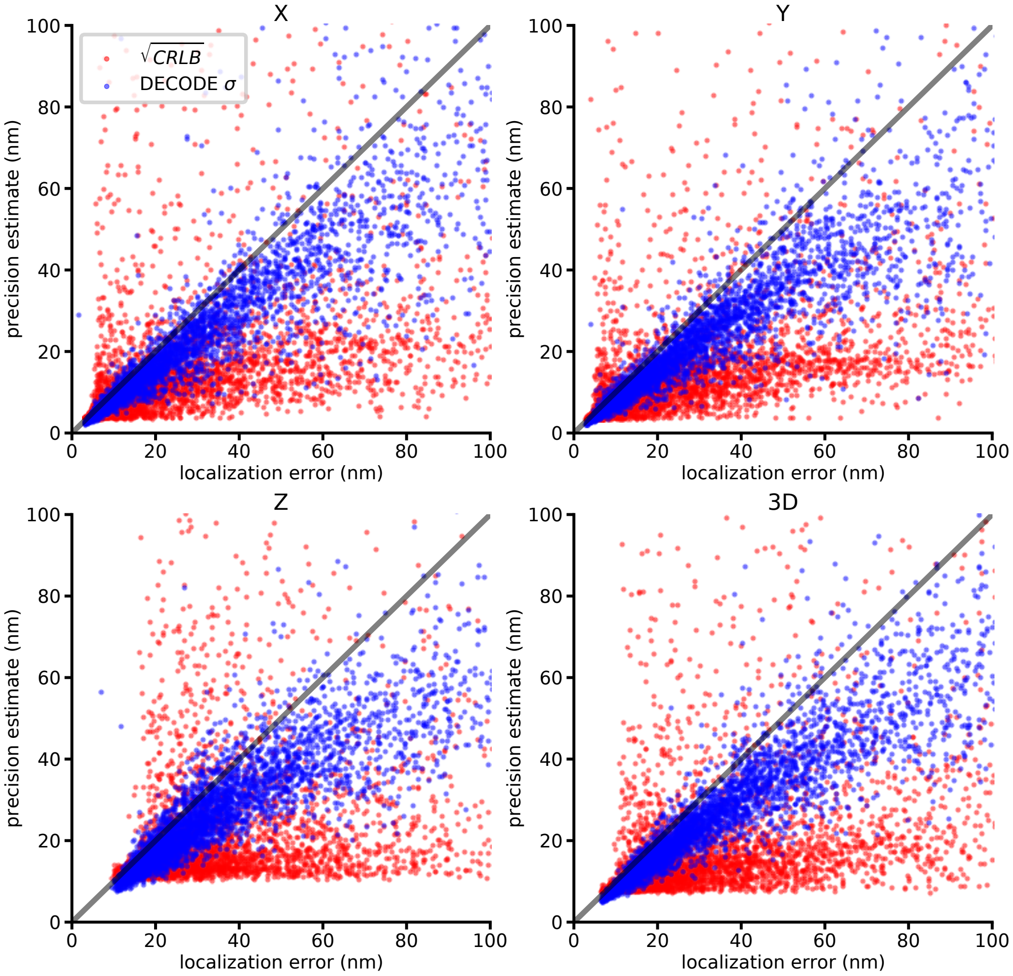
For imaging of microtubules at different activation densities (Fig. 4a), we used an exposure time of 15 ms and an excitation intensity at 640 nm of 15.5 kW cm^{-2}. We adjusted the UV pulse length to result in the desired density of activated fluorophores. As we started with the highest density, by the time we imaged the lowest density a large fraction of the fluorophores was bleached so that we could operate in the single-emitter regime.

For imaging microtubules with ultra-high labeling, we used an exposure time of 15 ms and an excitation intensity at 640 nm of 13.4 kW cm^{-2} and no UV activation.

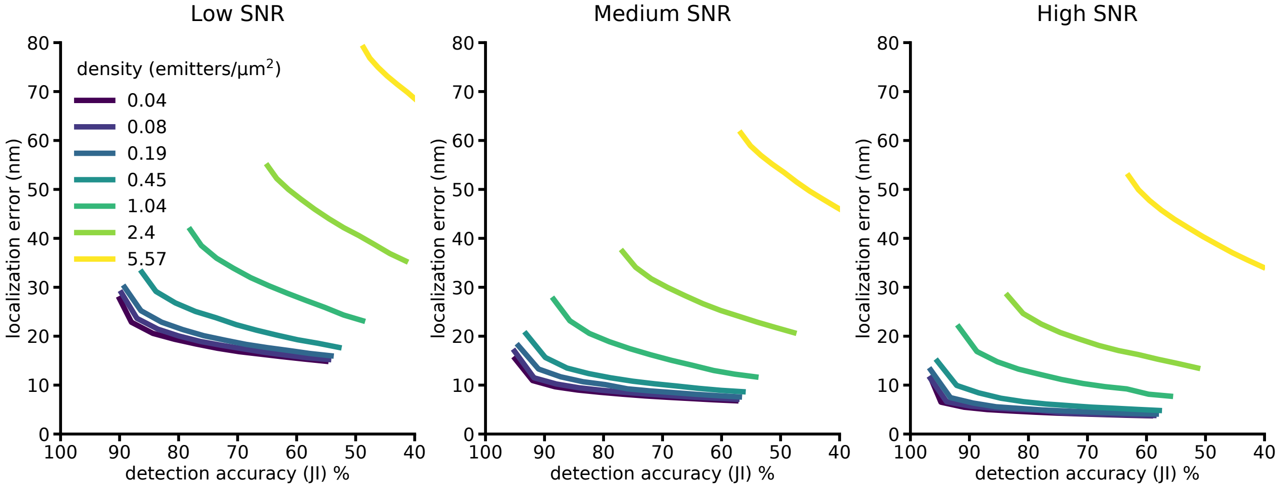
For live-cell imaging of Calnexin-mEos3.2 and MannII-mEos3.2 (Fig. 4d and e), the coverslips were washed briefly in PBS and subsequently mounted in 50 mM Tris/HCl pH 8 in 95 % (v/v) D_2O. The data were acquired with an exposure time of 15 ms, an excitation intensity of 22.6 kW cm^{-2} for the 561 nm laser, and a maximum intensity of 42 to 127 W cm^{-2} for the 405 nm laser. The pulse length of the 405 nm laser was adjusted manually to maintain a high emitter density and to allow imaging of all fluorophores in the field of view in about 1 min.

For the acquisition of live-cell data of Nup96-mMaple (Fig. 4f), coverslips containing Nup96-mMaple cells[11] (catalog no. 300461; CLS Cell Line Service, Eppelheim, Germany) were rinsed twice with warm PBS before they were mounted in 1 mL growth medium containing 20 mM HEPES buffer and imaged directly. During imaging, we used an excitation intensity at 561 nm of 16.7 kW cm^{-2} and a UV laser power of 80 W cm^{-2}. The exposure time was 12 ms and the pulse length of the UV laser was automatically adjusted from 1 to 12 ms to keep the density of localizations constant.

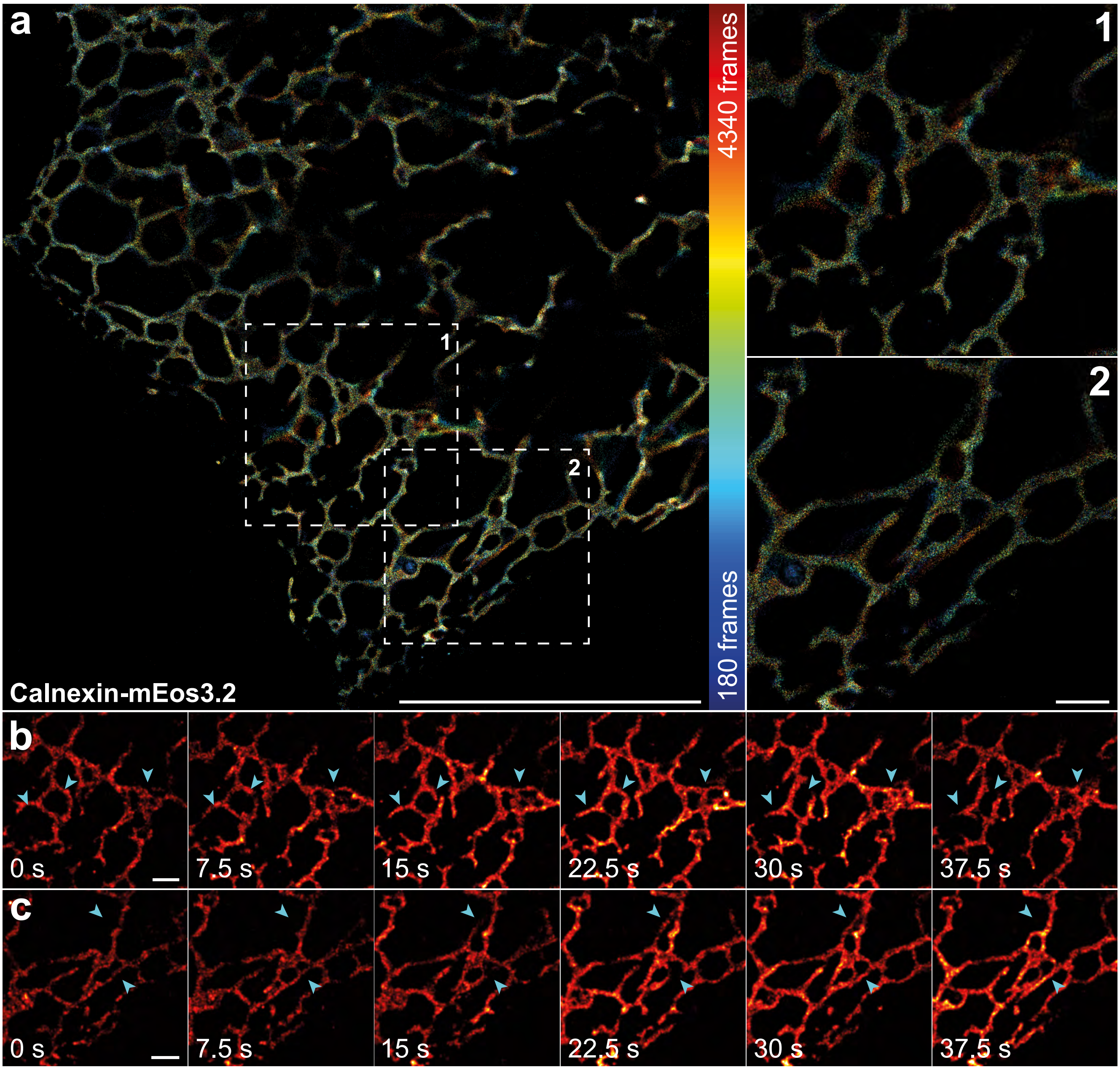
5 Supplementary figures



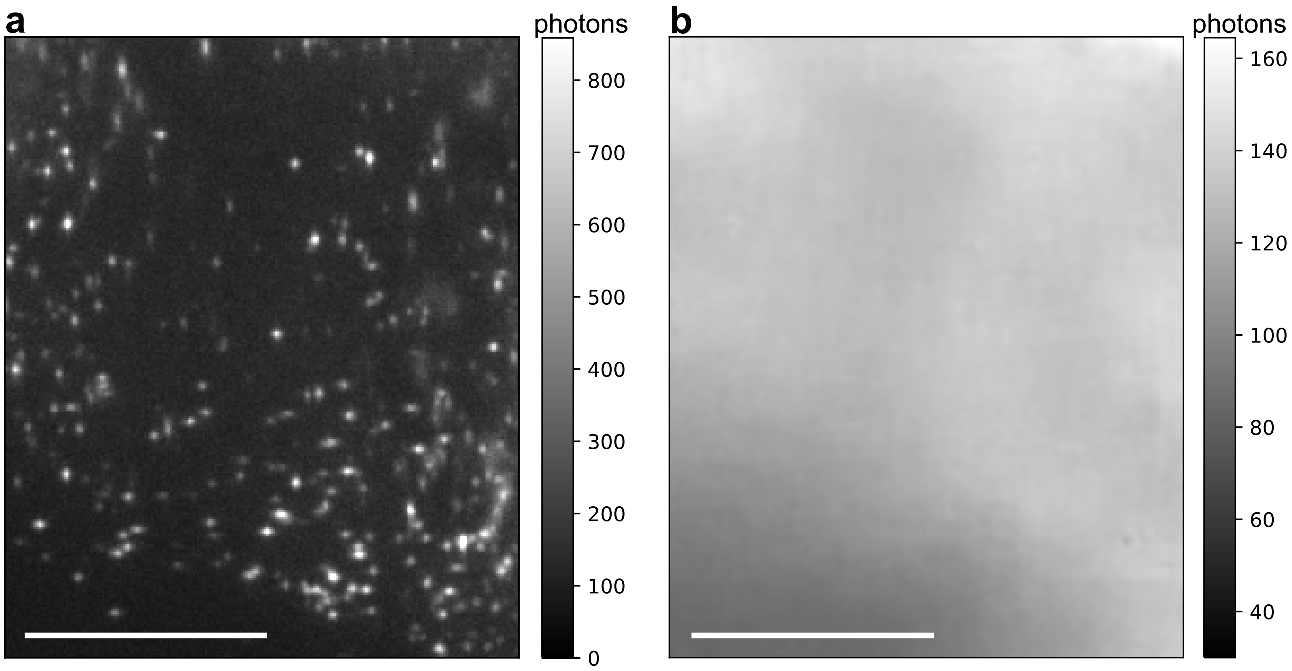
Supplementary Figure 1. Comparison of inferred uncertainty and single emitter CRLB in each dimension for dense data. DECODE's σ predictions correlate closely to the measured localization error in each dimension, i.e. much better than the single emitter CRLB estimate which assumes isolated emitters. We simulated the same dense emitter configuration 100 times and calculated the measured localization error as the RMSE of the predictions of the coordinates. See Methods and Supplementary Table 1 for additional details on training and evaluation.



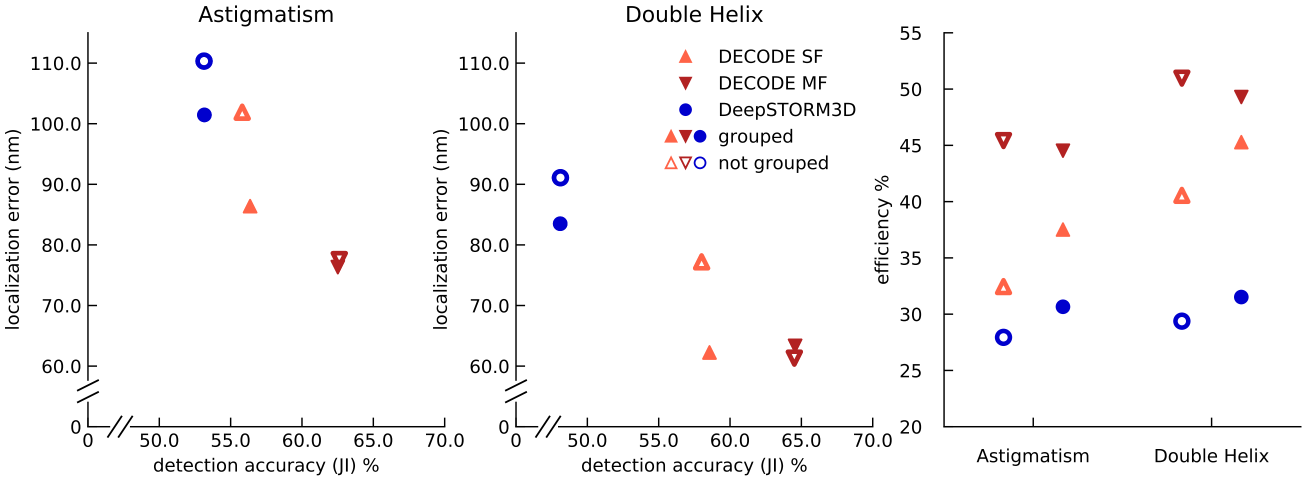
Supplementary Figure 2. Impact of filtering on localization error and detection efficiency. Each line corresponds to the two performance metrics evaluated for DECODE multi predictions for a given density with 0% - 40% of the worst predictions removed (ordered by the predicted DECODE σ). As the DECODE σ are well calibrated they allow to effectively trade off detection accuracy for a lower localization error. See methods and Supplementary Table 1 for additional details on training and evaluation.



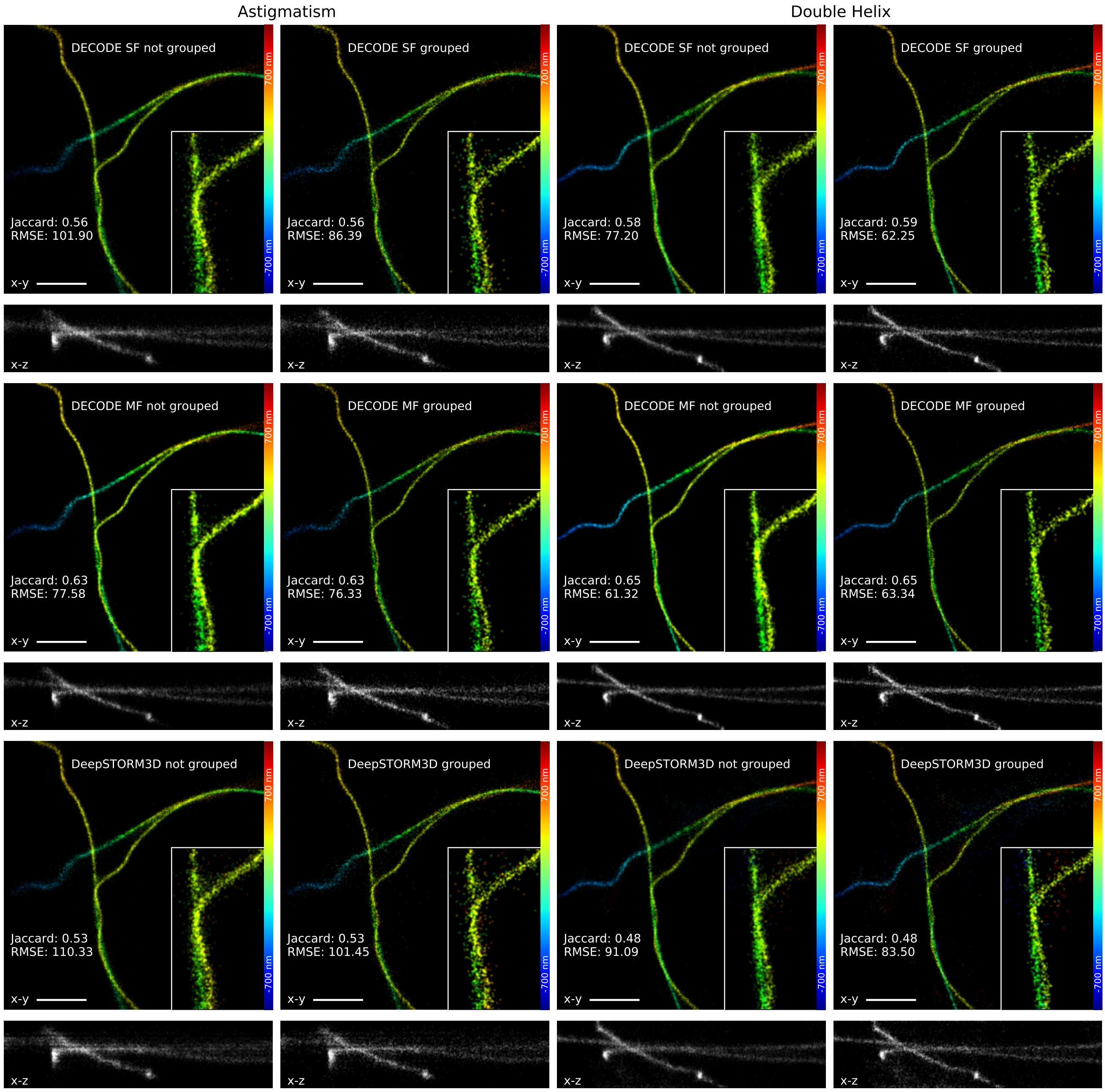
Supplementary Figure 3. Fast live-cell SMLM on the endoplasmic reticulum. a) Calnexin-mEos3.2 was imaged in U-2 OS cells and the individual localizations were color-coded by the frame they were observed in. On the right, 2 regions (dashed boxes in the overview image) are shown where dynamic changes could be observed during the time course of imaging. **b)** and **c)** Galleries of the two regions where in each reconstruction, localizations from 500 frames are shown. Arrows indicate regions with dynamic changes. See Supplementary Movie 3 and Supplementary Movie 4. Scale bars are 10 µm (overview image) and 1 µm (zoom-ins and galleries).



Supplementary Figure 4. Background prediction. Scale bar 100 pixels. Depicted are raw input frame from the data set corresponding to Fig. 4g (converted into photon units) and respective background prediction output of the model (rescaled into photon units).



Supplementary Figure 5. Performance evaluation of DeepSTORM3D and DECODE on the SMLM 2016 high density, low SNR training datasets with different modalities using the detection accuracy (Jaccard Index, JI, higher is better), localization error (lower is better) and efficiency (higher is better) as metrics. Both algorithms were evaluated with and without grouping. For DECODE we trained one model with (multi-frame, DECODE MF), and one without temporal context (single frame, DECODE SF). Even without temporal context, DECODE SF is superior to DS3D in the accuracy of both detection and localization, highlighting the importance of our network architecture, output representation, and loss function. Further improvements in DECODE performance come from temporal context.



Supplementary Figure 6. Comparison of reconstruction quality on SMLM 2016 challenge data. Reconstructions by DECODE and the DeepSTORM3D algorithm on the high density, low signal astigmatism (first two columns) and double helix (last two columns) challenge training data. For each setting x-y view, color coded by z coordinate, and x-z reconstructions are shown. The cut-outs show how the quantitative improvements given by the higher Jaccard and lower RMSE values of DECODE are reflected in better resolved details and less spurious localizations. Scale bars are 1 µm.

6 Supplementary tables

Figure	Data density [μm^{-2}]	Train Density [μm^{-2}]	SNR	DECODE filtering
Fig. 2a	1 emitter / frame	0.65	High	None
Fig. 2b	3.00	2.17	Medium	None
Fig. 2c	0.04 - 5.57	0.65, 2.17	Low, Medium, High	None
Fig. 2d	0.19 - 5.57	0.65, 2.17	Medium	None
Fig. 2e	0.04 - 5.57	0.65, 2.17	Medium	$N_{\text{emitter}}^{DECODE} = N_{\text{emitter}}^{\text{DeepSTORM3D}}$
Ext. Fig. 2	0.08, 2.40	0.65, 2.17	Medium	$N_{\text{emitter}}^{\text{DECODE}} = N_{\text{emitter}}^{\text{DeepSTORM3D}}$
Ext. Fig. 3	0.04 - 5.57	0.65, 2.17	Low, Medium, High	$N_{\text{emitter}}^{\text{DECODE}} = N_{\text{emitter}}^{\text{DeepSTORM3D}}$
Ext. Fig. 4	1 emitter / frame	0.65	High	None
Ext. Fig. 6	0.04 - 5.57	0.65, 2.17	Medium	None
Ext. Fig. 10	1.04	0.65	Medium	$N_{\text{emitter}}^{\text{DECODE}} = N_{\text{emitter}}^{\text{DeepSTORM3D}}$
Ext. Fig. 9	2.17	2.17	Inhomogeneous	None
Supp. Fig. S1	3.00	2.17	Medium	None
Supp. Fig. S2	0.04 - 5.57	0.65, 2.17	Low, Medium, High	0% - 40%

Table 1. Simulation and evaluation parameters for experiments based on our own simulations. Data density refers to the set of frames used for evaluation, while train density is the density of the simulated frames used for training the DECODE and DeepSTORM3D networks. For each emitter, we draw a photon flux from a Gaussian distribution $N(\mu_{\text{flux}}, \sigma_{\text{flux}})$. Low, medium and high SNR refer to mean photon counts of 1000, 5000 and 20,000 with background levels of 10, 50 and 200 photons per pixel respectively. To test predictions of inhomogenous backgrounds (Extended Data Fig. 9) we used a mean photon count of 7000 and background levels varying between 20 and 200 photons. The standard deviation of the intensity is calculated as $\sigma_{\text{flux}} = \mu_{\text{flux}}/20$. We used a mean on-time of 2 frames. For the CRLB comparisons in Fig. 2a and Extended Data Fig. 4 all emitters were instead simulated with exactly 20,000 photons and 200 background photons and did not persist across frames. To compare the DECODE's σ predictions with the measured localization uncertainty (Fig. 2b and Supp. Fig. S1) we simulated 100 frames and sampled the noise 100 times. For the CRLB comparisons (Fig. 2a and Ext. Fig. 4) we simulated 100k frames. For the remaining figures we simulated frames for each combination of data density and SNR until we acquired at least 20k emitters and 1k frames.

References

1. Chen, B.-C. *et al.* Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science* **346**, 1257998 (2014).
2. Legant, W. R. *et al.* High-density three-dimensional localization microscopy across large volumes. *Nat. methods* **13**, 359–365 (2016).
3. Bornschein, J. & Bengio, Y. Reweighted wake-sleep. *CoRR* **abs/1406.2751** (2014). 1406.2751.
4. Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).
5. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
6. Mund, M. *et al.* Systematic nanoscale analysis of endocytosis links efficient vesicle formation to patterned actin nucleation. *Cell* **174**, 884–896 (2018).
7. Deschamps, J., Rowald, A. & Ries, J. Efficient homogeneous illumination and optical sectioning for quantitative single-molecule localization microscopy. *Opt. express* **24**, 28080–28090 (2016).
8. Schröder, D., Deschamps, J., Dasgupta, A., Matti, U. & Ries, J. Cost-efficient open source laser engine for microscopy. *Biomed. Opt. Express* **11**, 609–623 (2020).
9. Edelstein, A., Amodaj, N., Hoover, K., Vale, R. & Stuurman, N. Computer control of microscopes using µManager. *Curr. protocols molecular biology* **Chapter 14**, Unit14.20 (2010).
10. Deschamps, J. & Ries, J. EMU: Reconfigurable graphical user interfaces for Micro-Manager. *BMC Bioinforma.* **21**, 456 (2020).
11. Thevathasan, J. V. *et al.* Nuclear pores as versatile reference standards for quantitative superresolution microscopy. *Nat. Methods* **16**, 1045–1053 (2019).